HAL open science

Vérification symbolique de modèles pour la logique épistémique dynamique probabiliste

Sébastien Gamblin, Alexandre Niveau, Maroua Bouzid

To cite this version:

Sébastien Gamblin, Alexandre Niveau, Maroua Bouzid. Vérification symbolique de modèles pour la logique épistémique dynamique probabiliste. Journées Francophones Planification, Décision et Apprentissage (PFIA) 2021, Jun 2021, Bordeaux, France. hal-03664513

HAL Id: hal-03664513

https://hal.science/hal-03664513v1

Submitted on 11 May 2022

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

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

Vérification symbolique de modèles pour la logique épistémique dynamique probabiliste

Sébastien Gamblin[1] Alexandre Niveau[1] Maroua Bouzid[1]

[1] Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

prénom.nom@unicaen.fr

Résumé

La logique épistémique dynamique probabiliste (PDEL) permet de raisonner sur les connaissances probabilistes imbriquées des agents et sur les changements de connaissances induits par l'occurrence d'événements. Bien que PDEL a été étudiée de manière théorique, elle n'a été appliquée qu'à des exemples jouets : en effet, l'explosion combinatoire des structures de Kripke ne permet pas d'utiliser directement le formalisme pour des applications réelles, comme des jeux de cartes.

Le présent article constitue un premier pas vers l'utilisation de PDEL en pratique : dans la continuité des récents travaux sur la vérification de modèles de DEL (non probabiliste), nous proposons une représentation "symbolique" des structures de Kripke probabilistes avec des fonctions pseudo-booléennes, représentables par des structures de données de la famille des diagrammes de décision, en particulier les diagrammes de décisions algébriques (ADDs). Nous montrons que les ADDs passent mieux à l'échelle que les structures de Kripke explicites, et permettent une vérification symbolique de modèles efficace, même sur un exemple réaliste du jeu de carte Hanabi, ouvrant ainsi la voie à l'application pratique de techniques de planification épistémique.

Mots Clef

Vérification symbolique de modèles, Logique épistémique dynamique probabiliste, PDEL, Structure de Kripke, Planification, Hanabi

Abstract

Probabilistic Dynamic Epistemic Logic (PDEL) is a formalism for reasoning about the higher-order probabilistic knowledge of agents, and about how this knowledge changes when events occur. While PDEL has been studied for its theoretical appeal, it was only ever applied to toy examples: the combinatorial explosion of probabilistic Kripke structures makes the PDEL framework impractical for realistic applications, such as card games.

This paper is a first step towards the use of PDEL in more practical settings: in line with recent work applying ideas from symbolic model checking to (non-probabilistic) DEL, we propose a "symbolic" representation of probabilistic Kripke structures as pseudo-Boolean functions, which can be represented with several data structures of the decision diagram family, in particular Algebraic Decision Diagrams (ADDs). We show that ADDs scale much better than explicit Kripke structures, and that they allow for efficient symbolic model checking, even on the realistic example of the Hanabi card game, thus paving the way towards the practical application of epistemic planning techniques.

Keywords

Symbolic model checking, Probabilistic dynamic epistemic logic, PDEL, Kripke structures, Planification, Hanabi

1 Introduction

The card game Hanabi Bauza [2010] has recently drawn interest from the AI community Bard *et al.* [2020]. Hanabi is a multiplayer cooperative game with imperfect information in which *higher-order knowledge* plays a very important role, i.e., players need to make decisions depending on what they know about what other players know, and so on. With the ultimate goal of computing strategies for games like Hanabi, our focus is on approaches based on Dynamic Epistemic Logic (DEL) Kooi [2003], a formalism allowing one to reason about the higher-order knowledge of agents, and about how this knowledge changes when events occur. DEL constructs allow for an elegant approach to multi-agent epistemic planning Bolander [2017], and more recently, to the problems of controller and distributed strategy synthesis in adversarial games Maubert *et al.* [2019].

However, in many realistic games, in particular with imperfect or incomplete information, such as Hanabi, there is clearly no winning strategy for any player; what one usually wants in these cases is an *optimal* strategy, one that maximizes the expectation of a victory. A natural direction is to study an adaptation of epistemic planning to Probabilistic Dynamic Epistemic Logic (PDEL), a generalization of DEL which is interpreted on Kripke

structures augmented with probabilistic information.
Yet, while there has been effort lately to make DEL useable in practice, by applying ideas from symbolic model checking, thus avoiding the combinatorial explosion of explicit Kripke structures Charrier and Schwarzentruber [2017]; van Benthem *et al.* [2017]; Gattinger [2018]; Charrier *et al.* [2019], there has been no such work for PDEL; to the best of our knowledge, it remains an entirely theoretical framework, only ever applied on toy examples.
This paper is a first step towards the use of PDEL in more practical settings: we propose a "symbolic" representation of stochastic Kripke structures as pseudo-Boolean functions, which can be represented with several data structures of the decision diagram family. Our experiments using Algebraic Decision Diagrams (ADDs) on a modelization of Hanabi show that the size of these representations scale much better than explicit Kripke structures, while allowing for efficient symbolic model checking, even for near-realistic game sizes. The next step is to generalize epistemic planning to a probabilistic setting, so as to study optimal strategy synthesis for games with imperfect or incomplete information; our results indicate that such a generalization would not only be of theoretical interest, but could also be useful in practice.
After presenting some background, such as Hanabi or PDEL (2), we introduce our symbolic representation and model checking procedure (3), then report about experimental results (4).

2 Background

2.1 Hanabi

Hanabi Bauza [2010] is a cooperative card game where players must play their cards in order; the specificity is that players can never look at their cards, although they see those of other players and can give them information. There are 5 card colors (red, blue, green, yellow, white) and 5 card values (numbers from 1 to 5, with the following repartition: $3 \times$ 1, $2 \times$ 2 3 4, and $1 \times$ 5) for each color. The goal is to play as many cards as possible on the table, but the cards of each color must be played in increasing order (an error costs one red token, out of three in total). At their turn, each player can either play a card on the table, give information to another player (which costs one blue token), or discard a card (which gives one blue token back). Giving information is limited to announcing all cards of a given color or value in a the hand of another player, by pointing to the cards. Figure 1 shows a game situation. See han [2021] for the precise rules of the game.
We will represent the "physical" state of the game (independant from the knowledge acquired by the players) using propositional variables, which basically describe where each card is located. For example, variable card^{ap}_{cvi} describes the fact that player a has the card of color $c \in \{W,R,B,Y,G\}$ and value $v \in \{1..5\}$ of id i (there are

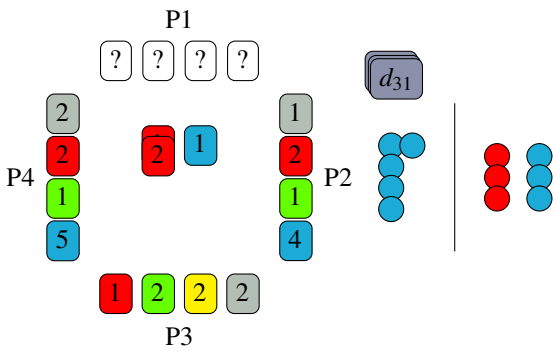

Figure 1: Presentation of a game situation as seen by player 1

several cards with the same color and value), at position p in their hand. Variable card^{z}_{cvi} is the same, but with $z \in \{Draw, Discard, Table\}$ and with no notion of position. Still other propositional variables count the number of tokens (R_i or B_i) and indicate which player is next (turn_a). Clearly, this is not sufficient to represent the real state of a game: the knowledge of each player about their cards evolves during a game, and in order to reason about how best to play, it is necessary to have a model of this knowledge – we use *epistemic logic* for this.

2.2 Probabilistic epistemic logic

We fix a finite set A of *agents* and a countable set PS of propositional symbols. We denote as WS the countable subset of PS containing the symbols used to describe possible worlds; for simplicity we consider it fixed, even though it depends on the application (for example, in the context of Hanabi, we consider that WS contains all and only the necessary symbols of the form card^{ap}_{cvi}, card^{z}_{cvi}, R_i, B_i and turn_a).
First, we define probabilistic Kripke structures (also called probabilistic Kripke/epistemic models; in the following we simply call them "Kripke structures" for short), which allows one to model the epistemic and probabilistic knowledge of agents.

