

Disjunctive Normal Form for Multi-Agent Modal Logics Based on Logical Separability

Liangda Fang,¹ Kewen Wang,² Zhe Wang,² Ximing Wen³

¹Department of Computer Science, Jinan University, China

²School of Information and Communication Technology, Griffith University, Australia

³Guangdong Institute of Public Administration, Guangzhou, China

fangld@jnu.edu.cn, {k.wang,zhe.wang}@griffith.edu.au, wenxim@mail2.sysu.edu.cn

Abstract

Modal logics are primary formalisms for multi-agent systems but major reasoning tasks in such logics are intractable, which impedes applications of multi-agent modal logics such as automatic planning. One technique of tackling the intractability is to identify a fragment called a normal form of multi-agent logics such that it is expressive but tractable for reasoning tasks such as entailment checking, bounded conjunction transformation and forgetting. For instance, DNF of propositional logic is tractable for these reasoning tasks. In this paper, we first introduce a notion of logical separability and then define a novel disjunctive normal form SDNF for the multi-agent logic K_n , which overcomes some shortcomings of existing approaches. In particular, we show that every modal formula in K_n can be equivalently casted as a formula in SDNF, major reasoning tasks tractable in propositional DNF are also tractable in SDNF, and moreover, formulas in SDNF enjoy the property of logical separability. To demonstrate the usefulness of our approach, we apply SDNF in multi-agent epistemic planning. Finally, we extend these results to three more complex multi-agent logics D_n , $K45_n$ and $KD45_n$.

1 Introduction

It is crucial for an intelligent agent system to be capable of representing and reasoning about high-order knowledge in the multi-agent setting. A general representative framework for these scenarios is multi-agent modal logics. However, some important reasoning tasks, including satisfiability checking and forgetting, are intractable in such logics (Halpern and Moses 1992; Bienvenu 2009). The intractability results impede applications of multi-agent modal logics, *e.g.*, multi-agent epistemic planning (Kominis and Geffner 2015; Huang et al. 2017).

Traditionally, the forward search algorithm is effective for agent planning (Bryce, Kambhampati, and Smith 2006), in which the search is performed in the space of knowledge bases (KBs) proceeding forward from the initial KB towards a goal KB entailing the goal formula. Two types of reasoning tasks are essential to the search algorithm, namely, progression and entailment check. The progression updates KBs according to the action effects while the entailment check is needed to decide if the current KB entails the goal formula

and the action preconditions. If a normal form is defined such that its entailment check, bounded conjunction and forgetting are tractable, then both progression and entailment check in the normal form are tractable (Bienvenu, Fargier, and Marquis 2010). As a result, in the normal form, an effective search algorithm is obtained. In the case of propositional logic, a normal form can be disjunctive normal form (DNF), conjunctive normal form (CNF) or prime implicate. Usually, a planner based on DNF is faster than those based on CNF and prime implicate (To, Pontelli, and Son 2011; To, Son, and Pontelli 2011). This is due to the fact that, in propositional logic, DNF possesses tractable entailment check, bounded conjunction and forgetting, but CNF and prime implicate do not (Darwiche and Marquis 2002).

Thus, in order to develop effective search algorithms to multi-agent epistemic planning, researchers aimed to develop suitable DNF-like normal forms for multi-agent modal logics such that major reasoning tasks, such as bounded conjunction, forgetting and entailment check, are tractable. Such a disjunctive normal form, named S5-DNF, is defined for modal logic S5 but only for the single-agent case (Bienvenu, Fargier, and Marquis 2010). When this DNF is extended to multi-agent modal logics, both entailment and forgetting will not be tractable any longer. Two other normal forms, *cover disjunctive normal forms (CDNFs)* (ten Cate et al. 2006) and *prime implicate normal forms (PINFs)* (Bienvenu 2008), have been introduced for description logic \mathcal{ALC} , a syntactic variant of the multi-agent modal logic K_n . However, these two normal forms have some shortcomings. CDNF is relatively less compact, that is, the CDNF representation is exponentially large for some simple formulas. Bounded conjunction for PINF is intractable and in the worst case a compiled formula in PINF is double exponentially large. In addition, Moss (2007) introduced a canonical form of modal formulas for K_n and D_n . While modal formulas can be equivalently transformed into Moss' canonical form, the complexity of the transformation is non-elementary. Hence, Moss' canonical formulas are not a practical normal form for epistemic planning.

In this paper, we first formulate the notion of *logic separability* in multi-agent modal logic, which was originally investigated for first-order logic (Levesque 1998). Informally, a conjunction ϕ of formulas is logically separable w.r.t. a form of reasoning if the reasoning for ϕ can be reduced to the

same reasoning for the conjuncts of ϕ . For example, the formula $\phi = (p \rightarrow q) \wedge (q \rightarrow r)$ is not logically separable since it logically implies a conjunct $p \rightarrow r$ that is not derived by any single conjunct of ϕ . By conjoining ϕ with the implicit conjunct, the new formula becomes logically separable.

Based on the logical separability, we introduce a disjunctive normal form, called *separability-based* DNF (or SDNF) for multi-agent modal logics K_n , D_n , $K45_n$ and $KD45_n$. Similarly, separability-based CNF (or SCNF) can also be defined. For these two normal forms and the two existing normal forms for multi-agent modal logics, we investigate the expressiveness, succinctness, queries and transformations (Darwiche and Marquis 2002). Thus we obtain an almost complete knowledge compilation map for multi-agent modal logics.

The main contributions of this paper contain:

1. We formulate the concept of logical separability for modal terms and formalise some desirable properties for them. Thanks to the notion of logical separability, we are able to define separability-based DNF and CNF, two novel normal forms for K_n .

2. For the two new normal forms as well as CDNF and PINF, we investigate the expressiveness, succinctness, queries and transformations. To the best of our knowledge, we are the first to construct this map for multi-agent modal logics. Interestingly, $SDNF_{\mathcal{L}_0}$ possesses all properties that propositional DNF has, *e.g.*, polytime satisfiability, bounded conjunction, forgetting and so on. In this sense, SDNF is a proper generalisation of propositional DNF for the multi-agent modal logic K_n .

3. As a case study, we illustrate the application of our results in multi-agent epistemic planning.

4. We extend the results of knowledge compilation for K_n to D_n without any modification on normal forms, and to $K45_n$ and $KD45_n$ under the condition that no consecutive modalities of the same agent appear in the given formula.

2 The multi-agent modal logic K_n

In this section, we first recall the syntax and semantics of the multi-agent epistemic logic K_n , and then introduce two normal forms of K_n , and major reasoning tasks in K_n .

Syntax and semantics Throughout this paper, we fix a set \mathcal{A} of n agents and a countable set P of variables. The set of modal formulas, written \mathcal{L}_{\square} , is obtained from the following grammar:

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \square_i\phi,$$

where $p \in P$ and $i \in \mathcal{A}$.

The formula $\square_i\phi$ means that agent i knows ϕ . The symbols \perp , \vee , \rightarrow , \leftrightarrow , and \diamond_i are defined as usual. We use i and j for agents, \mathcal{B} for sets of agents, p and q for variables, Q for finite sets of variables, Φ and Ψ for finite sets of formulas. For an \mathcal{L}_{\square} -formula ϕ , we use $|\phi|$ for the size of ϕ (*i.e.*, the number of occurrences of variables, logical connectives, and modalities in ϕ), $\delta(\phi)$ for the depth of ϕ (*i.e.*, the maximal number of nested occurrences of modalities in ϕ), and $P(\phi)$ for the set of variables appearing in ϕ . A formula ϕ is *smaller* than ψ , if $|\phi| < |\psi|$.

The notions of propositional literals, terms (TE), clauses (CL), disjunctive normal forms (DNF) and conjunctive normal forms (CNF) are defined as usual.

A modal literal is a formula of the form $\square_i\phi$ or $\diamond_i\phi$, where ϕ is a modal formula. A modal literal is positive (resp. negative), if it is of the form $\square_i\phi$ (resp. $\diamond_i\phi$). A formula is *basic*, if it is a propositional formula or modal literal. A *modal term* (resp. *clause*) is a conjunction (resp. disjunction) of basic formulas. For a modal term (resp. clause) ϕ , we use $\text{Prop}(\phi)$ for the set of propositional components (*i.e.*, maximal propositional subformulas) that are conjuncts (resp. disjuncts) of ϕ , $B_i(\phi)$ for the set of formulas ψ such that $\square_i\psi$ is a conjunct (resp. disjunct) of ϕ , and $D_i(\phi)$ for the set of formulas ψ such that $\diamond_i\psi$ is a conjunct (resp. disjunct) of ϕ . For example, consider the modal term $\phi = p \wedge \neg q \wedge \square_i \neg p \wedge \square_i q \wedge \diamond_i (\neg p \vee q) \wedge \diamond_i p$. Thus, $\text{Prop}(\phi) = \{p \wedge \neg q\}$, $B_i(\phi) = \{\neg p, q\}$, and $D_i(\phi) = \{\neg p \vee q, p\}$.

Definition 2.1. A Kripke model M is a tuple $\langle S, R, V \rangle$ where

- S is a non-empty set of possible worlds;
- $R = \{R_i \mid i \in \mathcal{A}\}$ where R_i is a binary relation on S ;
- V is a function assigning to each $s \in S$ a subset of P .

