Multi-agent Epistemic Planning with Common Knowledge

Qiang Liu and **Yongmei Liu**
Dept. of Computer Science, Sun Yat-sen University, Guangzhou 510006, China
liuq226@mail2.sysu.edu.cn, ymliu@mail.sysu.edu.cn

Abstract

In the past decade, multi-agent epistemic planning has received much attention from both dynamic logic and planning communities. Common knowledge is an essential part of multi-agent modal logics, and plays an important role in coordination and interaction of multiple agents. However, existing implementations of multi-agent epistemic planning provide very limited support for common knowledge, basically static propositional common knowledge. Our work aims to extend an existing multi-agent epistemic planning framework based on higher-order belief change with the capability to deal with common knowledge. We propose a novel normal form for multi-agent KD45 logic with common knowledge. We propose satisfiability solving, revision and update algorithms for this normal form. Based on our algorithms, we implemented a multi-agent epistemic planner with common knowledge called MEPC. Our planner successfully generated solutions for several domains that demonstrate the typical usage of common knowledge.

1 Introduction

Reasoning about knowledge and beliefs and their change plays an important role in many intelligent tasks, since actions may have preconditions involving agents' knowledge and beliefs, which may be changed by agents' actions. In some applications, even higher-order knowledge and beliefs, i.e., knowledge and beliefs about other agents' knowledge and beliefs, turn out to be insufficient, and common knowledge is needed. We say that ϕ is common knowledge of a group of agents if everybody knows ϕ, everybody knows everybody knows ϕ, and so on to infinity. For example, common knowledge is needed for agreement and coordination. To illustrate, suppose that Alice and Bob are trying to coordinate their actions. This involves the agents' agreeing on when to perform the actions, which we represent by ϕ. We expect that if Alice and Bob agree on something, then each of them knows that they have agreed on that. By induction, Alice and Bob have common knowledge of ϕ. The common knowledge modality adds a great deal of expressive power to multi-agent modal logics. As a result, deciding satisfiability becomes EXPTIME-complete [Halpern and Moses, 1992].

In the past decade, multi-agent epistemic planning has received much attention from both dynamic logic and planning communities. On the theory side, Bolander and Andersen [2011] formalized multi-agent epistemic planning (MEP) based on dynamic epistemic logic [Van Ditmarsch *et al.*, 2007], where both states and actions are represented as Kripke models. Very recently, Huang *et al.* [2017] proposed a general representation framework for MEP, where the initial knowledge base (KB) and the goal, the preconditions and effects of actions can be arbitrary multi-agent epistemic formulas, progression of KBs wrt actions is achieved through higher-order belief revision or update based on effects of actions, and the solution is an action tree branching on sensing results. On the implementation side, Kominis and Geffner [2015] and Muise *et al.* [2015] solved restricted versions of MEP problems by compiling them into classical planning, and Huang *et al.* [2017] implemented a contingent MEP planner based on AND/OR forward search. However, the implementation of [Muise *et al.*, 2015] does not support common knowledge, and those of [Kominis and Geffner, 2015] and [Huang *et al.*, 2017] provide very limited support for common knowledge, basically static propositional common knowledge. To the best of our knowledge, LoTREC [del Cerro *et al.*, 2001], a theorem prover for S5, is the only implementation capable of reasoning with common knowledge.

In this paper, we extend the MEP framework proposed by Huang *et al.* to incorporate general common knowledge. To support efficient reasoning in multi-agent KD45 logic, they made use of a normal form called alternating cover disjunctive formulas [Hales *et al.*, 2012]. However, this normal form cannot be directly generalized to support reasoning with common knowledge. Thus we propose a novel normal form for multi-agent KD45 with common knowledge. This normal form makes use of a new common knowledge modality $C_{\underline{a}}\phi$ which means that it is common knowledge that everybody except agent a knows ϕ. We propose a novel algorithm for checking satisfiability for this normal form. Also, we propose revision and update algorithms for this normal form. The essential idea is to change agents' common knowledge before changing agents' knowledge, and carry the changed common knowledge to change knowledge to ensure the consistency between knowledge and common knowledge. Based on our reasoning, revision, and update algorithms, we implemented a multi-agent epistemic planner called MEPC. Our planner successfully generated solutions for several domains that demonstrate the typical usage of common knowledge.

2 Preliminaries

In this section, we introduce the background work of our paper, i.e., the multi-agent modal logic and the epistemic planning modeling framework proposed by Huang *et al.*

2.1 Multi-agent Modal Logic

We fix a finite set of atoms $\mathcal{P}$ and a finite set of agents $\mathcal{A}$.

Definition 2.1. The language $\mathcal{L}_{KC}$ of multi-agent modal logic with common knowledge is generated by the BNF:

$$\phi \ ::= \ p \mid \neg\phi_1 \mid \phi_1 \wedge \phi_2 \mid K_a\phi_1 \mid C\phi_1,$$

where $p \in \mathcal{P}$, $a \in \mathcal{A}$, $\phi_1, \phi_2 \in \mathcal{L}_{KC}$.

Intuitively, $K_a\phi$ means agent a knows ϕ, and $C\phi$ means all agents commonly know ϕ. We let $L_a\phi$ and $D\phi$ abbreviate for $\neg K_a\neg\phi$ and $\neg C\neg\phi$, respectively. Intuitively, $L_a\phi$ means agent a thinks ϕ is possible, and $D\phi$ means all agents commonly think ϕ is possible.

We use ϕ and ψ to represent formulas, Φ and Ψ to represent sets of formulas, $\top$ and $\bot$ to denote $true$ and $false$, respectively. $\bigvee\Phi$ stands for the disjunction of members in Φ, while $L_a\Phi$ (resp. $D\Phi$) stands for the conjunction of $L_a\phi$ (resp. $D\phi$) where $\phi \in \Phi$. The modal depth of a formula $\phi \in \mathcal{L}_{KC}$ is the depth of nesting of modalities in ϕ.

Definition 2.2. A frame is a pair (W, R), where W is a nonempty set of possible worlds; for each agent $a \in \mathcal{A}$, R_a is a binary relation on W, called the accessibility relation for a.

When wR_aw', we say w' is an a-child of w. We say R_a is serial if for any $w \in W$, there is a $w' \in W$ s.t. wR_aw'; we say R_a is transitive if wR_au and uR_av imply wR_av; we say R_a is Euclidean if wR_au and wR_av imply uR_av. A $\mathrm{KD45}_n$ frame is a frame whose accessibility relations are serial, transitive and Euclidean.

Definition 2.3. A Kripke model is a triple $M = (W, R, V)$, where (W, R) is a frame, and $V : W \to 2^{\mathcal{P}}$ is a valuation map. A pointed Kripke model is a pair $s = (M, w)$, where M is a Kripke model and w is a world of M.

Definition 2.4. Let $s = (M, w)$ be a Kripke model where $M = (W, R, V)$. We interpret formulas in $\mathcal{L}_{KC}$ inductively:

- $M, w \models p$ iff $p \in V(w)$;
- $M, w \models \neg\phi$ iff $M, w \not\models \phi$;
- $M, w \models \phi \wedge \psi$ iff $M, w \models \phi$ and $M, w \models \psi$;
- $M, w \models K_a\phi$ iff for all v s.t. wR_av, $M, v \models \phi$;
- $M, w \models C\phi$ iff for all v s.t. $wR_{\mathcal{A}}v$, $M, v \models \phi$, where $R_{\mathcal{A}}$ is the transitive closure of the union of R_a for $a \in \mathcal{A}$.

Consider the context of K_n^C or $\mathrm{KD45}_n^C$. We say ϕ is satisfiable if there is a Kripke model (M, w) s.t. $M, w \models \phi$. We say ϕ entails ψ, written $\phi \models \psi$, if for any Kripke model (M, w), $M, w \models \phi$ entails $M, w \models \psi$. We say ϕ and ψ are equivalent, written $\phi \Leftrightarrow \psi$, if $\phi \models \psi$ and $\psi \models \phi$.

