Towards Symbolic and Succinct Perspective Shifts

Malvin Gattinger
University of Goningen
`malvin@w4eg.eu`

EpiP workshop at ICAPS 2020, Nancy / online[*]

Abstract

Recent work on Epistemic Planning uses Dynamic Epistemic Logic (DEL) to formalise and solve multi-agent planning problems. DEL allows agents to take into account knowledge of others by computing perspective shifts. So far, perspective shifts are usually defined on explicit Kripke models for S5.

Here we first generalise perspective shifts from S5 to K. We then show how perspective shifts can be computed without explicit Kripke models. Concretely, we define perspective shifting on symbolic structures and succinct models. Both are compact representations proposed in the literature to speed up model checking DEL. Our definitions can help to implement multi-agent epistemic planning more efficiently in the future.

1 Introduction

Starting with Löwe, Pacuit, and Witzel [LPW11] and Bolander and Andersen [BA11], researchers have argued for Dynamic Epistemic Logic (DEL) as a framework for epistemic planning. A next step was taken by Engesser et al. [Eng+17] who showed how to compute perspective shifts on Kripke models and defined *implicitly coordinated plans*, where agents not only know a plan, but also know that others can execute their part. The method was further optimised by partly the same authors using Monte-Carlo tree search [Rei+19]. This helps for planning problems with many steps, but the authors also say that the "most impactful limitation is the memory consumption of the epistemic states" and they "identified the size of the DEL models to be the reason why [their] method is not able to solve larger problems" [Rei+19].

The standard semantics for DEL are Kripke models and action models where possible worlds and events are listed explicitly "one by one". In recent work — at first sight unrelated to planning — researchers presented alternative representations and semantics for DEL which help to speed up model checking. The two main approaches are *symbolic structures* based on Binary Decision Diagrams (BDDs) [Ben+15] and *succinct models* based on Mental Programs [CS17]. While the work on symbolic structures focused on classical examples from the DEL literature (Muddy Children etc.) and implementing the model checker SMCDEL

[Gat19], the work on succinct models also includes theoretical results on the computational complexity. In particular, Charrier and Schwarzentruber [CS17] show that model checking is still in PSPACE when using succinct models. This is somewhat surprising, because many Kripke models and thus model checking inputs can be exponentially smaller when represented succinctly. Given this practical and theoretical success, it is natural to ask whether the new representations can be useful for epistemic planning.

Here we show how perspective shifts can be done directly within the compact representations. We hope that the resulting *symbolic and succinct perspective shifts* will allow us to implement DEL planning more efficiently in the future.

This article is structured as follows. We first recall the standard Kripke semantics for DEL in Section 2. Then we summarise how Kripke models can be encoded in symbolic structures and succinct models in Section 3 and 4, respectively. In Section 5 we then give definitions of perspective shifts for all three representations, with proofs that they are equivalent. We conclude with ideas for future work.

Throughout the article we use the following definition of Boolean formulas, substitution and quantification.

Definition 1. *We denote a finite set of atomic propositions as V and call it a* vocabulary. *To extend a vocabulary we define fresh atomic variables using primes, as for example in p'. Similarly, for a set of atoms V we define $V' := \{p' \mid p \in V\}$, hence for example $\{p, q\}' = \{p', q'\}$.*

The Boolean language *over a vocabulary V is denoted by $\mathcal{L}_B(V)$ and given by $\beta ::= \top \mid p \mid \neg\beta \mid \beta \wedge \beta$ where $p \in V$. We use the usual connectives $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ as abbreviations.*

We identify a Boolean valuation (also called assignment) with the set of atoms $s \subseteq V$ that it makes true and we write $s \vDash \beta$ if s satisfies β.

For any p, φ and ψ, let $[p \mapsto \varphi]\psi$ denote the result of replacing all occurrences of p with φ in ψ. Similarly, for ordered lists $P, Q \subseteq V$, let $[P \mapsto Q]\psi$ denote the result of simultaneously replacing each $p_i \in P$ with $q_i \in Q$ in ψ.

We use Boolean quantification as follows. For any p and φ, we define $\exists p\varphi := [p \mapsto \top]\varphi \vee [p \mapsto \bot]\varphi$ and $\forall p\varphi := [p \mapsto \top]\varphi \wedge [p \mapsto \bot]\varphi$. For any finite set $P = \{p_1, \ldots, p_n\} \subseteq V$ we define $\exists P\varphi := \exists p_1 \ldots \exists p_n\varphi$ and $\forall P\varphi := \forall p_1 \ldots \forall p_n\varphi$.

Finally, we also apply primes to formulas to replace atoms, for example $((p \wedge r) \rightarrow q)' = ((p' \wedge r') \rightarrow q')$.

---

[*]This file was last updated on 2020-10-08. The latest version is available at `https://malv.in/2020/EpiP-perspective-shifts.pdf`.

2 Dynamic Epistemic Logic

We now recall the syntax of DEL and the standard semantics based on Kripke models.

2.1 Epistemic Language

In the whole article we fix a finite set of agents I.

Definition 2. *Given a vocabulary V, the language of epistemic logic $\mathcal{L}(V)$ extends the Boolean language $\mathcal{L}_B(V)$ from Definition 1 and is given by*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi$$

where $p \in V$, $i \in I$.

As usual, $K_i\varphi$ should be read as "agent i knows that φ". For simplicity we do not include common knowledge here but refer to Ditmarsch, Hoek, and Kooi [DHK07] for more expressive variants of DEL including it.

2.2 Kripke Models

Definition 3. *A frame is a tuple $\mathcal{M} = (W, R)$, where W is a finite set of possible worlds and each $R_i \subseteq W \times W$ is the accessibility relation for agent i.*

A Kripke model for a vocabulary V is a tuple $\mathcal{M} = (W, R, \mathsf{Val})$, where (W, R) is a frame and $\mathsf{Val}\colon W \to \mathcal{P}(V)$ is called the valuation function.

A model is S5 iff all R_i are equivalence relations. In this case we also write $\sim_i$ for R_i.

A pointed model is a pair $(\mathcal{M}, w)$ where $w \in W$ and a multi-pointed model is a pair $(\mathcal{M}, \omega)$ where $\omega \subseteq W$.

Definition 4. *Semantics for $\mathcal{L}(V)$ on pointed Kripke models are given inductively as follows.*

1. *$(\mathcal{M}, w) \vDash \top$ always holds.*

2. *$(\mathcal{M}, w) \vDash p$ iff $p \in \mathsf{Val}(w)$.*

3. *$(\mathcal{M}, w) \vDash \neg\varphi$ iff not $(\mathcal{M}, w) \vDash \varphi$.*

4. *$(\mathcal{M}, w) \vDash \varphi \wedge \psi$ iff $(\mathcal{M}, w) \vDash \varphi$ and $(\mathcal{M}, w) \vDash \psi$*

5. *$(\mathcal{M}, w) \vDash K_i\varphi$ iff for all $v \in W$ we have that $R_i wv$ implies $(\mathcal{M}, v) \vDash \varphi$.*

Example 5. *Let $\mathcal{M} = (W, R, \mathsf{Val})$ be the Kripke model given by $W = \{0, 1, 2, 3\}$ and R (for agents a and b) and Val as shown:*
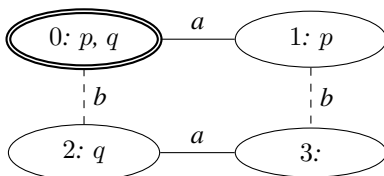
We have $(\mathcal{M}, 0) \vDash p \wedge q \wedge K_a p \wedge \neg K_b p \wedge K_b q \wedge \neg K_a q$.