A pointed Kripke model is a pair (M, s) , where M is a Kripke model and s is a world of M , called the *actual world*.

For convenience, we assume that Kripke models are pointed. Given a Kripke model (M, s) and an \mathcal{L}_{\square} -formula ϕ , the satisfaction relation $M, s \models \phi$ is defined as usual and in this case, we say that (M, s) is a model of ϕ . A modal formula ϕ is *satisfiable* if it has a model; ϕ *entails* ψ , written $\phi \models \psi$, if every model of ϕ is also a model of ψ ; ϕ and ψ are *equivalent*, written $\phi \equiv \psi$, if $\phi \models \psi$ and $\psi \models \phi$.

Throughout this paper, we use \mathcal{L} and \mathcal{L}' for fragments of \mathcal{L}_{\square} , and \mathcal{L}_0 and \mathcal{L}'_0 for propositional fragments. We also assume that every propositional term and clause has a polynomial representation in \mathcal{L}_0 and \mathcal{L}'_0 . All propositional fragments considered in (Darwiche and Marquis 2002) conform with this assumption except for full DNF, that is, if each of its variables appears exactly once in every term. We say \mathcal{L} and \mathcal{L}' are *dual*, if there is a polytime reduction f from \mathcal{L} to \mathcal{L}' s.t. for each $\phi \in \mathcal{L}$, $f(\phi) \equiv \neg\phi$, and vice versa. For example, DNF and CNF are dual in propositional logic.

Normal forms The cover disjunctive normal form (ten Cate et al. 2006) and the prime implicate normal form (Bivenvenu 2008) for the description logic \mathcal{ALC} can be seen as two normal forms for K_n since \mathcal{ALC} is a syntactic variant of K_n . We rephrase them in K_n .

Definition 2.2. A formula ϕ is in *cover disjunctive normal form* (CDNF), if it is defined by

$$\phi ::= \tau \wedge \bigwedge_{i \in \mathcal{B}} \nabla_i \Phi_i \mid \phi \vee \phi,$$

where τ is a satisfiable propositional term, each Φ_i is in CDNF, $\mathcal{B} \subseteq \mathcal{A}$, and $\nabla_i \Phi_i$ is a shorthand for $\square_i(\bigvee_{\phi \in \Phi_i} \phi) \wedge \bigwedge_{\phi \in \Phi_i} \diamond_i \phi$.

Definition 2.3. A modal clause¹ c is an *implicate* of ϕ , if $\phi \models c$. A modal clause c is a *prime implicate* of ϕ , if c is an implicate of ϕ and for all implicate c' of ϕ s.t. $c' \models c$, $c \models c'$.

Definition 2.4. A formula ϕ is in *prime implicate normal form* (PINF), if it is \top or \perp , or satisfies the following:

1. $\phi \not\models \top$ and $\phi \not\models \perp$;
2. ϕ is a conjunction $c_1 \wedge \dots \wedge c_n$ of modal clauses where
 - (a) $c_j \not\models c_k$ for $j \neq k$;
 - (b) each prime implicate of ϕ is equivalent to some conjunct c_j ;
 - (c) every c_j is a prime implicate of ϕ s.t. (i) if d is a disjunct of c_j , then $c_j \not\models c_j \setminus \{d\}$; (ii) $|D_i(c_j)| \leq 1$ for $i \in \mathcal{A}$; (iii) for every $i \in \mathcal{A}$, if $\beta \in B_i(c_j) \cup D_i(c_j)$ then β is in PINF; (iv) for every $i \in \mathcal{A}$, $\beta \in B_i(c_j)$ and $\gamma \in D_i(c_j)$, we have $\gamma \models \beta$.

Queries and Transformations Darwiche and Marquis (2002) enumerates a set of knowledge compilation properties for propositional logics, and classifies them into two categories: queries and transformations. The basic queries include polytime tests for satisfiability (**CO**), validity (**VA**), equivalence (**EQ**), sentential entailment (**SE**), clausal entailment (**CE**), implicant (**IM**), model counting (**CT**) and model enumeration (**ME**), while the basic transformations are (bounded) conjunction ($\wedge\mathbf{BC}/\wedge\mathbf{C}$), (bounded) disjunction ($\vee\mathbf{BC}/\vee\mathbf{C}$), negation ($\neg\mathbf{C}$), conditioning (**CD**), and (singleton) forgetting (**SFO/FO**). Most of them can be directly generalized to multi-agent modal logics except modal counting and enumeration since any satisfiable formula has infinitely many distinct models. Due to space limit, we only present the definition of satisfiability check, bounded conjunction and forgetting. For details, please refer to (Darwiche and Marquis 2002).

Definition 2.5. A language \mathcal{L} possesses the property **CO**, if there is a polytime algorithm for deciding its satisfiability.

Definition 2.6. A language \mathcal{L} satisfies $\wedge\mathbf{BC}$, if there is a polytime algorithm generating a formula of \mathcal{L} equivalent to $\phi \wedge \psi$ for each pair of formulas $\phi, \psi \in \mathcal{L}$.

A definition of forgetting for modal logics is defined in (French 2006) and it is further investigated in (Fang, Liu, and van Ditmarsch 2016).

Definition 2.7. Let $\phi \in \mathcal{L}_\square$ and $Q \subseteq P$. We say ψ is a *result of forgetting Q in ϕ* , if

1. $P(\psi) \subseteq P \setminus Q$;
2. for any formula η s.t. $P(\eta) \subseteq P \setminus Q$, $\phi \models \eta$ iff $\psi \models \eta$.

As the result of forgetting is unique up to logical equivalence, we use $\exists Q.\phi$ to denote the result of forgetting Q in ϕ .

Definition 2.8. A language \mathcal{L} satisfies **FO**, if there is a polytime algorithm that maps every formula $\phi \in \mathcal{L}$ and every subset $Q \subseteq P$ to an \mathcal{L} -formula that is equivalent to $\exists Q.\phi$.

¹The definition of modal clauses in (Bienvu 2008) is a bit different from that in this paper. It is defined as a disjunction of propositional literals and modal literals.

3 Separability-based DNF and CNF

In this section, based on logical separability, we introduce a general framework for defining normal forms DNF and CNF in \mathcal{K}_n .

One might wish to define a disjunctive normal form for \mathcal{K}_n as a disjunction of modal terms. However, this option would not work as none of satisfiability and forgetting is tractable for such a normal form. The problem lies in the definition of modal terms as a modal term can be logically inseparable. This can be seen from the following example.

Example 1. Consider the modal term $\phi = \square_i(p \vee q) \wedge \square_i(\neg p \vee q) \wedge \diamond_i\neg q$. Since ϕ is unsatisfiable, it holds that $\phi \models \perp$. But \perp cannot be derived from any modal literal of ϕ alone. Hence, the satisfiability problem of ϕ cannot be decomposed into its conjuncts. Informally, the formula ϕ is logically inseparable.

This example reveals that logical inseparable modal terms do not enjoy the modularity property for satisfiability, i.e., the satisfiability problem of a modal term ϕ cannot be reduced to those of each formula in $Prop(\phi)$, $B_i(\phi)$ and $D_i(\phi)$, leading to the impossibility result of tractable satisfiability check. Consequently, we consider only logically separable modal terms in the following. Based on the above observation, we can formalise the logical separability as follows.

Definition 3.1. A modal term ϕ is *logically separable*, iff for every basic formula η , if $\phi \models \eta$, then there is $\alpha \in Prop(\phi)$ or α is a modal literal of ϕ s.t. $\alpha \models \eta$.

Intuitively, logical separability requires that no logical puzzles are hidden within parts of modal terms.

Example 2. Continued with Example 1, the modal term ϕ is logically inseparable since $\phi \models \perp$ but no conjunct of ϕ entails \perp . The modal term $\psi = \square_i q \wedge \diamond_i \perp$, which is equivalent to ϕ , is logically separable. Thanks to the logical separability of ψ , it is easy to observe that ψ is unsatisfiable since \perp is derived from one conjunct $\diamond_i \perp$.

The logical separability of modal terms enforces the modularity property for major reasoning tasks such as satisfiability check and forgetting. The satisfiability problem of a logically separable term ϕ can be reduced to the satisfiability subproblems of each formula in $Prop(\phi)$ and $D_i(\phi)$.

Proposition 3.1. Let ϕ be a logically separable modal term. Then ϕ is satisfiable iff every formula $\alpha \in Prop(\phi) \cup \bigcup_{i \in \mathcal{A}} D_i(\phi)$ is satisfiable.

Proof. If ϕ is satisfiable, then each conjunct of ϕ is satisfiable. Thus each propositional formula $\alpha \in Prop(\phi)$ is satisfiable. It also holds for any epistemic literal $\diamond_i \gamma$ where $i \in \mathcal{A}$ and $\gamma \in D_i(\phi)$. Thus, each formula $\gamma \in D_i(\phi)$ is satisfiable.

