Master research Internship

Internship report

Expressiveness of single-rooted DEL presentations

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 finding a finite representation for the infinite structure; 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.

We have endeavoured to explore DEL structures' expressiveness, both in terms of languages and machine simulation, so as to try and classify them, in order to better delimit the expressive power of several DEL fragments. This paves the way for new attempts at solving some specific cases of the planning problem, left open for quite a while.

Contents

1 Introduction **1**
- 1.1 Research subject . . . 1
- 1.2 Contribution . . . 2

2 State of the art **3**
- 2.1 DEL framework . . . 3
  - 2.1.1 Epistemic logic and epistemic models . . . 3
  - 2.1.2 Event models . . . 4
  - 2.1.3 DEL presentations and DEL structures . . . 5
- 2.2 Automatic structures . . . 8
  - 2.2.1 Definition . . . 8
  - 2.2.2 Decidability of first-order logic and extensions . . . 9
- 2.3 The epistemic planning problem . . . 11
  - 2.3.1 Definition of the problem . . . 11
  - 2.3.2 A preliminary result on epistemic planning problem . . . 12
  - 2.3.3 Results with *ad hoc* proofs . . . 16
  - 2.3.4 Results based on automatic structures . . . 16
- 2.4 Research challenges . . . 17
  - 2.4.1 Open problems in DEL . . . 17
  - 2.4.2 Prospective methodology . . . 18

3 DEL presentations and language theory **18**
- 3.1 Regular languages . . . 19
  - 3.1.1 Non-ontic PDEL . . . 19
  - 3.1.2 Ontic PDEL . . . 19

3.2 Context-free languages . . . 23
3.2.1 Non-DEL-presentable . . . 23
3.2.2 DEL-presentable . . . 26
3.3 Context-sensitive languages . . . 27

4 DEL presentation and machine simulation **28**
4.1 Finite automata with $DEL(\mathcal{L}_K^{0,0}, \texttt{ontic})$ . . . 29
4.2 Pushdown automata with $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$ . . . 29
4.2.1 Formalism . . . 29
4.2.2 Simulation . . . 29
4.2.3 Configuration graphs . . . 31
4.3 Queue automata with $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$ . . . 32
4.3.1 Formalism . . . 33
4.3.2 Simulation . . . 33

5 Conclusion **35**
5.1 New results summary . . . 35
5.2 Future work . . . 35

1 Introduction

1.1 Research subject

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.

The (non-)automatic nature of several classes of DEL structures is already known, through the facts listed hereunder:

- for any number of agents, if the conditions are propositional and the event model is ontic, then the resulting DEL structure is automatic [Mau14, DT16, AMP14], which implies that the planning problem is decidable;
- for at least three agents, if modal depth is limited to 1 and the event model is ontic, then planning is undecidable [BA11], and by contraposition, the DEL structure is (generally) not automatic;
- for one agent, if alternation depth is at least 1 and the event model is non-ontic, then planning is once again undecidable [CMS16], which entails for the same reason that the corresponding DEL structure may well be non-automatic.

We have the following results depicted in Table 1 as to whether a DEL structure is automatic, relatively to pre- and postconditions; "md" stands for *modal depth*, while "ad" means *alternation depth* (see Definitions 3 and 4).

	ad=0				
pre / **post**	**md = 0**	**md ≤ 1**	**md ≤ 2**	···	**ad ≥ 1**
non-ontic	YES	?	?	?	NO
ontic	YES	NO	NO	NO	NO

Table 1: The double arrows convey implication regarding (un)decidability, while the dashed ones specify a polynomial reduction of the planning problem (see Section 2.3.2).

It should be noted that the very same table stays unchanged if we ask for (un)decidability of the epistemic planning problem; this suggests a strong link between the two notions, even though automaticity is only sufficient rather than necessary in general.

The non-ontic modal depth 1 case of special importance: should it be undecidable, then we can conclude the same for the remaining cases; otherwise, we are brought back to checking non-ontic modal depth 2.

At the same time, this seems to be a particularly difficult question, as we may exhibit a non-regular trace language while its DEL structure is (tree-)automatic; trying out some encodings (especially with finite trees) has revealed serious obstacles, although this process is inconclusive.

1.2 Contribution

In this report, as difficulties arise for defining the scope of the epistemic planning problem, especially for the above-mentioned case, we have intended to study the specified objects — namely, *DEL presentations* of infinite structures, called *DEL structures* — from the viewpoint of expressiveness, according to two orthogonal directions:

1. the concept of regular trees [DT16] suggests looking at trace **languages**, that are the class of languages that can be captured as the set of traces of the DEL structure induced by a DEL presentation, see Section 2.1;

2. reduction to a 2-counter machine [CMS16] conjures looking at what **machines** can be simulated through various classes of DEL structures.

Indeed, on the one hand, the simpler the trace language, the "closer" the structure to a description by some automaton; looking at the language grants a reckoning of the potential distance from the existence of an automatic presentation, hence the lack of any classical tool to solve the planning problem in the event that it is decidable. What's more, we can effectively generate at least one non-regular context-free language in this way. This gives rise to a first problem (defined formally in

Section 3): given some class of languages, what class of DEL structures may capture it through its trace languages? We show that, starting from a single world and using specific fragments of DEL, we can produce all prefix-closed regular languages, some prefix-closed non-regular context-free languages but not all, and perhaps some prefix-closed non-context-free context-sensitive languages. Notice that we will only judge those languages that are closed by prefix, as all trace languages have this property; this is a strong constraint compared to the Chomsky hierarchy.

On the other hand, the more complex the simulated machines, the more intrinsically difficult the epistemic planning problem: for instance, when possible, simulating 2-counter machines (equivalent to Turing machines) allowed to conclude that the planning problem is undecidable [CMS16]. We are aiming at reproducing the machine's execution tree as a DEL structure, which is our second problem (a formal definition can be found in Section 4). Such is the motivation for pinpointing DEL structures as global objects and classifying them by finding out a hierarchy; we are also interested in exploring the interplay between this hierarchy and the Chomsky hierarchy. We manage to simulate finite automata, pushdown automata and even queue automata under certain assumptions regarding which DEL fragment is used, and show a limit of propositional DEL's simulation power.

We start in Section 2 with a look at standard tools to enable a sufficient modelling of the problems, namely DEL presentations, DEL structures and epistemic temporal logic, introduce the theory of automatic structures with the particular intent of applying them to DEL structures, and then state the epistemic planning and protocol synthesis problems more formally (which gives us the opportunity to state and prove a first result on planning problem reduction in Subsection 2.3.2); we then explore single-rooted DEL's language expressiveness in Section 3; after that, we consider machine simulation in Section 4; finally, we sum up our main results and discuss our research prospects in Section 5.

2 State of the art

2.1 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.

2.1.1 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.

Definition 1 (Epistemic models)**.** *An* epistemic model *is a tuple* $\mathcal{M} = (W, (R_a)_{a \in \mathsf{Ag}}, V)$ *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* $\mathcal{M}$ *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 $\mathcal{M}$ may be distinguished (to highlight w as the "actual" world), in which case we note the resulting *pointed epistemic model* as $(\mathcal{M}, w)$.

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:

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

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.

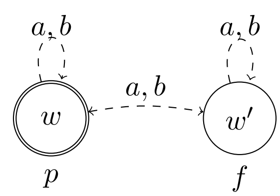


Figure 1: An epistemic model with two agents (a and b); the pointed world w is circled. Here, neither a nor b can distinguish between both worlds, while the real world is w.

2.1.2 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 *post-condition*. 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.

Definition 2 (Event models)**.** *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* $\overset{a}{\hookrightarrow}$ *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).

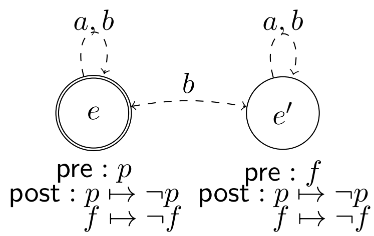


Figure 2: An event model with atomic propositions p and f, and simple preconditions and postconditions (the latter in matching form). When event e or e' happens, a knows which one did; quite the opposite, b cannot tell. Be wary of differing preconditions and of the "switching" effect postconditions have.

2.1.3 DEL presentations and DEL structures

In order to characterize relevant fragments of DEL, we will need the following concepts of modal depth and alternation depth Definitions 3 and 4.

Definition 3 (Modal depth)**.** *The* modal depth *of an epistemic formula* φ*, written* $md(\varphi)$*, is the maximum effective number of nested modal operators* K *in* φ*.*

Definition 4 (Alternation depth)**.** *The* alternation depth *of an epistemic formula* φ*, written* $ad(\varphi)$ *is number of switches between the modality* K *and its dual* $\neg K \neg$ *(often written* $\hat{K}$*) in* φ*.*

We now restrict the syntax of the formulas and collect a family of logics.

Definition 5 (Epistemic logic fragment)**.** *For every natural numbers* $n \geq m \geq 0$*, we let* $\mathcal{L}_K^{m,n}$ *denotes the fragment of* $\mathcal{L}_{\mathsf{K}}$ *of modal depth* m *and alternation depth* n*.*

This allows us to specify particular classes of event models.

Definition 6. *An event model whose postconditions are all required to be "ineffective" is called* non-ontic *(or* purely epistemic*), otherwise it is called* ontic*.*

Several event types of interest have been spotted [YWL13, BMS98]; we do not review them all, although some special cases have been handled in Sections 2.3.3 and 2.3.4[1], an only take a common example for the sake of illustration: 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 $\mathsf{pre}(e)$;

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

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

Definition 7 (Update product). *If $w_0 \not\models \mathsf{pre}(e_0)$, then the update product is undefined; otherwise,*

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

- $W^{\otimes} = \{(w, e) \in W \times E \mid (\mathcal{M}, w) \models \mathsf{pre}(e)\}$ *(each world satisfying an event's precondition generates a new world through that event);*
- $\forall a \in \mathsf{Ag}, \forall((w, e), (w', e')) \in (W^{\otimes})^2, (w, e) R_a^{\otimes} (w', e') \iff (w \overset{a}{\leadsto} w' \wedge e \overset{a}{\leadsto} e')$ *(agent a conflates new worlds iff he conflated both the associated old worlds and events);*
- $\forall (w, e) \in W^{\otimes}, V^{\otimes}(w, e) = \{p \in \mathsf{AP} \mid (\mathcal{M}, w) \models \mathsf{post}(e)(p)\}$ *(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.

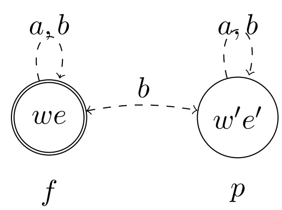


Figure 3: The update product of the two preceding models: now, b is the only one unable to judge which is the correct world. Atomic propositions' truth values have also been "swapped".