2.3 Actions and Product Updates

Definition 6. *An action model (sometimes called event model) is a tuple $\mathcal{A} = (A, R^{\mathcal{A}}, \mathsf{pre}, \mathsf{post})$ where A is a finite set of events, $R_i^{\mathcal{A}} \subseteq A \times A$ for each i, $\mathsf{pre}\colon A \to \mathcal{L}(V)$ assigns to each event a precondition and $\mathsf{post}\colon A \times V \to \mathcal{L}_B(V)$ at each event assigns to each atom a postcondition. An action model is S5 iff all $R_i^{\mathcal{A}}$ are equivalence relations.*

Given a Kripke model $\mathcal{M}$ and an action model $\mathcal{A}$, their product is $\mathcal{M} \times \mathcal{A} := (W^{new}, R_i^{new}, \mathsf{Val}^{new})$ where

- $W^{new} := \{(w, a) \in W \times A \mid \mathcal{M}, w \vDash \mathsf{pre}(a)\}$

- $R_i^{new} := \left\{ ((w_1, a_1), (w_2, a_2)) \mid \begin{array}{l} R_i w_1 w_2 \text{ and} \\ R_i^{\mathcal{A}} a_1 a_2 \end{array} \right\}$

- $\mathsf{Val}^{new}((w, a)) := \{p \in V \mid \mathcal{M}, w \vDash \mathsf{post}(a, p)\}$

An action is a pair $(\mathcal{A}, a)$ where $a \in A$. To update a pointed Kripke model with an action we define $(\mathcal{M}, w) \times (\mathcal{A}, a) := (\mathcal{M} \times \mathcal{A}, (w, a))$.

A multi-pointed action is a pair $(\mathcal{A}, \alpha)$ where $\alpha \subseteq A$. To update a multi-pointed Kripke model with a multi-pointed action we let $(\mathcal{M}, \omega) \times (\mathcal{A}, \alpha) := (\mathcal{M} \times \mathcal{A}, \{(w, a) \in \omega \times \alpha \mid (\mathcal{M}, w) \vDash \mathsf{pre}(a)\})$.

Definition 7. *Given a vocabulary V, the language of Dynamic Epistemic Logic $\mathcal{L}_D(V)$ extends $\mathcal{L}(V)$ with dynamic operators for action models and is given by*

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid [\mathcal{A}, a]\varphi$$

where $p \in V$, $i \in I$ and $(\mathcal{A}, a)$ is an action as in Definition 6.

**Definition 8.** *We interpret dynamic operators for action models as follows:*

$$(\mathcal{M}, w) \vDash [\mathcal{A}, a]\varphi \text{ iff } \begin{array}{l} \mathcal{M}, w \vDash \mathsf{pre}(a) \\ \text{implies } \mathcal{M} \times \mathcal{A}, (w, a) \vDash \varphi \end{array}$$

Example 9. *Let $\mathcal{M}$ be the Kripke model from Example 5 and let $\mathcal{A}$ be the action model shown below. This action model consists of two events and we denote their preconditions with $?\varphi$. We do not show any postconditions which is meant to indicate that $\mathsf{post}(a, p) = p$ for all a and all p.*
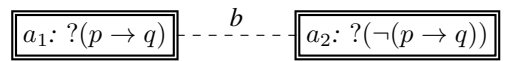
Intuitively, this action $(\mathcal{A}, \{a_1, a_2\})$ tells agent a whether $p \to q$ is true. Moreover, agent b does not learn the value of $p \to q$, but b still learns that a learns whether $p \to q$.

We show the resulting model $(\mathcal{M}, \{0\}) \times (\mathcal{A}, \{a_1, a_2\})$ below. Because in $(\mathcal{M}, 0)$ agent b already knew that q, we have $(\mathcal{M}, 0) \vDash [\mathcal{A}, \{a_1, a_2\}]K_b K_a(p \to q)$.
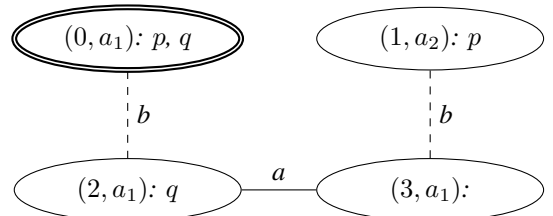
3 DEL on Symbolic Structures

Inspired by symbolic model checking methods developed for temporal logics, symbolic structures for DEL were first presented by Benthem et al. [Ben+15] for S5 with group announcements and later generalised to weaker logics and action models [Ben+18]. We now summarise the basic definitions, but refer to Gattinger [Gat18] for details.

The main idea of symbolic representation is to never spell out the set of all possibilities, in our case the set of all possible or reachable worlds in a Kripke model. Instead, we try to work with a compact representation that contains just enough information to evaluate all formulas, but does away with irrelevant information. For example, the valuation at a world matters, but whether a world is called 'v', 'w', '0' or '42' does not matter for the truth of any formula.

We use Boolean formulas to define and modify these compact representations, but this is only for simplicity. As described by Benthem et al. [Ben+18] we only care about the Boolean *function* and not the particular syntax of a formula. An actual implementation such as SMCDEL [Gat19] uses Binary Decision Diagrams (BDDs) as introduced by Bryant [Bry86], instead of formulas.

3.1 Knowledge and Belief Structures

Definition 10. *A* knowledge structure *is a tuple* $\mathcal{F} = (V, \theta, O)$ *where* V *is a finite vocabulary,* $\theta \in \mathcal{L}_B(V)$ *is the* state law *and* $O_i \subseteq V$ *for each* $i \in I$ *are the* observable variables.

A state *of* $\mathcal{F}$ *is a Boolean assignment* $s \subseteq V$ *such that* $s \vDash \theta$. *A pointed knowledge structure is a tuple* $(\mathcal{F}, s)$ *where* s *is a state of* $\mathcal{F}$. *A multi-pointed knowledge structure is a tuple* $(\mathcal{F}, \sigma)$ *where* $\sigma \in \mathcal{L}_B(V)$.

Definition 11. *For any knowledge structure* $\mathcal{F} = (V, \theta, O)$ *we define the* S5 *model* $\mathcal{M}(\mathcal{F}) := (W, R, \mathsf{Val})$ *by:*

- $W := \{s \subseteq V \mid s \vDash \theta\}$
- $\mathsf{Val}(s) := s$
- $R_i := \{(s, t) \in W \times W \mid s \cap O_i = t \cap O_i\}$

Definition 12. *A* belief structure *is a tuple* $\mathcal{F} = (V, \theta, \Omega)$ *where* V *and* θ *are as in Definition 10 and for each agent* i *we have an* observation law $\Omega_i \in \mathcal{L}_B(V \cup V')$. *Analogously to Definition 10 we call* $(\mathcal{F}, s)$ *a pointed belief structure and* $(\mathcal{F}, \sigma)$ *a multi-pointed belief structure.*

Definition 13. *For any belief structure* $\mathcal{F} = (V, \theta, \Omega)$ *we define the Kripke model* $\mathcal{M}(\mathcal{F}) := (W, R, \mathsf{Val})$ *where* W *and* Val *are defined as in Definition 11 and*

- $R_i s t$ *iff* $s \cup t' \vDash \Omega_i$.