Theorem 2.5. [Halpern and Moses, 1992] The satisfiability problems for K_n^C and $\mathrm{KD45}_n^C$ are EXPTIME-complete.

Halpern and Moses presented an algorithm for checking satisfiability in K_n^C. Given a formula $\phi \in \mathcal{L}_{KC}$, their algorithm first generates a set S consisting of all subformulas of ϕ and their negations. Then it computes a set W consisting of all subsets A of S that are propositionally consistent and maximal, *i.e.*, either $\psi \in A$ or $\neg\psi \in A$ for each $\psi \in S$. Finally it iteratively builds a model for ϕ with worlds in W.

We now define a normal form for K_n^C.

Definition 2.6. The set of modal terms is inductively defined:

- A propositional term, i.e., a conjunction of propositional literals, is a modal term;
- A formula of the form $\phi_0 \wedge \bigwedge_{a\in\mathcal{A}}(K_a\phi_a \wedge L_a\Psi_a) \wedge C\mu \wedge D\Lambda$ is a modal term, where ϕ_0 is a propositional term, Ψ_a, Λ are sets of modal terms, and ϕ_a, μ are conjunctions of disjunctions of modal terms.

A formula ϕ is in DNF if it's a disjunction of modal terms. A formula ϕ is in CDNF if it's a conjunction of DNFs.

So the knowledge and common knowledge parts of a modal term are in CDNF.

Proposition 2.7. In K_n^C, any formula in $\mathcal{L}_{KC}$ can be transformed to an equivalent DNF whose length is at most singly exponential in the length of the original formula.

Proof. For an arbitrary modal formula $\phi \in \mathcal{L}_{KC}$, we first put it into negation normal form. Then we treat modal atoms, ie, formulas of the form $K\psi$, $L\psi$, $C\psi$ and $D\psi$, as propositional atoms and transform the whole formula into propositional DNF. Then for the ψ in each modal atom, we repeat this process. By induction on the modal depth of ϕ, we can show that the length of the resulting DNF is at most singly exponential in the length of ϕ. □

In the rest of the paper, the logic we use is $\mathrm{KD45}_n^C$, except that in Section 3.1, we discuss satisfiability for K_n^C.

2.2 Epistemic Planning Modeling Framework

We follow the modeling framework for multi-agent epistemic planning (MEP) in [Huang *et al.*, 2017], and extend it with the support of general common knowledge.

We illustrate the framework with the classic muddy children example [Fagin *et al.*, 1995]. There are n children playing together and m of them get mud on their foreheads. Each can see the mud on others but not on his own forehead. The father announces "At least one of you has mud on your forehead". The father then keeps asking "Does any of you know whether you have mud on your own forehead?" It turns out the first $m-1$ times he asks the question, they will all say "No", but then the mth time the children with muddy foreheads will all answer "Yes".

Definition 2.8. A MEP problem is a tuple $\langle\mathcal{A}, \mathcal{P}, \mathcal{D}, \mathcal{S}, \mathcal{I}, \mathcal{G}\rangle$, where $\mathcal{A}$ is a set of agents, $\mathcal{P}$ is a set of atoms, $\mathcal{D}$ is a set of deterministic actions, $\mathcal{S}$ is a set of sensing actions, $\mathcal{I} \in \mathcal{L}_{KC}$ is the initial KB, and $\mathcal{G} \in \mathcal{L}_{KC}$ is the goal.

Example 1. Assume that there are three children ($\mathcal{A} = \{a, b, c\}$) and all of them are muddy. The atoms are: m_i, indicating the ith child is muddy. The deterministic actions are: $announce$, the father announces that at least one of them is muddy; $askno$, the father asks whether anyone knows that he is muddy, but no child answers "Yes". The initial KB is $m_a \wedge m_b \wedge m_c \wedge \bigwedge_{i \in \mathcal{A}}(K_i \bigwedge_{j \neq i} m_j)$. The goal is $\bigwedge_{i \in \mathcal{A}}(m_i \to K_i m_i)$.

Definition 2.9. A deterministic action is a pair $\langle pre, eff \rangle$, where $pre \in \mathcal{L}_{KC}$ is the precondition; eff is a set of conditional effects, each of which is a pair $\langle con, cef \rangle$, where con, $cef \in \mathcal{L}_{KC}$ indicate the condition and the effect, respectively.

Definition 2.10. A sensing action is a triple $\langle pre, pos, neg \rangle$, where pre, pos, $neg \in \mathcal{L}_{KC}$ indicate the precondition, the positive result, and the negative result, respectively.

For example,
$announce = \langle \top, \{eff\} \rangle$, where $eff = \langle \top, C \bigvee_i m_i \rangle$.
$askno = \langle pre, \{eff_1, eff_2\} \rangle$, where:

- $pre = \bigwedge_i (\neg K_i m_i \wedge \neg K_i \neg m_i)$,
- $eff_1 = \langle C \bigvee_i m_i, C \bigvee_{i \neq j}(m_i \wedge m_j) \rangle$,
- $eff_2 = \langle C \bigvee_{i \neq j}(m_i \wedge m_j), C \bigwedge_i m_i \rangle$.

Here pre says that no child knows if he is muddy, eff_1 says: under the condition that the children commonly know that at least one child is muddy, they commonly know at least two of them are muddy, and eff_2 says: under the condition that the children commonly know that at least two of them are muddy, they commonly know that all three children are muddy.

An action a is executable wrt a KB $\phi \in \mathcal{L}_{KC}$ if $\phi \models pre(a)$. Suppose a is executable wrt ϕ. The progression of ϕ wrt a is defined by resorting to belief change operators. Two main types of belief change are revision and update: revision concerns belief change about static environments due to partial and possibly incorrect information, whereas update concerns belief change about dynamic environments due to the performance of actions. We use a revision operator $\circ$ and an update operator $\diamond$ for $\mathcal{L}_{KC}$. We use update for progression wrt deterministic actions and revision for sensing actions.

Definition 2.11. Let $\phi \in \mathcal{L}_{KC}$ and a a deterministic action where $eff(a) = \{\langle \phi_1, \psi_1 \rangle, \ldots, \langle \phi_n, \psi_n \rangle\}$. Suppose $\phi_{i_1}, \ldots, \phi_{i_m}$ are all the ϕ_i's s.t. $\phi \models \phi_i$. Then the progression of ϕ wrt a is defined as $((\phi \diamond \psi_{i_1}) \ldots) \diamond \psi_{i_m}$.

Definition 2.12. Let $\phi \in \mathcal{L}_{KC}$ and a a sensing action. Then the progression of ϕ wrt a with positive (resp. negative) result is $\phi^+ = \phi \circ pos(a)$ (resp. $\phi^- = \phi \circ neg(a)$).

The progression of ϕ wrt a sequence of actions (with sensing results for sensing actions) is inductively defined as follows: $prog(\phi, \epsilon) = \phi$; $prog(\phi, (a; \sigma)) = prog(prog(\phi, a), \sigma)$ if $\phi \models pre(a)$, and undefined otherwise.

A solution of a MEP problem is an action tree branching on sensing results, such that the progression of the initial KB wrt each branch in the tree entails the goal.

3 Checking Satisfiability

In this section, we first present an algorithm for checking satisfiability in K_n^C, and then extend it to KD45_n^C.

3.1 K_n^C Satisfiability

The main idea of our algorithm for checking satisfiability in K_n^C is as follows. The input formula is in DNF, and we attempt to build a model for it recursively. When checking a model term $\delta = \phi_0 \wedge \bigwedge_{a \in \mathcal{A}}(K_a \phi_a \wedge L_a \Psi_a) \wedge C\mu \wedge D\Lambda$, if ϕ_0 is unsatisfiable, we immediately return $\bot$, otherwise we proceed as follows:

- For each agent a and each possibility $\psi_a \in \Psi_a$, we conjoin it with knowledge ϕ_a and common knowledge $\mu \wedge C\mu$, and check if the resulting formula is satisfiable.
- If not, we immediately return $\bot$, otherwise we check $\lambda \wedge \phi_a \wedge \mu \wedge C\mu$ and $D\lambda \wedge \phi_a \wedge \mu \wedge C\mu$ for each agent a and each common possibility $\lambda \in \Lambda$.