Conversely, assume that for every formula $\alpha \in Prop(\phi) \cup \bigcup_{i \in \mathcal{A}} D_i(\phi)$, α is satisfiable, but ϕ is unsatisfiable. Then, $\phi \models \perp$. Observe that ϕ is logically separable. Then there is a propositional formula $\alpha \in Prop(\phi)$ or a modal literal α of ϕ s.t. $\alpha \models \perp$. It is easy to prove that α must be an unsatisfiable propositional formula or unsatisfiable negative modal literal. By the assumption, it is impossible that α is

propositional. Hence, α is a negative modal literal $\diamond_i\gamma$. This implies that γ is unsatisfiable, a contradiction. \square

Similarly, forgetting a set Q of variables in a logically separable modal term ϕ can be accomplished by individually forgetting Q in each formula of $Prop(\phi)$, $B_i(\phi)$ and $D_i(\phi)$.

Proposition 3.2. *Let ϕ be a logically separable modal term and Q a set of variables. Then $\exists Q.\phi \equiv \bigwedge_{\alpha \in Prop(\phi)} (\exists Q.\alpha) \wedge \bigwedge_{i \in \mathcal{B}} [\bigwedge_{\beta \in B_i(\phi)} (\Box_i(\exists Q.\beta_i)) \wedge \bigwedge_{\gamma \in D_i(\phi)} (\Diamond_i(\exists Q.\gamma))]$.*

To prove Proposition 3.2, we need a lemma.

Lemma 3.1. *Let ϕ be a satisfiable logically separable modal term. Then, the following statements hold:*

1. For each propositional formula α' , $\phi \models \alpha'$ iff $\alpha \models \alpha'$ for some $\alpha \in Prop(\phi)$;
2. For each $i \in \mathcal{A}$ and each positive modal literal $\Box_i\beta'$, $\phi \models \Box_i\beta'$ iff $\beta \models \beta'$ for some $\beta \in B_i(\phi)$;
3. For each $i \in \mathcal{A}$ and each negative modal literal $\Diamond_i\gamma'$, $\phi \models \Diamond_i\gamma'$ iff $\gamma \models \gamma'$ for some $\gamma \in D_i(\phi)$.

Proof. We show only the second statement. The other proofs are similar.

\Leftarrow : Since $\beta \models \beta'$, $\Box_i\beta \models \Box_i\beta'$. Note that $\Box_i\beta$ is a conjunction of ϕ . Thus, $\phi \models \Box_i\beta'$.

\Rightarrow : Since ϕ is logically separable, $\eta \models \Box_i\beta'$, where $\eta \in Prop(\phi)$ or η is a modal literal of ϕ . Moreover, as $\Box_i\beta'$ is a positive modal literal, η must be a positive modal literal too. Suppose that $\eta = \Box_i\beta$ where $\beta \in B_i(\phi)$. It follows from $\Box_i\beta \models \Box_i\beta'$ that $\beta \models \beta'$. \square

We are ready to give the proof of Proposition 3.2.

Proof. Let ψ denote formula on the right-hand-side. Consider two possible cases:

Case 1. ϕ is unsatisfiable: Then $\exists Q.\phi$ is also unsatisfiable, matching the requirements given in the definition of forgetting (cf. Definition 2.7). By Proposition 3.1, there is an unsatisfiable formula $\alpha \in Prop(\phi)$, or for some $i \in \mathcal{A}$, there is $\gamma \in D_i(\phi)$ s.t. γ is unsatisfiable. Suppose that α is unsatisfiable. Then ψ is also unsatisfiable since ψ contains an unsatisfiable conjunct $\exists Q.\alpha$. Similarly, ψ is unsatisfiable in the case where $\gamma \in D_i(\phi)$ is unsatisfiable.

Case 2. ϕ is satisfiable: We only show the only-if direction of Condition 2 in Definition 2.7: for each formula η s.t. $P(\eta) \subseteq P \setminus Q$, if $\phi \models \eta$, then $\psi \models \eta$. It is easy to see the other conditions.

We prove the above proposition by induction on $\delta(\phi)$, the depth of ϕ . By Definition 2.7, the proposition holds for the base case $\delta(\phi) = 1$. Suppose that $\delta(\phi) > 1$. By Theorem 2.4.9 in (Bienvenu 2009), every \mathcal{L}_{\Box} -formula can be equivalently transformed into a conjunction of modal clauses. So we assume without loss of generality that η is a conjunction of modal clauses. Let c be a conjunct of η and of the form $\bigvee_{\alpha' \in Prop(c)} \alpha' \vee \bigvee_{i \in \mathcal{B}'} [\bigvee_{\beta' \in B_i(c)} (\Box_i\beta') \vee \bigvee_{\gamma' \in D_i(c)} (\Diamond_i\gamma')]$. It suffices to show that $\psi \models c$. For simplicity, we let $\mathcal{B} = \mathcal{B}'$. Since $\phi \models c$, it is easy to see that at least one of the following conditions holds.

1. $\bigwedge_{\alpha \in Prop(\phi)} \alpha \wedge \bigwedge_{\alpha' \in Prop(c)} (\neg\alpha')$ is unsatisfiable;

2. there exist $i \in \mathcal{B}$ and $\gamma \in D_i(\phi)$ s.t. $\gamma \wedge \bigwedge_{\beta \in B_i(\phi)} \beta \wedge \bigwedge_{\gamma' \in D_i(c)} (\neg\gamma')$ is unsatisfiable;
3. there exist $i \in \mathcal{B}$ and $\beta' \in B_i(c)$ s.t. $\neg\beta' \wedge \bigwedge_{\beta \in B_i(\phi)} \beta \wedge \bigwedge_{\gamma' \in D_i(c)} (\neg\gamma')$ is unsatisfiable.

Once the conditions in Definition 2.7 are satisfied, it follows that $\gamma \wedge \bigwedge_{\beta \in B_i(\phi)} \beta \models \bigvee_{\gamma' \in D_i(c)} \gamma'$. So $\Diamond_i(\gamma \wedge \bigwedge_{\beta \in B_i(\phi)} \beta) \models \Diamond_i(\bigvee_{\gamma' \in D_i(c)} \gamma')$. Since ϕ entails the former formula, we get that $\phi \models \Diamond_i(\bigvee_{\gamma' \in D_i(c)} \gamma')$. By Lemma 3.1, there is $\gamma^* \in D_i(\phi)$ s.t. $\gamma^* \models \bigvee_{\gamma' \in D_i(c)} \gamma'$. The formula $\exists Q.\gamma^*$ is the result of forgetting Q in γ^* . By the induction hypothesis, $\exists Q.\gamma^* \models \bigvee_{\gamma' \in D_i(c)} \gamma'$. Hence, $\Diamond_i(\exists Q.\gamma^*) \models \Diamond_i(\bigvee_{\gamma' \in D_i(c)} \gamma')$. Since $\Diamond_i(\exists Q.\gamma^*)$ is a conjunct of ψ , and $\Diamond_i \bigvee_{\gamma' \in D_i(c)} \gamma' \models \eta$, $\psi \models \eta$. \square

The following proposition gives the smallest logically separable modal term representation. In this normal form, there are at most one propositional part, and at most one positive modal literal for each agent. Moreover, every formula inside \Diamond_i entails that inside \Box_i .

Proposition 3.3. *The smallest logically separable modal term representation of a modal term ϕ satisfies*

1. $|Prop(\phi)| \leq 1$;
2. for each $i \in \mathcal{A}$, $|B_i(\phi)| \leq 1$;
3. for each $i \in \mathcal{A}$, $\beta \in B_i(\phi)$ and $\gamma \in D_i(\phi)$, $\gamma \models \beta$.

Proof. Observe that the smallest representation of unsatisfiable formula is \perp . Thus, it is easy to the statements hold when ϕ is unsatisfiable. We now assume that ϕ is satisfiable, and only verify Condition 1. The other two conditions can be proven similarly. On the contrary, suppose that $\alpha_1, \alpha_2 \in Prop(\phi)$ but they are distinct. If $\alpha_1 \models \alpha_2$ or $\alpha_2 \models \alpha_1$, then one of them is redundant, and ϕ is not the smallest form. Otherwise, $\alpha_1 \not\models \alpha_2$ and $\alpha_2 \not\models \alpha_1$. Thus, neither α_1 nor α_2 entails their conjunction $\alpha_1 \wedge \alpha_2$. This violates Lemma 3.1. \square

Forgetting in a logically separable modal term ϕ may not be tractably computed. This is because that some subformulas of ϕ may not be tractable for forgetting. To remedy this, we impose further conditions on modal terms. We not only require the logically separable modal term ϕ to be the smallest form, but also restrict the propositional part of ϕ to be in \mathcal{L}_0 , and every formula of $B_i(\phi)$ and $D_i(\phi)$ to be the disjunction of formulas in this form.

Definition 3.2. A modal term ϕ is a *separability-based term with \mathcal{L}_0* ($STE_{\mathcal{L}_0}$), if it is of the syntactic form $\alpha \wedge \bigwedge_{i \in \mathcal{B}} (\Box_i\beta_i \wedge \bigwedge_j \Diamond_i\gamma_{ij})$ s.t.

1. $\alpha \in \mathcal{L}_0$ and $\mathcal{B} \subseteq \mathcal{A}$;
2. β_i 's and γ_{ij} 's are disjunctions of $STE_{\mathcal{L}_0}$'s;
3. $\gamma_{ij} \models \beta_i$ for any i and j .