Example 14. *The following knowledge structure encodes the Kripke model given in Example 5:*

$$(V = \{p, q\},\ \theta = \top, O_a = \{p\},\ O_b = \{q\})$$

The same model is also encoded by this belief structure:

$$(V = \{p, q\},\ \theta = \top, \Omega_a = p \leftrightarrow p',\ \Omega_b = q \leftrightarrow q')$$

Suppose we add another atomic proposition r *that can be true or false independent of* p *and* q. *Moreover, suppose* r *is not known or observed by any of the two agents. This doubles the number of possible worlds in the Kripke model from* 4 *to* 8, *but only increases the size of the symbolic structures by one element in* V. *This illustrates that symbolic structures can be exponentially smaller than the Kripke model they encode.*

For simplicity, here we interpret $\mathcal{L}(V)$ on symbolic structures by defining $(\mathcal{F}, s) \vDash \varphi \ :\Leftrightarrow\ (\mathcal{M}(\mathcal{F}), s) \vDash \varphi$. In practice however, evaluating a formula on a symbolic structure by first 'unravelling' it to a Kripke model defeats the purpose of having a more compact representation. Instead, we should evaluate formulas on symbolic structures directly. This is possible because on a given knowledge structure any modal formula is equivalent to a purely Boolean formula.

Theorem 15. *For any belief or knowledge structure* $\mathcal{F}$ *there is a translation* $\| \cdot \|_\mathcal{F} \colon \mathcal{L}(V) \to \mathcal{L}_B(V)$ *such that for all* $\varphi \in \mathcal{L}(V)$ *and all states* $s \vDash \theta$ *we have* $\mathcal{F}, s \vDash \varphi$ *iff* $s \vDash \|\varphi\|_\mathcal{F}$.

Proof Sketch. For knowledge structures we translate K_i by quantifying over what i does not observe. Formally, let $\|K_i \varphi\|_\mathcal{F} := \forall (V \setminus O_i)(\theta \to \|\varphi\|_\mathcal{F})$.

Similarly, for belief structures we can define the translation $\|K_i \varphi\|_\mathcal{F} := \forall V'(\theta' \to (\Omega_i \to \|\varphi\|'_\mathcal{F}))$.

See Benthem et al. [Ben+18] for detailed proofs. $\square$

3.2 Transformers

The symbolic equivalent of action models are transformers. For brevity here we only consider *knowledge transformers* which encode S5 action models without post-conditions. Still, our definitions of perspective shifts easily transfer to the more general definitions given in Sections 2.7 and 2.8 of [Gat18].

Definition 16. *A* knowledge transformer *for the vocabulary* V *is a tuple* $\mathcal{F} = (V^+, \theta^+, O^+)$ *where* V^+ *is a finite set of fresh atoms disjoint with* V, $\theta^+ \in \mathcal{L}(V \cup V^+)$ *is the* event law *and* $O_i^+ \subseteq V$ *for each* $i \in I$ *are the* observable variables.

A multi-pointed transformer *is a tuple* $(\mathcal{X}, \sigma^+)$ *where* $\sigma^+ \in \mathcal{L}_B(V \cup V^+)$. *The transformation of a multi-pointed knowledge structure with a multi-pointed knowledge transformer is defined as:*

$$
\begin{array}{cclcccc}
 & ((V & , & \theta & , & O_i &),\ \sigma\ ) \\
\times & ((V^+ & , & \theta^+ & , & O^+ &),\ \sigma^+\ ) \\
:= & ((V \cup V^+, & & \theta \wedge \|\theta^+\|_\mathcal{F}, & & O_i \cup O_i^+), & \sigma \wedge \sigma^+)
\end{array}
$$

Similar to how symbolic structures for a vocabulary V encode Kripke models where the worlds are from $\mathcal{P}(V)$, transformers use the atoms from V^+ to encode action models where the events are elements of $\mathcal{P}(V^+)$. We give one example below and refer to Gattinger [Gat18] for translations to go back and forth between action models and transformers.

Example 17. *The* S5 *action model* $\mathcal{A}$ *from Example 9 can be encoded by the knowledge transformer*

$$\mathcal{X} = \left(V^+ = \{r\},\ \theta^+ = r \leftrightarrow (p \to q),\ \begin{array}{l} O_a^+ = \{r\} \\ O_b^+ = \varnothing \end{array} \right)$$

and $(\mathcal{A}, \{a_1, a_2\})$ *is encoded by* $(\mathcal{X},\ \sigma^+ = \top)$.

4 DEL on Succinct Models

Similar to symbolic structures, *succinct* models for DEL were introduced to represent Kripke models more efficiently. We will now summarise the main definitions needed to also define perspective shifts in this framework. For more details and explanations we refer to the references given in the following subsection, and especially Charrier [Cha18].

4.1 Mental Programs

Before defining succinct models we need to introduce mental programs. They describe which and how truth values of atoms can be changed to reach other states.

Mental programs use a syntax similar to Propositional Dynamic Logic (PDL) and were first presented by Charrier and Schwarzentruber [CS15] with this grammar:

$$\pi ::= p \leftarrow \top \mid p \leftarrow \bot \mid \beta? \mid \pi; \pi \mid \pi \cup \pi$$

Another grammar for programs is used by Charrier, Pinchinat, and Schwarzentruber [CPS19] where they are called 'accessibility programs':

$$\pi ::= p \leftarrow \beta \mid \beta? \mid \pi; \pi \mid \pi \cup \pi$$

Charrier and Schwarzentruber [CS17] use yet another grammar for mental programs. It includes an inversion operator that is used to define succinct product updates:

$$\pi ::= p \leftarrow \beta \mid \beta? \mid \pi; \pi \mid \pi \cup \pi \mid \pi \cap \pi \mid \pi^{-1}$$

Before discussing the differences between these grammars and deciding which one we will use here, we define semantics covering all three grammars.

Definition 18 (Semantics of Mental Programs). *We define the relation $\xrightarrow{\pi} \subseteq \mathcal{P}(V) \times \mathcal{P}(V)$ by induction over the structure of π.*

$$
\begin{aligned}
s \xrightarrow{p \leftarrow \top} t &\quad :\Longleftrightarrow \quad t = s \cup \{p\} \\
s \xrightarrow{p \leftarrow \bot} t &\quad :\Longleftrightarrow \quad t = s \setminus \{p\} \\
s \xrightarrow{\beta?} t &\quad :\Longleftrightarrow \quad t = s \text{ and } s \vDash \beta \\
s \xrightarrow{p \leftarrow \beta} t &\quad :\Longleftrightarrow \quad t = s \cup \{p\} \text{ and } s \vDash \beta \text{ or} \\
&\qquad\qquad\qquad t = s \setminus \{p\} \text{ and } s \nvDash \beta \\
s \xrightarrow{\pi^{-1}} t &\quad :\Longleftrightarrow \quad t \xrightarrow{\pi} s \\
s \xrightarrow{\pi_1; \pi_2} t &\quad :\Longleftrightarrow \quad \exists u \subseteq V : s \xrightarrow{\pi_1} u \xrightarrow{\pi_2} t \\
s \xrightarrow{\pi_1 \cup \pi_2} t &\quad :\Longleftrightarrow \quad s \xrightarrow{\pi_1} t \text{ or } s \xrightarrow{\pi_2} t \\
s \xrightarrow{\pi_1 \cap \pi_2} t &\quad :\Longleftrightarrow \quad s \xrightarrow{\pi_1} t \text{ and } s \xrightarrow{\pi_2} t
\end{aligned}
$$

We now show how two of the operators in the grammar of mental programs can be removed. Assignments of β and inverse do not add expressivity to mental programs.