Each of the above formulas is in CDNF, we call it a child of δ. When checking a CDNF ϕ, we put it into a DNF $\bigvee \Delta$, and recursively check each $\delta' \in \Delta$. We call δ' a derivant of ϕ.

However, the complication is that due to the presence of common knowledge, some newly generated modal terms might be the same as previous ones. To handle this issue, we maintain a graph whose nodes are modal terms. When a new modal term is generated, we mark it by $\top$, and when we know it is unsatisfiable, mark it by $\bot$. Also, when a new modal term δ' is derived from a child of a modal term δ, we add an edge from δ to δ', mark the edge by 1 if δ' is derived from $\psi_a \wedge \mu \wedge C\mu$, and 2 otherwise. Now, when checking a modal term δ which already exists, we return its current mark.

A formula δ may not be satisfiable if it's marked $\top$ by the above methods. We further update each modal term's mark by checking whether its common possibilities are satisfied. For each conjunct $D\lambda$ in each modal term δ, we check whether there is a modal term δ' reachable from δ s.t. λ is a conjunct of δ' and δ' is marked by $\top$. If not, we mark δ by $\bot$ and further propagate $\bot$ to its ancestors through 1-edges. We repeat this procedure until no mark is updated.

Definition 3.1. Let δ be a modal term. For $a \in \mathcal{A}$, let γ_a denote a's knowledge, *i.e.*, $\gamma_a = \phi_a \wedge \mu \wedge C\mu$. We define the set of δ's children, $Gen^K(\delta)$, as the union of the following:

- $S_1 = \{\gamma_a \wedge \psi_a \mid a \in \mathcal{A}, \psi_a \in \Psi_a\}$;
- $S_{\lambda_1} = \{\gamma_a \wedge \lambda \mid a \in \mathcal{A}\}$, and
  $S_{\lambda_2} = \{\gamma_a \wedge D\lambda \mid a \in \mathcal{A}\}$, where $\lambda \in \Lambda$.

Proposition 3.2. Let δ be a satisfiable modal term. Then the following hold for $Gen^K(\delta)$:

- for all $\phi \in S_1$, ϕ is satisfiable;
- for all $\lambda \in \Lambda$, there is $\phi \in S_{\lambda_1} \cup S_{\lambda_2}$ s.t. ϕ is satisfiable.

Algorithm 1: $Check^K(\phi)$

input: $\phi \in \mathcal{L}_{KC}$ is in DNF $\bigvee \Delta$ **output**: $\top$ / $\bot$

set $G = (W, R)$ to the empty graph;
if $Sat^K(\phi, \bot, a, 1) = \bot$ **then** return $\bot$;
call $Update$;
foreach δ in Δ **do**
  if $Mark(\delta) = \top$ **then** return $\top$;
return $\bot$;

Function $Sat^K(\phi, \phi', i, n)$

input: ϕ is in CDNF, ϕ' is a modal term, $i \in \mathcal{A}$, $n = 1, 2$

if ϕ *is not a modal term* **then**
  transform ϕ into DNF $\bigvee \Delta$;
  foreach δ in Δ **do**
    if $Sat^K(\delta, \phi', i, n) = \top$ **then** return $\top$;
  return $\bot$;
else
  if $\phi' \neq \bot$ **then** $R_i \leftarrow R_i \cup \{(\phi', \phi)_n\}$;
  if $\phi \in W$ **then** return $Mark(\phi)$;
  $W \leftarrow W \cup \{\phi\}$, $Mark(\phi) \leftarrow \top$;
  $Gen^K(\phi) = S_1 \cup \bigcup_{\lambda \in \Lambda}(S_{\lambda_1} \cup S_{\lambda_2})$;
  if *one of the following conditions holds*:
  1. ϕ_0 *is propositionally unsatisfiable*;
  2. $\exists \delta_a \in S_1$: $Sat^K(\delta_a, \phi, a, 1) = \bot$.
  then $Mark(\phi) \leftarrow \bot$, return $\bot$;
  foreach δ_a in $\bigcup_{\lambda \in \Lambda} S_{\lambda_1} \cup S_{\lambda_2}$ **do**
    call $Sat^K(\delta_a, \phi, a, 2)$;
  return $\top$;

Procedure $Update$

foreach δ in W and λ s.t. $D\lambda$ is a conjunct of δ **do**
  let $Child(\delta, \lambda)$ be a set of modal terms δ' where:
    1. δ' *is reachable from* δ; *2.* λ *is a conjunct of* δ';
    3. $Mark(\delta') = \top$.
$Updated \leftarrow \top$;
while $Updated = \top$ **do**
  $Updated \leftarrow \bot; UpdatedNodes \leftarrow \emptyset$;
  foreach δ in W and λ s.t. $D\lambda$ is a conjunct of δ **do**
    if $Mark(\delta) = \top$ *and* $Child(\delta, \lambda) = \emptyset$ **then**
      $Mark(\delta) \leftarrow \bot; Updated \leftarrow \top$;
      $UpdatedNodes \leftarrow UpdatedNodes \cup \{\delta\}$;
  foreach δ in $UpdatedNodes$ **do**
    call $Propagate(\delta)$;
  foreach δ in W and λ **do**
    delete δ' in $Child(\delta, \lambda)$ s.t. $Mark(\delta') = \bot$;

We demonstrate the procedure of $Check^K(\phi)$ by Figure 1. Let $\phi = Cp \wedge D\neg p$ and $\mathcal{A} = \{a\}$.

1. We calculate $Gen^K(\phi) = \{\delta_1, \delta_2\}$, where $\delta_1 = \neg p \wedge p \wedge Cp$ and $\delta_2 = D\neg p \wedge p \wedge Cp$, and set $W = \{\phi, \delta_1, \delta_2\}$ and $R_a = \{(\phi, \delta_1)_2, (\phi, \delta_2)_2\}$.
2. We check δ_1 next. Since its propositional part is unsatisfiable, we set $Mark(\delta_1) = \bot$.
3. We turn to δ_2, $Gen^K(\delta_2) = \{\delta_1, \delta_2\}$, $R_a \leftarrow R_a \cup \{(\delta_2, \delta_1)_2, (\delta_2, \delta_2)_2\}$, $Mark(\delta_2) = \top$. Thus $Mark(\phi) = \top$.
4. Since $D\neg p$ is a conjunct of ϕ and δ_2 but there isn't a reachable modal term δ' s.t. $\neg p$ is a conjunct of δ' and $Mark(\delta') = \top$, we set $Mark(\phi) = Mark(\delta_2) = \bot$. Eventually $Check^K(\phi)$ returns $\bot$.

Theorem 3.3. The complexity of $Check^K(\phi)$ is $O(4^{(d+1)cl+l})$, where l is the length of ϕ, d is the modal depth of ϕ, and c is the depth of nesting of C modalities in ϕ.