Dual to separability-based terms, it is natural to define separability-based clauses.

Definition 3.3. A modal clause ϕ is a *separability-based clause with \mathcal{L}_0* ($SCL_{\mathcal{L}_0}$), if it is of the syntactic form $\alpha \vee \bigvee_{i \in \mathcal{B}} (\Diamond_i\beta_i \vee \bigvee_j \Box_i\gamma_{ij})$ s.t.

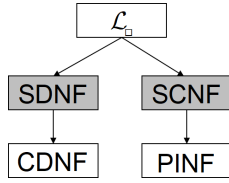


Figure 1: Succinctness of normal forms in K_n and D_n . An edge $\mathcal{L} \rightarrow \mathcal{L}'$ means that \mathcal{L} is strictly more succinct than \mathcal{L}' . No edge between \mathcal{L} and \mathcal{L}' indicates that they are incomparable.

1. $\alpha \in \mathcal{L}_0$ and $\mathcal{B} \subseteq \mathcal{A}$;
2. β_i 's and γ_{ij} 's are conjunctions of $SCL_{\mathcal{L}_0}$'s;
3. $\beta_i \models \gamma_{ij}$ for any i and j .

We are ready to define separability-based DNF and CNF.

Definition 3.4. A formula ϕ is in *separability-based disjunctive (resp. conjunctive) normal form with \mathcal{L}_0* ($SDNF_{\mathcal{L}_0}$ (resp. $SCNF_{\mathcal{L}_0}$)), if ϕ is a disjunction (resp. conjunction) of $STE_{\mathcal{L}_0}$'s (resp. $SCL_{\mathcal{L}_0}$'s).

We remark that the normal form $SDNF_{TE}$ is the form $SDNF$ of which propositional components are propositional terms (TE). The normal form $SCNF_{CL}$ are similarly defined. It is easy to verify that two existing normal forms $CDNF$ and $PINF$ are fragments of $SDNF_{TE}$ and $SCNF_{CL}$ respectively.

Proposition 3.4. $CDNF \subset SDNF_{TE}$ and $PINF \subset SCNF_{CL}$.

Proof. In the definition of $CDNF$ (Definition 2.2), each $\tau \wedge \bigwedge_{i \in \mathcal{B}} \nabla_i \Phi_i$ is an STE since each $\nabla_i \Phi_i$ is a shorthand for $\Box_i (\bigvee_{\phi \in \Phi_i} \phi) \wedge \bigwedge_{\phi \in \Phi_i} \Diamond_i \phi$, and each formula inside the modality \Diamond_i entails $\bigvee_{\phi \in \Phi_i} \phi$. Thus, $CDNF$ is a fragment of $SDNF$.

In the definition of $PINF$ (Definition 2.4), Condition 2-(c)-(ii) implies the condition of smallest logically separable modal clause representation. In addition, Conditions 2-(c)-(iii) and 2-(c)-(iv) correspond to Conditions 2 and 3 in Definition 3.3, respectively. So $PINF$ is a fragment of $SCNF$. \square

4 Expressiveness and Succinctness

In this section, we analyze spatial complexity of the four normal forms and compare them in terms of succinctness. Our main results include: (1) the sizes of the $SDNF$ and $SCNF$ representations of a given formula ϕ are single-exponential in the size of ϕ , and (2) we provide a full picture of the succinctness for the four normal forms $SDNF$, $SCNF$, $CDNF$ and $PINF$.

We first show that the transformations into $SDNF$ and $SCNF$ cause a single-exponential blowup and this upper bound is optimal.

Proposition 4.1. Any \mathcal{L}_\square -formula ϕ is equivalent to a formula in $SDNF_{\mathcal{L}_0}$ (resp. $SCNF_{\mathcal{L}_0}$) that is at most single-exponentially large in the size of ϕ . Moreover, there is an \mathcal{L}_\square -formula ϕ s.t. the size of $SDNF_{\mathcal{L}_0}$ (resp. $SCNF_{\mathcal{L}_0}$) for ϕ is single-exponential in $|\phi|$.

Proof. We provide a transformation for \mathcal{L}_\square into $SDNF_{\mathcal{L}_0}$: We first compile a modal formula into a $CDNF$ formula

and then replace all propositional components with their equivalent \mathcal{L}_0 -formulas. We note that every modal formula is equivalent to a formula in $CDNF$ that is at most single exponentially large in the original formula size (ten Cate et al. 2006). By Prop. 3.4, $CDNF$ is a subset of $SDNF_{TE}$. Thus, every modal formula can be equivalently transformed into a formula in $SDNF_{TE}$. As each propositional term can be turned into a polysize \mathcal{L}_0 -formula, we obtain the single-exponential upper bound of the transformation.

Since $SCNF$ is dual to $SDNF$, the transformation into $SCNF$ has the same upper bound.

To prove that the upper bound is optimal, we observe that the smallest $SDNF$ representation for the formula $\bigwedge_{j=1}^n (p_j \vee \Diamond_i q_j)$ has at least 2^n modal terms, each of which is equivalent to $\bigwedge_{j=1}^n r_j$ where r_j is either p_j or $\Diamond_i q_j$. Dually, $\bigvee_{j=1}^n (p_j \wedge \Box_i q_j)$ has also at least 2^n modal clauses. \square

By Theorem 4.1.13 in (Bienvenu 2009), the size of each $PINF$ formula equivalent to $\bigwedge_{j=1}^n [(\Box_i p_j \wedge \Diamond_i q_j) \vee (\Box_i r_j \wedge \Diamond_i s_j)]$ is at least double-exponential large in the size of the given formula. Therefore, our new normal forms have a better space complexity than $PINF$. We remark that, despite the worst case complexity is exponential, the transformation can be performed effectively for some practical applications. In particular, both progression and entailment check are tractable in $SDNF$ for multi-agent epistemic planning, and thus the cost of compilation phase is amortized over subsequent progression and entailment. The details will be discussed in Section 6.

We now analyze the relationships between different normal forms in terms of the succinctness.

Definition 4.1. A language \mathcal{L} is *at least as succinct as \mathcal{L}'* , written $\mathcal{L} \leq \mathcal{L}'$, if there is a polynomial f s.t. for any formula $\phi \in \mathcal{L}'$, there exists a formula $\psi \in \mathcal{L}$ s.t. $\psi \equiv \phi$ and $|\psi| \leq f(|\phi|)$.

The relation \leq is clear reflexive and transitive, i.e., a pre-order over languages. A language \mathcal{L} is *strictly more succinct than \mathcal{L}'* , written $\mathcal{L} < \mathcal{L}'$, if $\mathcal{L} \leq \mathcal{L}'$ and $\mathcal{L}' \not\leq \mathcal{L}$. Two languages \mathcal{L} and \mathcal{L}' are incomparable, if $\mathcal{L} \not\leq \mathcal{L}'$ and $\mathcal{L}' \not\leq \mathcal{L}$.

The results about succinctness are depicted in Figure 1. We explicate them in the following.

We note that $SDNF$ and $SCNF$ are subsets of \mathcal{L}_\square . By Proposition 4.1, \mathcal{L}_\square is strictly more succinct than both $SDNF$ and $SCNF$.

Proposition 4.2. $\mathcal{L}_\square < SDNF_{\mathcal{L}_0}$ and $\mathcal{L}_\square < SDNF_{\mathcal{L}_0}$.

We show that our new normal forms $SDNF$ and $SCNF$ are strictly more succinct than $CDNF$ and $PINF$, respectively.

Proposition 4.3. $SDNF_{\mathcal{L}_0} < CDNF$ and $SCNF_{\mathcal{L}_0} < PINF$.

Proof. Since $CDNF$ and $PINF$ are subsets of $SDNF_{TE}$ and $SCNF_{CL}$ respectively (cf. Prop. 3.4) and every propositional term and clause can be efficiently represented by \mathcal{L}_0 , we get that $SDNF_{\mathcal{L}_0} \leq CDNF$ and $SCNF_{\mathcal{L}_0} \leq PINF$. Every formula is equivalent to a $SCNF$ -formula that is at most single-exponentially larger, and converting the formula $\bigwedge_{j=1}^n [(\Box_i p_j \wedge \Diamond_i q_j) \vee (\Box_i r_j \wedge \Diamond_i s_j)]$ into $PINF$ causes an at least double exponential blowup. Thus, $PINF \not\leq SCNF_{\mathcal{L}_0}$.