By iterating the previous operation, starting with $(\mathcal{M}, w)$ and applying iteratively the event model $(\mathcal{E}, e)$, we get a (potentially infinite) sequence of epistemic models $(\mathcal{M}\mathcal{E}^n)_{n \in \mathbb{N}}$, each of which has $we_0 \dots e_n$ as its singled out world, and all its other worlds are also in $\mathcal{M}\mathcal{E}^n$. 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 $\mathcal{M}\mathcal{E}^*$; a sequence of such words is called a *history* (or *trace*) 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.

Definition 8 (Histories / Traces). *We note* $\mathrm{Hist}(\mathcal{M}\mathcal{E}^*)$ *the set of histories:* $\mathrm{Hist}(\mathcal{M}\mathcal{E}^*) \subseteq (2^{\mathsf{AP}} \times W)(2^{\mathsf{AP}} \times E)^*$.

Whence we define $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{M}\mathcal{E}^*)$ as the set of finite traces of events that can be obtained starting from w.

Definition 9 (Event histories / Event traces). *The set of event traces* $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{M}\mathcal{E}^*)$ *is obtained from* $\mathrm{Hist}(\mathcal{M}\mathcal{E}^*)$ *through the projection* $\left| \begin{array}{l} 2^{\mathsf{AP}} \times E \to E \\ (P, e) \mapsto e \end{array} \right.$ *(we "forget" valuations and initial worlds):*

$$\mathrm{Hist}_{\mathrm{ev}}(\mathcal{M}\mathcal{E}^*) = \bigcup_{n \in \mathbb{N}} \left\{ e_1 \dots e_n \;\middle|\; \begin{array}{c} \exists w \in W,\ \exists (P_i)_{0 \le i \le n} \in (2^{\mathsf{AP}})^{n+1}, \\ (P_0, w)(P_1, e_1) \dots (P_n, e_n) \in \mathrm{Hist}(\mathcal{M}\mathcal{E}^*) \end{array} \right\}.$$

Remark: in particular, $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{M}\mathcal{E}^*)$ *does not take into account the initial worlds; we demand this technical detail for convenience reasons, for unlike events, worlds cannot be iterated.*

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.

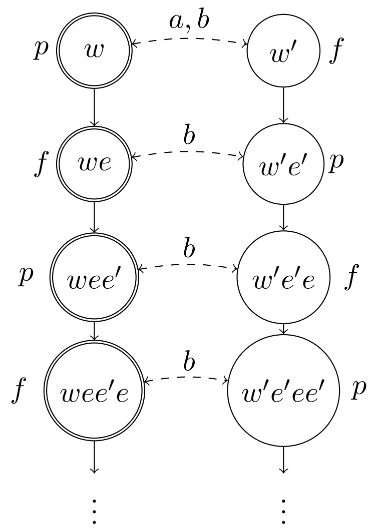


Figure 4: Glimpse of a simple DEL structure generated by the previous models; solid lines realise actions/time while dashed lines substantiate knowledge *(self loops were overlooked)*. There is an alternation for atoms; branching is notably absent — although we do succeed in generating an infinite pattern — 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 $(\mathcal{M}, w)$ is to hold iff $(\mathcal{M}, w) \models \mathsf{pre}(e)$ and $(\mathcal{M}, w) \otimes (\mathcal{E}, e) \models \varphi$ [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 2.3.1) 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.

Definition 10 (DEL presentations)**.** *Given an infinite structure S, a* DEL presentation *of S is a pair $(\mathcal{M}, \mathcal{E})$ where $\mathcal{M}$ is an epistemic model and $\mathcal{E}$ an event model such that $S \simeq \mathcal{M}\mathcal{E}^*$ (structure isomorphism).*

Depending on the nature of preconditions and postconditions of event $\mathcal{E}$, several subtypes of presentations can be defined. We will denote by $DEL(\mathcal{L}_K^{m,n}, \texttt{non-ontic})$ (resp. $DEL(\mathcal{L}_K^{m,n}, \texttt{ontic})$ the subclass of DEL presentations $(\mathcal{M}, \mathcal{E})$ where event model $\mathcal{E}$'s preconditions are in $\mathcal{L}_K^{m,n}$ and $\mathcal{E}$ is non-ontic (resp. ontic).

We introduce the specific sort of DEL presentations we will be working on in Sections 3 and 4, namely, *single-rooted DEL presentations.*

Definition 11 (Single-rooted DEL presentations)**.** *We call w some arbitrary element (world).*

For a set of agents Ag and $V \subseteq 2^A P$, see as a valuation $V : \{w\} \to \mathsf{AP}$, we define the epistemic model $\mathcal{S}_V^{\mathsf{Ag}} = (\mathcal{S}^{\mathsf{Ag}}, V)$, where $\mathcal{S}^{\mathsf{Ag}}$ is the frame[2] $(\{w\}, (\{w, w\})_{i \in Ag})$.

A single-rooted DEL presentation *with event model $\mathcal{E}$ is the DEL presentation $(\mathcal{S}_V^{\mathsf{Ag}}, \mathcal{E})$.*

2.2 Automatic structures

2.2.1 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.

[2] we borrow the standard vocabulary from modal logic, see for example [BDRV02].

We generally stick to finite words here, as the known results to be presented (see Sections 2.3.3 and 2.3.4) 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[3] 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 $R_i^{\mathcal{A}}$ 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 $\mathcal{A}$.

2.2.2 Decidability of first-order logic and extensions

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 2.3.4.

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.

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

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 1. *$\exists^\omega$ preserves regularity in the finite case, and we can conclusively affirm $\mathsf{FO}[\exists^\omega]$ 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 12 (Interpretations)**.** *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 $\mathcal{M}_k$ (see just below for a definition), since a powerful theorem asserts such a link to be the defining characteristic[4]; 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. $\mathcal{M}_k = (\Sigma^*, (\sigma_a)_{a\in\Sigma}, \preceq_p, \mathrm{el})$: $\Sigma = [\![0, k-1]\!]$, 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

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

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 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[5] 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 $\mathcal{M}_n$ on page 10: when looking at the full list of predicates used to extend chain-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.

2.3 The epistemic planning problem

Given a DEL presentation $((\mathcal{M}, w), (\mathcal{E}, e))$, the *epistemic planning problem* consists in determining whether some $\mathcal{L}_{\mathsf{EL}}$ objective formula φ is reachable via some history.

2.3.1 Definition of the problem

Definition 13 (Epistemic planning problem)**.**

- *Input:* $(\mathcal{M}, \mathcal{E}, \varphi)$ *with* $\varphi \in \mathcal{L}_{\mathsf{EL}}$.
- *Output: "yes" if there exists* h *in* $\mathcal{M}\mathcal{E}^*$ *such that* $(\mathcal{M}\mathcal{E}^*, h) \models \varphi$*, otherwise "no".*

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

We write $(\mathcal{M}, \mathcal{E}, \varphi) \in \mathsf{Planning}$ *when* $(\mathcal{M}, \mathcal{E}, \varphi)$ *is a positive instance.*

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).

2.3.2 A preliminary result on epistemic planning problem

We are not aware of any statement of the kind of Theorem 2.3.2 in the literature although it drastically reduces the class of DEL presentations one should focus on from a theoretical point of view

Theorem 1. *Let* $\mathcal{M}$ *be a Kripke model on* AP *and* $\mathcal{E} = (E_0, E, (R_a)_{a\in\mathsf{Ag}}, \mathsf{pre}, \mathsf{post})$ *be an event model on* AP *too. Then there exists* $\mathcal{E}' = (E'_0, E', (R'_a)_{a\in\mathsf{Ag}}, \mathsf{pre}', \mathsf{post}')$ *on* $\mathsf{AP}' \supseteq \mathsf{AP}$ *with formulas of modal-depth* ≤ 1 *only,* χ *on* $\mathsf{AP}'\backslash\mathsf{AP}$*, such that for all formulas* φ*:*

$$(\mathcal{M}, \varphi, \mathcal{E}) \in \mathsf{Planning} \text{ iff } (\mathcal{M}, (\chi \wedge \varphi), \mathcal{E}') \in \mathsf{Planning}$$

Moreover, the event model $\mathcal{E}'$ *has a size*[6] *polynomial in the size of* $\mathcal{E}$*.*

Proof. Let n be the maximal modal depth of the formulas appearing as preconditions in $\mathcal{E}$.

We define $\mathcal{E}'$ as $\mathcal{E}' = \mathcal{E}_{begin} \,\dot\cup\, \mathcal{E}_1 \,\dot\cup\, \ldots \,\dot\cup\, \mathcal{E}_n \,\dot\cup\, \mathcal{E}_{end}$, meant to mimic the occurrence of $\mathcal{E}$ by $n+2$ occurrences of $\mathcal{E}'$, as stated in Proposition 1 later. We first give the intuition of this contruction as follows.

- The first execution of $\mathcal{E}'$ uses $\mathcal{E}_{begin}$ and just initializes the execution by assigning proposition p_1 to true.
- The $i+1^{th}$ execution of $\mathcal{E}'$ for $i \in \{1, \ldots, n\}$ uses $\mathcal{E}_i$ and computes the value of sub-formulas of $\mathcal{E}$ of modal depth i. For instance, if $\mathsf{pre}(e) = \Diamond_a\Diamond_b p \wedge \Box_a q$ and $i = 1$, the values of $\Diamond_b p$ and $\Box_a q$ are computed. The values of such formulas ψ are stored into new atomic propositions p_ψ.
- The $n+2^{th}$ execution of $\mathcal{E}'$ uses $\mathcal{E}_{end}$. It mimics $\mathcal{E}$, replacing any modal formula ψ with p_ψ.

To force the order of execution, we introduce $n+1$ atomic propositions, $p_1, \ldots, p_{n+1}$:

- Formula $\bigwedge_{i=1}^{n+1} \neg p_i$ is true iff $\mathcal{E}_{begin}$ is to be executed next.
- For every $i \in \{1, \ldots, n\}$, we have p_i is true iff $\mathcal{E}_i$ is to be executed next.
- p_{n+1} is true iff $\mathcal{E}_{end}$ is to be executed next.

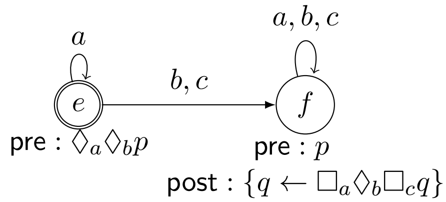
[6]The size is its mere description.

Therefore let $Sub(\mathcal{E})_{md\geq 1}$ be the set of all modal subformulas of $\mathcal{E}$, we define:

$$\mathsf{AP}' = \mathsf{AP} \cup \{p_\psi, \psi \in Sub(\mathcal{E})_{md\geq 1}\} \cup \{p_1, \dots, p_{n+1}\} \text{ and } \chi = \bigwedge_{i=1}^{n+1} \neg p_i$$

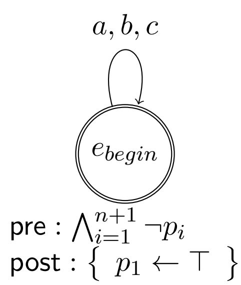
We now define $\mathcal{E}'$, the event model with modal depth at most 1 that mimics event model $\mathcal{E}$, as stated by Theorem .

