Master research Internship

Bibliographic report

Finitely presented infinite structures for strategic reasoning in multi-agent systems

Domain: Formal Languages and Automata Theory - Logic in Computer Science

Author:
Sébastien Lê Cong

Supervisor:
Sophie Pinchinat
François Schwarzentruber
LogicA — IRISA

Abstract: Prospective reasoning in a situation where several agents have some knowledge about the world and about each other's knowledge while several distinct events can take place is both a crucial and complex task in general; to better assess what may or may not happen, a useful framework lies in dynamic epistemic logic (DEL), which basically consists in unfolding jointly an epistemic and a time models so as to obtain a (generally) infinite structure. This allows us to state formally a planning problem, namely, how to satisfy some objective through applying some sequence of events, and the protocol synthesis problem, which involves indicating — possibly exhaustively — what precise sequence(s) are suitable.

Such DEL structures have been studied using various methods, generally reducing the infinite structure to a finite representation; one of these methods exploits automata theory: since we have, for instance, several decidability results and proof techniques for automatic structures at our disposal, bringing a DEL structure closer to an automatic presentation yields new theorems. Notwithstanding, this approach, although promising, has not yet been generalised.

Our objective is to systematise the use of automatic structures to derive properties such as (un)decidability in some cases, or even complexity estimates, and to explore further some less common automatic structures, such as those taking trees as their input rather than finite words, with the same focus in mind.

Contents

1 Introduction

In a situation involving several interacting thinking individuals — be they machines or not —, reasoning soundly about the dynamics of events according to what each one knows matters to any observer, whether he is an outsider or an insider. This configuration encompasses several practical and theoretical fields such as artificial intelligence (one can think of cooperating robots [DA16]), economics, or game theory [Aum99], and may deal with competitive contexts as well as collaborative ones. The present report inspects a generic problem known as (epistemic) planning: from a fixed starting situation and some objective (which can impose conditions on the state of knowledge), is it possible to execute some sequence of actions, or that some sequence of events occurs, so as to reach the predefined goal? Moreover, can we synthesise some or several various ways (if any) to achieve so? And what if we set our sights on infinite executions, and go after general temporal properties to be verified?

Once we are able to formalise the problem, we can see that the key issue is to deal with a class of typically infinite structures, named DEL structures (where DEL stands for *dynamic epistemic logic*) [VDvDHK07, BMS98]; the first idea that may come to mind would be to transform these entities into finite structures, where decidability and complexity analysis are tractable. This is the approach originally adopted for epistemic planning [BA11, YWL13], generating demonstrations (of decidability and undecidability) for restricted subclasses of models; even so, these methods do not appear to easily extend to more expressive objectives, while we hope for generalisation.

On the other hand, also using finiteness, another series of endeavours has turned to automatic structures: those are structures that can be described by a collection of finite automata — we call the latter an *automatic presentation* —, and therefore admit a specific finite representation in which some logical calculus, typically first-order logic [Blu99, BG00], can be ensured thanks to the procedural nature of these abstract machines. Such a viewpoint has been applied successfully, paying dividends either by identifying new (un)decidable subclasses of planning instances, giving new complexity bound estimates, or even exposing alternative proofs of prior findings [DT16, Mau14, AMP14]. This time, aiming for a general approach rather than *ad hoc* arguments seems promising, especially [DT16], and could lead towards a better understanding of DEL structures.

Since the DEL and automatic structure research communities are parallel to this day, it makes sense to delve into automatic presentations and into its settled theory to draw it closer to open epistemic problems. As a DEL structure is essentially a forest, we might take advantage of emphasising little or less operated automata whose input (such as trees) might reveal itself more adapted in this context.

We start in Section 2 with a look at standard tools to enable a sufficient modelling of the problems, namely DEL structures and epistemic temporal logic; we move along to Section 3 by introducing the theory of automatic structures and some of its main advances, with the particular intent of applying them to DEL structures, which is why our scope is biased towards decidability theorems and proof techniques; we then state the epistemic planning and protocol synthesis problems more formally in Section 4, and give a series of results already obtained, without and with making use of automatic structures, and point out unsolved questions which need be tackled; finally, we evaluate in Section 5 what approach could bear fruit, and some further targets we may try and attain.

2 DEL framework

In order to model both knowledge and time simultaneously, we make use of a construction stemming from dynamic epistemic logic (DEL) as our framework.

To achieve so, we firstly define a logic that allows us to consider agents' knowledge about the world, and secondly, a way to describe possible events; we then use these two notions jointly so as to produce a structure that relevantly caracterises what actions may happen in the future and how the various agents will comprehend any changes.

The following definitions are quite standard in the literature [AMP14, CMS16, DT16, YWL13, BA11, Mau14], although there may be minor variations in the selected formalisms.

2.1 Epistemic logic and epistemic models

Epistemic logic $\mathcal{L}_{\mathsf{K}}$ or $\mathcal{L}_{\mathsf{EL}}$ can be seen as a modal extension of propositional logic: given a finite set AP of atomic propositions as usual, we add a finite set of *agents* Ag, and then a new *knowledge* operator K_a for each agent a.