It remains to prove $\text{CDNF} \not\leq \text{SDNF}_{\mathcal{L}_0}$. We define a class of formulas ϕ_k inductively as follows: (1) $\phi_0 = p \vee q$; (2) $\phi_k = \phi_0 \wedge \Box_i \phi_{k-1}$. The size of ϕ_k is linear in k , more precisely, $3 + 5k$. Let f be a polynomial s.t. any clause c has a representation in \mathcal{L}_0 with size at most $f(|c|)$. Each ϕ_k has an $\text{SDNF}_{\mathcal{L}_0}$ representation with size $f(3) \cdot (k+1) + 2k$. The smallest CDNF formula ψ_k equivalent to ϕ_k is $(p \wedge \Box_i \psi_{k-1} \wedge \Diamond_i \psi_{k-1}) \vee (p \wedge \Box_i \perp) \vee (q \wedge \Box_i \psi_{k-1} \wedge \Diamond_i \psi_{k-1}) \vee (q \wedge \Box_i \perp)$. By induction on k , the size of ψ_k is single-exponential in k . \square

As mentioned before, no formula in SDNF (resp. SCNF) equivalent to the formula $\bigwedge_{j=1}^n (p_j \vee \Diamond_i q_j)$ (resp. $\bigvee_{j=1}^n (p_j \wedge \Box_i q_j)$) has size polynomial in the given formula. Moreover, the formula $\bigwedge_{j=1}^n (p_j \vee \Diamond_i q_j)$ is in PINF . The CDNF formula $\bigvee_{j=1}^n (p_j \wedge \Box_i q_j \wedge \Diamond_i q_j) \vee \bigvee_{j=1}^n (p_j \wedge \Box_i \perp)$ is equivalent to $\bigvee_{j=1}^n (p_j \wedge \Box_i q_j)$ and with polynomial size. Thus, we get

Proposition 4.4. $\text{SDNF}_{\mathcal{L}_0} \not\leq \text{PINF}$ and $\text{SCNF}_{\mathcal{L}_0} \not\leq \text{CDNF}$.

The remaining succinctness results shown in Figure 1 are easy consequences of Proposition 4.4 via exploiting the transitivity of \leq .

5 Queries and Transformations

In this section, we mainly examine $\text{SDNF}_{\mathcal{L}_0}$ w.r.t. the knowledge compilation properties proposed in (Darwiche and Marquis 2002), and identify conditions of \mathcal{L}_0 under which some useful properties hold in $\text{SDNF}_{\mathcal{L}_0}$. More importantly, we provide an almost complete picture for tractability of the four normal forms in Table 1. Interestingly, the normal form $\text{SDNF}_{\mathcal{L}_0}$ supports as many properties as propositional DNF. From the knowledge compilation perspective, SDNF is a proper alternative to the modal counterpart for DNF.

Queries It is well-known that the satisfiability of DNF is tractable (Darwiche and Marquis 2002). This positive result is still valid for $\text{SDNF}_{\mathcal{L}_0}$ if \mathcal{L}_0 allows polytime satisfiability check. We briefly introduce the procedure for the satisfiability of an SDNF formula ϕ . Suppose that we are given a polytime subprocedure $\text{SAT}_{\mathcal{L}_0}$ which returns a Boolean value stating whether or not an \mathcal{L}_0 -formula is satisfiable. If $\phi \in \mathcal{L}_0$, then it is satisfiable in K_n iff the subprocedure $\text{SAT}_{\mathcal{L}_0}$ returns yes. If ϕ is a disjunction of modal terms, then the satisfiability problem of ϕ can be reduced to that of each disjunct ϕ . Due to the modularity property (cf. Proposition 3.1), a logically separable modal term $\alpha \wedge \bigwedge_{i \in \mathcal{B}} (\Box_i \beta_i \wedge \bigwedge_j \Diamond_i \gamma_{ij})$ is satisfiable iff all of α and γ_{ij} 's are satisfiable. In summary, deciding if a SDNF formula ϕ is satisfiable can be tractably performed. Interestingly, even if the satisfiability of \mathcal{L}_0 is NP -Complete, the satisfiability of $\text{SDNF}_{\mathcal{L}_0}$ falls into Δ_2^P since the decision procedure calls the subprocedure $\text{SAT}_{\mathcal{L}_0}$ at most $|\phi|$ times.

Proposition 5.1. *If \mathcal{L}_0 satisfies CO , then $\text{SDNF}_{\mathcal{L}_0}$ satisfies CO .*

The negative results about other queries also carry forward from DNF to SDNF .

Proposition 5.2. *$\text{SDNF}_{\mathcal{L}_0}$ does not satisfy VA , SE , EQ , CE or IM unless $\text{P} = \text{NP}$.*

Proof. **VA:** Let $\tau_1 \vee \dots \vee \tau_n$ be a DNF where τ_k is a propositional term. For each τ_k , there exists $\psi_k \in \mathcal{L}_0$ s.t. $\psi_k \equiv \tau_k$ and $|\psi_k| < f(|\tau_k|)$ for some polynomial f . Clearly, $\psi_1 \vee \dots \vee \psi_n$ is in $\text{SDNF}_{\mathcal{L}_0}$. If we can decide whether this disjunction is valid in polytime, then the validity of DNF can be tractably accomplished. However, the latter problem is coNP -complete. A contradiction.

SE and **EQ:** Since **SE** implies **VA**, $\text{SDNF}_{\mathcal{L}_0}$ does not satisfy **SE**. Similarly, $\text{SDNF}_{\mathcal{L}_0}$ fails to satisfy **EQ**.

CE and **IM:** Let $\Box_i \phi$ be a modal literal where ϕ is propositional. Clearly, \top is in $\text{SDNF}_{\mathcal{L}_0}$ and $\Box_i \phi$ is a modal term. We get that $\top \models \Box_i \phi$ iff ϕ is valid. The validity problem of propositional logic is coNP -complete, and so is the problem that decides if $\top \models \Box_i \phi$. Hence, $\text{SDNF}_{\mathcal{L}_0}$ does not satisfy **CE**. Similarly, $\text{SDNF}_{\mathcal{L}_0}$ fails to satisfy **IM**. \square

Unlike DNF, even if \mathcal{L}_0 satisfies the polytime clause entailment check (**CE**), $\text{SDNF}_{\mathcal{L}_0}$ does not possess such a property. Moreover, it is impossible to propose a normal form permitting such a check. Let us illustrate this in the following. Suppose that $\Box_i \phi$ is a modal literal where ϕ is propositional. We get that $\top \models \Box_i \phi$ iff ϕ is valid. The validity problem of propositional logic is coNP -complete, and so is the problem that decides if $\top \models \Box_i \phi$.

This motivates us to propose a restricted version of clausal entailment check by restricting the form of propositional subformulas appearing in the modal clausal.

Definition 5.1. A language \mathcal{L} satisfies $\text{CE}_{\mathcal{L}_0}$ (resp. $\text{IM}_{\mathcal{L}_0}$), if there is a polytime algorithm that maps every formula $\phi \in \mathcal{L}$ and every $\text{SCL}_{\mathcal{L}_0}$ (resp. $\text{STE}_{\mathcal{L}_0}$) ψ to 1 if $\phi \models \psi$ (resp. $\psi \models \phi$) holds, and to 0 otherwise.

$\text{SDNF}_{\mathcal{L}_0}$ supports the restricted polytime clausal entailment check $\text{CE}_{\mathcal{L}'_0}$ under conditions that \mathcal{L}_0 satisfies both **CO** and $\wedge \text{BC}$, and \mathcal{L}'_0 is dual to \mathcal{L}_0 . Let $\phi \in \text{SDNF}_{\mathcal{L}_0}$ and $\psi \in \text{SCL}_{\mathcal{L}'_0}$. Due to the fact that $\phi \models \psi$ iff $\phi \wedge \neg \psi$ is unsatisfiable, we reduce the clausal entailment problem to the satisfiability problem. The main insight is first constructing an $\text{SDNF}_{\mathcal{L}_0}$ formula ϕ' equivalent to $\phi \wedge \neg \psi$, and then checking its satisfiability. This can be achieved by first obtaining an $\text{STE}_{\mathcal{L}_0}$ ψ' equivalent to $\neg \psi$, and then conjoining ψ' with ϕ . The facts that $\text{SCL}_{\mathcal{L}'_0}$ and $\text{STE}_{\mathcal{L}_0}$ are dual and that $\text{SDNF}_{\mathcal{L}_0}$ satisfies bounded conjunction (cf. Proposition 5.4) imply that the construction of ϕ' is tractable. Since the satisfiability of SDNF is tractable (cf. Proposition 5.1), we get that deciding if $\phi \models \psi$ can be accomplished in polytime. However, $\text{SDNF}_{\mathcal{L}_0}$ does not satisfy the restricted polytime implicant check even if \mathcal{L}'_0 is dual to \mathcal{L}_0 .