Definition 1 ((probabilistic) Kripke structure)**.** *A* (probabilistic) Kripke structure *is a tuple* $\mathscr{M} = \langle W, R, \mu, V \rangle$ *such that:*

- W *is a nonempty finite set of* worlds,
- R *associates to each agent* $a \in A$ *an* accessibility relation $R_a \subseteq W \times W$,
- μ *associates to each agent* $a \in A$ *and each world* $w \in W$ *a probability function* $\mu_a(w)$ *over all worlds, i.e., a function* $W \to [0,1]$ *such that* $\sum_{w' \in W} \mu_a(w)(w') = 1$, *and*
- V *is a* valuation function $V : W \to 2^{WS}$, *indicating the set of propositions that are true in each world.*

Note that there are more general definitions of probabilistic Kripke structures, using for example σ-algebras Fagin and Halpern [1994]; in this paper we use the simplest case of probability functions, but our approach would not be hard to generalize to richer settings. Intuitively, the accessibility relation of agent a links worlds that a considers as *undistinguishable*, and $\mu_a(w)(w')$ indicates the probability a assigns to being in world w' when it is actually in world w.

Example 2. *Let s and r be two propositions standing for "sun" and "rain"; assume that at any time at least one of the two propositions must be true. Figure 2 (left) displays an example of a very simple Kripke structure, in which there are two agents, whose accessibility relations are consistent with the probability functions. The probabilities associated with the arcs show that the black agent deems that there is a greater chance of rain. World w_1 is marked with a double circle to indicate that it is the real world. Black arcs show probabilities for agent a only, and orange arcs for agents a and b.*

Having both an accessibility relation and probability functions may seem redundant, but it has two advantages: (i) it allows one to model an unquantified uncertainty between possible worlds, which is not the same as assigning them a uniform probability, and (ii) it allows one to distinguish between having probability 0 to be true and being truly impossible. See Fagin and Halpern [1994] for an in-depth discussion.

Definition 3 (Probabilistic epistemic language $\mathscr{L}_{PEL}$)**.** *The language $\mathscr{L}_{PEL}$ of probabilistic epistemic logic is defined by the following Backus-Naur form :*

$$\phi ::= p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \mathrm{K}_a\phi \mid \alpha_1 \mathrm{Pr}_a(\phi_1) + \cdots + \alpha_k \mathrm{Pr}_a(\phi_k) \geq \beta$$

where $p \in WS$, $a \in A$, $\alpha_1, \ldots, \alpha_k, \beta$ are rational numbers, and $k \geq 1$.

We also use parentheses in formulas for disambiguation, and we recall the usual abbreviations: $p \vee q$ (or) is $\neg(\neg p \wedge q)$, $p \to q$ (implication) is $\neg p \vee q$ and $p \leftrightarrow q$ (equivalence) is $(p \to q) \wedge (q \to p)$. This language allows higher-order formulas so that we can write : $\mathrm{K}_i(\mathrm{Pr}_j(\phi) \geq \beta)$ or $\mathrm{Pr}_j(\mathrm{K}_i\phi) \geq \beta$. For example, in the first formula, agent i knows that agent j estimates the probability that ϕ holds to be greater than or equal to β. Next, we explain how one decides whether a formula in $\mathscr{L}_{PEL}$ holds in a given state of epistemic and probabilistic knowledge:

Definition 4 (semantics of PEL)**.** *Let $\mathscr{M}$ be a Kripke structure and w a world of $\mathscr{M}$. We define the semantics of PEL inductively as follows, where $p \in WS$, $\phi, \psi, \phi_1, \ldots \phi_k \in \mathscr{L}_{PEL}$, and $a \in A$:*

$\mathscr{M}, w \models p$ *iff* $V(w)(p) = \top$, *with* $p \in WS$

$\mathscr{M}, w \models \neg\phi$ *iff* $\mathscr{M}, w \not\models \phi$

$\mathscr{M}, w \models \phi \wedge \psi$ *iff* $\mathscr{M}, w \models \phi$ *and* $\mathscr{M}, w \models \psi$

$\mathscr{M}, w \models \mathrm{K}_a\phi$ *iff* $\mathscr{M}, w' \models \phi$ *for all* w' *s.t.* $(w, w') \in R_a$ *and* $\mathscr{M}, w \models \alpha_1 \mathrm{Pr}_a(\phi_1) + \cdots + \alpha_k \mathrm{Pr}_a(\phi_k) \geq \beta$ *iff* $\sum_{i=1}^{k} \alpha_i \mu_a(\mathscr{M}, w)(\phi_i) \geq \beta$, *with* $\mu_a(\mathscr{M}, w)(\phi_i) := \sum_{w' : \mathscr{M}, w' \models \phi_i} \mu_a(w)(w')$.

In our weather example, formulas $Pr_b(r) \geq 0.5$ and $K_b(Pr_a(s \vee r) \geq 0.8)$ are true. Now, Kripke structures, probabilistic or not, only represent a *static* state of knowledge. In order to take changes into account, e.g. due to player actions, they have to be coupled with other structures called *update models*.

2.3 Updating knowledge

Note that we use $\mathscr{L}_{WS}$ to refer to the propositional language.

Definition 5 ((probabilistic) update model)**.** *An* (probabilistic) update model *is a tuple $\mathscr{E} = \langle E, R^E, pre, post, \mu^E \rangle$, where*

- *E is nonempty set of* events*,*
- *R^E associates to each agent $a \in A$ an* accessibility relation *$R^E_a \subseteq E \times E$,*
- *pre is a* precondition *function pre: $E \to \mathscr{L}_{EL}$,*
- *post is a* postcondition *function post: $E \times WS \to \mathscr{L}_{WS}$ (with $\mathscr{L}_{WS}$ the language of propositional formulas),*
- *μ^E associates to each agent $a \in A$ and each event $e \in E$ a probability function $\mu_a(e)$ over all events.*

Example 6. *The update model in Figure 2 (middle) (in which we suppose that probability functions are the same for all agents) contains an event e_0 with precondition $s \wedge r$ which assigns true to the variable b ("rainbow"). If the precondition is true, there is a 60% chance that there is a rainbow. Nevertheless, this event is confused by agents with the event e_1, with precondition r. Would this second event occur, there would be an 80% chance that it be considered to occur.*

Note that using this definition, probabilistic update models are actually very close to the classical update models of DEL van Benthem *et al.* [2006b] – the only difference being the added probability functions. This makes them different from probabilistic update models as defined by van Benthem *et al.* [2009], for two reasons. First, they feature *postconditions*, which allow them to have *ontic effects*, i.e., to change the state of the world and not only the knowledge state of agents; this is indeed crucial if we want to use PDEL to reason about games such as Hanabi. By fixing $post(e,x) = x$ for all $e \in E$ et all $x \in WS$, we get a purely epistemic update model, with no ontic effect. The other difference is that in the update models of van Benthem *et al.* [2009], preconditions are not directly tied to events. They are defined as a set Φ of pairwise inconsistent sentences, together with a probability distribution $pre(\phi, e)$ over E for each sentence $\phi \in \Phi$ and each event $e \in E$. As the authors argue, this allows

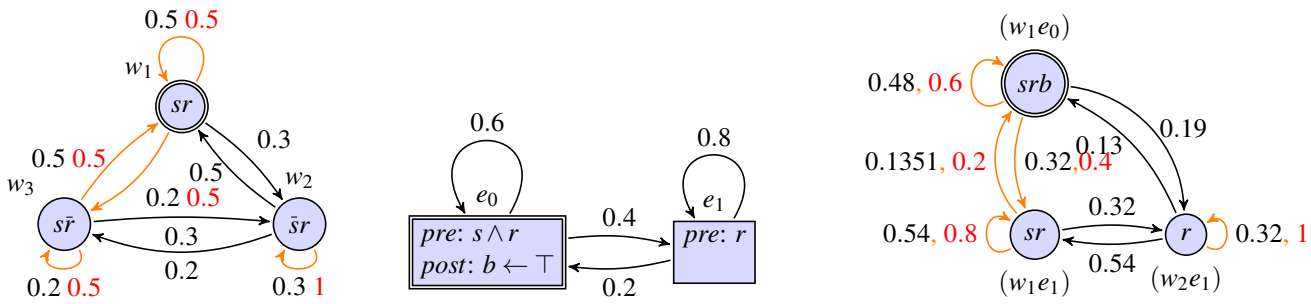

Figure 2: Examples of a probabilistic Kripke structure (left) and of an update model (middle), and their product update (right)

one to independantly consider *observation probabilities*, given by μ^E – quantifying the distinguishability of events – and *occurrence probabilities*, given by Φ and *pre* – indicating the probability with which each event can occur in each situation. Even though these "full update models" are more flexible, they are actually not more expressive than our simpler version (without postconditions). As remarked by the authors in a prepublished extended version of their work van Benthem *et al.* [2006a], any full update model has an equivalent so-called "observation model", which corresponds to our definition (again, ignoring ontic effects). Moreover, our choice of a conceptually simpler definition is harmless in practice, because transforming a full update model into an observation model can be done in quadratic time. However, it is important to remark that the converse is not true – it can be shown that the following observation model has no polynomial representation as a full update model: a clique of $|WS|$ events, with uniform probabilities, each event having a distinct variable $x \in WS$ as precondition. The problem is that full update models require pairwise inconsistent preconditions, so we need one for each valuation in 2^{WS}. This is a strong motivation for our choice of limiting update models to "observation models" because, although update models of this form do not seem very useful semantically, they become essential as soon as we add postconditions to them – such combinatorial, probabilistically uniform updates are frequent in the context of card games.