Lemma 19. *For any π_1 there is a π_2 that uses neither $p \leftarrow \beta$ nor π^{-1} such that for all states $s, t \subseteq V$, we have $s \xrightarrow{\pi_1} t$ iff $s \xrightarrow{\pi_2} t$.*

Proof. We define a translation by induction. For assignments of non-constant Boolean formulas, note that $p \leftarrow \beta$ is equivalent to $(\beta?; p \leftarrow \top) \cup (\neg\beta?; p \leftarrow \bot)$ according to the semantics in Definition 18. Hence in particular

the grammars from Charrier and Schwarzentruber [CS15] and Charrier, Pinchinat, and Schwarzentruber [CPS19] are equally expressive.

$$
\begin{aligned}
t(p \leftarrow \top) &\quad := \quad p \leftarrow \top \\
t(p \leftarrow \bot) &\quad := \quad p \leftarrow \bot \\
t(p \leftarrow \beta) &\quad := \quad (\beta?; p \leftarrow \top) \cup (\neg\beta?; p \leftarrow \bot) \\
t(\beta?) &\quad := \quad \beta? \\
t(\pi_1; \pi_2) &\quad := \quad t(\pi_1); t(\pi_2) \\
t(\pi_1 \cup \pi_2) &\quad := \quad t(\pi_1) \cup t(\pi_2) \\
t(\pi_1 \cap \pi_2) &\quad := \quad t(\pi_1) \cap t(\pi_2)
\end{aligned}
$$

To also remove the inversion operators we distinguish different subcases:

$$
\begin{aligned}
t((p \leftarrow \top)^{-1}) &\quad := \quad p?; (p \leftarrow \top) \cup (p \leftarrow \bot) \\
t((p \leftarrow \bot)^{-1}) &\quad := \quad \neg p?; ((p \leftarrow \top) \cup (p \leftarrow \bot)) \\
t((\beta?)^{-1}) &\quad := \quad \beta? \\
t((\pi_1; \pi_2)^{-1}) &\quad := \quad t(\pi_2^{-1}); t(\pi_1^{-1}) \\
t((\pi_1 \cup \pi_2)^{-1}) &\quad := \quad t(\pi_1^{-1}) \cup t(\pi_2^{-1}) \\
t((\pi_1 \cap \pi_2)^{-1}) &\quad := \quad t(\pi_1^{-1}) \cap t(\pi_2^{-1}) \\
t((\pi_1^{-1})^{-1}) &\quad := \quad t(\pi_1)
\end{aligned}
$$

One now checks that $s \xrightarrow{\pi} t$ iff $s \xrightarrow{t(\pi)} t$. $\qquad\square$

Eliminating the inverse operator in mental programs is similar to "pushing down" the converse operator in PDL with converse, also known as CPDL.[1] See for example [DM00, Section 3] where converse is restricted to atomic programs, using abbreviations similar to $t(\cdot)$ above. Whereas in PDL an atomic converse a^- cannot be rewritten further, in mental programs the atomic programs are assignments and $(p \leftarrow \beta)^{-1}$ can still be rewritten as above.

Motivated by Lemma 19, in the rest of this article we use the following grammar for mental programs. This is mainly to simplify our definitions in Section 5.3. For implementations it can still be more efficient to define other operators as primitives and not as abbreviations.

$$\pi ::= p \leftarrow \top \mid p \leftarrow \bot \mid \beta? \mid \pi; \pi \mid \pi \cup \pi \mid \pi \cap \pi$$

Intuitively, observation laws used in the previous section describe *what has to be true* at each of two states s and t for them to be connected. In contrast, mental programs focus on *which changes are allowed to go from s to t*. Mathematically, both are just different ways to describe the same thing: relations over $\mathcal{P}(V)$. But in actual model checking or DEL planning implementations the difference will matter.

4.2 Succinct Models

Definition 20. *A succinct model for a vocabulary V is a vector $\vec{\pi}$ where each π_i is a mental program over V.*

A pointed succinct model is a tuple $(\vec{\pi}, s)$ where $s \subseteq V$. A multi-pointed succinct model is a tuple $(\vec{\pi}, \sigma)$ where $\sigma \in \mathcal{L}_B(V)$.

We note that Charrier, Pinchinat, and Schwarzentruber [CPS19] call succinct models 'symbolic'. To avoid confusion with symbolic structures, here we follow Charrier and Schwarzentruber [CS17] and always call $\vec{\pi}$ a succinct model.

---

[1]This connection was pointed out to me by one of the anonymous reviewers.

Definition 21. *For any succinct model $\vec{\pi}$ we define the Kripke model $\mathcal{M}(\vec{\pi}) = (W, R, \mathsf{Val})$ by*

- $W := \mathcal{P}(V)$

- $R_i st \;:\Leftrightarrow\; s \xrightarrow{\pi_i} t$

- $\mathsf{Val}(s) := s$

Example 22. *The Kripke model from Example 5 is encoded by the succinct model $\vec{\pi}$ given by:*

$$\begin{aligned}
\pi_a &= (q \leftarrow \top) \cup (q \leftarrow \bot) \\
\pi_b &= (p \leftarrow \top) \cup (p \leftarrow \bot)
\end{aligned}$$

The multi-pointed model $(\mathcal{M}, \omega)$ with $\omega = \{0\}$ is encoded by $(\vec{\pi}, \; p \wedge q)$.

We encourage the reader to compare this with the symbolic structures in Example 14. Clearly, the π_i play the same role as R_i in Kripke models and O_i or Ω_i in symbolic structures. But succinct models do not have a counterpart of W in Kripke models or θ in symbolic structures. Succinct models always encode a set of worlds which is the powerset of the vocabulary, i.e. $W = \mathcal{P}(V)$, whereas in a symbolic structure the encoded set of worlds is given by θ, namely $W = \{\, s \subseteq V \mid s \vDash \theta \,\}$ which can be a proper subset of $\mathcal{P}(V)$. The fact that succinct models always encode the full set of worlds does not matter for the semantics: whether a world is unreachable or does not even exist does not affect the truth of any formula at reachable worlds. One might thus also say that in a succinct model the set of worlds (that are semantically relevant) consists of all the states that all π_i together can reach.

Just as with symbolic structures for simplicity we interpret $\mathcal{L}(V)$ on succinct models by defining $(\vec{\pi}, s) \vDash \varphi \;:\Leftrightarrow\; (\mathcal{M}(\vec{\pi}), s) \vDash \varphi$. In practice formulas should be evaluated directly on succinct models, not via the unravelling to a Kripke model — see Charrier and Schwarzentruber [CS17] and Charrier, Pinchinat, and Schwarzentruber [CPS19].

4.3 Succinct Actions

Action models can also be encoded succinctly. We only state the main definition here, adjusted to our notation.