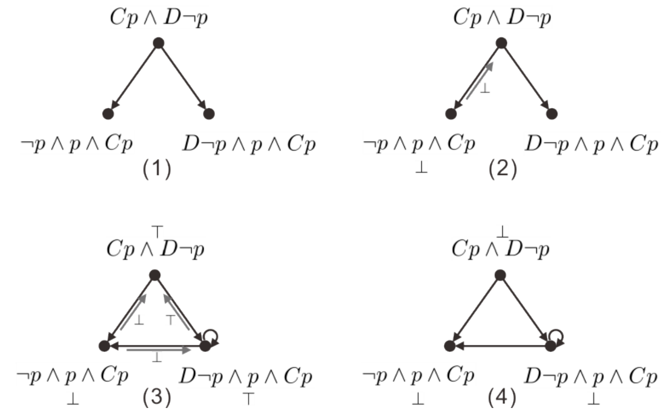

Figure 1: Procedure of $Check^K(Cp \wedge D\neg p)$

Procedure $Propagate(\delta)$

input: δ is a modal term

foreach δ' in W s.t. $(\delta', \delta)_1 \in R$ **do**
  if $Mark(\delta') = \top$ **then**
    $Mark(\delta') \leftarrow \bot$; call $Propagate(\delta')$;

Proof sketch. First consider ϕ without C. For any modal term δ generated during the algorithm, δ corresponds to a substring of ϕ. Hence the graph has at most 2^l nodes. Now consider that ϕ contains C modalities. Let $C\nu$ be the conjunction of all subformulas $C\mu$ in ϕ. The maximal length of $C\nu$ is cl. The children generated from $C\nu$ are in the form $\nu_1 \wedge C\nu$, where ν_1 doesn't contain common knowledge since $C\nu$ is maximal. For $1 \leq i < d$, we name the children generated from ν_i as ν_{i+1}. All ν_i corresponds to a substring of ν, thus the length of ν_i is less than cl. Since the children of $C\nu$ without common knowledge will be reduced to propositional formulas after at most $(d-1)$ recursions, the maximal length of the form $\bigwedge_{1 \leq i \leq d} \nu_i \wedge C\nu$ in a modal term generated during the algorithm is $cl + dcl$. Hence the maximum number of nodes is $2^{(d+1)cl+l}$. Let n be the number of nodes. Cycle checking and updating marks requires $O(n^2)$ time. Thus we obtain the complexity result $O(4^{(d+1)cl+l})$. $\square$

We proceed to prove soundness and completeness of $Check^K$. For soundness, when $Check^K(\phi)$ returns $\top$, we build a model M_ϕ for ϕ by modifying the frame $G = (W, R)$ as follows: remove $\delta \in W$ s.t. $Mark(\delta) = \bot$; rename $\delta \in W$ to w_δ; and create a valuation map V s.t. $V(w_\delta)$ satisfies ϕ_0 in δ. We show that for each w_δ in M_ϕ, $M_\phi, w_\delta \models \delta$. To facilitate the proof, we introduce the notion of implicants.

Definition 3.4. If the conjunction of a CDNF ϕ and another CDNF ϕ' is equivalent to a DNF $\bigvee \Delta$, we call each $\delta \in \Delta$ an implicant of ϕ, and write $\delta \models\!\!\models \phi$.

Obviously, if $\delta \models\!\!\models \phi$, then $\delta \models \phi$. If a modal term δ' is a subterm of a modal term δ, *i.e.*, the conjuncts of δ' is a subset of those of δ, then δ is an implicant of δ'. In Algorithm 1, when a modal term δ' is added as an a-child of a modal term $\delta = \phi_0 \wedge \bigwedge_{a \in \mathcal{A}}(K_a\phi_a \wedge L_a\Psi_a) \wedge C\mu \wedge D\Lambda$, δ' is an implicant of ϕ_a, μ and $C\mu$; similarly, δ' is an implicant of some $\psi_a \in \Psi_a$, or an implicant of λ or $D\lambda$ for some $\lambda \in \Lambda$.

Lemma 3.5. When $Check^K(\phi)$ returns $\top$, for all $w_\delta \in M_\phi$, we have $M_\phi, w_\delta \models \psi$ if $\delta \models \psi$.

Proof. We prove by structural induction on ψ. Consider ψ in the following forms:

- A propositional literal. Then ψ is a conjunct of the propositional term of δ. By the definition of $V(w_\delta)$, we have $M_\phi, w_\delta \models \psi$.
- $\psi_1 \wedge \psi_2$. Then $\delta \models \psi_1$ and $\delta \models \psi_2$. By induction, M_ϕ, $w_\delta \models \psi_1$ and $M_\phi, w_\delta \models \psi_2$. So $M_\phi, w_\delta \models \psi_1 \wedge \psi_2$.
- $\psi_1 \vee \psi_2$. Then $\delta \models \psi_1$ or $\delta \models \psi_2$. By induction, $M_\phi, w_\delta \models \psi_1$ or $M_\phi, w_\delta \models \psi_2$. So $M_\phi, w_\delta \models \psi_1 \vee \psi_2$.
- $K_a\psi'$. By Algorithm 1, for all $w_{\delta'}$ s.t. $w_\delta R_a w_{\delta'}$, $\delta' \models \psi'$. By induction, $M_\phi, w_{\delta'} \models \psi'$. So $M_\phi, w_\delta \models K_a\psi'$.
- $L_a\psi'$. By Algorithm 1, there is $w_{\delta'}$ s.t. $w_\delta R_a w_{\delta'}$ and $\delta' \models \psi'$. If not, δ will be marked $\perp$ by $Propagate$. By induction, $M_\phi, w_{\delta'} \models \psi'$. So $M_\phi, w_\delta \models L_a\psi'$.
- $C\psi'$. By Algorithm 1, for all $w_{\delta'}$ reachable from w_δ, $\delta' \models \psi'$. By induction, $M_\phi, w_{\delta'} \models \psi'$. So $M_\phi, w_\delta \models C\psi'$.
- $D\psi'$. By Algorithm 1, there is $w_{\delta'}$ reachable from w_δ s.t. $\delta' \models \psi'$. If not, δ will be marked $\perp$ by $Update$. By induction, $M_\phi, w_{\delta'} \models \psi'$. So $M_\phi, w_\delta \models D\psi'$. □

For completeness, we prove a modal term δ is unsatisfiable if it's marked $\perp$. The difficult case is when δ is marked $\perp$ during update. We introduce a lemma to handle this case.

Lemma 3.6. Let δ be a modal term s.t. $\delta \models D\lambda$. When δ is satisfiable, there is a modal term δ' reachable from δ s.t. $\delta' \models \lambda$ and δ' is satisfiable.

Proof. Since δ is satisfiable, there is a model (M, w) s.t. $M, w \models \delta$. Since $\delta \models D\lambda$, there is a path $w, w_1, \ldots, w_m, w'$ in M s.t. $M, w' \models \lambda$. Assume that $wR_{a_1}w_1$, $w_mR_{a'}w'$ and $w_iR_{a_{i+1}}w_{i+1}$ for $1 \leq i < m$. By Algorithm 1, there is a path $\delta, \delta_1, \ldots, \delta_m, \delta'$ in W where $\delta_i \models D\lambda$ for each i and $\delta' \models \lambda$. Moreover, we have $(\delta, \delta_1) \in R_{a_1}$, $(\delta_m, \delta') \in R_{a'}$ and $(\delta_i, \delta_{i+1}) \in R_{a_{i+1}}$. By induction, we can show that for each $i = 1, \ldots, m$, $M, w_i \models \delta_i$. So $M, w' \models \delta'$. □

Theorem 3.7. $Check^K(\phi) = \top$ iff ϕ is satisfiable in K_n^C.

Proof. $\Rightarrow$: Suppose $Check^K(\phi)$ returns $\top$. Then there is a world $w_\delta \in M_\phi$ s.t. $\delta \models \phi$. By Lemma 3.5, $M_\phi, w_\delta \models \phi$. Thus ϕ is satisfiable.

$\Leftarrow$: Suppose $Check^K(\phi)$ returns $\perp$. Let $\phi = \bigvee \Delta$. Then each modal term in Δ is marked $\perp$. We prove by induction on k that if δ is the kth modal term that $Check^K$ marks $\perp$, δ is unsatisfiable. There are 3 cases:

- The propositional term of δ is unsatisfiable.
- There is $\delta' \in S_1$ marked by $\perp$. By induction, δ' is unsatisfiable. By Proposition 3.2, δ is unsatisfiable.
- δ contains a conjunct $D\lambda$ and for each δ' reachable from δ where $\delta' \models \lambda$, δ' is marked $\perp$. By induction, δ' is unsatisfiable. By Lemma 3.6, δ is unsatisfiable. □

3.2 KD45$_n^C$ Satisfiability

The main idea of our algorithm for checking satisfiability in $\mathrm{KD45}_n^C$ is as follows. We require the input formula ϕ to be in a certain normal form. We apply to ϕ the algorithm for checking satisfiability in K_n^C. When the algorithm returns $\top$ and a model for ϕ, we add edges to the model to make it serial, transitive, and Euclidean. The normal form we use ensures that after adding edges, the model still satisfies ϕ. To motivate our normal form, let's consider two examples.