Here is a BNF grammar that provides the syntax for any proposition φ in $\mathcal{L}_{\mathsf{EL}}$:

$$\varphi \quad ::= \quad p \;\mid\; \neg\varphi \;\mid\; (\varphi \wedge \varphi) \;\mid\; K_a\varphi$$

There are other operators linked to knowledge, such as the *common knowledge* operator $C_{\mathcal{B}}$ where $\mathcal{B}$ is a subset of agents ($\mathcal{B} \subseteq \mathsf{Ag}$), which gives rise to another logic $\mathcal{L}_{\mathsf{KC}}$ [YWL13]; several studies restrict themselves to the K_a operator and thus $\mathcal{L}_{\mathsf{K}}$.

The semantics of $\mathcal{L}_{\mathsf{EL}}$ rests upon *epistemic models* (an application of Kripke models), where a finite set of worlds decorated by a subset of AP are linked through transitions associated to the agents: for agent a, an a-transition between worlds w and w' conveys that a reckons w' as the possible current world when the latter is effectively w.

Rigorously, $\mathcal{W} = (W, (R_a)_{a\in\mathsf{Ag}}, V)$ with W a finite set (of worlds), each R_a a binary relation on W, and $V : W \to 2^{\mathsf{AP}}$ a valuation on worlds. We may occasionally write $\overset{a}{\hookrightarrow}$ for R_a, the *accessibility relation* in $\mathcal{W}$ for a.

Usual terminology may talk about *doxastic logic* (a logic of *belief* rather than *knowledge*) in place of epistemic logic when accessibility relations are not required to be reflexive; we do not make that distinction here, putting no such constraints.

A specific world w of the epistemic model $\mathcal{W}$ may be distinguished (to highlight w as the "actual" world), in which case we note the resulting *pointed epistemic model* as $(\mathcal{W}, w)$.

The truth value of a formula is evaluated in a pointed epistemic model; as the semantics for the usual inductive cases is forthright, we only concern ourselves with the knowledge operator for arbitrary agent a and $\mathcal{L}_{\mathsf{EL}}$ formula φ:

$$(\mathcal{W}, w) \models K_a\varphi \text{ iff for all worlds } w' \text{ such that } w \overset{a}{\hookrightarrow} w',\ (\mathcal{W}, w') \models \varphi.$$

In layman's terms, $K_a\varphi$ means that proposition φ stands in all of the worlds thought by agent a to be possible when the real world is w.

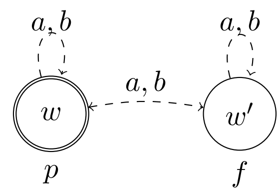

Figure 1: An epistemic model with two agents (a and b); the pointed world w is circled. Here, neither a nor b can distinguish between both worlds, while the real world is w.

2.2 Event models

Event models are akin to Kripke structures, except that worlds are replaced with *events*, which are no longer decorated with atomic propositions, but rather each with a *precondition* and a *postcondition*.

Preconditions are $\mathcal{L}_{\mathsf{EL}}$ formulae that define in which worlds — namely, those satisfying the precondition they assort with — the corresponding events can take place.

Postconditions tell which atomic propositions have to hold, not to hold or can stay unchanged after the event has taken place; an event model whose postconditions are all "ineffective" is called *purely epistemic*, while a postcondition that does have some actual effect is deemed *ontic*.

Precisely: an event model $\mathcal{E}$ is a tuple $(E, (R'_a)_{a\in\mathsf{Ag}}, \mathsf{pre}, \mathsf{post})$ where E is a finite set (of events), each R'_a is a binary (accessibility) relation on E, $\mathsf{pre} : E \to \mathcal{L}_{\mathsf{EL}}$ is a precondition function, and $\mathsf{post} : E \to \mathsf{AP} \to \mathcal{L}_{\mathsf{EL}}$ is a postcondition function (the last two mappings send their arguments to a formula selecting worlds). We might abuse notations and use $\overset{a}{\curvearrowright}$ too instead of R'_a.

Analogously as for worlds, we designate $(\mathcal{E}, e)$ as a *pointed event model* where e is some event from $\mathcal{E}$ (the "true" event).

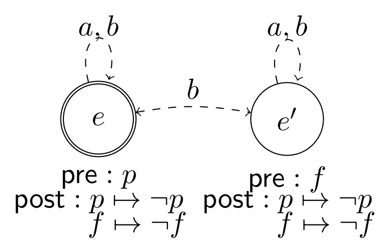

Figure 2: An event model with atomic propositions p and f, and simple preconditions and postconditions (the latter in matching form). When event e or e' happens, a knows which one did; quite the opposite, b cannot tell. Be wary of differing preconditions and of the "switching" effect postconditions have.

Depending on the nature of preconditions and postconditions, several subtypes of event models and thus subcases of the problem we will define in Section 4 arise — such as the non-ontic case, k-depth modality preconditions, or even propositional preconditions and postconditions — and yield specific results which will be described further down.