Now that we have update models to represent events that can occur in the real world (either by a voluntary action of an agent, or not), and how these events are perceived by the agents, we will explain how we can use them. The mechanism used in PDEL to compute the new knowledge state of agents after an event took place is called *product update*; it is more or less a simple Cartesian product of the Kripke structure representing the previous knowledge state and of the update model representing the event.

Definition 7 (probabilistic product update)**.** *Let $\mathscr{M}$ be a Kripke structure and $\mathscr{E}$ be an update model.*
The product update *of $\mathscr{M}$ by $\mathscr{E}$ is the Kripke structure $\mathscr{M} \otimes \mathscr{E} = \langle W^{\otimes}, R^{\otimes}, \mu^{\otimes}, V^{\otimes} \rangle$ defined as follows:*

- $W^{\otimes} = \{(w,e) \in W \times E \mid \mathscr{M}, w \models pre(e)\}$
- $(w,e) R_a^{\otimes} (w',e')$ *iff* $(w,w') \in R_a$ *and* $(e,e') \in R_a^E$
- $\mu_a^{\otimes}((w,e))((w',e')) = \frac{\mu_a(w)(w') \cdot \mu_a^E(e)(e')}{\sum_{(w'',e'') \in W^{\otimes}} \mu_a(w)(w'') \cdot \mu_a^E(e)(e'')}$ *if the denominator is nonzero, and* 0 *otherwise*
- $V^{\otimes}((w,e)) = \{p \in WS \mid \mathscr{M}, w \models post(e,p)\}$

Example 8. *Fig. 2 (right) shows the product update of the Kripke structure and update model. We can see that after the application of the event, agent a (in black) knows that it is raining: $K_a r$, but also that considers that there may be a rainbow with probability more than 45% : $Pr_a b \geq 0.45$.*

Thanks to the product update, it is then possible to do model checking by using a new operator that means "after the application of such event, will this formula be true?" This is how the logic becomes "dynamic".

Definition 9 (PDEL)**.** *The* probabilistic dynamic epistemic language *$\mathscr{L}_{PDEL}$ is defined by the BNF of $\mathscr{L}_{PEL}$ (Def. 3) augmented with the following production rule:*

$$\phi ::= [\mathscr{E}, e]\phi$$

where $\mathscr{E}$ is a probabilistic update model and e an event. We denote by $\mathscr{L}^0_{PDEL}$ the language in which all update models have propositional preconditions. The semantics of this additional operator is defined as follows: $\mathscr{M}, w \models [\mathscr{E}, e]\phi$ iff

$$\mathscr{M}, w \models pre(e) \implies \mathscr{M} \otimes \mathscr{E}, (w,e) \models \phi$$

We are interested in the *model checking* problem, which consists in deciding, given a pointed Kripke structure $(\mathscr{M}, w)$ and a formula ϕ in $\mathscr{L}_{PEL}$ or $\mathscr{L}_{PDEL}$, whether $\mathscr{M}, w \models \phi$.

3 Symbolic representation

Because of the combinatorial nature of Kripke structures, the size of such representations quickly becomes huge in practical examples, even moderately realistic. To avoid this problem, an idea is to use a "symbolic" representation of Kripke structures, in which redundancies are factored out, so that knowledge states remain scalable, while still allowing for reasonably efficient model checking. The

idea of symbolic model checking McMillan [1993] has recently been applied to dynamic epistemic logic, the symbolic representation of Kripke structures being either *accessibility programs* Charrier and Schwarzentruber [2017]; Charrier *et al.* [2019] or Binary Decision Diagrams (BDDs) van Benthem *et al.* [2017]; Gattinger [2018], BDDs being a well-known efficient representation of Boolean formulas Bryant [1986].

We now show how these concepts apply to probabilistic dynamic epistemic logic; contrary to Shirazi and Amir [2008], who also represent probabilistic Kripke structures in a factored way (using Bayesian networks), we are not only interested in static structures. In order to represent probabilities, we have to go beyond Boolean formulas, and use pseudo-Boolean functions. Given $X = \{x_1, \ldots, x_n\} \subseteq PS$ a finite set of propositional symbols, we call *pseudo-Boolean function (PBF) on variables* X a total function of the kind $f\colon 2^X \to \mathbb{R}$. There are several ways to represent pseudo-Boolean functions, notably generalizations of BDDs; let us mention Algebraic Decision Diagrams (ADDs) Bahar *et al.* [1997], Semiring-Labelled Decision Diagrams (SLDDs) Wilson [2005]; Fargier *et al.* [2013], Affine Algebraic Decision Diagrams (AADDs) Sanner and McAllester [2005], and Probabilistic Sentential Decision Diagrams (PSDDs) Kisa *et al.* [2014]. These languages are of varying *succinctness* (i.e., some are able to represent PBFs more compactly than others) and do not have the same efficiency for operations (such as summing PBFs or "forgetting" variables). There is a tradeoff to be found, depending on the application; systematically studying and quantifying such tradeoffs is the goal of the literature about the *knowledge compilation map* Darwiche and Marquis [2002]; Fargier *et al.* [2014]. In this paper, we remain as general as possible and only talk about PBFs; it should be implicitly understood that they are represented in some efficient language, such as ADD, the language we used in our experiments.

Before going on, we need to introduce some conventions and notations. We consider that Boolean functions, i.e., functions of the form $f\colon 2^X \to \mathbb{B}$, are particular pseudo-Boolean functions, and for simplicity we often identify propositional formulas ϕ over X with the Boolean function they represent; i.e., we see ϕ as the Boolean function $v \in 2^X \mapsto \begin{cases} 1 & \text{if } v \models \phi, \\ 0 & \textit{otherwise} \end{cases}$.

For a PBF f on $X \subseteq PS$ and a real number β, we denote $\mathrm{Cut}_{\geq\beta}(f)$ the Boolean function associating a valuation $v \in 2^X$ to 1 if $f(v) \geq \beta$, and to 0 otherwise.

We call support of f, written $\mathrm{supp}(f)$, the set $Y \subseteq 2^X$ where f is nonzero : $\mathrm{supp}(f) = \{x \in 2^X \mid f(x) \neq 0\}$. Note that if ϕ is a propositional formula over X, $\mathrm{supp}(\phi)$ is the model set of ϕ.

Finally, let $\odot\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ be an associative and commutative operation with a neutral element (such as addition, multiplication, min, max, or the Boolean connectives $\vee$ and $\wedge$), let $X = (Y \uplus Z) \subseteq PS$, and let f be a PBF on X. The $\odot$-*marginalization* of Y in f, denoted $\mathrm{Marg}_{\odot,Y}(f)$, is the function $g\colon 2^Z \to \mathbb{R}$ defined by $g(v) = \bigodot_{v' \in 2^X\,:\, v = v'_Z} f(v')$. Note that if f is a Boolean function, $\vee$-marginalization (resp. $\wedge$-marginalization) of Y corresponds to *existentially (resp. universally) forgetting* the variables in Y. We write $\mathrm{Forget}^{\exists}_Y(f) = \mathrm{Marg}_{\vee,Y}(f)$ and $\mathrm{Forget}^{\forall}_Y(f) = \mathrm{Marg}_{\wedge,Y}(f)$. Also we use for $X \subseteq PS$, f a boolean function and $v \in 2^X$, $f_{|v} = \mathrm{Forget}^{\exists}_X(f \wedge v)$

In this article, we use PBF everywhere. In fact, for simplification, if PBF is Boolean Function we will write it as Boolean formula. Also, a valuation, list of term, or a list or valuation can be considered as a Boolean formula. We can notice that, for v a valuation and f a boolean formula, notations $v \in \mathrm{supp}(f)$, $v \models f$ and $f(v) = 1$ are equivalent.

3.1 Static structure

Let us now show how to represent a Kripke structure with a PBF. The basic idea is to identify worlds with their valuations, so that propositional formulas over WS directly represent sets of worlds. Obviously, it is not possible in the general case: two distinct worlds in a Kripke structure can have the same valuation. The symbolic representation only works for structures whose valuation function is injective:

Definition 10 (V-injectivity)**.** *A Kripke structure* $\mathscr{M} = \langle W, R, \mu, V \rangle$ *is called* V-injective *if* $\forall w, w' \in W\colon w \neq w' \to V(w) \neq V(w')$.