Example 2. Let $\mathcal{A} = \{a, b\}$, $\mathcal{P} = \{p\}$, and $\phi = DK_ap$. Consider $M = (W, R, V)$ where $W = \{0, 1, 2\}$, $R_a = \{(0, 1), (1, 2)\}$, $R_b = \emptyset$, $V(0) = V(1) = \emptyset$, and $V(2) = \{p\}$. Then $(M, 0)$ is a model for ϕ in K_n^C, but it can't be made a model for ϕ in $\mathrm{KD45}_n^C$ by adding edges. Now suppose we have an modality $D_{\underline{a}}\phi$ meaning ϕ holds in some world reachable by a path whose last edge is not of agent a. Let M' be the same as M except $R_a = \{(1, 2)\}$ and $R_b = \{(0, 1)\}$. Then $(M', 0)$ is a model for $\phi' = D_{\underline{a}}K_ap$ in K_n^C, and it can be made a model for ϕ' in $\mathrm{KD45}_n^C$ by adding edges.

Example 3. Let $\phi = L_a(p \wedge K_a\neg p)$. Then ϕ is satisfiable in K_n^C but not in $\mathrm{KD45}_n^C$, thus K_n^C algorithm obtains the wrong answer. Note that ϕ entails $K_a\neg p$ in $\mathrm{KD45}_n^C$. If we add $K_a\neg p$ to ϕ, K_n^C algorithm could answer $false$ correctly.

In the following, we introduce our normal form.

Definition 3.8. The semantics of the subscripted common knowledge modality $C_{\underline{a}}$ where $a \in \mathcal{A}$ is defined as follows:

- $M, w \models C_{\underline{a}}\phi$ iff for all u, v s.t. $wR^*_{\mathcal{A}}u$ and uR_bv where $b \neq a$, we have $M, v \models \phi$, where $R^*_{\mathcal{A}}$ is the reflexive transitive closure of the union of R_a for $a \in \mathcal{A}$.

We let $D_{\underline{a}}\phi$ abbreviate for $\neg C_{\underline{a}}\neg\phi$. We use $\mathcal{L}_{KCS}$ to refer to $\mathcal{L}_{KC}$ where the $C_{\underline{a}}$ modality is used instead of C.

Definition 3.9. The set of modal S-terms (resp. normal S-terms) is inductively defined as follows:

- A propositional term is a modal (resp. normal) S-term;
- A formula of the form $\phi_0 \wedge \bigwedge_{a\in\mathcal{A}}(K_a\phi_a \wedge L_a\Psi_a \wedge C_{\underline{a}}\mu_a \wedge D_{\underline{a}}\Lambda_a)$ is a modal S-term (resp. normal S-term), where ϕ_0 is a propositional term, Ψ_a, Λ_a are sets of modal S-terms (resp. normal S-terms), Ψ_a is not empty, and ϕ_a, μ_a are conjunctions of disjunctions of modal S-terms (resp. ϕ_a, μ_a are disjunctions of normal S-terms).

A formula ϕ is in SDNF (resp. normal SDNF) if it's a disjunction of modal S-terms (resp. normal S-terms).

In the above definition, note the requirement that Ψ_a is not empty. Due to seriality, we can always replace an empty Ψ_a with the set $\{\top\}$ in $\mathrm{KD45}_n^C$.

Proposition 3.10. The following hold in $\mathrm{KD45}_n^C$:

- $C\phi \Leftrightarrow \bigwedge_{a\in\mathcal{A}}(K_a\phi \wedge C_{\underline{a}}K_a\phi)$;
- $D\phi \Leftrightarrow \bigvee_{a\in\mathcal{A}}(L_a\phi \vee D_{\underline{a}}L_a\phi)$.

Proof. We only prove Item 1. The $\Rightarrow$ direction is obvious. For the $\Leftarrow$ direction, suppose $M, w \models \bigwedge_{a\in\mathcal{A}}(K_a\phi \wedge C_{\underline{a}}K_a\phi)$. Let v be reachable from w and p a shortest path from w to v. If p is of length 1, since $M, w \models \bigwedge_{a\in\mathcal{A}} K_a\phi$,

$M, v \models \phi$. If p is of length > 1, then the last two edges on the path must be of different agents; otherwise, by transitivity, there is a shorter path from w to v. Let a be the agent of the last edge. Since $M, w \models C_{\underline{a}} K_a \phi$, $M, v \models \phi$. Thus $M, w \models C\phi$. □

By Proposition 3.10, in KD45_n^C, every formula is equivalent to a formula in SDNF and a formula in normal SDNF.

As motivated by Example 3, we present the following rules for unfolding nested knowledge.

Proposition 3.11. The following hold in KD45_n^C:

- $K_a(K_a\phi \wedge \phi^+ \vee \psi) \Leftrightarrow K_a(\phi \wedge K_a\phi \wedge \phi^+ \vee \psi)$;
- $K_a(C_{\underline{b}}\phi \wedge \phi^+ \vee \psi) \Leftrightarrow K_a(\phi \wedge C_{\underline{b}}\phi \wedge \phi^+ \vee \psi)$;
- $L_a(K_a\phi \wedge \phi^+) \Leftrightarrow K_a\phi \wedge L_a(K_a\phi \wedge \phi^+)$;
- $L_a(C_{\underline{b}}\phi \wedge \phi^+) \Leftrightarrow K_a(\phi \wedge C_{\underline{b}}\phi) \wedge L_a(C_{\underline{b}}\phi \wedge \phi^+)$.

Here $\phi \in \mathcal{L}_{KCS}$, ϕ^+ is a modal S-term and ψ is in SDNF.

Definition 3.12. Let ϕ be in SDNF. We apply the rules for unfolding knowledge to ϕ, from inside to outside, and then put the resulting formula into an SDNF ϕ'. We say that ϕ' is in unfolded SDNF. Unfolded normal SDNF is similarly defined.

For example, $L_aK_a\phi \Leftrightarrow K_a\phi \wedge L_aK_a\phi$, and $K_aK_aK_a\phi \Leftrightarrow K_a(\phi \wedge K_a(\phi \wedge K_a\phi))$.

Proposition 3.13. Let ϕ be in unfolded SDNF. Given any formula ψ and agent a, if $\phi \models K_a\psi$ in KD45_n^C, then we also have $\phi \models K_a\psi$ in K_n^C.

Now we define the set of children of a modal S-term. Note that due to transitivity, $K_a\phi_a$ entails $K_aK_a\phi_a$ in KD45_n^C.

Definition 3.14. Let $\delta = \phi_0 \wedge \bigwedge_{a\in\mathcal{A}}(K_a\phi_a \wedge L_a\Psi_a \wedge C_{\underline{a}}\mu_a \wedge D_{\underline{a}}\Lambda_a)$ be a modal S-term. For $a \in \mathcal{A}$, we let γ_a denote the knowledge of a, *i.e.*, $\gamma_a = \phi_a \wedge K_a\phi_a \wedge C_{\underline{a}}\mu_a \wedge \bigwedge_{b\neq a}(\mu_b \wedge C_{\underline{b}}\mu_b)$. We define the set of δ's children, written $Gen^{KD45}(\delta)$, as the union of the following sets:

- $S_1 = \{\gamma_a \wedge \psi_a \mid a \in \mathcal{A}, \psi_a \in \Psi_a\}$,
- $S_{b\lambda 1} = \{\gamma_a \wedge \lambda \mid a \in \mathcal{A}, a \neq b\}$, and $S_{b\lambda 2} = \{\gamma_a \wedge D_{\underline{b}}\lambda \mid a \in \mathcal{A}\}$, where $b \in \mathcal{A}, \lambda \in \Lambda_b$.

Proposition 3.15. Let δ be a satisfiable modal S-term. Then the following hold for $Gen^{KD45}(\delta)$:

- for all $\delta' \in S_1$, δ' is satisfiable;
- for all $b \in \mathcal{A}$ and $\lambda \in \Lambda_b$, there is $\delta' \in S_{b\lambda_1} \cup S_{b\lambda_2}$ s.t. δ' is satisfiable.