Several event types of interest have been spotted [YWL13, BMS98]; we do not review them all, as we concentrate on some special cases that have been handled in Sections 4.2 and 4.3[1]:

- An event model with events set E is said to be *globally deterministic* iff for all (e, e') in E^2, $\mathsf{pre}(e) \wedge \mathsf{pre}(e') \equiv \bot$; with all the more reason, in any world, no two distinct events can be performed;

- A *public announcement* is a non-ontic event e such that for every agent a, $e \overset{a}{\hookrightarrow} e'$ iff $e' = e$: enacting e amounts to remove all worlds that violate $\mathsf{pre}(e)$; this conveys having all agents know $\mathsf{pre}(e)$;

- A *functional dependent* event model is one whose accessibility relations are functional. *Secret communication* event models — where there are exactly two events, some agents perfectly distinguish between the two (accessibility is identity), and all the others only see the same event — fall into this class.

2.3 DEL structures

The intuitive effect of events on worlds is formalised by means of the *update product*: given a pointed epistemic model $(\mathcal{W}, w_0)$ and a pointed event model $(\mathcal{E}, e_0)$, their update product $(\mathcal{W}, w_0) \otimes (\mathcal{E}, e_0)$ is a new epistemic model (hence new worlds) defined as follows:

Definition. *If $w_0 \not\models \mathsf{pre}(e_0)$, then the update product is undefined; otherwise,*

$$(\mathcal{W}, w_0) \otimes (\mathcal{E}, e_0) = ((W^{\otimes}, (R_a^{\otimes})_{a \in \mathsf{Ag}}, V^{\otimes}), (w_0, e_0)) \text{ with:}$$

- $W^{\otimes} = \{(w, e) \in W \times E \mid (\mathcal{W}, w) \models \mathsf{pre}(e)\}$ *(each world satisfying an event's precondition generates a new world through that event);*

- $\forall a \in \mathsf{Ag}, \forall((w, e), (w', e')) \in (W^{\otimes})^2, (w, e) R_a^{\otimes} (w', e') \iff (w \overset{a}{\hookrightarrow} w' \wedge e \overset{a}{\hookrightarrow} e')$ *(agent a conflates new worlds iff he conflated both the associated old worlds and events);*

- $\forall (w, e) \in W^{\otimes}, V^{\otimes}(w, e) = \{p \in \mathsf{AP} \mid (\mathcal{W}, w) \models \mathsf{post}(e)(p)\}$ *(any new world observes the postcondition that the associated event required from the old world).*

We identify the new worlds (w', e') with the words $w'e'$, for ease of notation and for the sake of the iteration process described below.

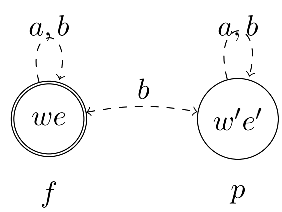

Figure 3: The update product of the two preceding models: now, b is the only one unable to judge which is the correct world. Atomic propositions' truth values have also been "swapped".

[1]Actually, they are both simple enough to apprehend and likely to show up in modelling.

By iterating the previous operation, starting with $(\mathcal{W}, w)$ and applying successively event models $(\mathcal{E}_i, e_i)$ indexed by i, we get a (potentially infinite) sequence of epistemic models $(\mathcal{W}\mathcal{E}^n)_{n\in\mathbb{N}}$, each of which has $we_0 \dots e_n$ as its singled out world, and all its other worlds are also in $\mathcal{W}\mathcal{E}^n$. This way, each original world unfolds as a *tree*, and the original epistemic model spreads out as a *forest*; they will make ground for rephrasing the as we shall see a little further.

These words constitute a subset of $\mathcal{W}\mathcal{E}^*$; a sequence of such words is called a *history* and informs us on what series of events took place until the final world it denotes; histories are the main subject of inquiry in planning.

The whole structure (consisting of all worlds) obtained through this process is called a *DEL structure*; it may well be infinite, and, as a direct result of the way it was contructed, exhibits both (vertical) time relations and (horizontal) knowledge relations.

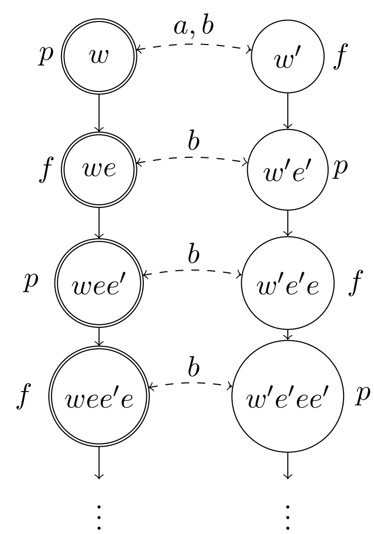

Figure 4: Glimpse of a simple DEL structure generated by the previous models; solid lines realise actions/time while dashed lines substantiate knowledge *(self loops were overlooked)*. There is an alternation for atoms; branching is notably absent — although we do succeed in generating an infinite pattern — for we have only shown a limited situation for practical purposes, to illustrate principles.

Dynamic epistemic logic $\mathcal{L}_{\mathsf{DEL}}$ ensues from the adjunction of an action operator $\langle \mathcal{E}, e\rangle$ for each event model: from formula φ, we get $\langle \mathcal{E}, e\rangle\varphi$, whose semantics under some pointed epistemic model $(\mathcal{W}, w)$ is to hold iff $(\mathcal{W}, w) \models \mathsf{pre}(e)$ and $(\mathcal{W}, w) \otimes (\mathcal{E}, e) \models \varphi$ [CMS16, BMS98].