Definition 23. *A* succinct action model *for the vocabulary V is a tuple $(V^+, \xi_A, \vec{\pi}_A, \mathsf{post})$ where V^+ is a set disjoint from V, ξ_A is a formula over $V \cup V^+$, $\vec{\pi}_A$ is a vector of mental programs over $V \cup V^+$, and post is a mental program over $V \cup V^+$.*

For translations between explicit and succinct action models, see Section 3.3, and for the succinct product update, see Section 3.4 of Charrier and Schwarzentruber [CS17].

5 Perspective Shifts

To use DEL for multi-agent epistemic planning it is necessary to consider the perspective of different agents. We recall an example from Engesser et al. [Eng+17]: Anne wants to let Bob into her flat while she is away. She can put the keys under the door mat, but for an *implicitly coordinated plan* she also has to inform Bob that this is where

he will find the keys. To find such plans in general we need a way to compute perspective shifts: What is the situation as seen by Anne or Bob? After this action, what will Anne know about what Bob knows?

5.1 Explicit Perspective Shifts

To formalise the idea of 'taking the perspective of another agent' Engesser et al. [Eng+17] define the local state of an agent in a model as follows.

Definition 24. *Given a multi-pointed $\mathsf{S5}$ model $(\mathcal{M}, \omega)$, the* local state *of agent i is:*

$$\omega^i := \{ v \in W \mid \exists w \in \omega : w \sim_i v \}$$

Example 25. *In $(\mathcal{M}, \{0\})$ where $\mathcal{M}$ is the model from Example 5, the local states of agents a and b are:*

$$\{0\}^a = \{0, 1\} \qquad \{0\}^b = \{0, 2\}$$

We can also nest the operator $(\cdot)^i$ to talk about higher-level perspectives. Starting with $\{0\}$ we have

$$(\{0\}^a)^b = \{0, 1\}^b = \{0, 1, 2, 3\}$$

This means that from the perspective of agent a the local state of b is the whole model. To make it explicit: If 0 is the actual world, then a considers 0 and 1 possible and thereby considers it possible that b considers any of 0, 1, 2, and 3 possible.

The local state and the knowledge of an agent are clearly related: an agent knows that something is true iff it is true at all worlds in their local state.

Proposition 26 (Proposition 1 from Engesser et al. [Eng+17])**.** *For any state ω we have $\omega^i \vDash \varphi$ iff $\omega \vDash K_i$.*

Engesser et al. [Eng+17] only consider $\mathsf{S5}$ models where all accessibility relations are equivalence relations. It is easy to generalise the definition of local states to Kripke models where not all R_i have to be equivalence relations.

Definition 27. *Given a multi-pointed Kripke model $(\mathcal{M}, \omega)$, the* local state *of agent i is*

$$\omega^i := \{ v \in W \mid \exists w \in \omega : (w, v) \in R_i \}$$

Intuitively, the local state ω^i is given by $R_i(\omega)$ and consists of all those worlds an agent considers possible if the actual world is an element of ω. However, some natural properties of local states no longer hold in the general setting.

Fact 28. *For $\mathsf{S5}$ models we have $\omega \subseteq \omega^i$, but this is not the case when R_i is not reflexive. Moreover, $(\cdot)^i$ is idempotent for $\mathsf{S5}$ models, but it is not if R_i is not transitive.*

Still, there are non-$\mathsf{S5}$ settings where it makes sense to talk about perspective shifts and even nested ones.

Example 29. *In the following Kripke model agent a has a false belief that p is true. In fact, a believes that p is common knowledge among a and b.*
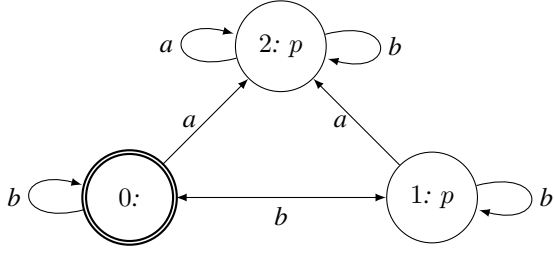
Given the actual state $\omega = \{0\}$, the local states are:

$$\omega^a = \{2\}$$
$$\omega^b = \{0,1\}$$
$$(\omega^b)^a = \{0,1\}^a = \{2\}$$

Kripke models only cover the static "EL" part of DEL. For the dynamic part we also need to describe how an action is perceived by an agent. Given the similarity between Kripke models and action models, this is quite easy. The following is essentially Definition 27 for action models.

Definition 30. *Given a multi-pointed action model $(\mathcal{A}, \alpha)$, the* local action *of i is $\{b \in A \mid \exists a \in \alpha : R_i^{\mathcal{A}} ab\}$.*

5.2 Symbolic Perspective Shifts

To work efficiently with local states in a planning tool, we want to avoid iterating over explicit lists of worlds. We now show how perspective shifts can be computed on symbolic structures from Section 3.

In the S5 setting we can exploit symmetry as follows.

Definition 31. *Given a multi-pointed knowledge structure $(\mathcal{F}, \sigma)$ where $\mathcal{F} = (V, \theta, O)$, the* symbolic local state *of i is given by:*
$$\sigma^i := \exists (V \setminus O_i)(\theta \wedge \sigma)$$

Example 32. *Consider $(\mathcal{F}, \sigma = p \wedge q)$ where $\mathcal{F}$ is the knowledge structure from Example 14, encoding the Kripke model from Example 5. The local states here are:*

$$\sigma^a \equiv \exists (V \setminus O_a)(\sigma \wedge \theta) \equiv \exists q(p \wedge q) \equiv p$$
$$\sigma^b \equiv \exists (V \setminus O_b)(\sigma \wedge \theta) \equiv \exists p(p \wedge q) \equiv q$$

The attentive reader will note how Definition 31 corresponds to Definition 24. Concretely, σ corresponds to ω, the O_i play the role of $\sim_i$ and θ is the restriction to W.

We could also have defined $\sigma^i := \theta \wedge \exists (V \setminus O_i)(\theta \wedge \sigma)$, but this would be redundant: For any σ the tuples $(\mathcal{F}, \sigma)$ and $(\mathcal{F}, \sigma \wedge \theta)$ denote the same multi-pointed knowledge structure. In fact, SMCDEL [Gat19] uses the opposite direction to minimise knowledge structures: we can restrict the BDD of σ to what is not already implied by θ. In belief structures the same optimisation can be applied to each Ω_i.

We now state and prove that Definition 31 not only works for simple examples such as the above, but that it is always truthful, i.e. that explicit and symbolic perspective shifts agree with each other.