Proposition 5.3. *Let \mathcal{L}_0 and \mathcal{L}'_0 be dual. If \mathcal{L}_0 satisfies **CO** and $\wedge \text{BC}$, then $\text{SDNF}_{\mathcal{L}_0}$ satisfies $\text{CE}_{\mathcal{L}'_0}$. However, $\text{SDNF}_{\mathcal{L}_0}$ does not satisfy $\text{IM}_{\mathcal{L}'_0}$ unless $\text{P} = \text{NP}$.*

Proof. The proof for the $\text{CE}_{\mathcal{L}_0}$ property is illustrated above, and the proof for the $\text{IM}_{\mathcal{L}_0}$ property is similar to that for the **IM** property (cf. Proposition 5.2). \square

Transformations We now present the results about the transformations. It follows from Definition 3.4 that the disjunction of SDNF formulas can be generated efficiently. However, SDNF supports neither polytime conjunction nor

\mathcal{L}	CO	VA	SE	EQ	CE	CE $_{\mathcal{L}'_0}$	IM	IM $_{\mathcal{L}'_0}$	$\neg\mathbf{C}$	$\wedge\mathbf{C}$	$\wedge\mathbf{BC}$	$\vee\mathbf{C}$	$\vee\mathbf{BC}$	CD	FO	SFO
SDNF $_{\mathcal{L}_0}$	✓*	○	○	○	○	✓*	○	○	×	×	✓*	✓	✓	✓*	✓*	✓*
SCNF $_{\mathcal{L}_0}$	○	✓*	○	○	○	○	○	✓*	×	✓	✓	×	✓*	✓*	○	?
CDNF	✓	○	○	○	○	✓*	○	○	×	×	✓	✓	✓	✓	✓	✓
PINF	✓	✓	✓	✓	○	✓*	○	✓*	×	×	×	×	✓	?	✓	✓

Table 1: Queries and transformations for normal forms in \mathbf{K}_n and \mathbf{D}_n . The occurrence of ✓ (or ✓*) in the cell of row r and column c means that “the normal form \mathcal{L}_r given in row r satisfies the property P_c given in column c (under specific conditions in the case of ✓*)”. The symbol × means that “ \mathcal{L}_r does not satisfy P_c ”, ○ means that “ \mathcal{L}_r does not satisfy P_c unless $\mathbf{P} = \mathbf{NP}$ ”, and ? means that “the issue remains open”.

polytime negation. Fortunately, it supports bounded conjunction. The following proposition states the above results.

Proposition 5.4.

- SDNF $_{\mathcal{L}_0}$ satisfies $\vee\mathbf{C}$, and does not satisfy $\wedge\mathbf{C}$ or $\neg\mathbf{C}$.
- If \mathcal{L}_0 satisfies $\wedge\mathbf{BC}$, then SDNF $_{\mathcal{L}_0}$ satisfies $\wedge\mathbf{BC}$.

Proof. • $\wedge\mathbf{C}$: By Proposition 4.1, any SDNF representation of $\phi_n = \bigwedge_{j=1}^n (p_j \vee \diamond_i q_j)$ has size exponential in n . But the size of ϕ_n is linear in n . Hence, $\wedge\mathbf{C}$ does not hold for SDNF $_{\mathcal{L}_0}$.

• $\neg\mathbf{C}$: On the contrary, assume that SDNF $_{\mathcal{L}_0}$ satisfies $\neg\mathbf{C}$. This, together with the fact that $\vee\mathbf{C}$ holds for SDNF $_{\mathcal{L}_0}$, imply that SDNF $_{\mathcal{L}_0}$ satisfy $\wedge\mathbf{C}$, a contradiction.

• $\wedge\mathbf{BC}$: By assumption, there exists a polytime algorithm that maps every pair of \mathcal{L}_0 -formulas α and α' to an \mathcal{L}_0 -formula that is equivalent to $\alpha \wedge \alpha'$. Let the polynomial f be the time complexity, k the degree of f , and c the sum of the coefficients of f . So $f(|\alpha|, |\alpha'|) \leq c|\alpha|^k |\alpha'|^k$. Suppose that $\phi = \bigvee_{j=1}^m \psi_j$ and $\phi' = \bigvee_{l=1}^m \psi'_l$ are two formulas in SDNF $_{\mathcal{L}_0}$. Thus, any ψ_j and ψ'_l are in STE $_{\mathcal{L}_0}$. We can construct a formula ϕ'' in SDNF $_{\mathcal{L}_0}$ that is equivalent to $\phi \wedge \phi'$ by simply taking the disjunction of all modal terms ψ''_{jl} where $\psi''_{jl} \equiv \psi_j \wedge \psi'_l$ for $1 \leq j \leq m$ and $1 \leq l \leq m$. It is easy to see that $|\phi''| \leq c|\phi|^k |\phi'|^k$ if $|\psi''_{jl}| \leq c|\psi_j|^k |\psi'_l|^k$.

It remains to prove that for any two STE's ψ and ψ' , there is an STE ψ'' s.t. $\psi'' \equiv \psi \wedge \psi'$ and $|\psi''| \leq c|\psi|^k |\psi'|^k$. Suppose that $\psi = \alpha \wedge \bigwedge_{i \in \mathcal{B}} (\Box_i \beta_i \wedge \bigwedge_{j=1}^{m_i} \diamond_i \gamma_{ij})$ and $\psi' = \alpha' \wedge \bigwedge_{i \in \mathcal{B}'} (\Box_i \beta'_i \wedge \bigwedge_{l=1}^{n_i} \diamond_i \gamma'_{il})$. We assume without loss of generality that $\mathcal{B} = \mathcal{B}'$. We construct a formula $\psi'' = \alpha'' \wedge \bigwedge_{i \in \mathcal{B}} (\Box_i \beta''_i \wedge \bigwedge_{j=1}^{m_i} \diamond_i \gamma''_{ij} \wedge \bigwedge_{l=1}^{n_i} \diamond_i \gamma''_{il})$, where $\alpha'' \equiv \alpha \wedge \alpha'$, $\beta''_i \equiv \beta_i \wedge \beta'_i$, $\gamma''_{ij} \equiv \beta'_i \wedge \gamma_{ij}$ and $\gamma''_{il} \equiv \beta_i \wedge \gamma'_{il}$. Then ψ'' is an STE with size at most $c|\psi|^k |\psi'|^k$. □

It is easy to design procedures for generating the results of conditioning and forgetting of SDNF formulas respectively. They are similar to the procedure for the satisfiability problem of SDNF formulas. Thus, we get

Proposition 5.5. *If \mathcal{L}_0 satisfies CD (resp. FO/SFO), then SDNF $_{\mathcal{L}_0}$ satisfies CD (resp. FO/SFO).*

Since SCNF is dual to SDNF, it is similar to obtain corresponding results for SCNF $_{\mathcal{L}_0}$. CDNF supports as many knowledge compilation properties as SDNF on the ground that it is a fragment of SDNF $_{\text{TE}}$ and propositional term satisfies the specific conditions. Most results for PINF presented

in this section originate from (Darwiche and Marquis 2002; Bienvenu 2009). Table 1 summarizes the query and transformation properties of the four normal form.

We conclude this section by briefly discussing our main results. Given a proper propositional fragment \mathcal{L}_0 , SDNF $_{\mathcal{L}_0}$ is tractable for all of queries and transformations that DNF admits, and thus being a proper alternative to the modal counterpart for DNF. For the similar reason, SCNF and PINF can be viewed as the modal counterpart for CNF and prime implicate, respectively. It is worth noting that SDNF supports polytime clausal entailment checking (CE $_{\mathcal{L}_0}$), bounded conjunction ($\wedge\mathbf{BC}$) and forgetting (FO), which are important for multi-agent epistemic planning. SCNF does not satisfy polytime entailment check or forgetting while bounded conjunction does not hold for PINF. Though CDNF satisfy all of CE $_{\mathcal{L}_0}$, $\wedge\mathbf{BC}$ and FO, SDNF has two major advantages over CDNF. In theory, it is strictly more succinct than CDNF (cf. Proposition. 4.3). On the practice side, SDNF is more flexible than CDNF since the former can be designed via making use of efficient and compact propositional representations, e.g., OBDD. From the knowledge compilation perspective, SDNF is more suitable for multi-agent epistemic planning than the other three normal forms.

6 Application to Multi-Agent Epistemic Planning

Based on the notion of canonical formulas, Aucher (2011) gave a syntactic representation of progression w.r.t. epistemic actions. Due to the high complexity of their canonical formulas, this approach is not practical for implementation of multi-agent epistemic planning. Bienvenu, Fargier, and Marquis (2010) proposed a tractable approach to progression and entailment check for single-agent epistemic planning. It is challenging to extend their approach to multi-agent case in that it is necessary to consider not only first-order knowledge (*i.e.*, to know what is the world), but also high-order knowledge, (*i.e.*, to know what other agents know).

In this section, we briefly explain how to apply our results in multi-agent epistemic planning. Especially, two essential procedures (progression and entailment check) can be accomplished efficiently by using the form SDNF.

A multi-agent epistemic planning problem consists of the initial KB, the goal formula, and ontic and epistemic actions. The main distinction between epistemic planning and traditional planning are that (1) the initial KB and the goal formula are expressed in modal logics other than propositional logics, and (2) the former involves not only ontic actions

changing the world but also epistemic actions modifying the mental attitude of agents. We begin with a multi-agent epistemic planning problem adapted from (Kominis and Geffner 2015) to explain the progression and entailment check.

Example 3. *There are four rooms r_1, r_2, r_3 and r_4 in a row from left to right on a corridor. Each of two boxes b_1 and b_2 is located in a room. Two agents i and j can move from one room to its left and right adjacent room. When an agent is in a room, she can sense if a box is in the room. Initially, agent i is in r_1 , j is in r_4 , box b_1 is in r_2 and b_2 is in r_3 . Each agent only knows where herself is (i.e., $\Box_i at(i, r_1) \wedge \Box_j at(j, r_4)$). The goals for agents i and j are to determine the rooms in which b_1 and b_2 respectively.*

In the following, we discuss the progression w.r.t. ontic and epistemic actions.