This model could be stretched to take extra information into account, such as each action's performer and cost, ending up with new metrics to coerce our objective and problem [YWL13], but we discard this sophistication as in the reviewed material.

We can linearly reduce a set of events to a single one by replacing them by their disjoint union, as [Mau14] has proceeded. In this setting, reframing the construction in *epistemic temporal logic* (ETL) becomes feasible. This logic's primary objects are labelled trees closed by non-empty prefixes (they are the equivalent of histories), giving birth to forests, whence it is possible to establish the $\mathsf{CTL}^*\mathsf{K}_n$ logic [AMP14]: indeed, ETL is in relation with DEL as was first brought to light by earlier efforts [vBGP07, vBGHP09].

Syntactically speaking, $\mathsf{CTL}^*\mathsf{K}_n$ is merely the classic temporal logic CTL^* (we shall give a flavour of CTL^* later, in Section 4.1) with modal knowledge operators K_a (with $n = |\mathsf{Ag}|$ = number of agents) as an option for forming state formulae; from the semantic outlook, the novelty resides in how K_a links trees together: node x for tree t satisfies formula $K_a\varphi$ (where φ is a $\mathsf{CTL}^*\mathsf{K}_n$ state formula) iff any node y agent a conflates with x ($x \overset{a}{\leadsto} y$) does satisfy φ for the greatest tree that comprises y.

3 Automatic structures

3.1 Definition

We introduce the parallel concept of *automatic structure*, which is a special case of logical structure; informally, an automatic structure is a structure whose domain and relations, including equality, can be represented by an encoding and automata operating on said encoding.

We look into a generalised notion of automaton, namely, a *synchronous n-tape transducer* [Ber13, RS97] or *synchronous n-tape automaton* [Rub08].

Such machines can be given various candidate types to read, like words, infinite words (named ω-words), trees, or infinite trees (likewise: ω-trees) [Blu99]; the previous list does not aim for completeness (for instance, the notion of transfinite words can also be encountered). Nonetheless, (finite-)word presentations have been more studied [Rub08].

In the finite case, words (or trees) may be completed with a special "void" padding character (denoted by $\square$ or $\perp$, for example) so as to define their *convolution*, which in turn is used as the input for the automaton [Blu99]: for words w_k ($1 \leq k \leq n$) of lengths l_k, writing out $w_k = w_{k,1} \dots w_{k,l_k}$ and $l = \max\limits_{1 \leq k \leq n} l_k$, we define $\bigoplus\limits_{k=1}^{n} w_k$ as an n-tape word of length l:

$$\bigoplus_{k=1}^{n} w_k = \begin{bmatrix} w'_{1,1} \\ \vdots \\ w'_{n,1} \end{bmatrix} \dots \begin{bmatrix} w'_{1,l} \\ \vdots \\ w'_{n,l} \end{bmatrix}$$
$$\text{where } w'_{k,i} = \begin{cases} w_{k,i} & \text{if } i \leq l_k \\ \square & \text{if } i > l_k \end{cases}$$

We write $(\Sigma^*)^{\oplus n}$ for the set of n-convoluted words on base alphabet Σ.

We generally stick to finite words here, as the known results to be presented (see sections 4.2 and 4.3) derive from this case; as an aside, a similar convolution can be defined for finite trees [Blu99].

(Notice such an n-tape automaton on alphabet Σ is almost a classical 1-tape automaton on $(\Sigma \cup \{\square\})^n$, except the acceptable form for the entries has to be restricted a bit.)

Now, given a finite or countably infinite relational[2] signature τ with $\tau = (R_i)_i$ (where R_i has arity r_i), we consider the τ-structure $\mathcal{A}$ defined as $(A, (R_i^{\mathcal{A}})_i)$ where A is some set, and each $R_i^{\mathcal{A}}$

[2]Let us remark any function can be viewed as a relation, therefore we can tolerate functions in signatures without loss of generality.

an interpretation of R_i on A; recall the equality predicate has to be featured (it is often implicit in received formalism).

Using some alphabet Σ (assumed to be finite), we require the existence of some *encoding function* ν, which has to be a **surjective partial function** from Σ^* to A with $\mathrm{dom}(\nu)$ a regular language; furthermore, we demand that the "inverse images" under ν for each relation (specially equality) be regular languages:

For relation symbol R_i amongst τ, the language L_{R_i} defined as follows is regular:

$$L_{R_i} = \{\bigoplus_{k=1}^{r_i} w_k \in (\Sigma^*)^{\oplus r_i} \mid (\nu(w_k))_{1 \leq k \leq r_i} \in R_i^{\mathcal{A}}\}$$

$(\nu, \Sigma, \mathrm{dom}(\nu), (L_{R_i})_i)$ is then called an *automatic presentation* of $\mathcal{A}$.

3.2 Decidability of the model checking

In the finite case, any automatic presentation boils down to an injective one: as a proof, we can use the "alphabetical order", whose pivotal property is being a recognisable well-order, to choose the smallest encoding for each element of the structure [Blu99].