We will use the following event model $\mathcal{F}$ to illustrate the construction (notice that $\mathcal{F}$ has alternation 2, because of event f).



- $\mathcal{E}_{begin} = (E_{0,begin}, E_{begin}, (R_{a,begin})_{a\in\mathsf{Ag}}, \mathsf{pre}_{begin}, \mathsf{post}_{begin})$ with:
  - $E_{0,begin} = E_{begin} = \{e_{begin}\}$.
  - $\mathsf{pre}_{begin}(e_{begin}) = \bigwedge_{i=1}^{n+1} \neg p_i$.
  - $R_{a,begin} = \{(e_{begin}, e_{begin})\}$.
  - $\mathsf{post}_{begin}(e_{begin}) = \{\ p_1 \leftarrow \top\ \}$

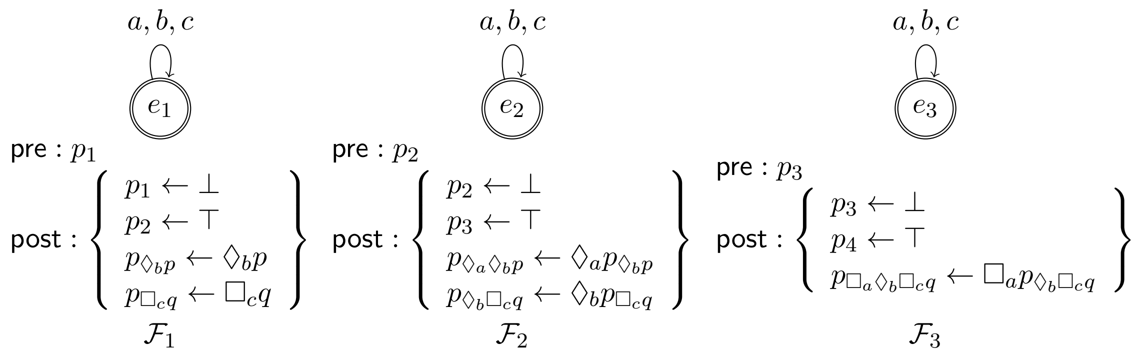
  For instance, $\mathcal{F}_{begin}$ is:



- $\mathcal{E}_i = (E_{0,i}, E_i, (R_{a,i})_{a\in\mathsf{Ag}}\mathsf{pre}_i, \mathsf{post}_i)$ with:
  - $E_{0,i} = E_i = \{e_i\}$.
  - $\mathsf{pre}_i(e_i) = p_i$.
  - $\mathsf{post}_i(e_i) = \left\{ \begin{array}{l} p_i \leftarrow \bot \\ p_{i+1} \leftarrow \top \\ p_\psi \leftarrow \psi' \end{array} \right\}$

    for all formulas ψ with $md(\psi) = i$. ψ' is defined by replacing the modal sub-formulas γ of ψ with $md(\psi) = i - 1$ by p_γ.

For instance, $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ are defined as follows:



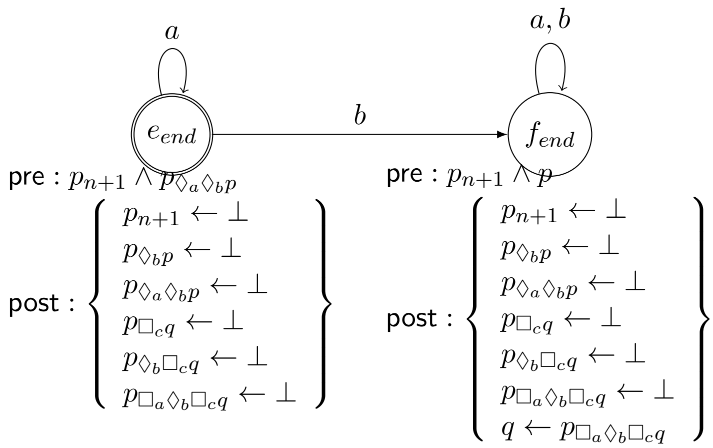
pre : p_1 post : $\left\{\begin{array}{l} p_1 \leftarrow \bot \\ p_2 \leftarrow \top \\ p_{\Diamond_b p} \leftarrow \Diamond_b p \\ p_{\Box_c q} \leftarrow \Box_c q \end{array}\right\}$ $\mathcal{F}_1$

pre : p_2 post : $\left\{\begin{array}{l} p_2 \leftarrow \bot \\ p_3 \leftarrow \top \\ p_{\Diamond_a \Diamond_b p} \leftarrow \Diamond_a p_{\Diamond_b p} \\ p_{\Diamond_b \Box_c q} \leftarrow \Diamond_b p_{\Box_c q} \end{array}\right\}$ $\mathcal{F}_2$

pre : p_3 post : $\left\{\begin{array}{l} p_3 \leftarrow \bot \\ p_4 \leftarrow \top \\ p_{\Box_a \Diamond_b \Box_c q} \leftarrow \Box_a p_{\Diamond_b \Box_c q} \end{array}\right\}$ $\mathcal{F}_3$

- $\mathcal{E}_{end} = (E_{0,end}, E_{end}, (R_{a,end})_{a \in \mathsf{Ag}}, \mathsf{pre}_{end}, \mathsf{post}_{end})$ with:
  - $E_{end} = \{e_{end}, e \in E\}$;
  - $E_{0,end} = \{e_{end}, e \in E_0\}$;
  - $R_{a,end} = \{(e_{end}, f_{end}), (e, f) \in R_a\}$;
  - $\mathsf{pre}_{end}(e_{end}) = \left\{\begin{array}{l} p_{n+1} \wedge p_{\mathsf{pre}(e)} \text{ if } \mathsf{pre}(e) \text{ is modal} \\ p_{n+1} \wedge \mathsf{pre}(e) \text{ otherwise.} \end{array}\right\}$;
  - $\mathsf{post}_{end}(e_{end}) = \left\{\begin{array}{l} p_{n+1} \leftarrow \bot \\ p_\psi \leftarrow \bot \text{ for all } \psi \in Sub(\mathcal{E})_{md \geq 1} \\ p \leftarrow \left\{\begin{array}{l} p_{\mathsf{post}(e,p)} \text{ if } \mathsf{post}(e,p) \text{ is modal} \\ \mathsf{post}(e,p) \text{ otherwise} \end{array}\right. \quad \text{for } p \in \mathsf{AP}. \end{array}\right\}$.

For instance, $\mathcal{F}_{end}$ is defined as follows:



pre : $p_{n+1} \wedge p_{\Diamond_a \Diamond_b p}$ post : $\left\{\begin{array}{l} p_{n+1} \leftarrow \bot \\ p_{\Diamond_b p} \leftarrow \bot \\ p_{\Diamond_a \Diamond_b p} \leftarrow \bot \\ p_{\Box_c q} \leftarrow \bot \\ p_{\Diamond_b \Box_c q} \leftarrow \bot \\ p_{\Box_a \Diamond_b \Box_c q} \leftarrow \bot \end{array}\right\}$

pre : $p_{n+1} \wedge p$ post : $\left\{\begin{array}{l} p_{n+1} \leftarrow \bot \\ p_{\Diamond_b p} \leftarrow \bot \\ p_{\Diamond_a \Diamond_b p} \leftarrow \bot \\ p_{\Box_c q} \leftarrow \bot \\ p_{\Diamond_b \Box_c q} \leftarrow \bot \\ p_{\Box_a \Diamond_b \Box_c q} \leftarrow \bot \\ q \leftarrow p_{\Box_a \Diamond_b \Box_c q} \end{array}\right\}$

It is clear that every preconditions in $\mathcal{E}$ has a model depth at most one.

We now prove the correctness of the construction.

Proposition 1. *For any Kripke model $\mathcal{M}$ and any event model $\mathcal{E}$ on AP with maximal modal depth n, $\mathcal{M} \otimes \mathcal{E}'^{n+2}$ and $\mathcal{M} \otimes \mathcal{E}$ are isomorphic.*

Proof. $\mathcal{M}$, when seen as a Kripke model on AP', is such that $p_1, \ldots, p_{n+1}$ are all false in every world of $\mathcal{M}$. Therefore, $\mathcal{M} \otimes \mathcal{E}' = \mathcal{M} \otimes \mathcal{E}_{begin}$, because any other components of $\mathcal{E}'$ but $\mathcal{E}_{begin}$ require some precondition p_i, and are thus not executable. Next, $\mathcal{E}_1$ is the only executable component of $\mathcal{E}'$ that is executable so $\mathcal{M} \otimes \mathcal{E}' \otimes \mathcal{E}' = \mathcal{M} \otimes \mathcal{E}_{begin} \otimes \mathcal{E}_1$, and so on and so forth. Finally, we obtain that:

$$\mathcal{M} \otimes \mathcal{E}'^{n+2} = \mathcal{M} \otimes \mathcal{E}_{begin} \otimes \mathcal{E}_1 \otimes \cdots \otimes \mathcal{E}_n \otimes \mathcal{E}_{end}.$$

Note that $\mathcal{E}_{begin}, \mathcal{E}_1, \ldots, \mathcal{E}_n$ are single-pointed and that when executed, their precondition is true in all worlds of $\mathcal{M}$. Therefore, $\mathcal{M} \otimes \mathcal{E}_{begin} \otimes \mathcal{E}_1 \otimes \cdots \otimes \mathcal{E}_n$ is isomorphic to $\mathcal{M}$, with the value of atomic propositions of $\mathsf{AP}' \backslash \mathsf{AP}$ changed.

Lemma 1. *For every $1 \leq i \leq n$ and for any $\psi \in Sub(\mathcal{E})_{1 \leq md \leq i}$, we have*

$$\mathcal{M}, w \models \psi \textit{ iff } \mathcal{M} \otimes \mathcal{E}_{begin} \otimes \mathcal{E}_1 \otimes \cdots \otimes \mathcal{E}_n, we_{begin}e_1 \ldots e_i \models p_\psi$$

Proof. to be done by induction on i. ☐

In particular for $i = n$, and for every $\psi \in Sub(\mathcal{E})$,

$$\mathcal{M}, w \models \psi \text{ iff } \mathcal{M} \otimes \mathcal{E}'^{n+1}, we_{begin}e_1 \ldots e_n \models p_\psi$$

Therefore $V(we_{begin}e_1 \ldots e_n) = V(w) \cup \{p_{n+1}\} \cup \{p_\psi \mid \psi \in Sub(\mathcal{E}) \text{ and } \mathcal{M}, w \models \psi\}$.