Theorem 33. *Suppose we have a multi-pointed knowledge structure $(\mathcal{F}, \sigma)$. Consider the encoded multi-pointed S5 model $(\mathcal{M}(\mathcal{F}), \omega)$ where $\omega := \{s \subseteq V \mid s \vDash \theta \wedge \sigma\}$. Then we have for all $s \subseteq V$:*

$$s \text{ is a state of } \mathcal{F} \text{ and } s \vDash \sigma^i \iff s \in \omega^i$$

Proof. We have the following chain of equivalences:

$$
\begin{aligned}
& s \text{ is a state of } \mathcal{F} \text{ and } s \vDash \sigma^i \\
\Leftrightarrow\quad & s \vDash \theta \wedge \sigma^i \\
\Leftrightarrow\quad & s \vDash \theta \wedge \exists (V \setminus O_i)(\theta \wedge \sigma) \\
\Leftrightarrow\quad & s \in W \text{ and } s \vDash \exists (V \setminus O_i)(\theta \wedge \sigma) \\
\Leftrightarrow\quad & s \in W \text{ and } \exists t \subseteq V : t \vDash \theta \wedge \sigma \text{ and } t \cap O_i = s \cap O_i \\
\Leftrightarrow\quad & s \in W \text{ and } \exists t \in \omega : t \sim_i s \\
\Leftrightarrow\quad & s \in \omega^i
\end{aligned}
$$

$\square$

For the general setting of K instead of S5, recall that we encode the relation of each agent with a boolean formula over a double vocabulary: $\Omega_i \in \mathcal{L}_B(V \cup V')$. To define local states using these formulas we need to invert the encoded relation. For this we simultaneously (un)prime all atomic propositions. Formally, let $\Omega_i^{\smile} := [V \cup V' \mapsto V' \cup V]\Omega_i$. This includes a slight abuse of notation: '$\cup$' should be read as concatenation of ordered lists here, instead of a union of sets. For example, we have $((p \wedge \neg q') \vee q)^{\smile} = ((p' \wedge \neg q) \vee q')$.

Definition 34. *Given a multi-pointed belief structure $(\mathcal{F}, \sigma)$ where $\mathcal{F} = (V, \theta, \Omega)$, the* symbolic local state *of agent i is given by:*
$$\sigma^i := \exists V'(\theta' \wedge \sigma' \wedge \Omega_i^{\smile})$$

Example 35. *The Kripke model from Example 29 is encoded by this belief structure:*

$$(V = \{p, q\},\ \theta = q \rightarrow p,\ \Omega_a = q',\ \Omega_b = q \leftrightarrow q')$$

Note that we need an extra atomic variable q here, because the Kripke model contains two worlds with the same valuation (1 and 2). There are many ways to add extra variables, yielding different state and observation laws. Our choice here is to make q true only at world 2.

Consider the actual state $\sigma = \neg p \wedge \neg q$ corresponding to $\{0\}$ in Example 29. We can now compute the local states and perspective shifts mentioned above symbolically:

$$
\begin{aligned}
\sigma^a &= \exists V'\,(\theta' \wedge \sigma' \wedge \Omega_a^{\smile}) \\
&= \exists \{p,q\}'\,((q \rightarrow p)' \wedge (\neg p \wedge \neg q)' \wedge (q')^{\smile}) \\
&= \exists p'\, \exists q'\,((q' \rightarrow p') \wedge (\neg p' \wedge \neg q') \wedge q) \\
&= q
\end{aligned}
$$

$$
\begin{aligned}
\sigma^b &= \exists V'\,(\theta' \wedge \sigma' \wedge \Omega_b^{\smile}) \\
&= \exists \{p,q\}'\,((q \rightarrow p)' \wedge (\neg p \wedge \neg q)' \wedge (q \leftrightarrow q')^{\smile}) \\
&= \exists p'\, \exists q'\,((q' \rightarrow p') \wedge (\neg p' \wedge \neg q') \wedge (q' \leftrightarrow q)) \\
&= \neg q
\end{aligned}
$$

$$
\begin{aligned}
(\sigma^b)^a &= (\neg q)^a \\
&= \exists V'\,(\theta' \wedge (\neg q)' \wedge \Omega_a^{\smile}) \\
&= \exists V'\,((q \rightarrow p)' \wedge \neg q' \wedge (q')^{\smile}) \\
&= \exists V'\,((q' \rightarrow p') \wedge \neg q' \wedge q) \\
&= \exists V'\,(q \wedge \neg q') \\
&= q
\end{aligned}
$$

This corresponds to the explicit local states: $\sigma^a = q$ encodes $\omega^a = \{2\}$ and $\sigma^b = \neg q$ encodes $\omega^b = \{0,1\}$.

Similar to Theorem 33 the following theorem states that Definition 34 does what we want and is correct not only in Example 35, but in general.

Theorem 36. *Suppose we have a multi-pointed belief structure* $(\mathcal{F}, \sigma)$. *Consider the encoded multi-pointed model* $(\mathcal{M}(\mathcal{F}), \omega)$ *where* $\omega := \{s \subseteq V \mid s \models \theta \land \sigma\}$. *Then for all* $s \subseteq V$ *we have:*

$$s \text{ is a state of } \mathcal{F} \text{ and } s \models \sigma^i \iff s \in \omega^i$$

Proof. Intuitively, a state s satisifes σ^i iff it can be reached via the relation encoded by Ω_i from a state t which satisfies σ. We have the following chain of equivalences:

$$
\begin{aligned}
& s \models \theta \land \sigma^i \\
\Leftrightarrow\ & s \models \theta \land \exists V'(\theta' \land \sigma' \land \Omega_i^{\smile}) \\
\Leftrightarrow\ & s \models \theta \text{ and } s \models \exists V'(\theta' \land \sigma' \land \Omega_i^{\smile}) \\
\Leftrightarrow\ & s \in W \text{ and } \exists t' \subseteq V' : s \cup t' \models (\theta' \land \sigma' \land \Omega_i^{\smile}) \\
\Leftrightarrow\ & s \in W \text{ and } \exists t \subseteq V : t \cup s' \models (\theta \land \sigma \land \Omega_i) \\
\Leftrightarrow\ & s \in W \text{ and } \exists t \subseteq V : t \models \theta \land \sigma \text{ and } t \cup s' \models \Omega_i \\
\Leftrightarrow\ & s \in W \text{ and } \exists t \in \omega : R_i st \\
\Leftrightarrow\ & s \in \omega^i
\end{aligned}
$$

$\square$

Before we conclude this section, it remains to define local states for transformers, the symbolic analogue of Definition 30.

Definition 37. *Given a multi-pointed knowledge transformer* $(\mathcal{X}, \sigma^+)$ *where* $\mathcal{X} = (V^+, \theta^+, O^+)$*, the* symbolic local action *for i is* $\exists (V \setminus O^+)(\theta^+ \land \sigma^+)$.

Clearly, this is analogous to Definition 31, using components of $\mathcal{X}$ instead of $\mathcal{F}$. For more general (belief) transformers the definition would be analogous to Definition 34.

5.3 Succinct Perspective Shifts

We now turn again to succinct models. Here we can also compute perspective shifts directly, without going back and forth to an explicit Kripke model.

Definition 38. *Given a succinct multi-point σ and a mental program π, we define the* localisation σ^π *by induction over the structure of π:*

$$
\begin{aligned}
\sigma^{p \leftarrow \top} &:= \exists p(\sigma) \land p \\
\sigma^{p \leftarrow \bot} &:= \exists p(\sigma) \land \neg p \\
\sigma^{\beta?} &:= \sigma \land \beta \\
\sigma^{(\pi_1 ; \pi_2)} &:= (\sigma^{\pi_1})^{\pi_2} \\
\sigma^{(\pi_1 \cup \pi_2)} &:= \sigma^{\pi_1} \lor \sigma^{\pi_2} \\
\sigma^{\pi_1 \cap \pi_2} &:= \bigvee_{s \subseteq V} \left((\beta_s \land \sigma)^{\pi_1} \land (\beta_s \land \sigma)^{\pi_2} \right)
\end{aligned}
$$

where $\beta_s := \bigwedge s \land \bigwedge \{\neg p \mid p \in V \setminus s\}$ *for any* $s \subseteq V$.