Algorithm 2: $Check^{KD45}(\phi)$

input: ϕ is in unfolded SDNF **output:** $\top$ / $\bot$

$Check^K(\phi)$ where Gen^K is replaced by Gen^{KD45}

When $Check^{KD45}(\phi)$ returns $\top$, we first build a model M_ϕ for ϕ as we do for $Check^K(\phi)$, and then set each accessibility relation to its transitive and Euclidean closure. Note that seriality is ensured by requiring Ψ_a non-empty. The normal form we use and the definition of Gen^{KD45} ensure that the resulting KD45_n^C model M'_ϕ still satisfies ϕ.

Lemma 3.16. Let $a, b \in \mathcal{A}$. When $Check^{KD45}(\phi)$ returns $\top$, for all $w_\delta, w_{\delta'} \in M'_\phi$ s.t. $w_\delta R_a w_{\delta'}$, we have:

- when $\delta \models K_a\phi$, $\delta' \models \phi \wedge K_a\phi$;
- when $\delta' \models K_a\phi$, $\delta \models K_a\phi$;
- when $\delta \models C_{\underline{b}}\phi$, $\delta' \models C_{\underline{b}}\phi$;
- when $\delta' \models C_{\underline{b}}\phi$, either $\delta \models C_{\underline{b}}\phi$ or $\delta \models K_aC_{\underline{b}}\phi$.

Proof. We prove by induction on the order $(w_\delta, w_{\delta'})$ is added to R_a. The induction cases are straightforward. We prove the base case where $(w_\delta, w_{\delta'})$ is in R_a before edges are added. Item 1 and Item 3 follow from the definition of Gen^{KD45}. From $\delta' \models K_a\phi$ we can infer that δ is an implicant of $K_a\phi$, $K_aK_a\phi$ or $L_aK_a\phi$. Since δ is in unfolded SDNF, we have $\delta \models K_a\phi$. From $\delta' \models C_{\underline{b}}\phi$ we can infer that δ is an implicant of $C_{\underline{b}}\phi$, $K_aC_{\underline{b}}\phi$ or $L_aC_{\underline{b}}\phi$. If δ is an implicant of $L_aC_{\underline{b}}\phi$, we have $\delta \models K_aC_{\underline{b}}\phi$ by unfolding rules. □

Lemma 3.17. When $Check^{KD45}(\phi)$ returns $\top$, for all $w_\delta \in M'_\phi$, we have $M'_\phi, w_\delta \models \psi$ if $\delta \models \psi$.

Proof. We only prove the cases whose proofs are different from those for K_n^C. Consider ψ in the following forms:

- $K_a\psi'$. By Lemma 3.16, for all $w_{\delta'}$ s.t. $w_\delta R_a w_{\delta'}$, $\delta' \models \psi'$. By induction, $M_\phi, w_{\delta'} \models \psi'$. So $M_\phi, w_\delta \models K_a\psi'$.
- $C_{\underline{a}}\psi'$. By Lemma 3.16, for all $w_{\delta''}$ s.t. $w_\delta R^*_{\mathcal{A}} w_{\delta''}$, $\delta'' \models C_{\underline{a}}\psi'$. Then by the definition of Gen^{KD45}, for all $w_{\delta'}$ s.t. $w_{\delta''} R_b w_{\delta'}$ where $b \neq a$, we have $\delta' \models \psi'$. By induction, $M'_\phi, w_{\delta'} \models \psi'$. So $M'_\phi, w_\delta \models C_{\underline{a}}\psi'$. □

Theorem 3.18. $Check^{KD45}(\phi)$ returns $\top$ iff ϕ is satisfiable in KD45_n^C.

The proof for soundness follows from Lemma 3.17, while the proof for completeness is identical to that in Theorem 3.7.

Theorem 3.19. The complexity of $Check^{KD45}(\phi)$ is $O(4^{(d+1)(c+d)nl+l})$, where n is the number of agents, l, c, and d are as in Theorem 3.3.

Proof sketch. The complexity follows from Theorem 3.3 with two increasing from two factors: the common knowledge operator C is transformed into n copies of the subscripted version; knowledge of each agent is preserved when generating children of a modal S-term. □

4 Belief Revision and Update

In this section, we introduce our algorithms for higher-order belief change involving common knowledge. As Huang *et al.* [2017], we reduce change of epistemic formulas to that of lower-order epistemic formulas, and as basis we resort to change of propositional formulas. The essential difference between revision and update is: revision satisfies the conjunction property that when $\phi \wedge \phi'$ is satisfiable, $\phi \circ \phi' \Leftrightarrow \phi \wedge \phi'$, while update satisfies the distribution property that when both ϕ_1 and ϕ_2 are satisfiable, $(\phi_1 \vee \phi_2) \diamond \phi' \Leftrightarrow \phi_1 \diamond \phi' \vee \phi_2 \diamond \phi'$. We illustrate our main ideas with three examples.

Example 4. Revise $K_a(\neg p \wedge K_a\neg p)$ with Cp. When we recursively revise $\neg p \wedge K_a \neg p$, we cannot simply revise it with p, which will give us the incorrect result $K_a(p \wedge K_a\neg p) \wedge Cp$;

we will have to revise it with p and carry Cp, which gives us the correct result $K_a(p \wedge K_a p) \wedge Cp$.

Example 5. Revise $K_a(\neg p \wedge q)$ with $K_a \neg q \wedge L_a p$. We cannot simply revise old knowledge with new knowledge, which gives us $\neg p \wedge \neg q$, inconsistent with new possibility p. We will take the disjunction of $\neg p \wedge \neg q$ and the revision of old knowledge with the conjunction of new knowledge and new possibility. This gives us $\neg p \wedge \neg q \vee p \wedge \neg q$, equivalent to $\neg q$. The same idea applies to the revision of common knowledge.

Example 6. Revise $L_a(p \wedge q) \wedge L_a(\neg p \wedge q)$ with $K_a p \wedge L_a r$. Since there are old possibilities consistent with new knowledge, we will only keep such old possibilities and revise them with new knowledge. Thus we get $L_a(p \wedge q) \wedge K_a p \wedge L_a r$. However, consider revising $L_a(p \wedge \neg q) \wedge L_a(\neg p \wedge \neg q)$ with $K_a q \wedge L_a r$. Since all old possibilities are inconsistent with new knowledge, we revise all of them with new knowledge. The result is $L_a(p \wedge q) \wedge L_a(\neg p \wedge q) \wedge K_a q \wedge L_a r$. The same idea applies to the revision of common possibilities.

Motivated by Example 4, we define a "revision with carry" operator $\phi \circ_\gamma \phi'$ where γ is the carry. The difference between ϕ' and γ is that the revision result must entail ϕ' while the result only need to be consistent with γ.

Definition 4.1. Let $\circ$ be a revision operator. The revision of ϕ with ϕ' and carry γ, written $\phi \circ_\gamma \phi'$ is defined as follows:

- $\phi \circ_\gamma \phi' = \phi \circ \phi'$, if $\phi \wedge \gamma$ is satisfiable;
- $\phi \circ_\gamma \phi' = \phi \circ (\phi' \wedge \gamma)$, otherwise.

Similarly, we can define the "update with carry" operator.

Motivated by Example 6, we define the $*$ operator to restrict attention to consistent pairs of formulas if possible.

Definition 4.2. $\Phi * \Phi' =$

- $\{(\phi, \phi') \mid \phi \in \Phi, \phi' \in \Phi', \phi \wedge \phi'$ is satisfiable$\}$, if there are $\phi \in \Phi$ and $\phi' \in \Phi'$ s.t. $\phi \wedge \phi'$ is satisfiable;
- $\{(\phi, \phi') \mid \phi \in \Phi, \phi' \in \Phi'\}$, otherwise.