While V-injective Kripke structures are considerably restricted in terms of expressiveness (e.g., the satisfiable S5 formula $\mathrm{K}_a 1p \wedge \neg \mathrm{K}_a 2(\mathrm{K}_a 1p \vee \mathrm{K}_a 2\neg p)$ has no V-injective S5 model), the setting still applies to a lot of games – notably those in which there are no private announcements or secret actions, so that all uncertainty is reducible to uncertainty about the "physical state" of the game. This is the case for Hanabi: for example, even though Alice does not know what the other players know about her cards, she can enumerate the possible game states (this boils down to enumerating her possible hands), and for each one, assuming it is the actual game state, she knows exactly what other players know. This would not be the case if there were private announcements (e.g., one player gives an information to another one without being heard by the others) or secret actions (e.g., two players switch one of their cards without others noticing). Moreover, for cases requiring more expressiveness, it is always possible to distinguish between worlds by labelling them with fresh symbols. Finally, note that this restriction is not unusual; it also holds for existing approaches to symbolic model checking for epistemic logic.

In order to represent relations over worlds, we will use the classical trick of *duplicating our vocabulary* Gattinger [2018]: we introduce a fresh set of symbols $WS' \subseteq PS \setminus WS$, in one-to-one correspondence with those in WS (we fix some bijection $\mathrm{succ}\colon WS \to WS'$, denoting $\mathrm{pred} = \mathrm{succ}^{-1}$), and see each valuation $v \in 2^{WS \cup WS'}$ as a pair of valuations (v_1, v_2), where $v_1 = v \cap WS$ and $v_2 = \{\mathrm{pred}(x) \mid x \in v \cap WS'\}$.

For $v \in 2^{WS}$, we write $\mathrm{succ}(v) := \{\mathrm{succ}(x) \mid x \in v\}$, and for f a PBF on WS, we write $\mathrm{succ}(f)$ for the PBF $v \in 2^{WS} \mapsto \mathrm{succ}(v)$; we introduce similar overloadings for pred.
We can define symbolic Kripke structure and the function to transform a explicit Kripke structure into symbolic one.

Definition 11 (Symbolic Kripke structure)**.** *A* symbolic Kripke structure *is a tuple $F = \langle W^r, R^r, \mu^r \rangle$ such that*

- *W^r is a Boolean function on WS, called the* law of worlds*;*
- *R^r associates to each agent $a \in A$ a Boolean function R^r_a on $WS \cup WS'$, called the* law of knowledge*;*
- *μ^r associates to each agent $a \in A$ a PBF μ^r_a on $WS \cup WS'$, called the* law of probabilities.

This representation allows any Kripke structure to be modelled, whether it is S5 or KD45.

Definition 12. *Each explicit Kripke structure $\mathscr{M} = \langle W, R, \mu, V \rangle$ corresponds to a symbolic Kripke structure* $\mathrm{symb}(\mathscr{M}) = \langle W^r, R^r_a, \mu^r_a \rangle$ *defined as follows:*

- $W^r := \bigvee_{w \in W} V(w)$*;*
- $R^r_a := \bigvee_{(w,w') \in R_a} V(w) \wedge \mathrm{succ}(V(w'))$ *;*
- $\mu^r_a = \sum_{(w,w') \in W^2} \mu_a(w)(w') \times V(w) \wedge \mathrm{succ}(V(w'))$.

Here, it is important to see that the "sums" are disjoint if $\mathscr{M}$ is V-inj. Indeed, whether it is for the or logical on R^r or for the sum on μ^r, if $\mathscr{M}$ is V-inj, the relation $V(w) \wedge \mathrm{succ}(V(w'))$ is unique. If it is not, there is a loss of information by collasping on the structure.

Definition 13 (Symbolic Kripke into explicit : $\mathrm{symb}^{-1}(F)$)**.** *Each symbolic Kripke structure $F = \langle W^r, R^r, \mu^r \rangle$ corresponds to the unique Kripke structure* $\mathrm{symb}^{-1}(F) = \langle W, R, \mu, V \rangle$ *defined as follows:*

- $W := \mathrm{supp}(W^r)$*;*
- $V := (v \in W \mapsto v)$*;*
- $R_a := \{(v, v') \in W^2 \mid v \cup \mathrm{succ}(v') \in \mathrm{supp}(R^r_a)\}$*;*
- $\mu_a := (v \in W \mapsto (v' \in W \mapsto \mu^r_a(v \cup \mathrm{succ}(v')) \times \mathrm{norm}_a(v)))$*, where* $\mathrm{norm}_a(v) = 1/\sum_{v' \in W} \mu^r_a(v \cup \mathrm{succ}(v'))$ *if the denominator is nonzero, and* 0 *otherwise.*

Lemma 14. *If $\mathscr{M}$ is V-injective, $\mathscr{M} \simeq$* $\mathrm{symb}^{-1}(\mathrm{symb}(\mathscr{M}))$.

Proof. Let's suppose $\mathscr{M}_l = \langle W_l, R_l, \mu_l, V_l \rangle$ V-inj and $\mathscr{M}_r = \mathrm{symb}^{-1}(\mathrm{symb}(\mathscr{M}_a)) = \langle W_r, R_r, \mu_r, V_r \rangle$. Let f be $W_l \to W_r$ such that $f(w_l) = V(w_l)$.
$W_r = \mathrm{supp}(\bigvee_{w \in W_l} V(w))$ and V_l is injective. So for all $w_l \in W_l$ there is one only $w_r \in W_r$ such that $f(w_l) = V(w_l) = w_r$.

Let $w_r, w'_r \in W^2_r$ and $w_l, w'_l \in W^2_l$ s.t. $w_r = f^{-1}(w_l) = V(w_l)$, $w'_r = f^{-1}(w'_l) = V(w'_l)$,

$$
\begin{aligned}
&(w_r, w'_r) \in R_r \\
&\iff w_r \cup \mathrm{succ}(w'_r) \models \mathrm{supp}(\bigvee_{(w,w') \in R_l} V(w) \wedge \mathrm{succ}(V(w'))) \\
&\iff V(w_l) \cup \mathrm{succ}(V(w'_l)) \models \mathrm{supp}(\bigvee_{(w,w') \in R_l} V(w) \wedge \mathrm{succ}(V(w'))) \\
&\iff (w_l, w'_l) \in R_l
\end{aligned}
$$

$\mu_r(w_r, w'_r)$

- $(\sum_{w_l, w'_l \in W^2_l} \mu_l(w_l)(w'_l) \times V(w_l) \times \mathrm{succ}(V(w'_l)))(V(w_l) \cup \mathrm{succ}(V(w'_l)) \times norm_a(w'_l)$
  with $norm_a(w'_l) = 1/\sum_{w_l \in W_l} \sum_{w,w' \in W^2_l} \mu_l(w)(w') \times V(w) \times \mathrm{succ}(V(w'))$
- $(\sum_{w \in f^{-1}(w_r), w' \in f^{-1}(w'_r)} \mu_l(w_l)(w'_l) \times V(w_l) \times \mathrm{succ}(V(w'_l)))(V(w_l) \cup \mathrm{succ}(V(w'_l)) \times norm_a(w'_l)$
  with V_l injective
- $(\mu_l(w_l)(w'_l) \times V(w_l) \times \mathrm{succ}(V(w'_l)))(V(w_l) \cup \mathrm{succ}(V(w'_l)) \times norm_a(w'_l)$
- $\mu_l(w_l)(w'_l) \times norm_a(w'_l)$
  We have by definition of explicit Kripke structure : $\forall w \in W_l$, $\sum_{w'} \mu_l(w)(w') = 1$ and with distinct sums, we have $norm_a(w_r) = 1$.
- $\mu_l(w_l)(w'_l)$

□

Though a symbolic Kripke structure represents a unique (V-injective) Kripke structure, the converse is not true. When building a symbolic representation of a V-injective Kripke structure, the law of worlds is fixed, but there are two degrees of freedom: (i) the law of probabilities does not have to be normalized, and (ii) the laws of knowledge and probabilities can assign values to any assignment in $2^{WS \cup WS'}$, even when it does not correspond to any pair of worlds.
Moreover, these "symbolic" Kripke structures are very generic, since, as we already mentioned, there is no constraint as to how the Boolean and pseudo-Boolean functions defining the three "laws" are represented. The choice of concrete representations can thus depend on the intended tradeoff between spatial and temporal efficiency for various applications. Anyway, it should be clear that symbolic Kripke structure can yield exponential space savings: a trivial example is $F = \langle \top, \top, 1 \rangle$ (where 1 here is the constant PBF), which represents a Kripke structure with $2^{|WS|}$ distinct worlds.

3.2 Model checking on static structures

In order to decide whether (the Kripke structure represented by) a given symbolic Kripke structure is a model of an $\mathscr{L}_{PEL}$ formula, we can build a Boolean

function on WS representing the set of all worlds of the structure in which the formula holds. This can be done using dynamic programming thanks to the following inductive definition.