This property is of special interest as it is made extensive use of by the theory: many further propositions' demonstrations are built on it, including results on to-be-defined "regular automatic trees" in Section 4.3.

In addition, we can even generically trade any alphabet for a binary equivalent by adjusting the encoding, even in the infinite (words or trees) case [Blu99], but this property seems less significant so far.

As a prerequisite, let us say that *first-order logic* (FO) or *predicate calculus* extends propositional logic with quantification over elements (for an introduction to mathematical logic, we refer to [EFT13]); we will introduce another logic, chain-MSO, and the associated type of quantification later on.

Two fundamental theorems surface [Rub08]:

1. *Definability*: in an automatic structure, any FO formula defines a "decoded" regular relation;

2. *Decidability*: in any automatic strucure, FO is decidable.

Actually, regularity for relations is preserved through many elementary operations; such a distinctive feature permits the practical shaping of new automatic presentations from older ones.

The two most usual quantifiers ($\forall$ and $\exists$) preserve regularity; using a general definition for quantifiers, such as the "there exists infinitely many" quantifier ($\exists^\omega$ or $\exists^\infty$), we can ask which ones possess the same property, and in the finite word case, others have been found to be regularity-preserving as well [Rub08]. This notion is useful as it allows us to build new decidable formulae upon preexisting ones (and thus to extend the logic under consideration).

Example. *$\exists^\omega$ preserves regularity in the finite case, and we can conclusively affirm FO[$\exists^\omega$] to be decidable.*

Another useful notion is that of *interpretation* [Blu99]: from an automatic structure, finding an appropriate exterior one where the former can be described by means of FO logic formulae from the latter guarantees the second structure's automaticity. In other words, we can *embed* a structure into another one. The general concept of interpretation goes past FO.

Definition. *For a logic $\mathcal{L}$, an $\mathcal{L}$-interpretation of dimension k of structure $\mathcal{A}$ of base set A and signature $(R_i)_{i\in I}$ (where equality belongs to the predicates, and r_i is R_i's arity) in structure $\mathcal{B}$ of base set B and signature $(R'_j)_{j\in J}$ is a tuple $(\delta, (\varphi_{R_i})_{i\in I})$ where:*

- *all formulae (to wit, $\delta(\bar{x})$ and each $\varphi_{R_i}(\bar{x}_1, \ldots, \bar{x}_{r_i})$) are $\mathcal{L}$ formulae taking k-tuples as parameters ($\bar{x}$ and each $\bar{x}_i$), and whose predicates are taken amongst $(R'_j)_{j\in J}$;*
- *$\delta^{\mathcal{B}}$ defines the domain B' ($B' \subseteq B^k$) of a surjective map $h : B' \to A$;*
- *for all i in I, $\mathcal{B} \models \varphi_{R_i}(\bar{x}_1, \ldots, \bar{x}_{r_i})$ iff $(h(\bar{x}_1), \ldots, h(\bar{x}_{r_i})) \in R_i^{\mathcal{A}}$.*

Infinite signatures demand a supplementary condition of computability for the indexing $i \mapsto \varphi_{R_i}$ [Rub08].

General proving techniques emerge straight away [Blu99]; for a start, interpretation can be used both ways: we can either try and relate to a canonical interpretation in $\mathcal{N}_k$ or $\mathcal{W}_k$ (see just below for a definition), since a powerful theorem asserts such a link to be the defining characteristic[3]; or conversely, we can make profit from structures known not to be automatic (such as Peano arithmetic $(\mathbb{N}, +, .)$, which has undecidable FO theory) to deduce another one cannot admit an automatic presentation.

The mentioned canonical (automatic) structures seem important enough to be elucidated here:

1. $\mathcal{N}_k = (\mathbb{N}, +, |_k)$: $n|_k m$ iff n is a power of k that divides m;
2. $\mathcal{W}_k = (\Sigma^*, (\sigma_a)_{a\in\Sigma}, \preceq_p, \mathrm{el})$: σ_a is the function appending the letter a to the end of a word, $w \preceq_p w'$ means that w is a prefix for w', and el is a binary predicate testing for equal length.

This means we can take either an arithmetic or a word-based point of view.

A "quantitative" tool can be mentioned too ([Blu99, Proposition 5.1]): in the finite-word case, we can express a relative upper bound on the encoding of functional relations; consequently, violating this condition signals non-automaticity. This criterion follows from an argument reminiscent of the traditional demonstration of the pumping lemma for regular languages, as we stumble upon a repeted state in the function's automaton, which enables iterating multiple encodings for the function's value and thus contradicts the presentation's injectivity.

As a reminder for the next result, *monadic second-order logic* (MSO) expands first-order logic by introducing quantification over sets, through *second-order variables* (conventionally capitalised), in opposition to *first-order variables* (ordinarily in lower case), the latter affecting elements; a small example: $\exists X, (X \neq \varnothing \wedge \forall x, \forall y, (P(x, y) \implies P(y, x)))$ means binary predicate P is symmetric on some non-empty set.

Focus has been dispensed [DT16] on chain-MSO, whose very definition is a semantic restriction of MSO based on automatic presentations: indeed, second-order quantification now ranges over chains rather than generic sets, and the appraisal of a set as a chain is determined by the encoding. Intuitively, there is a natural connection binding this logic with trees, as exploring a branch is similar to scouring a chain.