Given a multi-pointed succinct model $(\vec{\pi}, \sigma)$*, the* succinct local state of i *is given by* σ^{π_i}.

The clause for $\cap$ in Definition 38 can unfortunately yield exponentially longer formulas. At first sight it seems that $\sigma^{\pi_1 \cap \pi_2} := \sigma^{\pi_1} \land \sigma^{\pi_2}$ would be a better choice, but this would invalidate Lemma 41 below. The β_s formulas are needed to ensure that whenever $t \models \sigma^{\pi_1 \cap \pi_2}$, the state t can not just be reached via π_1 and π_2 from some (possibly different) states s_1 and s_2 satisfying σ, but it can be reached from one and the same s via both mental programs.

The mental programs π_i for agents in succinct models are usually $\cap$-free anyway: Charrier and Schwarzentruber [CS17] only use $\cap$ to define postconditions. Hence we only include $\cap$ here for the sake of completeness and do not worry about the resulting complexity of the translation.

Example 39. *Consider the succinct model $\vec{\pi}$ from Example 22, encoding the Kripke model from Example 5. Given the state $\sigma = p \land q$, the succinct local state of a can then be computed as follows:*

$$
\begin{aligned}
\sigma^{\pi_a} &= (p \land q)^{((q \leftarrow \top) \cup (q \leftarrow \bot))} \\
&= (p \land q)^{(q \leftarrow \top)} \lor (p \land q)^{(q \leftarrow \bot)} \\
&= (\,(\exists q(p \land q)) \land q\,) \lor (\,(\exists q(p \land q)) \land \neg q\,) \\
&= (p \land q) \lor (p \land \neg q) \\
&= p
\end{aligned}
$$

Analogously we have $\sigma^{\pi_b} = (p \land q)^{((p \leftarrow \top) \cup (p \leftarrow \bot))} = q$.

Example 40. *As a second example we encode the Kripke model from Example 29 as a succinct model. First note that similar to the symbolic encoding given in Example 35 we need an additional atom q to be able to encode the two worlds with the same valuation. The vocabulary of the succinct model is thus $V = \{p, q\}$ and $\vec{\pi}$ is given by:*

$$\pi_a = (p \leftarrow \top); (q \leftarrow \top)$$

$$\pi_b = (q?) \cup (\neg q?; ((p \leftarrow \top) \cup (p \leftarrow \bot)))$$

Again note that the actual state is given by $\sigma = \neg p \land \neg q$. The succinct local states of the two agents and the nested local state can thus be computed by:

$$
\begin{aligned}
& \sigma^{\pi_a} \\
=\ & (\neg p \land \neg q)^{((p \leftarrow \top); (q \leftarrow \top))} \\
=\ & \left((\neg p \land \neg q)^{(p \leftarrow \top)} \right)^{(q \leftarrow \top)} \\
=\ & (\,(\exists p(\neg p \land \neg q)) \land p\,)^{(q \leftarrow \top)} \\
=\ & (\neg q \land p)^{(q \leftarrow \top)} \\
=\ & (\exists q(\neg q \land p)) \land q \\
=\ & p \land q
\end{aligned}
$$

$$
\begin{aligned}
& \sigma^{\pi_b} \\
=\ & (\neg p \land \neg q)^{((q?) \cup (\neg q?; ((p \leftarrow \top) \cup (p \leftarrow \bot))))} \\
=\ & (\neg p \land \neg q)^{(q?)} \lor (\neg p \land \neg q)^{(\neg q?; ((p \leftarrow \top) \cup (p \leftarrow \bot)))} \\
=\ & ((\neg p \land \neg q) \land q) \lor \left((\neg p \land \neg q)^{(\neg q?)} \right)^{((p \leftarrow \top) \cup (p \leftarrow \bot))} \\
=\ & ((\neg p \land \neg q) \land \neg q)^{((p \leftarrow \top) \cup (p \leftarrow \bot))} \\
=\ & (\neg p \land \neg q)^{(p \leftarrow \top)} \lor (\neg p \land \neg q)^{(p \leftarrow \bot)} \\
=\ & (\exists p(\neg p \land \neg q) \land p) \lor (\exists p(\neg p \land \neg q) \land \neg p) \\
=\ & (\neg q \land p) \lor (\neg q \land \neg p) \\
=\ & \neg q
\end{aligned}
$$

$$
\begin{aligned}
& (\sigma^{\pi_b})^{\pi_a} \\
=\ & (\neg q)^{\pi_a} \\
=\ & (\neg q)^{((p \leftarrow \top); (q \leftarrow \top))} \\
=\ & \left((\neg q)^{(p \leftarrow \top)} \right)^{(q \leftarrow \top)} \\
=\ & (\,(\exists p(\neg q)) \land p\,)^{(q \leftarrow \top)} \\
=\ & (\neg q \land p)^{(q \leftarrow \top)} \\
=\ & (\exists q(\neg q \land p)) \land q \\
=\ & p \land q
\end{aligned}
$$

To conclude this section we prove that succinct perspective shifts agree with the explicit definition not only for these examples but in general.

Lemma 41. *For all multi-points σ and mental programs π and states $s, t \subseteq V$ we have:*

$$t \models \sigma^\pi \iff \exists s : s \models \sigma \text{ and } s \xrightarrow{\pi} t$$

Proof. By induction over π. For the base case where $\pi = p \leftarrow \top$ we have this chain of equivalences:

$$
\begin{aligned}
& t \vDash \sigma^{(p \leftarrow \top)} \\
\Leftrightarrow \quad & t \vDash \exists p(\sigma) \wedge p \\
\Leftrightarrow \quad & t \vDash \exists p(\sigma) \text{ and } t \vDash p \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } t = s \cup \{p\} \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{p \leftarrow \top} t
\end{aligned}
$$

The second base case $\pi = p \leftarrow \bot$ is similar. For the third case $\pi = \beta?$ we have:

$$
\begin{aligned}
& t \vDash \sigma^{\beta?} \\
\Leftrightarrow \quad & t \vDash \sigma \wedge \beta \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } t = s \text{ and } s \vDash \beta \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{\beta?} t
\end{aligned}
$$

For the induction step, first consider the ;-case. We have the following chain of equivalences, where the second and third step follow from the induction hypothesis.

$$
\begin{aligned}
& t \vDash \sigma^{(\pi_1; \pi_2)} \\
\Leftrightarrow \quad & t \vDash (\sigma^{\pi_1})^{\pi_2} \\
\Leftrightarrow \quad & \exists u : \ u \vDash \sigma^{(\pi_1)} \text{ and } u \xrightarrow{\pi_2} t \\
\Leftrightarrow \quad & \exists u \, \exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{\pi_1} u \xrightarrow{\pi_2} t \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{\pi_1; \pi_2} t
\end{aligned}
$$

Second, consider $\cup$. In the following chain the third step follows from the induction hypothesis.

$$
\begin{aligned}
& t \vDash \sigma^{(\pi_1 \cup \pi_2)} \\
\Leftrightarrow \quad & t \vDash \sigma^{(\pi_1)} \vee \sigma^{(\pi_2)} \\
\Leftrightarrow \quad & t \vDash \sigma^{(\pi_1)} \text{ or } t \vDash \sigma^{(\pi_2)} \\
\Leftrightarrow \quad & (\exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{\pi_1} t) \text{ or } (\exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{\pi_2} t) \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } (s \xrightarrow{\pi_1} t \text{ or } s \xrightarrow{\pi_2} t) \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{\pi_1 \cup \pi_2} t
\end{aligned}
$$