Definition 15 (Boolean translation of formula)**.** *Let $F = \langle W^r, R^r, \mu^r \rangle$ be a symbolic Kripke structure and ϕ be a formula in $\mathscr{L}_{PEL}$. The* Boolean translation of ϕ in F, *denoted $||\phi||_F$, is the Boolean function defined inductively as follows:*

$$||p||_F := p$$
$$||\neg\phi||_F := 1 - ||\phi||_{\text{symb}(\mathscr{M})}$$
$$||\phi \wedge \psi||_F := ||\phi||_{\text{symb}(\mathscr{M})} \times ||\psi||_{\text{symb}(\mathscr{M})}$$
$$||\mathrm{K}_a\phi||_F := \mathrm{Forget}^{\forall}_{WS'}(\mathrm{succ}(W^r) \wedge R^r_a \rightarrow \mathrm{succ}(||\phi||_{\text{symb}(\mathscr{M})}))$$
$$||\alpha_1 \mathrm{Pr}_a(\phi_1) + \cdots + \alpha_k \mathrm{Pr}_a(\phi_k) \geq \beta||_F := \mathrm{Cut}_{\geq\beta}(\textstyle\sum_{i=1}^k \alpha_i \times ||\mathrm{Pr}_a(\phi_i)||_F) \textit{ where :}$$

$$||\mathrm{Pr}_a(\phi)||_F = \mathrm{Marg}_{\Sigma,WS'}(\mathrm{succ}(W^r) \times \mathrm{succ}(||\phi||_{\text{symb}(\mathscr{M})}) \times \mathit{norm}(\mu^r_a))$$
$$\mathit{norm}(\mu^r_a) = \frac{\mu^r_a}{\mathrm{Marg}_{\Sigma,WS'}(\mu^r_a)}$$

We said that the law of probabilities does not have to be normalized, but for modelcheck a Pr_a formula, we need to find normalized PBFs to be able to compare the PBF values with the β value of the formula. In fact, we just need before applying a model checking to normalize the F structure in this way: $\mathit{norm}(F) = \langle W^r, R^r, \mathit{norm}(\mu^r) \rangle$
Remark that using $||\mathrm{Pr}_a(\phi)||_F$ is a bit of an notational abuse, since $\mathrm{Pr}_a(\phi)$ is not a formula, and indeed $||\mathrm{Pr}_a(\phi)||_F$ is not a Boolean function. Now, it should be clear that the complexity of building the Boolean translation of a formula depends on the concrete representations used. Nonetheless, note that all operations need – sum, product, renamings, cut, marginalization/forgetting – can be considered as elementary operations on PBF representations; they are notably used by Fargier *et al.* [2014] to compare the efficiency of several languages of the decision diagram family.
The following result can be proved by induction :

Proposition 16 (symbolic model checking on PEL)**.** *Let $\mathscr{M} = \langle W, R, \mu, V \rangle$ be a V-inj Kripke structure. For any formula ϕ in $\mathscr{L}_{PEL}$ any $w \in W$, it holds that $\mathscr{M}, w \models \phi \iff ||\phi||_{\text{symb}(\mathscr{M})}(V(w)) = 1$.*

Proof. The proof is by induction on ϕ. Cases of atomic propositions, negation and conjunction are immediate.

$\mathscr{M}, w \models \neg\phi$
- $\iff \mathscr{M}, w \not\models \phi$
- $\iff ||\phi||_{\text{symb}(\mathscr{M})}(V(w)) \neq 1$ (hyp)
- $\iff 1 - ||\phi||_{\text{symb}(\mathscr{M})}(V(w)) \neq 0$
- $\iff ||\neg\phi||_{\text{symb}(\mathscr{M})} = 1$ (Domain of $||\phi||_{\text{symb}(\mathscr{M})}(V(w))$ is boolean : [0,1])

$\mathscr{M}, w \models \phi \wedge \psi$
- $\iff \mathscr{M}, w \models \phi \wedge \mathscr{M}, w \models \psi$
- $\iff ||\phi||_{\text{symb}(\mathscr{M})}(V(w)) = 1$ and $||\psi||_{\text{symb}(\mathscr{M})}(V(w)) = 1$ (hyp)
- $\iff ||\phi||_{\text{symb}(\mathscr{M})}(V(w)) \times ||\psi||_{\text{symb}(\mathscr{M})}(V(w)) = 1$ (Domain of $||\phi||_{\text{symb}(\mathscr{M})}(V(w))$ and $||\psi||_{\text{symb}(\mathscr{M})}(V(w))$ are boolean : [0,1])
- $\iff (||\phi||_{\text{symb}(\mathscr{M})} \times ||\psi||_{\text{symb}(\mathscr{M})})(V(w)) = 1$
- $\iff ||\phi \wedge \psi||_{\text{symb}(\mathscr{M})}(V(w)) = 1$

$\mathscr{M}, w \models \mathrm{K}_a\phi$
- $\iff \forall w' \in W$ s.t. $(w, w') \in R_a : \mathscr{M}, w' \models \phi$
- $\iff \forall w' \in W$ s.t. $V(w) \cup \mathrm{succ}(V(w')) \in \mathrm{supp}(R^r_a) : ||\phi||_{\text{symb}(\mathscr{M})}(V(w')) = 1$ (definition 12 + hyp)
- $\iff \forall w' \in W : R^r_a(V(w) \cup \mathrm{succ}(V(w')) = 1 \rightarrow ||\phi||_{\text{symb}(\mathscr{M})}(V(w')) = 1$
- $\iff \forall v' \in 2^{WS'} : v' \models \mathrm{succ}(W^r) \rightarrow (R^r_a(V(w) \cup v') = 1 \rightarrow \mathrm{succ}(||\phi||_{\text{symb}(\mathscr{M})}(v')) = 1))$
- $\iff \forall v' \in 2^{WS'} : \neg v' \models \mathrm{succ}(W^r) \vee \neg V(w) \cup v' \models R^r_a \vee \mathrm{succ}(||\phi||_{\text{symb}(\mathscr{M})}(v')) = 1$
- $\iff \forall v' \in 2^{WS'} : (V(w) \cup v' \models \mathrm{succ}(W^r) \wedge R^r_a) \rightarrow \mathrm{succ}(||\phi||_{\text{symb}(\mathscr{M})}(v')) = 1$
- $\iff \mathrm{Forget}^{\forall}_{WS'}(\mathrm{succ}(W^r) \wedge R^r_a \rightarrow \mathrm{succ}(||\phi||_{\text{symb}(\mathscr{M})}))(V(w)) = 1$
- $\iff ||\mathrm{K}_a\phi||_{\text{symb}(\mathscr{M})}(V(w)) = 1$

We suppose that μ^r_a is normalized.
$\mathscr{M}, w \models \alpha_1 \mathrm{Pr}_a(\phi_1) + \cdots + \alpha_k \mathrm{Pr}_a(\phi_k) \geq \beta$

- $\iff \sum_{i=1}^k \alpha_i \mu_a(\mathscr{M}, w)(\phi_i) \geq \beta$ with $\mu_a(\mathscr{M}, w)(\phi_i) := \sum_{w' : \mathscr{M}, w' \models \phi_i} \mu_a(w)(w')$.
- $\iff \sum_{i=1}^k \alpha_i \sum_{w' : \mathscr{M}, w' \models \phi_i} \mu_a(w)(w') \geq \beta$ (rewrite)
- $\iff \sum_{i=1}^k \alpha_i \sum_{w' : \mathscr{M}, w' \models \phi_i} \mu^r_a(V(w) \cup \mathrm{succ}(V(w'))) \geq \beta$ (with definition 12)
- $\iff \sum_{i=1}^k \alpha_i \sum_{v \in 2^{WS} : \mathrm{succ}(W^r) \times \mathrm{succ}(||\phi||_{\text{symb}(\mathscr{M})}(v)) = 1} \mu^r_a(V(w) \cup \mathrm{succ}(v)) \geq \beta$ (with the induction hypothesis that $\mathscr{M}, w \models \phi \iff ||\phi||_{\text{symb}(\mathscr{M})}(V(w)) = 1$ with $v = V(w')$ and the addition of W^r in order to respect $w' \in W$ which become $v \models W^r$)
- $\iff \sum_{i=1}^k \alpha_i \sum_{v \in 2^{WS}} (\mu^r_a(V(w) \cup \mathrm{succ}(v)) \times \mathrm{succ}(W^r) \times \mathrm{succ}(||\phi||_{\text{symb}}(\mathscr{M})))(V(w) \cup \mathrm{succ}(v)) \geq \beta$ ($\mathrm{succ}(W^r) \times \mathrm{succ}(||\,\text{symb}(\phi)||_(v))$ are purely boolean, and that's why we remove the $= 1$. It works as a filter under $\sum$ and also in multiplication of PBF.)
- $\iff \sum_{i=1}^k \alpha_i \mathrm{Marg}_{\Sigma,WS}((\mu^r_a(V(w) \cup \mathrm{succ}(v)) \times \mathrm{succ}(W^r) \times \mathrm{succ}(||\phi||_{\text{symb}(\mathscr{M})})))(V(w)) \geq \beta$ (definition of Marginalisation)
- $\iff \sum_{i=1}^k \alpha_i ||\mathrm{Pr}_a(\phi_i)||_{\text{symb}(\mathscr{M})}(V(w)) \geq \beta$ (definition 15)
- $\iff \mathrm{Cut}_{\geq\beta}(\sum_{i=1}^k \alpha_i \times ||\mathrm{Pr}_a(\phi_i)||_{\text{symb}(\mathscr{M})})(V(w))$

$$\overset{\text{(definition 15)}}{\iff} ||\alpha_1 \Pr_a(\phi_1) + \cdots + \alpha_k \Pr_a(\phi_k) \geq \beta||_{\text{symb}(\mathscr{M})}(V(w))$$