Now, we can only execute component $\mathcal{E}_{end}$ of $\mathcal{E}'$ which, by construction, is a copy of $\mathcal{E}$ where modal formulas ψ in preconditions are replaced by p_ψ, and which sets proposition p_{n+1} and all propositions p_ψ, where $\psi \in Sub(\mathcal{E})$, to false thereafter. Therefore $\mathcal{M} \otimes \mathcal{E}'^{n+1} \otimes \mathcal{E}_{end} = \mathcal{M} \otimes \mathcal{E}'^{n+2}$ and $\mathcal{M} \otimes \mathcal{E}$ are isomorphic. ☐

Corollary 1. *For any φ on AP, any $\mathcal{M}$ and any $\mathcal{E}$, every plan for $(\mathcal{M}, \chi \wedge \varphi, \mathcal{E}')$ is of the form $(\mathcal{E}_{begin}, e_{begin}), (\mathcal{E}_1, e_1), \ldots, (\mathcal{E}_n, e_n), (\mathcal{E}_{end}, e^1_{end}) \ldots (\mathcal{E}_{begin}, e_{begin}), (\mathcal{E}_1, e_1), \ldots, (\mathcal{E}_n, e_n), (\mathcal{E}_{end}, e^m_{end})$.*

Proof. The proof can be done by induction over m. It will use tha fact that in order to achieve χ, the plan can only either be empty (thus $m = 0$) or end with some $(\mathcal{E}_{end}, e^m_{end})$, such that $e^m \in E$, otherwise some p_i would be true which does not achieve formula $\chi = \bigwedge_{i=1}^{n+1} \neg p_i$. ☐

Combining Proposition 1 and Corollary 1 entails the following.

Proposition 2. *For any φ on AP, any $\mathcal{M}$ and any $\mathcal{E}$,*

$$\begin{gathered}(\mathcal{E}, e^1) \ldots (\mathcal{E}, e^m) \textit{ is a plan for } (\mathcal{M}, \varphi, \mathcal{E}) \\ \textit{iff} \\ (\mathcal{E}_{begin}, e_{begin}), (\mathcal{E}_1, e_1), \ldots, (\mathcal{E}_n, e_n), (\mathcal{E}_{end}, e^1_{end}) \ldots (\mathcal{E}_{begin}, e_{begin}), (\mathcal{E}_1, e_1), \ldots, (\mathcal{E}_n, e_n), (\mathcal{E}_{end}, e^m_{end}) \\ \textit{is a plan for } (\mathcal{M}, (\chi \wedge \varphi), \mathcal{E}).\end{gathered}$$

Theorem 2.3.2 is then an immediate corollary of Proposition 2.

It is easy to establish that the size of $\mathcal{E}'$ is polynomial in the size of $\mathcal{E}$. ☐

We can therefore focus on a subclass of DEL presentations as stated by the following reduction and which justifies the dashed arrows in Table 1.

Corollary 2. *The epistemic planning problem can be polynomially reduced to the epistemic planning problem over DEL presentations in $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$.*

2.3.3 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[7]) 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).

2.3.4 Results based on automatic structures

Returning to the case of propositional events (PDEL), building upon the ascertained bridge between DEL and ETL [vBGHP09], $\mathcal{ME}^*$'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 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

[7] 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.

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 the DEL structures induced by DEL presentations in $DEL(\mathcal{L}_K^{0,0}, \texttt{ontic})$ are regular automatic trees, even extended with events that are exclusively public announcements, are automatic structures; this list may not be exhaustive.

2.4 Research challenges

2.4.1 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).

	non-ontic	ontic
$md = 0$	PSPACE-complete	Decidable
$ad = 0$	?	Undecidable
$ad \geq 1$	Undecidable	Undecidable

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

$(\mathcal{M}, \mathcal{E}, \varphi)$	*Decidable classes*
$\lvert\mathsf{Ag}\rvert = 1$	$\mathcal{M}$ in S5, KD45, K45
$\mathcal{E}$	$\mathcal{E}$ is public announcement
$\mathcal{E}$ under $\mathcal{L}_{\mathsf{KC}}$	several specific frames [YWL13]

Table 3: Epistemic planning: state of the art (with respect to frames)

One particular situation that needs attention is the 0-alternation k-depth case.

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[8] 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

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

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?

2.4.2 Prospective methodology

2.4.2.1 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.

2.4.2.2 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.

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.

3 DEL presentations and language theory

As the planning problem brings (finite) traces of events into play, we would like to capture traces through a more global approach: namely, we aim for a classification of DEL structures based on their sets of traces, seen as languages; this enables us to talk about DEL presentations' expressivity, according to what classes of language they bring about in the Chomsky hierarchy.

More specifically, we often restrict our study to $\mathcal{S}_V^{\mathsf{Ag}}$; thus, we look at $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{S}_V^{\mathsf{Ag}}\mathcal{E}^*)$ where $\mathcal{E}$ is an event model.

A summary

ASSE À PRÉCISER $<$ DEL propositional

1. regular languages (finite-state automaton): $\leq$ DEL propositional, ontic
2. context-free languages: $<$ DEL

   but DEL $\cap$ (CFL $\setminus$Reg) $\neq \varnothing$ DEL modal depth 1, non-ontic

Definition 14 (Trace language problem)**.**

Let us state once again that since any language of histories is closed by prefix, we limit ourselves only to prefix-closed languages.

3.1 Regular languages

We write *PDEL* for the subclasses of DEL presentations where preconditions are in $\mathcal{L}_K^{0,0}$; we enquire PDEL with regard to postconditions.

3.1.1 Non-ontic PDEL

We show that the prefix closure of $(ab)^*$, $\mathrm{Pref}((ab)^*)$, cannot be a trace language.

Theorem 2. *For any non-ontic propositional event model $\mathcal{E}$ and any valuation V to AP,* $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{S}_V^{\mathsf{Ag}}\mathcal{E}) \neq \mathit{Pref}((ab)^*)$*.*

Proof. Let us write $\mathcal{M} = \mathcal{S}_V^{\mathsf{Ag}}$ and $L = \mathrm{Pref}((ab)^*)$; we suppose $\mathcal{E}$ is a non-ontic propositional event model such that $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{M}\mathcal{E}^*) = L$. Let w be $\mathcal{M}$'s world, and let the events be a and b: $E = \{a; b\}$; since knowledge has no repercussion, we can suppose that there is no agent: $\mathsf{Ag} = \varnothing$.

$a \in L$, so we must have $wa \in \mathcal{M}\mathcal{E}^*$. Then, since the model is non-ontic, we have $V(w) = V(wa)$; since $aa \notin L$, we have $(\mathcal{M}\mathcal{E}, wa) \not\models \mathsf{pre}(a)$. Yet $(\mathcal{M}, w) \not\models \mathsf{pre}(a)$, which raises a contradiction. $\square$

3.1.2 Ontic PDEL

We now switch to ontic propositional DEL and achieve the following result.

We match through an explicit construction an arbitrary prefix-closed regular language given in the form of an automaton with a propositional DEL structure $\mathcal{M}\mathcal{E}^*$, which establishes that the trace languages of propositional DEL structures without epistemic relations are exactly the prefix-closed regular languages.

3.1.2.1 Formalism

Let L be a prefix-closed regular language on the (finite) alphabet Σ; it is recognised by a finite automaton $\mathcal{A}$ where any state is final (which caracterises closure by prefix).

Remark: $\mathcal{A}$ is not supposed to be deterministic with a view to generalising to context-free languages, where pushdown automata's determinism is not granted.

$\mathcal{A} = (\Sigma, Q, I, \Delta, F)$ with:

- the set of states is called Q, that of initial states I;
- $\Delta \subseteq Q \times \Sigma \times Q$ (transitions set) and $\delta^* : Q \times \Sigma^* \to 2^Q$ transition function extended from $\delta : Q \times \Sigma \to 2^Q$;
- $F = Q$ (any state is final).

We note $q_1 \xrightarrow{\alpha} q_2$ the existence of the transition from state q_1 towards state q_2 labelled by the letter α of Σ:

$$q_1 \xrightarrow{\alpha} q_2 \overset{\text{def}}{\equiv} (q_1, \alpha, q_2) \in \Delta$$

and $\delta(P, \alpha) = \bigcup_{q \in P} \delta(q, \alpha)$ ($P \mapsto \delta(P, \alpha)$ is the "set-based transition function" associated to the letter α, that is to say, the fuction that returns the image of subset P by $\delta(\cdot, \alpha)$).

We now state the theorem.

Theorem 3. *If L is a prefix-closed rational langage, then there exist an epistemic model $\mathcal{M}$ and a propositional event model $\mathcal{E}$ such that $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{ME}^*) \simeq L$ (monoid isomorphism).*

Since we know moreover [Mau14, Lemma 22] that for any propositional DEL structure $\mathcal{ME}^*$, $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{ME}^*) \in \mathrm{Reg} \cap \mathrm{PrefClosed}$ (where PrefClosed designates the class of prefix-closed languages), we deduce the following corollary.

Corollary 3. *The class of event histories associated to propositional DEL structures is exactly the class of prefix-closed regular languages:*

$$\mathit{PrefixReg} = \{\mathrm{Hist}_{\mathrm{ev}}(\mathcal{ME}^*) \mid (\mathcal{M}, \mathcal{E}) \in \mathit{PDEL}\}$$

3.1.2.2 Construction

We associate to each subset P of Q a fresh atomic proposition P and we let $\mathsf{AP} = 2^Q$.

To each initial state q, we match a world w_q of valuation $\{\{q\}\}$:

$$W = \{w_q \mid q \in I\} \;;\; \forall q \in I, V(w_q) = \{\{q\}\}.$$

To any letter α from Σ, we associate an event e_α:

$$E = \{e_\alpha \mid \alpha \in \Sigma\}.$$

Remark: in fact, rigorously, we could or even should rather write $W = I$ and $E = \Sigma$, but we introduce these notations for more clarity, in order to distinguish well between the automaton $\mathcal{A}$ stricly speaking and the DEL structure.

We then define the events' preconditions and postconditions in the following way: for all α in Σ and all X in 2^Q, denoting $T_\alpha(X)$ the set of predecessors for any state of X by α, namely $\{q \in Q \mid \exists q' \in X, q \xrightarrow{\alpha} q'\}$:

- $\mathsf{pre}(e_\alpha) = \bigvee_{P \cap T_\alpha(F) \neq \varnothing} P$ (another more explicit form is possible[9]);
- $\mathsf{post}(e_\alpha)(P') = \bigvee_{\delta(P,\alpha)=P'} P$.

We do not take any agent: $\mathsf{Ag} = \varnothing$, as the potential accessibility relations do not take part in the propositional case to determine the world language.