Below is the formal definition of our revision operator. Item 4 needs some explanation. In order to obtain a satisfiable result, we revise different parts of the old S-term in the following order, using the ideas behind the three motivating examples:

1. When old common knowledge is consistent with new knowledge and new possibilities, revise it with new common knowledge; otherwise simply assume new common knowledge.
2. Revise knowledge using revised common knowledge.
3. Revise possibilities using revised common knowledge and knowledge.
4. Keep old common possibilities that are consistent with revised common knowledge, knowledge and possibilities.

Definition 4.3. Let ϕ and ϕ' be in unfolded normal SDNF. The revision of ϕ with ϕ', written $\phi \circ \phi'$, is defined recursively:

1. When ϕ and ϕ' are propositional formulas, $\phi \circ \phi' = \phi \circ_s \phi'$, where $\circ_s$ is Satoh [1988]'s revision operator;
2. When $\phi = \bigvee \Delta$ and $\phi' = \bigvee \Delta'$, $\phi \circ \phi' = \bigvee \{\delta \circ \delta' \mid (\delta, \delta') \in \Delta * \Delta'\}$;
3. When ϕ and ϕ' are normal S-terms and $\phi \wedge \phi'$ is satisfiable, $\phi \circ \phi'$ is $\phi \wedge \phi'$ converted to a normal S-term;
4. Otherwise, ϕ and ϕ' are normal S-terms, and $\phi \wedge \phi'$ is not satisfiable. Let $\phi = \phi_0 \wedge \bigwedge_{a \in \mathcal{A}}(K_a \phi_a \wedge L_a \Psi_a \wedge C_{\underline{a}} \mu_a \wedge D_{\underline{a}} \Lambda_a)$, and $\phi' = \phi'_0 \wedge \bigwedge_{a \in \mathcal{A}}(K_a \phi'_a \wedge L_a \Psi'_a \wedge C_{\underline{a}} \mu'_a \wedge D_{\underline{a}} \Lambda'_a)$. Then $\phi \circ \phi' = \phi'' = \phi''_0 \wedge \bigwedge_{a \in \mathcal{A}}(K_a \phi''_a \wedge L_a \Psi''_a \wedge C_{\underline{a}} \mu''_a \wedge D_{\underline{a}} \Lambda''_a)$ where:
   (a) $\phi''_0 = \phi_0 \circ \phi'_0$;
   (b) If $C_{\underline{a}} \mu_a \wedge \bigwedge_{b \neq a}(K_b \phi'_b \wedge L_b \Psi'_b)$ is satisfiable, $\mu''_a = (\mu_a \circ_{\gamma_1} \mu'_a) \vee \bigvee_{\lambda'_a \in \Lambda'_a}(\mu_a \circ_{\gamma_2} (\mu'_a \wedge \lambda'_a))$ where $\gamma_1 = C_{\underline{a}} \mu'_a$ and $\gamma_2 = C_{\underline{a}} \mu'_a \wedge D_{\underline{a}} \lambda'_a$; otherwise $\mu''_a = \mu'_a$;
   (c) $\phi''_a = (\phi_a \circ_{\gamma_1} (\phi'_a \wedge \bigwedge_{b \neq a} \mu''_b)) \vee \bigvee_{\psi'_a \in \Psi'_a}(\phi_a \circ_{\gamma_2} (\psi'_a \wedge \phi'_a \wedge \bigwedge_{b \neq a} \mu''_b))$, where $\gamma_1 = K_a \phi'_a \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a$ and $\gamma_2 = K_a \phi'_a \wedge L_a \psi'_a \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a$;
   (d) $\Psi''_a = \{\psi \circ_\gamma \psi' \mid (\psi, \psi') \in \Psi_a * \{\phi''_a \wedge \bigwedge_{b \neq a} \mu''_b\}\} \cup \Psi'_a$, where $\gamma = K_a \phi''_a \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a$;
   (e) $\Lambda''_a = \{\lambda \mid \lambda \in \Lambda_a$ and $D_{\underline{a}} \lambda \wedge \bigwedge_{a \in \mathcal{A}}(K_a \phi''_a \wedge L_a \Psi''_a \wedge C_{\underline{a}} \mu''_a)$ is satisfiable$\} \cup \Lambda'_a$.

We now state properties of our revision operator.

Definition 4.4. The set of disjunct-wise satisfiable (d-sat) normal SDNFs is inductively defined:

- A disjunction $\bigvee \Delta$ of normal S-terms is d-sat if each $\delta \in \Delta$ is d-sat;
- A normal S-term $\phi_0 \wedge \bigwedge_{a \in \mathcal{A}}(K_a \phi_a \wedge L_a \Psi_a \wedge C_{\underline{a}} \mu_a \wedge D_{\underline{a}} \Lambda_a)$ is d-sat if it is satisfiable and each disjunct in each ϕ_a or μ_a is d-sat.

Proposition 4.5. An unfolded normal S-term without D modalities $\phi = \phi_0 \wedge \bigwedge_{a \in \mathcal{A}}(K_a \phi_a \wedge L_a \Psi_a \wedge C_{\underline{a}} \mu_a)$ is satisfiable if the following hold:

1. ϕ_0 is propositionally satisfiable;
2. For all $a \in \mathcal{A}$ and for all $\psi_a \in \Psi_a$, $\phi_{\psi_a} = \psi_a \wedge \phi_a \wedge K_a \phi_a \wedge \bigwedge_{b \neq a} \mu_b \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu_a$ is satisfiable.

Proof. We construct a model (M, w) for ϕ. By Item 1, we create a world w where $V(w)$ satisfies ϕ_0. By Item 2, for each $a \in \mathcal{A}$ and $\psi_a \in \Psi_a$, there is a $KD45_n^C$ model (M_{ψ_a}, w_{ψ_a}) satisfying ϕ_{ψ_a}. We add each (M_{ψ_a}, w_{ψ_a}) to M and let $w R_a w_{\psi_a}$. Then (M, w) is a $KD45_n^C$ model for ϕ after calculating the transitive and Euclidean closures. □

Proposition 4.6. Let ϕ and ϕ' be two d-sat unfolded normal SDNFs. Then $\phi \circ \phi'$ is a d-sat normal SDNF, and $\phi \circ \phi' \models \phi'$. Moreover, when $\phi \wedge \phi'$ is satisfiable, $\phi \circ \phi' \Leftrightarrow \phi \wedge \phi'$.

Proof. The conjunction property follows from Definition 4.3 Item 3 directly. Now we assume that $\phi \wedge \phi'$ is unsatisfiable.

We first prove $\phi \circ \phi' \models \phi'$ by induction on the modal depth of ϕ. Let $md(\phi)$ denote the modal depth of ϕ. When $md(\phi) = 0$, we consider ϕ' in two cases:

- $md(\phi') = 0$. Then $\phi \circ_s \phi' \models \phi'$ by Satoh's revision.

- $md(\phi') > 0$. Let $\phi = \bigvee \Delta$ and $\phi' = \bigvee \Delta'$. By Definition 4.3 Item 2, $\phi \circ \phi' = \bigvee \{\delta \circ \delta' \mid (\delta, \delta') \in \Delta * \Delta'\}$. For each $(\delta, \delta') \in \Delta * \Delta'$, since higher-order subformulas in δ' are directly conjoined into $\delta \circ \delta'$, we have $\delta \circ \delta' \models \delta'$. Let $\phi \circ \phi' = \bigvee \Delta''$. For each $\delta'' \in \Delta''$, there is a $\delta' \in \Delta'$ s.t. $\delta'' \models \delta'$. Therefore, $\phi \circ \phi' \models \phi'$.

When $md(\phi) = n + 1$, we consider ϕ and ϕ' in two cases:

- ϕ and ϕ' are normal S-terms. Let $\phi'' = \phi \circ \phi'$. By Items (a), (d) and (e), we have $\phi''_0 \models \phi'_0$, $\bigwedge_{a \in \mathcal{A}} L_a \Psi''_a \models \bigwedge_{a \in \mathcal{A}} L_a \Psi'_a$ and $\bigwedge_{a \in \mathcal{A}} D_{\underline{a}} \Lambda''_a \models \bigwedge_{a \in \mathcal{A}} D_{\underline{a}} \Lambda'_a$. By Item (b) and induction, for each agent a we have $\mu''_a \models \mu'_a$, thus $\bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a \models \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu'_a$. Similarly, by Item (c) we have $\bigwedge_{a \in \mathcal{A}} K_a \phi''_a \models \bigwedge_{a \in \mathcal{A}} K_a \phi'_a$. Take the conjunction of the above results, we have $\phi'' \models \phi'$.
- Otherwise, we follow the proof in the base case.