□

3.3 Symbolic updates

Now that we have defined a symbolic Kripke structure and showed how PEL model checking can be done on it, we will show how these structures can be updated through symbolic update models, which finally yields a model checking algorithm for PDEL (restricted to propositional preconditions).

Symbolic representation of update models. We want to use the same principle as for Kripke structures to represent the accessibility relation and the probability functions. But an event is not labeled with a valuation, but with precondition and postcondition functions. If we consider only update models with propositional formulas as preconditions (which are a lot less expressive, but still sufficient for most real-life games, in which the applicability of actions only depends on the objective state of the game and not on what players know about it), it is possible to view precondition and postcondition functions as valuations, by representing them into propositional *action theories* from classical planning. An action theory is a propositional formula θ on $WS \cup WS+$ (where $WS+$ is, once again, a set of fresh propositional symbols in one-to-one correspondance with those in WS via some bijection after, of which we denote before the reciprocal, and that we overload in the same way as succ and pred) that describes the effects of an action (or rather, here, an event) in the following way: given two valuations $v, v' \in 2^{WS}$, world state v' is a possible outcome of applying the action in world state v if and only if $v \cup \text{after}(v') \models \theta$.

Definition 17 (propositional event theory)**.** *Let $\mathscr{E} = \langle E, R^E, \mu^E, pre, post\rangle$ be an update model with propositional preconditions. The* propositional event theory *of event $e \in E$ is the Boolean function θ_e defined by the propositional formula $pre(e) \wedge \theta_e^{post}$, where* $\theta_e^{post} = \bigwedge_{x \in WS} (post(e)(x) \leftrightarrow \text{after}(x))$

We can break down the θ_e definition of a e event with the following lemma, with $post_e(v) := \{x \in WS \mid v \models post(e)(x)\}$.

Lemma 18. *For $v \in 2^{WS}$, $\theta_e^{post}{}_{|v} = \text{Forget}^{\exists}_{WS}(\theta_e^{post} \wedge v) = \text{after}(post_e(v))$*

In other words, as θ_e permit to associate to each $v \in 2^{WS}$ a unique $post_e(v)$ and we can see that $v \wedge \theta_e^{post} = v \wedge \text{after}(post_e(v))$

We say that an event e in an update model is *atomic* if $|\text{supp}(pre(e))| = 1$, i.e., the event only applies to a single valuation in 2^{WS}: it only describes the transition from one complete valuation to another. We call *event-atomic* an update model of which all events are atomic. Note that the propositional event theory of atomic events can be represented by a complete term over $WS \cup WS+$, or equivalently, by a single valuation in $2^{WS \cup WS+}$. Thus, an event-atomic update model can simply be represented by a (static) Kripke structure on $WS \cup WS+$ (of which the valuation of each world can be interpreted as an atomic event theory) – and to any Kripke structure on $WS \cup WS+$ corresponds a unique event-atomic update model. Moreover, it can be shown (it is a consequence of our results) that any update model (with propositional preconditions) can be represented as an equivalent event-atomic update model, by separating each event into as many atomic events as needed, duplicating all accessibility arcs and probabilities (modulo a final normalization step).
We can now show how an update model can be translated into a symbolic representation, and finally show how to compute the product update on symbolic structures.

Definition 19 (symbolic update model)**.** *Let $\mathscr{E} = \langle E, R^E, pre, post, \mu^E\rangle$ be a update model with propositional preconditions; its symbolic representation is the symbolic Kripke structure* $\text{symb}(\mathscr{E}) = \langle W^r, R^r, \mu^r\rangle$ *on $WS \cup WS+$ defined as follows:*

- $W^r := \bigvee_{e \in E} \theta_e$;
- $R^r_a := \bigvee_{(e,e') \in R^E} \theta_e \wedge \text{succ}(\theta_{e'})$;
- $\mu^r_a := \sum_{(e,e') \in E^2} \theta_e \times \text{succ}(\theta_{e'}) \times \mu^E_a(e)(e')$.

Thanks to the symbolic representation definitions of the symbolic Kripke structures and the symbolic update models, we can now define the symbolic product update.
By the way, to keep model checking valid, we need to don't loose information with collapsing. So we need to keep in mind that $\mathscr{M} \otimes \mathscr{E}$ must be V-injective.
So, we need to describe some sufficient conditions for everything to go well. We define a new constraint on the event structure: the Transition-injectivity.

Definition 20 ($\mathscr{E}$ Transition-injective)**.** *$\mathscr{E}$ is said to be transition-injective (T-inj) wrt ϕ iff $\forall e, e' \in E^2$, $\forall v, v' \in 2^{WS}$ s.t. $v \models pre(e) \wedge \phi$ and $v' \models pre(e') \wedge \phi$, $post_e(v) = post_{e'}(v') \implies v = v', e = e'$.*

In other words we can describe the Transition-injectivity as follow : $\forall v \neq v' \in 2^{WS}$ or $e \neq e' \in E^2 \implies post_e(v) \neq post_{e'}(v')$.
It allows us, regardless of the starting worlds responding to a ϕ formula, to guarantee that there will not be two worlds of the same valuation : $post_e(v) = post_{e'}(v')$ only if $v = v'$ and $e = e'$, and v are all differents by the V-injectivity of $\mathscr{M}$, so $V((w,e))$ can't have duplicate (i.e.to have a Valuation-injective structure). Lemma 23 and Definition 20 permit to get $\mathscr{M} \otimes \mathscr{E}$ V-inj avoiding a collapsing of worlds of the same valuation with the symb() function, so we can have a symbolic product update, and later a valid model checking.

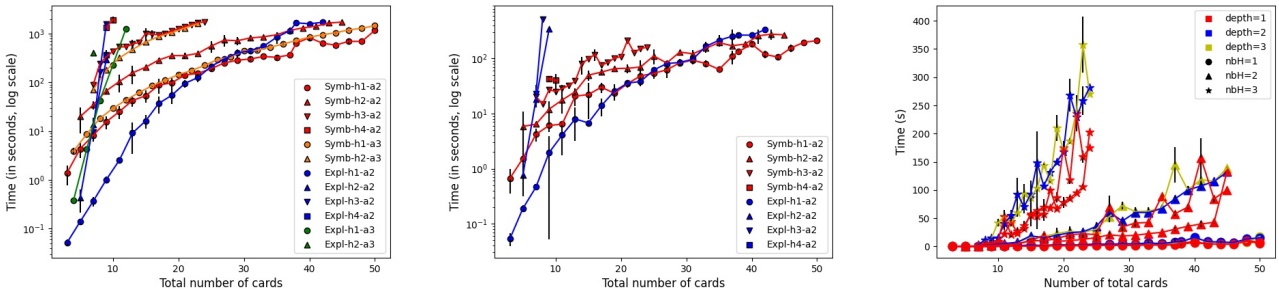

Figure 3: Computation time for the creation of Kripke structures (left) and for updating knowledge (middle) in their explicit and symbolic versions, and for model checking PEL formulas on symbolic structures. "hX-aY" means Y agents with X cards in hand. "depth" is the modal depth of the formula, "nbH" is the number of cards in each player's hand.$\mathscr{M}$

Proposition 21. *Forall $\mathscr{M}$ V-inj s.t. $W \models \phi$ and forall $\mathscr{E}$ T-inj wrt ϕ, $\mathscr{M} \otimes \mathscr{E}$ is V-inj.*

Proof. Let $w, w' \in W$ with $V(w) \models \phi$ and $V(w') \models \phi$ and by the definition of $T-injectivity$, $\forall V(w), V(w')' \in 2^{WS}$, $\forall e, e' \in E$, $\forall V(w) \neq V(w')$ or $e \neq e' \implies post_e(V(w)) \neq post_{e'}(V(w'))$, so $V^{\otimes}((w,e)) \neq V^{\otimes}((w',e'))$. Consequently, $\mathscr{M} \otimes \mathscr{E}$ is V-inj. □