As chain-MSO model checking is demonstrated to be decidable in automatic structures, the notion of *regular automatic tree* is drawn: in a tree with finite branching degree — let us assume

[3] We are abreast of highly resembling theorems for ω-words, finite trees, and ω-trees [Blu99], should we resort to these types of data.

bounded by n — and successor predicates S_i ($1 \leq i \leq n$), if we number each edge with the index i of the corresponding predicate S_i, the path joining the root to a certain node can be unequivocally described by the sequence of its edges' numbers; this sequence is the address of the node. For example, if we decided to number the edges from left to right (and supposing the following nodes and all possible edges before the leaves exist), the leftmost node of depth 3 would have address 111, the rightmost node of depth 2 would be attributed address nn, and the node reached by taking the leftmost, rightmost, right-to-leftmost and left-to-rightmost edges one after another would earn address $1n2(n-1)$.

To rule a tree[4] as regular automatic, the "canonical" addressing function for the nodes as just drafted has to be the decoding function associated to an automatic presentation (which is a stronger condition than simply having *some* automatic representation for the tree proposition 5).

Recall structure $\mathcal{W}_n$ on page 8: when looking at the full list of predicates used to extend $\mathsf{chain\text{-}MSO}$ in the paper, we can directly find a 1-dimensional embedding.

There are also several complexity results for *FO-queries* (namely, FO propositions to be verified) in automatic structures [Blu99], whose estimates depend on how entry size is specified (input structure, formula size, or both combined); this could be interesting for complexity estimates of any problem correctly reduced to a decision in an automatic structure.

4 Epistemic planning

4.1 Decision problems

Given a DEL structure as defined above (induced by $(\mathcal{W}, w)$, $(\mathcal{E}_i, e_i)_i$, and all possible applications of the update product), the *epistemic planning problem* consists in determining whether some $\mathcal{L}_{\mathsf{EL}}$ objective formula φ is reachable via some history.

A more general *planning problem* introduces temporality [AMP14] and deals with supposing φ's belonging to the extended logic $\mathsf{CTL}^*\mathsf{K}_n$: this latter approach of looking at infinite histories allows to take the (unbounded) future into consideration instead of restraining the scope to finite-time steps and properties; stating safety and liveness properties to be verified is possible in this language.

CTL^* [HT87] is a temporal logic, interpreted in trees, that amounts to make statements about nodes and (infinite) branches (with state formulae and path formulae respectively), possibly quantifying (with $\exists$ and $\forall$) over paths; modal operators grant finite-time and infinite-time expressivity (*next*: $\bigcirc$, *until*: U; also, *eventually*: $\Diamond$ and *generally*: $\Box$ as syntactic sugar).

The logic CTL^* is strictly broader than both linear-time logic (LTL: checks only linear execution paths; no quantification: we do not look at what happens "sideways") and computation tree logic (CTL: inspects all possible paths, but with compulsory quantification at each node step). For more details about this logic, we refer to [Eme90].

We present two examples to illustrate CTL^*'s meaning:

1. $\exists\Box\Diamond a$: "there exists an execution path where there are infinitely many nodes where atom a is true";

2. Here is a CTL^* formula that exceeds LTL and CTL's expressive powers (note it can be viewed as a disjunction bringing into play an LTL proposition and a CTL proposition):

[4] Will *all* its predicates, possibly exceeding sheer $(S_i)_{1 \leq i \leq n}$.

$\Diamond(a \wedge \bigcirc a) \vee \forall\Box\exists\Diamond a$: "someday, there is a step where a stands both now and at the next step, or for all executions, there is always a path that eventually leads to a's being true".

Another subsequent generalisation lies in *protocol synthesis*, which entails finding all (infinite) histories that satisfy a given planning problem.

4.2 Results with *ad hoc* proofs

Various studies have led to a host of results (including decidable fragments) and associated techniques.

The first thing to underline is the undecidability of the general epistemic planning problem; as a matter of fact, even without postconditions and propositional preconditions, letting several agents suffices for undecidability, as we can achieve a reduction to the halting problem for a Turing machine [BA11] (in the same spirit as that of the sketching below).

When preconditions accept modal depth 2 and no postconditions intervene, employing a 2-counter machine and relating it to models so as to provide a counter-example (undecidability) [CMS16]: considering a program for the aforementioned machine, by matching each program state (line number and counters' values) with an epistemic model and each instruction with an event model, a "simulated" instance of the epistemic planning problem is reduced to the undecidable halting problem for 2-counter machines.

On the undecidability side, we know that, on the contrary, non-void posconditions (even only propositional) result in undecidability as soon as preconditions reach modal depth 1 [BA11].

Concentrating on finiteness up to isomorphism in the globally deterministic case thanks to bisimulation contraction (and enumerative combinatorics), the problem is shown to be decidable in single-agent and specific frame (S5, KD45, K45[5]) cases [BA11]. The key point there is bisimulation's accordance with epistemic formulae: since two bisimilar states are equivalent under $\mathcal{L}_{\mathsf{EL}}$, the bisimulation contraction of any epistemic model must preserve .