Finally, consider $\cap$. In the following chain, the third equivalence is by induction hypothesis. The fourth equivalence holds because whenever $s_n \vDash \beta_s$ then we have $s_n = s$.

$$
\begin{aligned}
& t \vDash \sigma^{(\pi_1 \cup \pi_2)} \\
\Leftrightarrow \quad & t \vDash \bigvee_{s \subseteq V} \left((\beta_s \wedge \sigma)^{\pi_1} \wedge (\beta_s \wedge \sigma)^{\pi_2} \right) \\
\Leftrightarrow \quad & \exists s : \ t \vDash (\beta_s \wedge \sigma)^{\pi_1} \wedge (\beta_s \wedge \sigma)^{\pi_2} \\
\Leftrightarrow \quad & \exists s : \left(\begin{array}{l} \exists s_1 : s_1 \vDash \beta_s \wedge \sigma \text{ and } s_1 \xrightarrow{\pi_1} t \text{ and} \\ \exists s_2 : s_2 \vDash \beta_s \wedge \sigma \text{ and } s_2 \xrightarrow{\pi_2} t \end{array} \right) \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{\pi_1} t \text{ and } s \xrightarrow{\pi_2} t \\
\Leftrightarrow \quad & \exists s : \ s \vDash \sigma \text{ and } s \xrightarrow{\pi_1 \cap \pi_2} t
\end{aligned}
$$

$\square$

The following Theorem connects succinct with explicit perspective shifts, similar to Theorems 33 and 36 for symbolic perspective shifts.

Theorem 42. *Take any multi-pointed succinct model $(\vec{\pi}, \sigma)$ and consider the encoded Kripke model $(\mathcal{M}(\pi), \omega)$ where $\omega := \{s \subseteq V \mid s \vDash \sigma\}$. Then we have for all $s \subseteq V$:*

$$
s \vDash \sigma^{\pi_i} \iff s \in \omega^i
$$

Proof. By Lemma 41. $\square$

6 Conclusion

We presented definitions to compute perspective shifts directly on symbolic structures and succinct models for DEL. For each definition we gave examples and proved its correctness, i.e. that it agrees with the original definition on explicit Kripke models.

Admittedly, this work is mainly an exercise in finding the right definitions and notation. In hindsight, nothing here is surprising. Still, it is educational to see which operations can be done without 'unravelling'.

The next step is clear: the definitions of symbolic and succinct perspective shifts should be implemented and used in an epistemic planning tool. As a first step we extended SMCDEL [Gat19] with symbolic perspective shifts based on Definitions 31 and 34.

Computing perspective shifts directly on compact representations should make DEL planning tools more efficient and allow us to solve larger problems. For example, in the cooperative card game Hanabi used by Reifsteck et al. [Rei+19] the number of possible worlds grows exponentially in the number of agents and cards per hand. With compact representations it should be possible to deal with larger examples, and possibly even the real game with up to 60 cards.

Bibliography

[BA11] Thomas Bolander and Mikkel Birkegaard Andersen. "Epistemic planning for single- and multi-agent systems". In: *Journal of Applied Non-Classical Logics* 21.1 (2011), pages 9–34. https://doi.org/10.3166/jancl.21.9-34.

[Ben+15] Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. "Symbolic Model Checking for Dynamic Epistemic Logic". In: *Proceedings of the 5th International Workshop on Logic, Rationality, and Interaction (LORI 2015)*. Edited by Wiebe van der Hoek, Wesley H. Holliday, and Wen-fang Wang. 2015, pages 366–378. https://doi.org/10.1007/978-3-662-48561-3_30.

[Ben+18] Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. "Symbolic Model Checking for Dynamic Epistemic Logic – S5 and Beyond". In: *Journal of Logic and Computation* 28 (2 2018), pages 367–402. https://doi.org/10.1093/logcom/exx038.

[Bry86] Randal E. Bryant. "Graph-Based Algorithms for Boolean Function Manipulation". In: *IEEE Transaction on Computers* C-35.8 (1986), pages 677–691. https://doi.org/10.1109/TC.1986.1676819.

[Cha18] Tristan Charrier. "Complexité théorique du raisonnement en logique épistémique dynamique et étude d'une approche symbolique". English title: Theoretical complexity of reasoning in dynamic epistemic logic and study of a symbolic approach. PhD thesis. 2018. https://www.theses.fr/en/2018REN1S124.

[CPS19] Tristan Charrier, Sophie Pinchinat, and François Schwarzentruber. "Symbolic model checking of public announcement protocols". In: *Journal of Logic and Computation* 29 (8 2019), pages 1211–1249. https://doi.org/10.1093/logcom/exz023.

[CS15] Tristan Charrier and François Schwarzentruber. "Arbitrary Public Announcement Logic with Mental Programs". In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. 2015, pages 1471–1479. https://dl.acm.org/citation.cfm?id=2772879.2773340.

[CS17] 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 '17. São Paulo, Brazil: International Foundation for Autonomous Agents and Multiagent Systems, 2017, pages 123–131. http://www.aamas2017.org/proceedings/pdfs/p123.pdf.

[DHK07] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Springer, 2007. ISBN: 978-1-4020-5838-7. https://doi.org/10.1007/978-1-4020-5839-4.

[DM00] Giuseppe De Giacomo and Fabio Massacci. "Combining Deduction and Model Checking into Tableaux and Algorithms for Converse-PDL". In: *Information and Computation* 162.1 (2000), pages 117–137. https://doi.org/10.1006/inco.1999.2852.

[Eng+17] Thorsten Engesser, Thomas Bolander, Robert Mattmüller, and Bernhard Nebel. "Cooperative Epistemic Multi-Agent Planning for Implicit Coordination". In: *Methods for Modalities 2017*. Edited by Sujata Ghosh and R. Ramanujam. 2017, pages 75–90. https://doi.org/10.4204/EPTCS.243.6.

[Gat18] Malvin Gattinger. "New Directions in Model Checking Dynamic Epistemic Logic". PhD thesis. University of Amsterdam, 2018. ISBN: 978-94-028-1025-7. https://malv.in/phdthesis.

[Gat19] Malvin Gattinger. *SMCDEL — An Implementation of Symbolic Model Checking for Dynamic Epistemic Logic with Binary Decision Diagrams*. Version 1.1.0. 2019. https://doi.org/10.5281/zenodo.3568325.

[LPW11] Benedikt Löwe, Eric Pacuit, and Andreas Witzel. "DEL Planning and Some Tractable Cases". In: *LORI 2011: Logic, Rationality, and Interaction*. Edited by Hans van Ditmarsch, Jérôme Lang, and Shier Ju. 2011, pages 179–192. https://doi.org/10.1007/978-3-642-24130-7_13.

[Rei+19] Daniel Reifsteck, Thorsten Engesser, Robert Mattmüller, and Bernhard Nebel. "Epistemic Multi-agent Planning Using Monte-Carlo Tree Search". In: *KI 2019: Advances in Artificial Intelligence*. Edited by Christoph Benzmüller and Heiner Stuckenschmidt. Volume 11793. 2019, pages 277–289. ISBN: 978-3-030-30178-1. https://doi.org/10.1007/978-3-030-30179-8_24.