Definition 22 (symbolic product update)**.** *Let $F = \langle W_F^r, R_F^r, \mu_F^r \rangle$ be a symbolic Kripke structure on WS and $\chi = \langle W_\chi^r, R_\chi^r, \mu_\chi^r \rangle$ be a symbolic update model on $WS \cup WS+$ of a Transition-injective Event model (with propositional precondition, as required in definition of T-injective). The* symbolic product update *of F by χ is the symbolic Kripke structure on WS defined as*

- $W^r = \text{before}(\text{Forget}^{\exists}_{WS}(W_F^r \wedge W_\chi^r))$
- $R^r = \text{before}(\text{Forget}^{\exists}_{WS}(R_F^r \wedge R_\chi^r))$
- $\mu^r = \frac{\alpha}{\text{Marg}_{\Sigma,WS'}(\alpha)}$ *with* $\alpha = \text{before}(\text{Marg}_{\Sigma,WS}(\mu_F^r \times \mu_\chi^r))$

Note that this definition implies that WS is "quadruplicated", since we need fresh symbols to represent $\text{succ}(\text{after}(x))$ for $x \in WS$. We implicitly extend succ to range over $WS \cup WS+$ and after to range over $WS \cup WS'$, making sure that $\text{succ} \circ \text{after} = \text{after} \circ \text{succ}$. The resulting symbolic Kripke structure is on WS because the worlds in the product update are pairs (w,e), on $WS \cup WS'$, which is being forgotten and renamed.
With goods properties on $\mathscr{M}$ (V-injective) and on $\mathscr{E}$ (T-injective), we have :

Lemma 23. *For any V-inj $\mathscr{M}$ and any T-inj $\mathscr{E}$ with propositional preconditions,* $\text{symb}(\mathscr{M} \otimes \mathscr{E}) = \text{symb}(\mathscr{M}) \otimes \text{symb}(\mathscr{E})$

The following result, of which we omit the straightforward but tedious proof, puts all pieces together; then we complete Def. 15 with the case of the update operator.

We note $\mathscr{L}_{PDEL}^{0\text{-}T\text{-}inj}$ the language in which update models have only propositional preconditions and are Transition-injective.

Definition 24. *Let $\mathscr{M} = \langle W, R, \mu^r \rangle$ be a V-inj Kripke structure, $\mathscr{E}$ be a Transition-injective update model, and ϕ be a formula in $\mathscr{L}_{PDEL}^{0\text{-}T\text{-}inj}$.*
The Boolean translation of ϕ in $\text{symb}(\mathscr{M})$, *denoted $||\phi||_{\text{symb}(\mathscr{M})}$, is the Boolean function defined inductively in the same way as in Def. 15, with the following additional case:*
$||[\mathscr{E}, e]\phi||_{\text{symb}(\mathscr{M})} := \text{Forget}^{\exists}_{WS+}(pre(e) \rightarrow (\theta_e^{post} \wedge \text{after}(||\phi||_{\text{symb}(\mathscr{M}) \otimes \text{symb}(\mathscr{E})})))$

We can see $\theta_e^{post} \wedge \text{after}(||\phi||_{\text{symb}(\mathscr{M}) \otimes \text{symb}(\mathscr{E})})$ as $\{w, w^+ \mid w^+ = post_e(w),\ F \times \text{symb}(\mathscr{E}), \text{before}(w^+) \models \phi\}$. So w^+ are worlds after product update that satisfy ϕ, and w the worlds which led to w^+, and forgetting the $WS+$ allows to keep the worlds that led to it. With this new definition on translation, we obtain a proposition to do symbolic model checking on PDEL.

Proposition 25 (symbolic model checking on PDEL)**.** *Let $\mathscr{M}$ be a V-inj Kripke structure. For any formula ϕ in $\mathscr{L}_{PDEL}^{0\text{-}T\text{-}inj}$ and any $w \in W$, it holds that $\mathscr{M}, w \models \phi \iff ||\phi||_{\text{symb}(\mathscr{M})}(V(w)) = 1$.*

Proof. The proof is by induction on ϕ. For case of $\phi \in \mathscr{L}_{PEL}$, see proposition 16.

$\mathscr{M}, w \models [\mathscr{E}, e]\phi$
$\iff \mathscr{M}, w \models pre(e) \implies \mathscr{M} \otimes \mathscr{E}, (w,e) \models \phi$. (Definition 9)
$\iff \mathscr{M}, w \models pre(e) \implies ||\phi||_{\text{symb}(\mathscr{M} \otimes \mathscr{E})}(V^{\otimes}((w,e))) = 1$ (Hyp and Proposition 21)
$\iff pre(e)(V(w)) = 1 \rightarrow ||\phi||_{\text{symb}(\mathscr{M} \otimes \mathscr{E})}(V^{\otimes}((w,e))) = 1$ (preconditons are propositional)
$\iff pre(e)(V(w)) = 1 \rightarrow ||\phi||_{\text{symb}(\mathscr{M}) \otimes \text{symb}(\mathscr{E})}(V^{\otimes}((w,e))) = 1$ (Lemma 23)

$\iff pre(e)(V(w)) = 1 \rightarrow ||\phi||_{\text{symb}(\mathscr{M})\otimes\text{symb}(\mathscr{E})}(post_e(V(w))) = 1$ (Def of $post_e(v)$)

$\iff pre(e)(V(w)) = 1 \rightarrow \text{after}(||\phi||_{\text{symb}(\mathscr{M})\otimes\text{symb}(\mathscr{E})})(\text{after}(post_e(V(w)))) = 1$

$\iff pre(e)(V(w)) = 1 \rightarrow \text{after}(||\phi||_{\text{symb}(\mathscr{M})\otimes\text{symb}(\mathscr{E})})(\theta_e^{post}{}_{|V(w)}) = 1$ (Lemma 18)

$\iff pre(e)(V(w)) = 1 \rightarrow (\theta_e^{post} \wedge \text{after}(||\phi||_{\text{symb}(\mathscr{M})\otimes\text{symb}(\mathscr{E})}))(V(w) \wedge \theta_e^{post}{}_{|V(w)}) = 1$ (Lemma 18) (translation still only work on $\theta_e^{post}{}_{|V(w)}$ and $\theta_e^{post} \models V(w) \wedge \theta_e^{post}{}_{|V(w)}$)

$\iff \text{Forget}^{\exists}_{WS+}(pre(e))(V(w)) = 1 \rightarrow \text{Forget}^{\exists}_{WS+}((\theta_e^{post} \wedge \text{after}(||\phi||_{\text{symb}(\mathscr{M})\otimes\text{symb}(\mathscr{E})})))(\text{Forget}^{\exists}_{WS+}(V(w) \wedge \theta_e^{post}{}_{|V(w)})) = 1$ (forget is free in *pre* which is on *WS*, on forgetting on right is possible because of the Transition-injectivity of the event model : there is one valuation on $WS \cup WS+$)

$\iff \text{Forget}^{\exists}_{WS+}(pre(e))(V(w)) = 1 \rightarrow \text{Forget}^{\exists}_{WS+}((\theta_e^{post} \wedge \text{after}(||\phi||_{\text{symb}(\mathscr{M})\otimes\text{symb}(\mathscr{E})})))(V(w)) = 1$

$\iff \text{Forget}^{\exists}_{WS+}(pre(e) \rightarrow (\theta_e^{post} \wedge \text{after}(||\phi||_{\text{symb}(\mathscr{M})\otimes\text{symb}(\mathscr{E})})))(V(w)) = 1$

$\iff ||[\mathscr{E}, e]\phi||_F(V(w)) = 1$

□

4 Experiments on Hanabi

We now report on experiments we ran on our Python implementation of the PDEL framework, with both on "explicit" and symbolic Kripke structures. The concrete language we used to represent PBFs is that of ADDs Bahar *et al.* [1997], which, while strictly less succinct than other decision diagram languages such as SLDDs, has efficient algorithms for almost all the operators we need Fargier *et al.* [2014]. We used our own Python implementation of an ADD package. The variable order in ADDs is of crucial importance to get small representations; we use a natural order that arise when compiling formulas, after having experimentally checked that varying the position of variable groups (e.g., putting variables from *WS* closer to WS' or to $WS+$) did not seem to yield much better results.

On the ADD structure, basic operations are polynomial in the size of the structure. For example applying an operator (or, and, sum, product) between 2 adds is quadratic. The problem is for Marginalisation or Forget: it's quadratic in the worst case, for *a single variable*, so it can explode when you need to forget a lot. However (1) it doesn't explode if the variables to be forgotten are at the end of the order, and (2) in practice it remains reasonable because there are a lot of symmetries in the structures, before and after marginalisations. The aim of the experiments is precisely to see that in practice, at scales sufficient to represent Hanabi, it works.