An ontic action a_o is associated with a pair of functions $\langle pre, eff \rangle$ where $pre \in \mathcal{L}_{\Box}$ specifies the precondition and eff is the effect. In order to express the effect, we consider two versions p and p' of each variable p . For each $p \in P$, the unprimed version p means that p holds before performing the action a_o , and the primed one p' states that p holds after. The effect is a conjunction of formulas of the form: $p' \equiv \delta^+ \vee (p \wedge \neg \delta^-)$. Two propositional formulas δ^+ and δ^- are conditions that make p true and false respectively. Intuitively, the effect means that p holds after executing a_o iff δ^+ holds, or δ^- does not hold and p holds initially. In Example 3, if agent i knows that she is not in the rightmost room r_4 , then she can move right, and thus $pre(right(i)) = \Box_i(\neg at(i, r_4))$; after moving right, agent i will be in room r_{n+1} if she is in r_n initially, and therefore $eff(right(i)) = \bigwedge_{n=1}^3 [at'(i, r_{n+1}) \equiv at(i, r_n)]$.

In this paper, we assume that all ontic actions are public and that there is no sort of imperfect information in them. This assumption was proposed in (Kominis and Geffner 2015). To exactly capture progression under this assumption, it is necessary to progress KBs according to the action effect via high-order knowledge shared by all agents.

Definition 6.1. Let k be a natural number and ϕ a formula. The formula $\blacksquare^k \phi$ is inductively defined:

- $\blacksquare^1 \phi = \bigwedge_{i \in \mathcal{A}} \Box_i \phi$;
- $\blacksquare^k \phi = \blacksquare^{k-1} \phi \wedge \bigwedge_{i \in \mathcal{A}} \Box_i (\blacksquare^{k-1} \phi)$.

Intuitively, $\blacksquare^1 \phi$ means that every agent knows that ϕ holds, i.e., ϕ is the everyone knowledge; and $\blacksquare^k \phi$ means that ϕ is the depth k everyone knowledge.

The computation of the progression ξ of the KB ϕ w.r.t. an ontic action a_o involves three steps:

1. Construct the formula ψ by conjoining ϕ with the depth k everyone knowledge about the effect of a_o where k is the depth of ϕ (i.e., $\psi = \phi \wedge \blacksquare^k eff(a_o)$).
2. Obtain the formula η via forgetting the set of variables Q in ψ where Q is the set of unprimed variables appearing in $eff(a_o)$ (i.e., $\eta = \exists Q. \psi$).
3. Replace each occurrence of primed variables with their unprimed counterpart in η (i.e., $\xi = \eta[P'/P]$).

By expressing the initial KB and the action effects of ontic actions in SDNF $_{\mathcal{L}_0}$ with \mathcal{L}_0 satisfying polytime bounded conjunction ($\wedge BC$) and forgetting (FO), which is always possible due to Propositions 4.1, the above three steps can be accomplished in polytime, and thus the progression w.r.t. ontic actions can be tractably computed.

The progression of epistemic actions is relatively simple. An epistemic action a_e is associated with a triple of functions $\langle pre, pos, neg \rangle$ of \mathcal{L}_{\Box} -formulas, where pre, pos and neg indicate the precondition, positive and negative sensing results, respectively. For example, consider the action $sense(i, b_1, r_2)$ which means that agent i sense if box b_1 in room r_2 , its precondition, positive and negative sensing results are as follows: $pre(sense(i, b_1, r_2)) = \Box_i at(i, r_2)$, $pos(sense(i, b_1, r_2)) = \Box_i in(b_1, r_2)$ and $neg(sense(i, b_1, r_2)) = \Box_i \neg in(b_1, r_2)$. The computation of the progression is done via firstly making two copies of ϕ , and then conjoining them with positive and negative results respectively, i.e., $\phi \wedge pos(a_e)$ and $\phi \wedge neg(a_e)$. Again, SDNF is suitable to perform such progression since it supports polytime $\wedge BC$.

Continued with Example 3, we illustrate the computation of progression.

Example 4. *Suppose that agent i first moves right, and then senses whether b_1 in room r_2 .*

The progression of the initial KB $\phi = \Box_i at(i, r_1) \wedge \Box_j at(j, r_4)$ w.r.t. the ontic action $right(i)$ is obtained as follows:

1. Conjoin ϕ with $\blacksquare^1 eff(a_o)$, and the resulting formula ψ is $\Box_i at(i, r_1) \wedge \Box_j at(j, r_4) \wedge \blacksquare^1 [\bigwedge_{n=1}^3 (at'(i, r_{n+1}) \equiv at(i, r_n))]$;
2. The set of unprimed variables appearing in $eff(a_o)$ is $\{at(i, r_1), at(i, r_2), at(i, r_3)\}$. Forgetting them in ψ results in the formula $\eta = \Box_i at'(i, r_2) \wedge \Box_j at(j, r_4)$;
3. Substitute $at'(i, r_2)$ with $at(i, r_2)$ in η , leading to $\xi = \Box_i at(i, r_2) \wedge \Box_j at(j, r_4)$.

After agent i moves right, she knows that her position is r_2 .

Then, the progression w.r.t. $sense(i, b_1, r_2)$ splits ξ into the two KBs: $\Box_i [at(i, r_2) \wedge in(b_1, r_2)] \wedge \Box_j at(j, r_4)$ and $\Box_i [at(i, r_2) \wedge \neg in(b_1, r_2)] \wedge \Box_j at(j, r_4)$. These two formulas together means that agent i knows whether b_2 is in r_2 .

Besides progression, another major computation effort lies in the reasoning to decide if the current KB entails the goal formula and the action preconditions. It follows from Proposition 5.3 that the entailment check is tractable if the current KB is expressed in SDNF $_{\mathcal{L}_0}$ and both the goal formula and the preconditions are in SCNF $_{\mathcal{L}'_0}$, where \mathcal{L}_0 and \mathcal{L}'_0 are dual, and \mathcal{L}_0 satisfies CO and $\wedge BC$. Thanks to the tractability of both progression and entailment check, the whole planning process can be done effectively.

7 Extension to D_n , $K45_n$ and $KD45_n$

In the area of philosophy, it is ideal to assume that each agent does not know a contradiction or/and she has introspection about her own knowledge. The former assumption can be captured by the consistency axiom D ($\neg \Box_i \perp$) while the latter can be described as the introspection axioms 4 ($\Box_i \phi \rightarrow \Box_i \Box_i \phi$) and 5 ($\Diamond_i \phi \rightarrow \Box_i \Diamond_i \phi$).

\mathcal{L}	CO	VA	SE	EQ	CE	ACE $_{\mathcal{L}'_0}$	IM	AIM $_{\mathcal{L}'_0}$	$\neg C$	$\wedge C$	$\wedge BC$	$\vee C$	$\vee BC$	CD	FO	SFO
ASDNF $_{\mathcal{L}_0}$	✓*	○	○	○	○	✓*	○	○	×	×	✓*	✓	✓	✓*	✓*	✓*
ASCNF $_{\mathcal{L}_0}$	○	✓*	○	○	○	○	○	✓*	×	✓	✓	×	✓*	✓*	○	?
ACDNF	✓	○	○	○	○	✓*	○	○	×	×	✓	✓	✓	✓	✓	✓

Table 2: Queries and transformations for normal forms in $K45_n$ and $KD45_n$

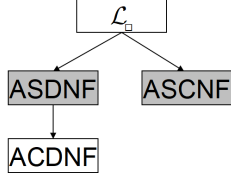


Figure 2: Succinctness of normal forms in $K45_n$ and $KD45_n$.

In this section, we are concerned about modal logics D_n , $K45_n$ and $KD45_n$ that are extensions to K_n . The logic D_n is K_n extended by the axiom **D**, $K45_n$ is the logic that extends K_n with axioms **4** and **5**, and $KD45_n$ is the modal logic containing the above three axioms.

Let us begin with the D_n case. Both SDNF and SCNF previously defined (cf. Def. 3.4) are suitable to D_n . This is because that the knowledge compilation results about these two forms (cf. Fig. 1 and Tab. 1) still hold for D_n , and thus SDNF and SCNF support as many queries and transformations as propositional DNF and CNF respectively.

Theorem 7.1. *The results in Fig. 1 and Tab. 1 hold for D_n .*

Proof. The proofs of results about K_n in Figure 1 and Table 1 are applicable to the logic D_n except the following statements:

1. If \mathcal{L}_0 satisfies **CO**, then SDNF $_{\mathcal{L}_0}$ satisfies **CO**.
2. If \mathcal{L}_0 satisfies **VA**, then SCNF $_{\mathcal{L}_0}$ satisfies **VA**.

Since SDNF and SCNF are dual, the above two statements are equivalent. It suffices to prove the first statement. The proof is similar to that of Proposition 5.1 except the case for a logically separable modal term $\alpha \wedge \bigwedge_{i \in B} (\Box_i \beta_i \wedge \bigwedge_j \Diamond_j \gamma_{ij})$. It is satisfiable iff non only the formulas α and γ_{ij} 's but also β_i 's are satisfiable. \square

Now, we turn to consider the logic $K45_n$. It is non-trivial to extend our proposed results for K_n to $K45_n$ since the definition of separability-based term cannot be applied in $K45_n$. We show this in an illustrative example.