3.1.2.3 Example

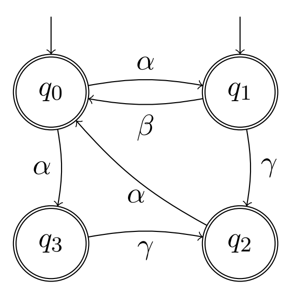


Figure 5: A (non-deterministic) automaton with only final states and recognising the prefix-closed rational language $\mathrm{Pref}((\varepsilon + \beta + \gamma\alpha)(\alpha\beta + \alpha\gamma\alpha)^*\alpha\gamma)$ on alphabet $\{\alpha, \beta, \gamma\}$ (explicitly, $L = \varepsilon + \gamma + (\varepsilon + \beta + \gamma\alpha)(\alpha\beta + \alpha\gamma\alpha)^*(\varepsilon + \alpha + \alpha\gamma)$).

$$\begin{array}{l} W = \{w_{q_0}; w_{q_1}\} \ ; \left\{ \begin{array}{l} V(w_{q_0}) = \{\{q_0\}\} \\ V(w_{q_1}) = \{\{q_1\}\} \end{array} \right. . \\ E = \{e_\alpha; e_\beta; e_\gamma\} \end{array}$$

Notation: so as to lighten the writing, in what follows, for a set X, the notation $\bigvee X$ represents the disjunction $\bigvee_{x \in X} x$.

For preconditions and postconditions, we only consider the subsets $\{q_0\}$, $\{q_1\}$, $\{q_1; q_3\}$ and $\{q_2\}$ (for only them are reachable).

$\mathsf{pre}(e_\alpha) = \bigvee\{P \mid P \subseteq Q \wedge (q_0 \in P \vee q_2 \in P)\} = \{q_0\} \vee \{q_2\} \vee \ldots$
$\mathsf{pre}(e_\beta) = \bigvee\{P \mid P \subseteq Q \wedge q_1 \in P\} = \{q_1\} \vee \{q_1; q_3\} \vee \ldots$
$\mathsf{pre}(e_\gamma) = \bigvee\{P \mid P \subseteq Q \wedge (q_1 \in P \vee q_3 \in P)\} = \{q_1\} \vee \{q_1; q_3\} \vee \ldots$

[9]Indeed, could be replaced with $\mathsf{pre}(e_\alpha) = \bigvee_{P \cap T_\alpha(F) \neq \varnothing} P \wedge \bigwedge_{P \cap T_\alpha = \varnothing} \neg P$, since the Ps are two by two distinct.

$\mathsf{post}(e_\alpha)(\{q_0\}) = \{q_2\} \vee \dots$
$\mathsf{post}(e_\alpha)(\{q_1; q_3\}) = \{q_0\} \vee \dots$
$\mathsf{post}(e_\beta)(\{q_0\}) = \{q_1\} \vee \dots$
$\mathsf{post}(e_\gamma)(\{q_2\}) = \{q_1\} \vee \{q_1; q_3\} \vee \dots$

3.1.2.4 Correctness of the construction

The following lemma ensures

Lemma 2. *For all $P \in 2^Q \setminus \{\varnothing\}$, for all $q_0 \in I$,*

$$w_{q_0} e_{\alpha_1} \dots e_{\alpha_n} \models P \text{ iff } \delta^*(q_0, \alpha_1 \dots \alpha_n) = P.$$

In particular, it is guaranteed that P's uniqueness is an invariant at each step thanks to this property, which could have been written $V(w_{q_0} e_{\alpha_1} \dots e_{\alpha_n}) = \delta^*(q_0, \alpha_1 \dots \alpha_n)$.

Writing the history tacitly assumes its validity with regard to preconditions and postconditions.

Proof. We reason by induction on the history's length n:

- if $n = 0$: $w_{q_0} \models P$ if and only if $P = \{q_0\}$, since $V(w_{q_0}) = \{\{q_0\}\}$; but $\delta^*(q_0, \varepsilon) = \{q_0\}$;
- we assume the property at rank n.
  (a) If $w_{q_0} e_{\alpha_1} \dots e_{\alpha_{n+1}} \models P$, then $w_{q_0} e_{\alpha_1} \dots e_{\alpha_n} \models \mathsf{post}(e_{\alpha_{n+1}})(P)$ *a fortiori*. From $\mathsf{post}(e_{\alpha_{n+1}})(P) = \bigvee_{\delta(X, \alpha_{n+1}) = P} X$, we deduce the existence of X such that $w_{q_0} e_{\alpha_1} \dots e_{\alpha_n} \models X$ with $\delta(X, \alpha_{n+1}) = P$; in particular, $X \neq \varnothing$ since $P \neq \varnothing$, which guarantees that the induction hypothesis applies: $\delta^*(q_0, \alpha_1 \dots \alpha_n) = X$. We thus have $\delta(\delta^*(q_0, \alpha_1 \dots \alpha_n), \alpha_{n+1}) = P$, that is $\delta^*(q_0, \alpha_1 \dots \alpha_{n+1}) = P$.

  (b) Conversely, we suppose $\delta^*(q_0, \alpha_1 \dots \alpha_{n+1}) = P$ with $P \neq \varnothing$; yet $\delta^*(q_0, \alpha_1 \dots \alpha_{n+1}) = \delta(\delta^*(q_0, \alpha_1 \dots \alpha_n), \alpha_{n+1})$.
  From this, we deduce $\delta^*(q_0, \alpha_1 \dots \alpha_n) \neq \varnothing$ in a first phase, which allows for applying the induction hypothesis: $V(w_{q_0} e_{\alpha_1} \dots e_{\alpha_n}) = \delta^*(q_0, \alpha_1 \dots \alpha_n)$; at the same time, we deduce $\delta^*(q_0, \alpha_1 \dots \alpha_n) \cap T_{\alpha_{n+1}}(F) \neq \varnothing$ (since all the states are final), and thus $w_{q_0} e_{\alpha_1} \dots e_{\alpha_n} \models \mathsf{pre}(e_{\alpha_{n+1}})$ (1).
  Besides, we also deduce $\delta^*(q_0, \alpha_1 \dots \alpha_n) \implies \bigvee_{\delta(X, \alpha_{n+1}) = P} X$, hence $w_{q_0} e_{\alpha_1} \dots e_{\alpha_n} \models \mathsf{post}(e_{\alpha_{n+1}})(P)$ (2).
  (1) and (2) together amount precisely to $w_{q_0} e_{\alpha_1} \dots e_{\alpha_{n+1}} \models P$.

□

Lemma 3. $e_{\alpha_1} \dots e_{\alpha_n} \in \mathrm{Hist}_{\mathrm{ev}}(\mathcal{ME}^*)$ *iff there exist $P \in 2^Q \setminus \{\varnothing\}$ and $q_0 \in I$ such that $w_{q_0} e_{\alpha_1} \dots e_{\alpha_n} \models P$.*

Proof. The leftward implication holds, since the right-hand property presupposes that the history under consideration is valid.

From left to right: we reason by induction on the histories' length n.

- If $n = 0$: we both have $\varepsilon \in \text{Hist}_{\text{ev}}(\mathcal{M}\mathcal{E}^*)$ and $w_{q_0} \models \{q_0\}$, so the implication is verified.
- We assume the property for n and we suppose $e_{\alpha_1} \dots e_{\alpha_{n+1}} \in \text{Hist}_{\text{ev}}(\mathcal{M}\mathcal{E}^*)$. We have $e_{\alpha_1} \dots e_{\alpha_n} \in \text{Hist}_{\text{ev}}(\mathcal{M}\mathcal{E}^*)$, thus (by induction hypothesis) there exist a non-empty subset of states P and an initial state q_0 such that $w_{q_0} e_{\alpha_1} \dots e_{\alpha_n} \models P$; according to Lemma 2, we even have $V(w_{q_0} e_{\alpha_1} \dots e_{\alpha_n}) = P$. As moreover $w_{q_0} e_{\alpha_1} \dots e_{\alpha_n} \models \mathsf{pre}(e_{\alpha_{n+1}})$ with $\mathsf{pre}(e_{\alpha_{n+1}}) = \bigvee_{X \cap T_{\alpha_{n+1}}(F) \neq \varnothing} X$, we deduce in particular $P \cap T_{\alpha_{n+1}}(F) \neq \varnothing$.

  As $\mathsf{post}(e_{\alpha_{n+1}})(P') = \bigvee_{\delta(X, \alpha_{n+1}) = P'} X$ for all P', it then suffices to write down $P' = \delta(P, \alpha_{n+1}) = \delta(P \cap T_{\alpha_{n+1}}(F$ to get $w_{q_0} e_{\alpha_1} \dots e_{\alpha_{n+1}} \models P'$ with $P' \neq \varnothing$, since indeed, $P \cap T_{\alpha_{n+1}}(F)$ is non-empty and sent to F by $\delta(\cdot, \alpha_{n+1})$ as a subset of $T_{\alpha_{n+1}}(F)$.

□

These two lemmas now established, we present below a proof of the theorem.

Proof. We define $\varphi : \left| \begin{array}{l} \Sigma \to E \\ \alpha \mapsto e_\alpha \end{array} \right.$, which is a bijection extending to a monoid isomorphism $\Sigma^* \to E^*$; we show $\text{Hist}_{\text{ev}}(\mathcal{M}\mathcal{E}^*) = \varphi(L)$.

$$
\begin{aligned}
w \in L &\iff \exists q_0 \in I, \exists P \in 2^Q \setminus \{\varnothing\}, \delta^*(q_0, w) = P \\
&\overset{\text{lemme1}}{\iff} \exists q_0 \in I, \exists P \in 2^Q \setminus \{\varnothing\}, w_{q_0} \varphi(w) \models P \\
&\overset{\text{lemme2}}{\iff} \varphi(w) \in \text{Hist}_{\text{ev}}(\mathcal{M}\mathcal{E}^*)
\end{aligned}
$$

Consequently, $\text{Hist}_{\text{ev}}(\mathcal{M}\mathcal{E}^*) = \varphi(L)$.

□

3.2 Context-free languages

3.2.1 Non-DEL-presentable

We show that a particular prefix-closed context-free non-regular language does not have any single-rooted DEL presentation.

We consider the two-letter language L defined by $L = \{a^n b^m \mid n \geq m \geq 0\}$; L is closed by prefix and context-free, as it is generated by the following grammar:

$$\{S \to \varepsilon \mid aS \mid aSb$$

L is not regular, as can be shown through the pumping lemma. [HU79]

We now state the main result we intend to prove:

Theorem 4. *Let $L = \{a^n b^m \mid n \geq m \geq 0\}$; then, supposing $|\mathsf{Ag}| = 1$, for any event model $\mathcal{E}$ and any valuation V to AP,* $\text{Hist}_{\text{ev}}(\mathcal{S}_V^{\mathsf{Ag}} \mathcal{E}) \neq L$.

Remark: we use a and b to name the events rather than e_a and e_b for the sake of simplification, since confusion is unlikely.