The same philosophy was applied [YWL13] to take on multi-agent with propositional preconditions: using k-bisimilarity (a weak bisimilarity preserving $\mathcal{L}_{\mathsf{EL}}$ up to k-depth modality), we fulfil a fitting contraction.

This approach manages to broach other fragments, even when common knowledge is authorised, with propositional preconditions of specific kinds; seizing commutativity and other "algebraic" attributes modulo bisimulation, we conclude in favour of decidability. [YWL13]

Finally, making the most of commutativity just as aforesaid and integer linear programming [CMS16], one concludes that with propositional preconditions and no postconditions, epistemic planning is not only decidable but squarely in PSPACE(-complete).

4.3 Results based on automatic structures

Returning to the case of propositional events (PDEL), building upon the ascertained bridge between DEL and ETL [vBGHP09], $\mathcal{WE}^*$'s **regularity** is initially proven and then translated to a game arena background, for which a theory of uniform strategies is proposed [Mau14]. As a consequence, epistemic planning with an $\mathcal{L}_{\mathsf{EL}}$ objective is proven decidable anew; this assertion is not

[5] Those are conditions on accessibility relations; they step in during the combinatorial part of the proof, which we voluntarily omit for the sake of clarity owing to its technical nature.

unprecedented in itself, but the proof method is. What's more, the epistemic planning problem under a $\mathsf{CTL}^*\mathsf{K}_n$ goal is also decidable,

A direct application of the previous methodology even ensures partial complexity analysis, as we get upper bounds in terms of models size, and protocol synthesis for $\mathcal{L}_{\mathsf{EL}}$ and $\mathsf{CTL}^*\mathsf{K}_n$ [AMP14].

Going back to [DT16], when compounding knowledge relations to regular automatic trees, what we could call $\mathsf{chain} - \mathsf{MSOK}_n$ (we leave out some additional predicates for the examined logics to alleviate notations) is more expressive than several other logics, including linear-time epistemic μ-calculus $\mu\mathsf{CTLK}_n$ — which subsumes $\mathsf{CTL}^*\mathsf{K}_n$—, which guarantees they are decidable too in such structures.

It turns out that some subclasses of DEL structures are regular automatic trees, namely 0-DEL structures (which means the propositional events case) and those whose events are exclusively public announcements; this list may not be exhaustive.

5 Research challenges

5.1 Open problems in DEL

The charts below sum up our current understanding of epistemic planning; we separate the classes of instances in two, according to whether accessibility relations are involved or not, as this consideration results in instances of different natures (propositions versus model structure).

	no postcondition	with postconditions
$md = 0$	PSPACE-complete	Decidable
$ma = 0$	?	Undecidable
$ma \geq 1$	Undecidable	Undecidable

Table 1: Epistemic planning: state of the art (with respect to propositions)

Frame considered	*Decidable classes*
$\mathcal{W}$ with 1 agent	S5, KD45, K45
$\mathcal{E}$	public announcement
$\mathcal{E}$ under $\mathcal{L}_{\mathsf{KC}}$	public, almost-mutex transitive, possibly oblivious, functional dependent, binary, triple dichotomous

Table 2: Epistemic planning: state of the art (with respect to frames)

One particular situation that needs attention is the 0-alternation k-depth case, where (modal) alternation (ma) refers to the number of switches between the modality K and its dual $\neg K \neg$ (often written $\hat{K}$), while modal depth (md) relates to the maximum effective number of nested modal operators K.

Decidability (and complexity should the affirmative stand) for preconditional modal depth at most 1 and no postconditions has yet to be contemplated. [CMS16]

As for the complexity class for propositional events, we are aware of its lying somewhere between PSPACE and EXPTIME; this categorisation could be refined.

Another subject worth of consideration would be synthesising maximal permissive[6] epistemic protocols, a notion that regards safety properties in particular [AMP14].

On a higher level, even though the latest attempts have not been pointless, we can discern limits coming up hindering progress.

We have seen regular automatic trees' function; nevertheless, as generic as this setting is, it does not assure that the converse be true: only exposing a DEL structure as not being a regular automatic tree prevents us from jumping to conclusions. For instance, the non-ontic case (even with only one agent) [DT16, Theorem 3] as well as mixing public announcements with propositional conditions [DT16, Proposition 13] do not necessarily bring about a regular automatic tree. Our undertaking is to shed light on DEL structures: how much can automata capture, i.e. what DEL structures can be explained as automatic structures? If not all, then is there another way to account for them?

5.2 Prospective methodology

5.2.1 Finitely-presented structures

On regular automatic trees specifically, as revealed [DT16] through (two) counter-examples serving as proof for non-regularity of the address language (i.e. the set of finite histories), non-rational algebraic domains can arise; this could steer us towards resorting to tree-automatic structures. (Notice trees appear naturally in such a context, as ETL relies on it [Mau14, AMP14].) Actually, as the orthodox address encoding cannot portray histories subtly enough, we feel justified in swapping the encoding basis in order to meet higher standards.