Example 5. *Suppose that a logical separable modal term $\phi = \Diamond_i(p \wedge \Box_i \neg p)$. By Proposition 3.1, ϕ is satisfiable in K_n . However, it is not the case in $K45_n$ since ϕ implies that $\Diamond_i(p \wedge \neg p)$, which is equivalent to \perp . This is due to the additional axioms **4** and **5**.*

From the above example, we know that, in $K45_n$, there exist logical entanglements between two propositional subformulas on different depth of formulas. Thus, the crux is that some separability-based terms are logically inseparable in $K45_n$. To achieve logical separability, we need to prohibit any consecutive occurrence of modalities of the same agent.

Definition 7.1. A formula has *the alternating agent modality property* if no modalities of an agent directly occur inside those of the same agent.

We say ϕ is an *alternating separability-based term* (ASTE), if it is an STE with the alternating agent modality property. Similarly, we can define the following notions: *alternating separability-based clause* (ASCL), DNF (ASDNF) and CNF (ASCNF). In addition, the alternating version of CDNF (ACDNF) was proposed in (Hales, French, and Davies 2012). For example, the formula $\Box_i \Diamond_i p$ is not an ASTE since \Diamond_i occurs directly within the \Box_i modality. But the formula $\Box_i \Diamond_j \Box_i p$ is an ASTE since there is a \Diamond_j modality inbetween two \Box_i modalities.

We remark that all knowledge compilation results, stated in Sec. 4 and 5, also hold for ASDNF, ASCNF and ACDNF in the logic $K45_n$ except the following. Firstly, transforming into ASDNF or ASCNF causes an at most double exponential blowup in the size of the original formula since stripping out any occurrence of consecutive modalities with the same agent leads to an extra single-exponential blowup. Although the aforementioned transformation for arbitrary formulas may cause a double-exponential blowup, its complexity falls into single-exponential if the original formula possesses the alternating agent modality property. In addition, the definitions of polytime tests for restricted clausal entailment (ACE $_{\mathcal{L}_0}$) and implicant (AIM $_{\mathcal{L}_0}$) are slightly adjusted by using ASTE and ASCL instead of STE and SCL respectively. Similar to Prop. 5.3, if \mathcal{L}_0 and \mathcal{L}'_0 are dual and \mathcal{L}_0 supports **CO** and $\wedge BC$, then ASDNF $_{\mathcal{L}_0}$ satisfies ACE $_{\mathcal{L}'_0}$.

The above knowledge compilation results for $K45_n$ are also hold for $KD45_n$. We summarize them in Figure 2 and Table 2. The meaning of symbols in Figure 2 and Table 2 are the same as Figure 1 and Table 1 respectively.

Theorem 7.2. *The results in Figure 2 and Table 2 hold for $K45_n$ and $KD45_n$.*

To prove Theorem 7.2, we need the following lemma. We use \models_{\perp} to denote the entailment relation in the logic L .

Lemma 7.1. *For every alternating formula ϕ , ϕ is satisfiable in K_n iff it is satisfiable in $K45_n$.*

Proof. By Lemma 5 in (Lakemeyer and Lespérance 2012), we have $\perp \models_{K_n} \phi$ iff $\perp \models_{K45_n} \phi$. It follows that ϕ is satisfiable in K_n iff it is satisfiable in $K45_n$. \square

We are ready to present the proof of Theorem 7.2.

Proof. The formulas, introduced in Section 4 and used to prove the relative succinctness between two languages, are with depth 1, and alternating formulas. Hence, the succinctness results about K_n are applicable to $K45_n$ and $KD45_n$. For

a similar reason, the negative results on queries and transformations for K_n also hold for $K45_n$ and $KD45_n$.

The positive result on queries for $K45_n$ and $KD45_n$ can be proved using Lemma 7.1. By Proposition 5.1 and Lemma 7.1, $ASDNF_{\mathcal{L}_0}$ supports polytime satisfiability check (CO) if \mathcal{L}_0 supports CO. Similarly, $ASDNF_{\mathcal{L}_0}$ satisfies polytime test for restricted clausal entailment ($ACE_{\mathcal{L}_0}$) under specific conditions.

It is easy to see that SDNF supports polytime arbitrary disjunction ($\vee C$) and bounded conjunction ($\wedge BC$). Finally, the computation of forgetting and conditioning preserves the alternating agent modality property. Hence, $ASDNF_{\mathcal{L}_0}$ supports polytime forgetting (FO) and conditioning (CD).

Since SCNF is dual to SDNF, it is similar to obtain corresponding results for $SCNF_{\mathcal{L}_0}$. CDNF supports as many queries and transformations as SDNF on the ground that it is a fragment of $SDNF_{TE}$ and propositional term satisfies the specific conditions. \square

8 Conclusions and Future Work

We have introduced a notion of logical separability for modal terms, which is a key property to guarantee that the satisfiability check and forgetting can be computed in a modular way. Based on the logical separability, we have defined a normal form SDNF for the multi-agent modal logic K_n , which can be seen as a generalization of the well-known propositional normal form DNF. As a dual to SDNF, we can define the SCNF for K_n . More importantly, we have constructed a knowledge compilation map on four normal forms SDNF, SCNF, CDNF and PINF in terms of their succinctness, queries and transformations. Interestingly, bounded conjunction, forgetting and restricted clausal entailment check are all tractable for $SDNF_{\mathcal{L}_0}$ formulas under some restrictions on \mathcal{L}_0 . These three properties are crucial to effective implementations of multi-agent epistemic planning. Although SDNF and CDNF admit tractability for the same queries and transformations, the former is a better choice of the target compilation language for multi-agent epistemic planning since SDNF is more succinct and flexible than CDNF. Finally, we extend the above results to modal logics D_n , $K45_n$ and $KD45_n$.

For future work, we plan to implement a multi-agent epistemic planner based on SDNF. It is also interesting to identify tractable normal forms in other multi-agent modal logics (e.g., T_n and $S5_n$) and expressive description logics (e.g., $ALCO$ and $ALCOT$). Since the description logic ALC is highly-related to K_n , the results proposed in this paper is also applicable to ALC .

9 Acknowledgments

We are grateful to Yongmei Liu, Dacheng Wen and Fang Lv for their helpful discussions on the paper. This work was supported by the Natural Science Foundation of China (Nos. 61603152, 61463044, 61703182, 61772232 and 61773179) and Guangxi Key Laboratory of Trusted Software project (Nos. kx201604 and kx201606). Liangda Fang and Ximing Wen are also affiliated to Guangxi Key Laboratory of

Trusted Software, Guilin University of Electronic Technology, Guilin, China. This work was partly supported by Australian Research Council (ARC) under DP130102302.

References

- Aucher, G. 2011. DEL-sequents for progression. *Journal of Applied Non-Classical Logics* 21(3-4):289–321.
- Bienvenu, M.; Fargier, H.; and Marquis, P. 2010. Knowledge Compilation in the Modal Logic S5. In *AAAI*, 261–266.
- Bienvenu, M. 2008. Prime Implicate Normal Form for ALC Concepts. In *AAAI*, 412–417.
- Bienvenu, M. 2009. *Consequence Finding in Modal Logic*. Ph.D. Dissertation, Université de Toulouse.
- Bryce, D.; Kambhampati, S.; and Smith, D. E. 2006. Planning Graph Heuristics for Belief Space Search. *Journal of Artificial Intelligence Research* 26:35–99.
- Darwiche, A., and Marquis, P. 2002. A Knowledge Compilation Map. *Journal of Artificial Intelligence Research* 17:229–264.
- Fang, L.; Liu, Y.; and van Ditmarsch, H. 2016. Forgetting in Multi-Agent Modal Logics. In *IJCAI*, 1066–1073.
- French, T. N. 2006. *Bisimulation Quantifiers for Modal Logics*. Ph.D. Dissertation, University of Western Australia.
- Hales, J.; French, T.; and Davies, R. 2012. Refinement Quantified Logics of Knowledge and Belief for Multiple Agents. In *AiML*, 317–338.
- Halpern, J. Y., and Moses, Y. 1992. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence* 54:319–379.
- Huang, X.; Fang, B.; Wan, H.; and Liu, Y. 2017. A General Multi-agent Epistemic Planner Based on Higher-order Belief Change. In *IJCAI*, 3327–3334.
- Kominis, F., and Geffner, H. 2015. Beliefs in Multiagent Planning: From One Agent to Many. In *ICAPS*, 147–155.
- Lakemeyer, G., and Lespérance, Y. 2012. Efficient Reasoning in Multiagent Epistemic Logics. In *ECAI*, 498–503.
- Levesque, H. 1998. A Completeness Result for Reasoning with Incomplete First-Order Knowledge Bases. In *KR*, 14–23.
- Moss, L. S. 2007. Finite models constructed from canonical formulas. *Journal of Philosophical Logic* 36(6):605–640.
- ten Cate, B.; Conradie, W.; Marx, M.; and Venema, Y. 2006. Definitorially Complete Description Logics. In *KR*, 79–89.
- To, S. T.; Pontelli, E.; and Son, T. C. 2011. On the Effectiveness of CNF and DNF Representations in Contingent Planning. In *IJCAI*, 2033–2038.
- To, S. T.; Son, T. C.; and Pontelli, E. 2011. Conjunctive Representations in Contingent Planning: Prime Implicates versus Minimal CNF Formula. In *AAAI*, 1023–1028.