Here is a representation of a few first nodes of the desired full tree from the root, disregarding transverse edges:

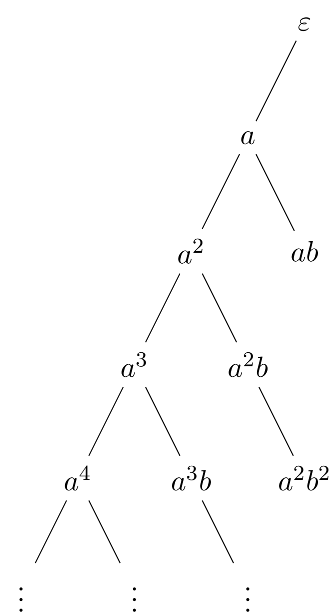


Figure 6: Wanted DEL structure with neglected epistemic relations.

Proof. We suppose that there is a single agent 1, whose associated accessibility relation in $\mathcal{E}$ is R'.

We require $H = L$.

Lemma 4. $R' \neq \varnothing$

Proof. If there are no arrows in the event model, then there is no epistemic link between $\mathcal{M}\mathcal{E}^*$'s nodes, as the update product can only either prune or preserve previous arrows. (Similarly, no loop on w leads to the same situation, which justifies $|\mathsf{Ag}| > 0$.)

Consequently, the problem reduces to the propositional case (as any formula can be replaced with a propositional one); we already know this implies $\mathcal{M}\mathcal{E}^*$ be automatic, and in particular, its event traces form a regular language, while L is not regular. $\square$

Lemma 5. *$\mathcal{E}$ is ontic.*

Proof. Suppose the event model is not ontic (*i.e.* there are no postconditions); since $H = L$, event b can be applied in world wa, but not in w: $\begin{cases} wa & \models & \mathsf{pre}(b) \\ w & \not\models & \mathsf{pre}(b) \end{cases}$.

w and wa are thus not bisimilar; since they have the same valuations (as $\mathcal{E}$ is purely epistemic) and $\mathcal{M}\mathcal{E}$ is reduced to wa, this implies that wa has no loop. By the same argument as in the previous point, the event traces language H' starting from wa would be regular, and H too, because $H = aH'$. Once again, this is incompatible with $H = L$ as L is not regular. $\square$

Lemma 6. *Event b has at least one successor: $R'_1(b) \neq \varnothing$.*

Proof. Let us once more reason by contradiction: we suppose that b has no successor. Since the set of all possible valuations is finite ($|2^{\mathsf{AP}}| = 2^{|\mathsf{AP}|} < +\infty$), there exist two positive integers i and j such that $i < j$ and $V(wa^i b) = V(wa^j b)$; this grants that $\forall k \in [\![0, i]\!], (wa^i b^k \models \mathsf{pre}(b) \iff wa^j b^k \models \mathsf{pre}(b))$, as no arrow leaves any node that has b as the last event in its history.

At the same time, we have $\begin{cases} wa^i b^i & \not\models & \mathsf{pre}(b) \\ wa^j b^i & \models & \mathsf{pre}(b) \end{cases}$, giving rise to a contradiction. $\square$

Lemma 7. *b cannot have b as its only successor: $R'_1(b) \neq \{b\}$. (Thus, a is a successor of b: bR'_1a.)*

Proof. For every given $wa^i b^k$ node, any proposition is equivalent to its K-free version (*i.e.* all $\Box$ and $\Diamond$ get removed); we are brought back to the previous case, and the same contradiction holds. $\square$

Lemma 8. *b is among a's successors: aR'_1b.*

Proof. If there is no arrow from a to b, then, as before, since there are two non-negative integers i and j such that $i < j$ with $V(wa^i) = V(wa^j)$, we deduce $wa^i \simeq wa^j$, which cannot be (b can be applied j times from wa^j but not at wa^i). $\square$

Lemma 9. *There is no loop on b: $b \notin R'_1(b)$.*

Proof. Let us suppose the existence of such a loop; both $\mathcal{M}$ and $\mathcal{E}$ are then complete models (*i.e.* the relation encompasses all couples of nodes).

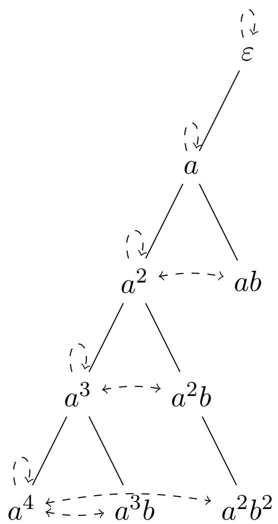
There exist two positive integers i and j such that $i < j$ with $V(wa^i) = V(wa^j)$; this entails $(\mathcal{M}\mathcal{E}^i, wa^i) \simeq (\mathcal{M}\mathcal{E}^j, wa^j)$ (by virtue of the Lemma below), which is impossible.

Lemma 10. *If $\mathcal{M}$ and $\mathcal{E}$ are complete models, then for all h and h' in $\mathcal{M}\mathcal{E}^*$, $V(h) = V(h')$ iff $(\mathcal{M}\mathcal{E}^*, h) \simeq (\mathcal{M}\mathcal{E}^*, h')$.*

$\square$

According to the previous properties, we are left with the following models (where pre- and postconditions are not indicated and w's valuation is indifferent):

Which results in the following DEL structure (each node is labeled by its event history; we only show the beginning of the tree):



Calling d the maximal modal depth, we can now once again consider two left nodes a^i and a^j with $i < j$ such that $V(a^i) = V(a^j)$ and $\{V(h) \mid a^i R_1 h\} = \{V(h) \mid a^j R_1 h\}$: this makes a^i and a^j bisimilar, which is a contradiction.

We are left with no candidate, hence the conclusion.

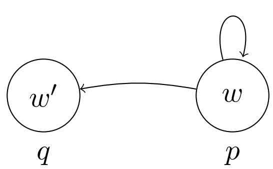
□

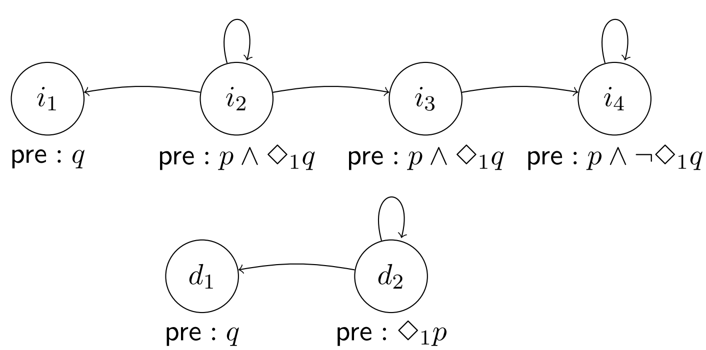
3.2.2 DEL-presentable

It turns out that one of the example from [DT16, Section B.8] generates a non-regular context-free language; stricly speaking, the DEL presentation is not single-rooted, but the trace stemming from one world is still non-regular context-free.

Theorem 5 (Strict CFL generation)**.** *There exists a non-regular context-free language L_{CF} and a $(\mathcal{L}_K^{1,0}, \texttt{non-ontic})$-DEL presentation $(\mathcal{M}, \mathcal{E})$ such that L_{CF} is a subtree of $\mathcal{M}\mathcal{E}^*$ rooted in $\mathcal{M}$.*

Proof. The models $\mathcal{M}$ and $\mathcal{E}$, with only one agent 1 and $\mathsf{AP} = \{p, q\}$, run as follows:





In order to prove that L_{CF} is context-free, it suffices to define a pushdown automaton that recognises the trace language stemming from w.

As we can show by induction the explicit behaviour of the model, namely, that of a counter (with the i component of $\mathcal{E}$ as the incrementor and d as the decrementor), we can take the following automaton:

- states: $Q = \{q_0; q_1\}$;
- initial state: q_0;
- stack bottom symbol[10]: ω;
- stack alphabet: $Z = \{\omega; I_2\}$;
- acceptance by final state with $T = Q$: every state is final *(characteristic of closure by prefix)*;
- transition rules, which are written according to the pattern (current state, read letter, stack top) $\to$ (new state, production replacing the stack top):

$$\begin{cases} (q_0, i_2, X) & \to & (q_0, I_2 X) \text{ where } X \in \{\omega; I_2\} \\ (q_0, d_2, \omega) & \to & (q_0, \omega) \\ (q_0, d_2, I_2) & \to & (q_0, \varepsilon) \\ (q_0, i_3, X) & \to & (q_1, X) \text{ where } X \in \{\omega; I_2\} \\ (q_1, i_4, X) & \to & (q_1, X) \text{ where } X \in \{\omega; I_2\} \\ (q_1, d_2, I_2) & \to & (q_1, \varepsilon) \end{cases}.$$

We can prove L_{CF} is indeed non-regular using a sublanguage and the pumping lemma, as in [DT16, Section B.8].

□

Remark 1. *Actually, in this case, we can also extend the automaton and show that* $\mathrm{Hist}_{\mathrm{ev}}(\mathcal{ME}^*)$ *is context-free (and still non-regular) as a whole.*

3.3 Context-sensitive languages

We will say *strictly* when qualifying a language in relation to the Chomsky hierarchy; for instance, a stricly context-free language is a non-regular context-free language.

Building upon the previous example in Section 3.2.2 for generating a strictly context-free language, we have tried to expand the models with an additional agent, worlds and events so as to form a strictly context-sensitive language, while staying in $DEL(\mathcal{L}_K^{[,\texttt{non-ontic}}, \texttt{non-ontic})]10$.

The idea is to generalise the previous counter behaviour to a double-counter behaviour: instead of having only two event components managing incrementation and decrementation of a single chain of worlds (thus generating a trace language with a distinct subset akin to the typically strictly CFL $\{a^n b^m \mid n \geq m\}$), we have been aiming at designings *three* event components taking care of *two* "orthogonal" chains, each with essentially one agent for control, such that:

- the first component increments the length of the first chain of worlds;
- if the first chain of worlds is not empty, the second component decrements the first chain and increments the second chain;

[10] Convention: the top of the stack is to the left (and thus, pushing and popping operate to the left).

- finally, the third component decrements the second chain if the second chain is not empty.

This design is intended at obtaining a distinct a trace language, which we call L_{CS}, with a distinct subset similar to $\{a^n b^m c^k \mid n \geq m \geq k\}$, which is stricly context-sensitive.

Conjecture 1.

$$L_{\mathsf{CS}} \in \text{PrefixCSL} \setminus \text{PrefixCFL}$$

We still have to confirm

4 DEL presentation and machine simulation

Knowing what classes of machines can be simulated through DEL leaks information on DEl's expressive power; for instance, several undecidability proofs for certain DEL fragments rely on simulating a Turing machine.

Here, we will take into consideration finite automata (FA), pushdown automata (PA) and the less studied class of queue automata (QA). We give a fairly abstract definition of the *Machine simulation problem* due to the fact that we will consider several classes of machines.