All experiments have been performed on a server with 4 AMD Opteron 6282SE 2.6GHz processors, 64 cores and 512G RAM (but no experiment exploited any parallelism or multithreading. This power has not been exploited and the calculation times are roughly the same on a laptop). Each experiment has been run five times; the graphs show the average time, with a black line indicating standard deviation.

In Fig. 3 (left) , we can compare the creation times of static Kripke structures with event structures, in the explicit and symbolic cases, with varying total number of cards and number of cards in each player's hand, with a timeout of 1800 seconds. For 2 agents, we can compare the red curves (symbolic) and blue curves (explicit). For nbH=1, the gain appears only from 25 cards. For nbH=2, the difference is more drastic: with the symbolic approach we can generate the game with 46 cards, versus only 11 for the explicit one. We can make similar observations for 3 agents. Let us note that the creation time of the symbolic structures is mainly used for creating the update models, which have twice as many variables as the static structures (because of the variables after(var)). For testing the application of the product update, we used the most complex action of the game: playing a card. It is programmed as an "action pipeline": play a card, then shift the cards in hand to fill the "hole", draw a new card, change turns. Results (Fig. 3, middle) are similar to those about creation time; the gain becomes flagrant for nbH=2 or 3.

For model checking on structures (only with 2 agents), we use a number of formulas, up to a depth of 3, using K and μ operators, and apply them on different Kripke structures with varying *nbH*. For the explicit structures that could be generated, model checking takes less than 5 seconds for almost all formulas. For symbolic structures (Fig. 3, right), propositional formulas are instantaneous, but the model checking of formulas with K or μ operators requires forgetting variables and time-consuming calculations. Yet, it is feasible in practice, and it compares very advantageously to explicit structures when those cannot even be generated.

5 Experiments with a KBP

A KBP can be seen as a plan in a form of program with conditions and loops presented by **?**. We implemented a small one, presented in algorithm 1 and ran it to test a basic type of planification in realistic contexte.

By varying the distribution of the cards at the beginning of the game, here is what we were able to obtain as a sequence of actions, for nbA=2, nbC=10, with pointed world as agent *a* has {W5, W2, W4} and *b* has {W2, W1, W4} in hand:

- b-has-v=1-at-1 because of [!event] K_b he-can-play-sth
- b-plays-p1, because K_b b-can-play-1

Algorithm 1 KBP

Require: $(\mathscr{M}, w)$ epistemic model, E list of events, a an agent, nbH number of cards in hands, A list of agents, $nbred$ and $nbblue$ number of current red and blue tokens
Ensure: An action of E
  for all $p \in \{1..nbH\}$ **do**
    if $\mathscr{M}, w \models \mathrm{K}_a$ a-can-play(p) **then**
      return a-plays-p
  for all $event \in E$ in announcement **do**
    for all $b \in A \backslash \{a\}$ **do**
      if $\mathscr{M}, w \models [!event]K_b$ b-can-play-something **then**
        return event
  $\gamma \leftarrow$ 0.66 if $nbred$=0, 0.80 if $nbred$=1, 0.90 if $nbred$=2
  for all $p \in \{1..nbH\}$ **do**
    if $\mathscr{M}, w \models \mathrm{Pr}_a$ a-can-play-(p) $\geq \gamma$ **then**
      return a-plays-p
  if $nbblue = 0$ **then**
    return a-discards-nbH
  return random-annoucement

- b-has-v=2-at-1 because of [!event] K_b he-can-play-sth
- b-plays-p1, because K_b b-can-play-1
- random-announcement
- b-plays-p2, because of Pr_b b-can-play-2 ≥ 0.66

The relatively low number of cards allows here to have only yellow cards. Thus, players already know a large part of the information on their cards. All they have to do now is learn the value of their cards. Nevertheless, we can see from the sequence of actions generated by the KBP that the actions are well targeted and that the model checking with probabilities even allows agent b to take risks by playing a card that enters the game with a 66% chance.
With differents experiences with nbA=2, nbH=3, we have some results. With nbC=10 and 20, 49 and 56 are respectivly the maximal number of possible model checking in this KBP. For 3 runs with nbC=10, is 18.88, with standard deviation as 19.49 and the average time to apply a model checking is 2.99 seconds with standard deviation 3.56. For 2 runs with nbC=20, the average number of model checking necessary in the KBP is 21.08, with standard deviation as 22.42 and the average time to apply a model checking is 19.53 seconds with standard deviation 32.07.

6 Conclusion

In this article we have seen that we can do model checking on probabilistic Kripke structures using pseudo-boolean functions, implemented with structures such as ADDs.
The primary interest to test model checking with PDEL with Hanabi as example is to see the card game aspect with inherent probabilities. It seems quite justified to ask "what is the probability that this card is 2 green", "is this probability greater than that this card is 2 blue"? In this game it is not always possible to make a safe move, so probabilistic reasoning is still present. So our future work will rely on symbolic structures to apply planning techniques in order to compute strategies for Hanabi.

References

R. Iris Bahar, Erica A. Frohm, Charles M. Gaona, Gary D. Hachtel, Enrico Macii, Abelardo Pardo, and Fabio Somenzi. Algebraic decision diagrams and their applications. *Formal Methods in System Design*, 10(2/3):171–206, 1997.

Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The hanabi challenge: A new frontier for AI research. *Artif. Intell.*, 280:103216, 2020.

Antoine Bauza. Hanabi. `http://www.antoinebauza.fr/?tag=hanabi`, 2010. accessed 2018-06-11.

Thomas Bolander. A gentle introduction to epistemic planning: The DEL approach. In *Proceedings of the Ninth Workshop on Methods for Modalities, M4M@ICLA 2017, Indian Institute of Technology, Kanpur, India, 8th to 10th January 2017*, pages 1–22, 2017.

Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Computers*, 35(8):677–691, 1986.

Tristan Charrier and François Schwarzentruber. A succinct language for dynamic epistemic logic. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 123–131, 2017.

Tristan Charrier, Sophie Pinchinat, and François Schwarzentruber. Symbolic model checking of public announcement protocols. *J. Log. Comput.*, 29(8):1211–1249, 2019.

Adrian Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, pages 229–264, 2002.

Ronald Fagin and Joseph Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367, 1994.

Hélène Fargier, Pierre Marquis, and Nicolas Schmidt. Semiring labelled decision diagrams, revisited: Canonicity and spatial efficiency issues. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 884–890. IJCAI/AAAI, 2013.

Hélène Fargier, Pierre Marquis, Alexandre Niveau, and Nicolas Schmidt. A knowledge compilation map for ordered real-valued decision diagrams. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1049–1055. AAAI Press, 2014.

Malvin Gattinger. *New directions in Model Checking Dynamic Epistemic Logic*. PhD thesis, Universiteit van Amsterdam, 2018.

Hanabi rules. https://www.ultraboardgames.com/hanabi/game-rules.php, 2021. accessed 2021-01-16.

Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014.

Barteld P. Kooi. Probabilistic dynamic epistemic logic. *Journal of Logic, Language and Information*, 12(4):381–408, 2003.

Bastien Maubert, Sophie Pinchinat, and François Schwarzentruber. Reachability games in dynamic epistemic logic. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 499–505. ijcai.org, 2019.

Kenneth L. McMillan. *Symbolic model checking*. Kluwer, 1993.

Scott Sanner and David A. McAllester. Affine algebraic decision diagrams (aadds) and their application to structured probabilistic inference. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 1384–1390. Professional Book Center, 2005.

Afsaneh Shirazi and Eyal Amir. Factored models for probabilistic modal logic. In Dieter Fox and Carla P. Gomes, editors, *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 541–547. AAAI Press, 2008.

Johan van Benthem, Jelle Gerbrandy, and Barteld P. Kooi. Dynamic update with probabilities. Prepublication/extended version PP-2006-21 (March 15th, 2006) [https://www.illc.uva.nl/Research/Publications/Reports/PP/#PP-2006-21], ILLC (Institute for Logic, Language and Computation), University of Amsterdam, 2006.

Johan van Benthem, Jan van Eijck, and Barteld P. Kooi. Logics of communication and change. *Inf. Comput.*, 204(11):1620–1662, 2006.

Johan van Benthem, Jelle Gerbrandy, and Barteld P. Kooi. Dynamic update with probabilities. *Studia Logica*, 93(1):67–96, 2009.

Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. Symbolic model checking for Dynamic Epistemic Logic — S5 and beyond. *Journal of Logic and Computation*, 28(2):367–402, 11 2017.

Nic Wilson. Decision diagrams for the computation of semiring valuations. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 331–336. Professional Book Center, 2005.