More generally, we wish to look into ω-words, trees, and ω-trees as the new foundational elements acting as automata input; in particular, with respect to encoding, we may question the relation this mapping maintains with its image. Indeed, we should note that a host of theorems heretofore displayed [Blu99, DT16, Rub08] for finite words hold on condition of an injective encoding, which is no longer secured in the ω case (seeing as we cannot express an analogue for the alphabetical order without length), both for words and trees — although it is still valid for *finite* trees.

[Blu99] has some considerations to offer as a starting point, such as the greater expressiveness of these types of elements (and even a hierarchical classification picturing other general classes), or a link between countable ω-automatic presentations and finite-word automatic presentations under injectivity hypothesis.

Should automatic representations fail as a characterisation instrument for DEL structures, we could potentially draw inspiration from this track to design a new sort of finite representation.

5.2.2 Infinite structures classification

Exploring beyond, we would like to delimit DEL structures amidst the class of infinite structures so as to enhance our grasp of their nature. From this perspective, we could resolve to the alternative of catching DEL structures by discovering relationships to already established classes of structures and whether it could hint at decidability results.

[6] A notion referring to synthesising a strategy whose policy cannot be relaxed without violating the expected property.

As an illustration, the Caucal hierarchy [Cau02, Cau03] may outline a trail to pursue because of its salient property of MSO-decidability (shared by all of its enclosed structures) and link to (pushdown) automata [CW03].

To give a quick insight, the Caucal hierarchy is a classification starting with finite trees and then iterating alternately MSO-interpretations and "unfoldings"; this procedure comes to a class of graphs.

Even a negative result would help clarifying the status of DEL structures — as in the event that MSO is too strong a logic compared to some undecidable fragments of epistemic planning.

References

[AMP14] Guillaume Aucher, Bastien Maubert, and Sophie Pinchinat. Automata Techniques for Epistemic Protocol Synthesis. In Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi, editors, *Proceedings 2nd International Workshop on Strategic Reasoning, SR 2014, Grenoble, France, April 5-6, 2014.*, volume 146 of *EPTCS*, pages 97–103, 2014.

[Aum99] Robert J. Aumann. Interactive epistemology I: Knowledge. *International Journal of Game Theory*, 1999.

[BA11] Thomas Bolander and Mikkel Birkegaard Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–33, 2011.

[Ber13] Jean Berstel. *Transductions and context-free languages*. Springer-Verlag, 2013.

[BG00] Achim Blumensath and Erich Grädel. Automatic Structures. In *Proceedings of 15th IEEE Symposium on Logic in Computer Science LICS 2000*, pages 51–62. IEEE, 2000.

[Blu99] Achim Blumensath. Automatic Structures. Diploma thesis, RWTH-Aachen, 1999.

[BMS98] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements and common knowledge and private suspicions. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98), Evanston, IL, USA, July 22-24, 1998*, pages 43–56, 1998.

[Cau02] Didier Caucal. *On Infinite Terms Having a Decidable Monadic Theory*, pages 165–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[Cau03] Didier Caucal. On infinite transition graphs having a decidable monadic theory. *Theor. Comput. Sci.*, 290(1):79–115, January 2003.

[CMS16] Tristan Charrier, Bastien Maubert, and François Schwarzentruber. On the Impact of Modal Depth in Epistemic Planning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1030–1036, 2016.

[CW03] Arnaud Carayol and Stefan Wöhrle. *The Caucal Hierarchy of Infinite Graphs in Terms of Logic and Higher-Order Pushdown Automata*, pages 112–123. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[DA16] Sandra Devin and Rachid Alami. An implemented theory of mind to improve human-robot shared plans execution. In *The Eleventh ACM/IEEE International Conference on Human Robot Interation, HRI 2016, Christchurch, New Zealand, March 7-10, 2016*, pages 319–326, 2016.

[DT16] Gaëtan Douéneau-Tabot. Propriétés régulières des arbres. Internship report, École Normale Supérieure de Cachan, 2016.

[EFT13] H-D Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical logic*. Springer Science & Business Media, 2013.

[Eme90] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol. B*, chapter 16, pages 995–1072. 1990.

[HT87] T. Hafer and W. Thomas. Computation tree logic CTL* and path quantifiers in the monadic theory of the binary tree. pages 269–279, July 1987.

[Mau14] Bastien Maubert. *Logical foundations of games with imperfect information: uniform strategies*. PhD thesis, Université Rennes 1, 2014.

[RS97] Grzegorz Rozenberg and Arto Salomaa, editors. *Handbook of Formal Languages, Vol. 3: Beyond Words*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[Rub08] Sasha Rubin. Automata Presenting Structures: A Survey of the Finite String Case. *Bulletin of Symbolic Logic*, 14(2):169–209, 006 2008.

[vBGHP09] Johan van Benthem, Jelle Gerbrandy, Tomohiro Hoshi, and Eric Pacuit. Merging frameworks for interaction. *Journal of Philosophical Logic*, 38(5):491–526, 2009.

[vBGP07] Johan van Benthem, Jelle Gerbrandy, and Eric Pacuit. Merging frameworks for interaction: DEL and ETL. In *TARK*, 2007.

[VDvDHK07] Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.

[YWL13] Quan Yu, Ximing Wen, and Yongmei Liu. Multi-Agent Epistemic Explanatory Diagnosis via Reasoning about Actions, 2013.