Definition 15 (Machine simulation problem)**.**

- *Input: M a machine in some class $\mathcal{C}$ (among FA, PA, QA).*
- *Output: A single-rooted DEL presentation $(\mathcal{S}_V^{\mathsf{Ag}}, \mathcal{E})$ such that $\mathcal{S}_V^{\mathsf{Ag}}\mathcal{E}^*$ and the execution tree of M are isomorphic.*

We state the further results under a generic theorem regarding abstract machines.

In the following, given an alphabet Σ, we let $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$.

Theorem 6. *Let $\mathcal{C}$ be a class of machines (among FA, PA, QA).*

For any machine M in $\mathcal{C}$, we can compute $\mathcal{E}_M$ with $\mathcal{E}_M = (E, R'_a, \mathsf{pre}, \mathsf{post})$, a one-world epistemic model $\mathcal{M}$ where $\mathcal{M} = (\{w\}, \{(w,w)\}, V)$, a function $\lambda : \Sigma_\varepsilon \times \mathsf{Op} \to E$ and a formula φ such that for any word s in Σ^*, for any sequence $(t_1, \ldots, t_n)$ in Op^n and decomposition $(s_1, \ldots, s_n)$ of s in $(\Sigma_\varepsilon)^n$, the following holds:

1. *$(t_1, \ldots, t_n)$ is a computation of M on $w_1 \ldots w_n$ if and only if $\lambda(w_1, t_1) \ldots \lambda(w_n, t_n) \in \mathrm{Hist}_{\mathrm{ev}}(\mathcal{S}_V^{\mathsf{Ag}}\mathcal{E}^*)$;*

   What's more, for any property P among $\{\mathsf{be\ accepting}, \mathsf{stack\ top} = a, \ldots\}$, there exists φ_P such that $P_L = \{(q,u) \mid u \in L\}$ where L is a regular star-free language.
2. *$c_0 \overset{t_1,w_1}{\vdash} c_1 \vdash \ldots \overset{t_n,w_n}{\vdash} c_n$ with $P(c_n)$ if and only if $\lambda(w_1,t_1) \ldots \lambda(w_n,t_n) \models \varphi_P$.*

Remark: in the following sections, we use a "mechanical" model for the machines: we do not consider any input alphabet, but merely transitions associated to operations. If such an alphabet comes into play, each of our constructions can be adapted by duplicating the event models and associating them the corresponding letters; the word matching an execution is then to be found as the projection of its event trace.

4.1 Finite automata with $DEL(\mathcal{L}_K^{0,0}, \texttt{ontic})$

DEL presentions in $DEL(\mathcal{L}_K^{0,0}, \texttt{ontic})$ are sufficient to simulate any finite automaton. Indeed, we can simply encode the current state as a single specific atomic proposition, and associate to each transition $q \to q'$ an event model with a single event having precondition q and postcondition q'

We suspect the class of finite automata *cannot* be simulated without ontic events (see 5.2), although this is yet to be proven.

4.2 Pushdown automata with $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$

We exhibit a simulation, which allows us to reduce the reachability problem for a pushdown automaton to the epistemic planning problem for DEL presentations in $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$.

4.2.1 Formalism

Beforehand, we define for some alphabet Σ the set of operations $\mathrm{Op}(\Sigma)$ by

$$\mathrm{Op}(\Sigma) = \{\mathsf{push}_x; \mathsf{pop}_x \mid x \in \Sigma\}$$

A *pushdown automaton* $\mathcal{A}$ is defined by $\mathcal{A} = (\Sigma, Q, q_0, F, \Delta)$ where:

- Σ is a finite alphabet;
- Q is a set of states;
- $q_0 \in Q$ (initial state) ;
- $F \subseteq Q$ (set of final states) ;
- $\Delta \subseteq Q \times \mathrm{Op}(\Sigma) \times Q$ (set of transitions). We write $q \overset{\mathrm{op}}{\to} q'$ for $(q, \mathrm{op}, q') \in \Delta$.

The initial configuration is (q_0, ε) (initial state and empty stack).

push_x is always possible and amounts to add x as the stack's top; pop_x is possible only when the stack's top is x, and removes the top.

The *reachability problem* consists in determining whether there exists an execution leading the automaton to a final state.

4.2.2 Simulation

Theorem 7. *The unfolding of the configuration graph of a pushdown automaton admits a $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$-presentation.*

Corollary 4. *The reachability problem for a pushdown automaton reduces to the epistemic planning problem in the ontic case with modal depth 1 and one agent.*

Proof. We consider models with a single agent, which we refrain from naming since there is no ambiguity: $|\mathsf{Ag}| = 1$.

We define atomic propositions in the following manner:

$$\mathsf{AP} = Q \mathbin{\dot\cup} \Sigma$$

We define the next epistemic model $\mathcal{M}$, which comprises a unique (pointed) world:

- $W = \{w\}$;
- $V(w) = \{q_0\}$;
- $R = \{(w, w)\}$.

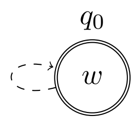


Figure 7: Initial epistemic model $\mathcal{M}$.

For each transition t of Δ, we define the event model $\mathcal{E}_t$ in the following manner:

1. Si $t = q \overset{\mathsf{push}_x}{\rightarrow} q'$:

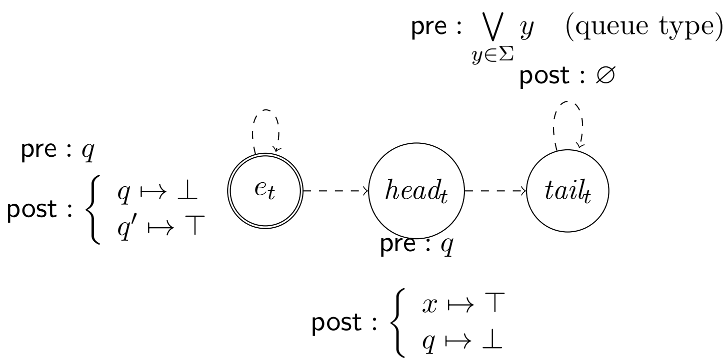


Figure 8: Event model $\mathcal{E}_t$: pushing.

2. If $t = q \overset{\mathsf{pop}_x}{\rightarrow} q'$:

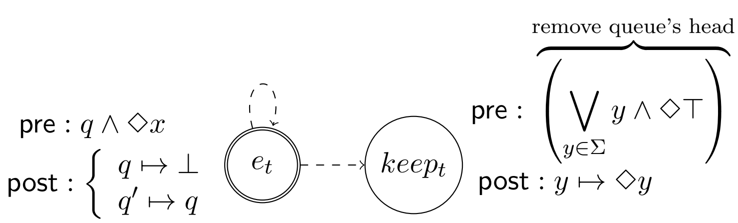


Figure 9: Event model $\mathcal{E}_t$: popping.

We then take as the goal formula φ defined as follows:

$$\varphi = \bigvee_{q \in F} q$$

□

Intuitively, the pointed world acts as both the stack "generator" and the state storage.

The node chain embodies the state of the stack, whose head is always "alongside" the pointed world, while the current state matches the pointed world's valuation.

4.2.3 Configuration graphs

We address the unfolding of a pushdown automaton's configuration graph; more specifically, we prove that the propositional fragment of DEL is too weak even for expressing an elementary pushdown automaton's execution tree.

We display a pushdown automaton whose configuration graph is not isomorphic to any propositional $\mathcal{M}\mathcal{E}^*$ structure.

Theorem 8. *There exists a pushdown automaton $\mathcal{A}$ such that, for all couples of epistemic model $\mathcal{M}$ and event model $\mathcal{E}$, the unfolding of the configuration graph of $\mathcal{A}$ not be isomorphic to $\mathcal{M}\mathcal{E}^*$.*

In particular, it is not isomorphic to $\mathcal{S}_V^{\mathsf{Ag}}\mathcal{E}^*$.

Proof. We take a pushdown automaton $\mathcal{A}$ with a single state q; we do not concern ourselves with the state's possibly final nature, nor with the acceptance mode, as this is irrelevant for the configuration graph.

- Set of states: $Q = \{q\}$
- Input alphabet: $A = \{a\}$
- Stack alphabet: $Z = \{\perp\}$
- Transition function $\delta : Q \times (A \cup \{\varepsilon\}) \times Z \to \mathcal{P}(Q \times Z^*)$:

$$\delta(q, a, \perp) = \{(q, \perp\perp); (q, \varepsilon)\}$$

- Initial stack symbol: $\perp$
- Initial state: q

Informally, we can only push and pop $\perp$ (in particular, nothing can be done once the stack is empty, which introduces a notion of counting).

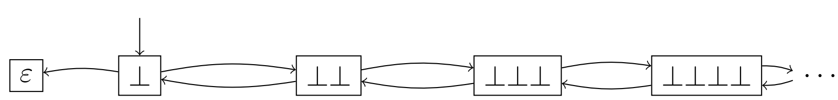


Figure 10: Configuration graph for $\mathcal{A}$; the states as well as the transitions' labels were omitted.

We now proceed to show that the previous pushdown automaton constitutes a counter-example.

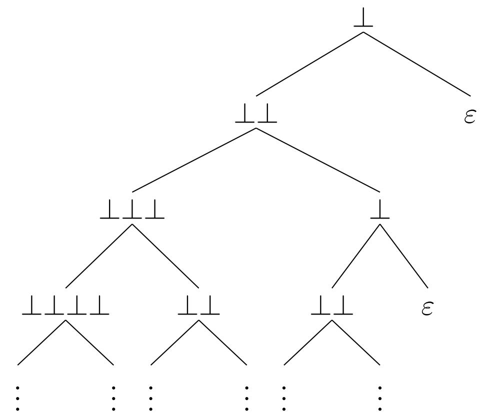


Figure 11: Unfolding of the previous graph.

The nodes from the (unfolded) graph have labels of the form $\perp^n$ (neglecting state q) where $n \in \mathbb{N}$, and all words of this form do feature; we can note that, since the automaton's behaviour depends exclusively on the stack, the subtrees with the same root label are isomorphic.

We suppose the unfolded graph isomorphic to a propositional $\mathcal{ME}^*$ structure and we call w_n the world of depth n with label $\perp^{n+1}$ for all n in $\mathbb{N}$ (we can confirm through induction that this definition is correct; visually, we are talking of the nodes on the left branch in the diagram from Figure 11. We demonstrate that each of these worlds must have a distinct valuation, which will lead to a contradiction.

Since the preconditions and postconditions are propositional, two worlds having the same valuations must correspond to two isomorphic subtrees; it is thus sufficient to establish that the subtrees whose roots are nodes from the left branch are non-isomorphic two by two. To show this, it is enough to note that the node matching w_n is linked to a node with label ε by a path of length exactly n (by just taking n times the right arrow) but not by a path of stricly inferior length (which can be proven by induction, considering the non-unfoled graph, for example): consequently, each of the w_n matches a subtree non-isomorphic to the others.

We deduce $|\{V(n) \mid n \in \mathbb{N}^*\}| = +\infty$; yet, $\{V(n) \mid n \in \mathbb{N}^*\} \subseteq 2^{\mathsf{AP}}$, which implied $|\{V(n) \mid n \in \mathbb{N}^*\}| \leq |2^{\mathsf{AP}}|$, then $|2^{\mathsf{AP}}| = +\infty$ and as a consequence $|\mathsf{AP}| = +\infty$: it would take an infinity of propositional atoms, which is ruled out by hypothesis.

□

Remark 2. *As a sidenote, both simulations of stack and queue automata were tested with some software developed by the team, Hintikka's World*[11]*, to visualise user-specified DEL structures.*

4.3 Queue automata with $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$

We exhibit a simulation, which allows us to reduce the reachability problem for a queue automaton to the epistemic planning problem for DEL presentations in $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$.

A queue automaton, as the name suggests, is essentially the same as a pushdown automaton, except the stack is replaced with a queue.

[11] http://people.irisa.fr/Francois.Schwarzentruber/hintikkasworld/

As a class, they are as powerful as Turing machines, which explains why they are comparatively less studied. [VFZ80]

We exhibit a simulation, which allows us to reduce the reachability problem for a queue automaton to an epistemic planning problem in the ontic case with modal depth 1; this yields a new undecidability proof (of an already known result), and can possibly be used to try the same approach in the purely epistemic case.

4.3.1 Formalism

The formalism is analogous to that of pushdown automata.

Beforehand, we define for some alphabet Σ the set of operations $\mathrm{Op}(\Sigma)$ by

$$\mathrm{Op}(\Sigma) = \{\mathsf{enqueue}_x; \mathsf{dequeue}_x \mid x \in \Sigma\}$$

A *queue automaton* $\mathcal{A}$ is defined by $\mathcal{A} = (\Sigma, Q, q_0, F, \Delta)$ where:

- Σ is a finite alphabet;
- Q is a set of states;
- $q_0 \in Q$ (initial state) ;
- $F \subseteq Q$ (set of final states) ;
- $\Delta \subseteq Q \times \mathrm{Op}(\Sigma) \times Q$ (set of transitions). We write $q \overset{\mathrm{op}}{\to} q'$ for $(q, \mathrm{op}, q') \in \Delta$.

The initial configuration is (q_0, ε) (initial state and empty queue).

$\mathsf{enqueue}_x$ is always possible and amounts to add x as the queue's head; $\mathsf{dequeue}_x$ is possible only when the last element of the queue is x, and removes the last element.

The *reachability problem* consists in determining whether there exists an execution leading the automaton to a final state.

4.3.2 Simulation

Theorem 9. *The unfolding of the configuration graph of a queue automaton admits a $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$-presentation.*

Corollary 5. *The reachability problem for a queue automaton reduces to the epistemic planning problem in the ontic case with modal depth 1 and one agent.*

As the expressive power of the class of queue automata is the same as that of Turing machines, we get the following result as a corollary.

Corollary 6. *The epistemic planning problem in the ontic case with modal depth 1 and one agent is undecidable.*

Theorem 9. We consider models with a single agent, which we refrain from naming since there is no ambiguity: $|\mathsf{Ag}| = 1$.

We define atomic propositions in the following manner:

$$\mathsf{AP} = Q \,\dot{\cup}\, \Sigma$$

We define the next epistemic model $\mathcal{M}$, which comprises a unique (pointed) world:

- $W = \{w\}$;
- $V(w) = \{q_0\}$;
- $R = \{(w, w)\}$.

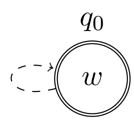


Figure 12: Initial epistemic model $\mathcal{M}$.

For each transition t of Δ, we define the event model $\mathcal{E}_t$ in the following manner:

1. If $t = q \overset{\mathsf{enqueue}_x}{\rightarrow} q'$:

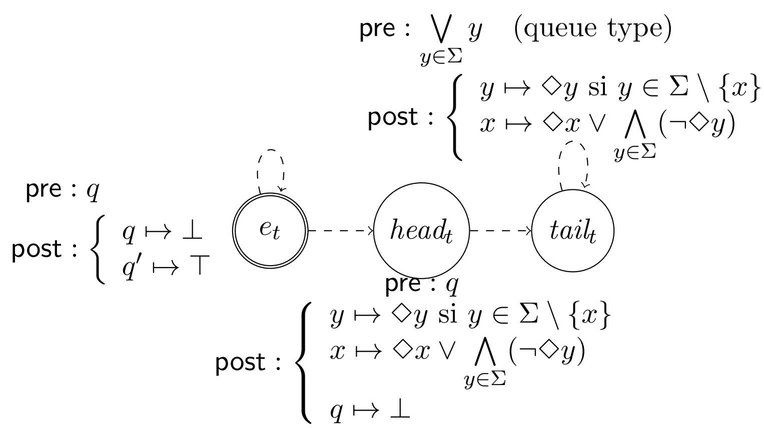


Figure 13: Event model $\mathcal{E}_t$: enqueueing.

2. If $t = q \overset{\mathsf{dequeue}_x}{\rightarrow} q'$:

We then take as the goal formula φ defined as follows:

$$\varphi = \bigvee_{q \in F} q$$

$\square$

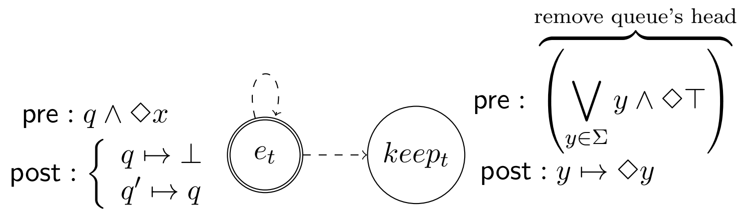


Figure 14: Event model $\mathcal{E}_t$: dequeueing.

Intuitively, the pointed world acts as both the queue "generator" and the state storage.

The node chain embodies the state of the queue, whose head is always "alongside" the pointed world, while the current state matches the pointed world's valuation.

The linking allows to "repel" the last generated queue world at the tip of the chain through propagation via the postconditions.

5 Conclusion

5.1 New results summary

Our results concerning languages are summarised in Table 15 and Diagram 17, while those regarding machines are summed up in Table 1.

In particular, we have shown undecidability for the $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$ case with 1 agent, which is an improvement on the $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$ case with 3 agents [BA11].

Languages	DEL presentations
Reg	Theorem 3
CFL	Theorem 4 , Theorem 5
CSL	Conjecture 1

Figure 15: Summary for languages.

Diagram summarising a comparison between single-rooted DEL's expressiveness and the Chomsky hierarchy for prefix-closed languages.

The Theorem 2.3.2 showing a reduction of $DEL(\mathcal{L}_K^{md\geq 2,ad}, \texttt{non-ontic})$ to $DEL(\mathcal{L}_K^{1,0}, \texttt{ontic})$ is novel to the best of our knowledge.

5.2 Future work

In the time span left for this internship, we have some leads to investigate:

- simulating pushdown automata with only 1 state for $DEL(\mathcal{L}_K^{md,0}, \texttt{non-ontic})$;
- simulating higher-order pushdown automata;

Machines	DEL presentations
FA	$\mathbb{E}(\mathcal{L}_K^{0,0}, \texttt{ontic})$ and no agent
PA	$\mathbb{E}(\mathcal{L}_K^{0,1}, \texttt{ontic})$ and one agent
QA	$\mathbb{E}(\mathcal{L}_K^{0,1}, \texttt{ontic})$ and one agent

Figure 16: Summary of minimal classes of DEL presentations to simulate machines.

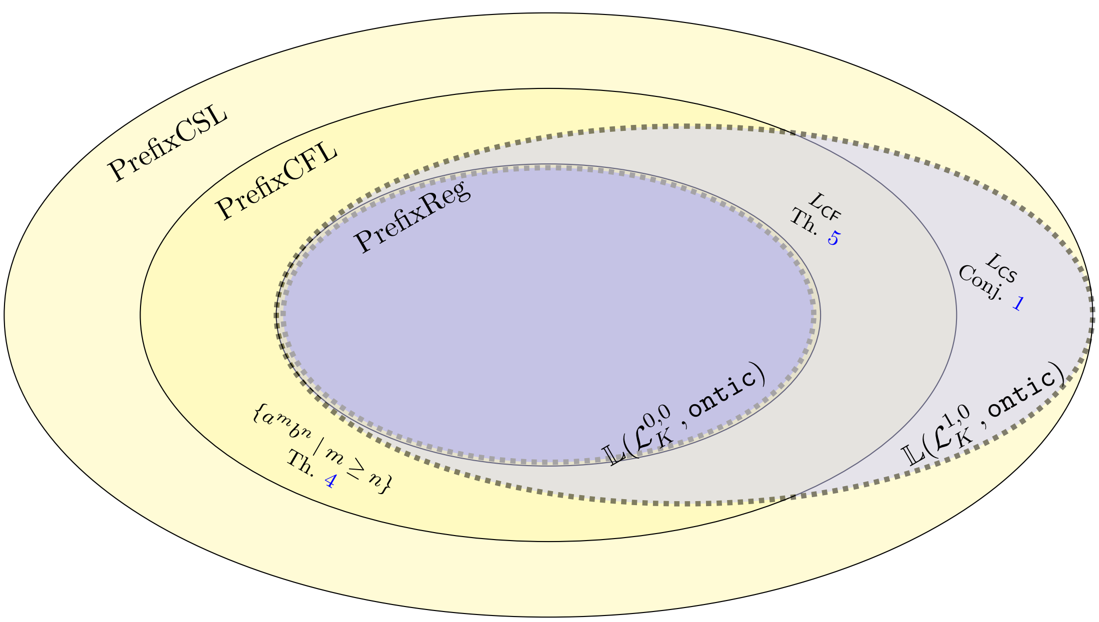


Figure 17: Euler diagram depicting DEL hierarchy and Chomsky hierarchy

- showing a finite automaton cannot be simulated with $DEL(\mathcal{L}_K^{1,0}, \texttt{non-ontic})$ (which would suggest a weak expressiveness for the non-alternating fragment, and maybe hint at decidability for the planning problem);
- complete the hierarchy;
- evaluate complexity estimates for the epistemic planning problem thanks to machine simulations;
- try and interpret game arenas as DEL structures, which appears naturally as the converse approach to that of [Mau14] (from PDEL to arenas), and where epistemic relations correspond to synchronous transducers; some thought has been given to this question, but we do not have enough matter for the moment.

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.

[BDRV02] Patrick Blackburn, Maarten De Rijke, and Yde Venema. *Modal Logic: Graph. Darst*, volume 53. Cambridge University Press, 2002.

[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.

[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.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[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.

[VFZ80] Bernard Vauquelin and Paul Franchi-Zannettacci. Automates a file. *Theoretical Computer Science*, 11(2):221 – 225, 1980.

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