Let γ be a modal S-term s.t. $\phi' \wedge \gamma$ is d-sat and $\phi'' = \phi \circ_\gamma \phi'$. Now we prove that ϕ'' and $\phi'' \wedge \gamma$ are d-sat by induction on the modal depth of ϕ. When $md(\phi) = 0$, $\phi \wedge \gamma$ is d-sat and $\phi \circ_\gamma \phi' = \phi \circ \phi'$. We consider ϕ' in two cases:

- $md(\phi') = 0$. Then $\phi \circ_s \phi'$ is d-sat by Satoh's revision.
- $md(\phi') > 0$. Let $\phi = \bigvee \Delta$ and $\phi' = \bigvee \Delta'$. We have $\phi \circ \phi' = \bigvee \{\delta \circ \delta' \mid (\delta, \delta') \in \Delta * \Delta'\}$. For each $(\delta, \delta') \in \Delta * \Delta'$, let $\delta = \phi_0$ and $\delta' = \phi'_0 \wedge \bigwedge_{a \in \mathcal{A}} (K_a \phi'_a \wedge L_a \Psi'_a \wedge C_{\underline{a}} \mu'_a \wedge D_{\underline{a}} \Lambda'_a)$. Thus $\delta \circ \delta' = (\phi_0 \circ_s \phi'_0) \wedge \bigwedge_{a \in \mathcal{A}} (K_a \phi'_a \wedge L_a \Psi'_a \wedge C_{\underline{a}} \mu'_a \wedge D_{\underline{a}} \Lambda'_a)$. Again by Satoh's revision, $\delta \circ \delta'$ is d-sat. Therefore, ϕ'' and $\phi'' \wedge \gamma$ are d-sat.

When $md(\phi) = n + 1$, we consider ϕ and ϕ' in two cases:

- ϕ and ϕ' are normal S-terms. By Item (a) and Satoh's revision, ϕ''_0 is d-sat. By Item (b) and induction, μ''_a is d-sat. By definitions of "revision with carry" and unfolded formulas, we can show that $C_{\underline{a}} \mu''_a$ is d-sat. Similarly, we have that ϕ''_a and $K_a \phi''_a$ are d-sat. Since $\phi''_a \models \bigwedge_{b \neq a} \mu''_b$, $\phi''_a \wedge \bigwedge_{b \neq a} \mu''_b \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a$ is d-sat. By Item (d) and induction, $\psi''_a \wedge \phi''_a \wedge K_a \phi''_a \wedge \bigwedge_{b \neq a} \mu''_b \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a$ is d-sat for each $\psi''_a \in \Psi''_a$. By Proposition 4.5 and Item (e), ϕ'' is d-sat. By Definition 4.1, we can show that $\phi'' \wedge \gamma$ is d-sat.
- Otherwise, we follow the proof in the base case. □

Finally, we present the formal definition of our update operator. For Item 4, the difference with that of the revision operator lies with Item (d) where we update each possibility. Recall that for Item 4 (d) of the revision operator, when there are possibilities consistent with new common knowledge and knowledge, we only keep them and revise them.

Definition 4.7. Let ϕ and ϕ' be in unfolded normal SDNF. The update of ϕ with ϕ', written $\phi \diamond \phi'$, is defined recursively:

1. When ϕ and ϕ' are propositional formulas, $\phi \diamond \phi' = \phi \diamond_w \phi'$, where $\diamond_w$ is Winslett [1988]'s update operator;
2. When $\phi = \bigvee \Delta$, $\phi \diamond \phi' = \bigvee_{\delta \in \Delta} \delta \diamond \phi'$;
3. When ϕ is an S-term, and $\phi' = \bigvee \Delta'$, $\phi \diamond \phi' = \bigvee \{\phi \diamond \delta' \mid (\phi, \delta') \in \{\phi\} * \Delta'\}$;
4. Otherwise, ϕ and ϕ' are normal S-terms, $\phi \diamond \phi' = \phi''$ where ϕ, ϕ', ϕ'' have the form as in Definition 4.3, and:
   (a) $\phi''_0 = \phi_0 \diamond \phi'_0$;
   (b) If $C_{\underline{a}} \mu_a \wedge \bigwedge_{b \neq a} (K_b \phi'_b \wedge L_b \Psi'_b)$ is satisfiable, $\mu''_a = (\mu_a \diamond_{\gamma_1} \mu'_a) \vee \bigvee_{\lambda'_a \in \Lambda'_a} (\mu_a \diamond_{\gamma_2} (\mu'_a \wedge \lambda'_a))$ where $\gamma_1 = C_{\underline{a}} \mu'_a$ and $\gamma_2 = C_{\underline{a}} \mu'_a \wedge D_{\underline{a}} \Lambda'_a$; otherwise $\mu''_a = \mu'_a$;
   (c) $\phi''_a = (\phi_a \diamond_{\gamma_1} (\phi'_a \wedge \bigwedge_{b \neq a} \mu''_b)) \vee \bigvee_{\psi'_a \in \Psi'_a} (\phi_a \diamond_{\gamma_2} (\psi'_a \wedge \phi'_a \wedge \bigwedge_{b \neq a} \mu''_b))$, where $\gamma_1 = K_a \phi'_a \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a$ and $\gamma_2 = K_a \phi'_a \wedge L_a \psi'_a \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a$;
   (d) $\Psi''_a = \{\psi \diamond_\gamma (\phi''_a \wedge \bigwedge_{b \neq a} \mu''_b) \mid \psi \in \Psi_a\} \cup \Psi'_a$, where $\gamma = K_a \phi''_a \wedge \bigwedge_{a \in \mathcal{A}} C_{\underline{a}} \mu''_a$;
   (e) $\Lambda''_a = \{\lambda \mid \lambda \in \Lambda_a \text{ and } D_{\underline{a}} \lambda \wedge \bigwedge_{a \in \mathcal{A}} (K_a \phi''_a \wedge L_a \Psi''_a \wedge C_{\underline{a}} \mu''_a) \text{ is satisfiable}\} \cup \Lambda'_a$.

Proposition 4.8. Let ϕ and ϕ' be two d-sat unfolded normal SDNFs. Then $\phi \diamond \phi'$ is a d-sat normal SDNF, and $\phi \diamond \phi' \models \phi'$. Moreover, $(\phi_1 \vee \phi_2) \diamond \phi' \Leftrightarrow \phi_1 \diamond \phi' \vee \phi_2 \diamond \phi'$.

The distribution property can be obtained from Definition 4.7 Item 2 directly. We omit the proofs for the other properties of update since they are similar to those of revision.

5 Implementation and Experimental Results

Based on our reasoning, revision and update algorithms, we implemented a planner called MEPC for multi-agent epistemic planning with common knowledge. Our planning algorithm supports contingent planning by extending breadth-first search to AND/OR graphs. We evaluate MEPC with five domains which use common knowledge in different ways.

Collaboration-and-Communication: CC(n). There are 4 rooms, 2 agents and n boxes. Agents can enter rooms and sense boxes in it. Also, agents can share information. The goal is to let agents know the positions of boxes. The common knowledge is that each box is in exactly one room.

Muddy-Children: MC(n, m). There are n children and m of them are muddy.

Public-Announcing: PA(n). There are n agents in room 1. Agent 1 can sense whether the book is in room 2 and take it away. Each agent can share his belief about the book to others, while agent n can make a public announcement. The goal is to achieve common knowledge that agent 1 belives the book is missing.

Selective-Communication: SC(n). There are n rooms and n agents in different rooms. A secret is false, but initially it's common knowledge that agent 1 believes the secret is true. Agent 1 can find out that the secret is false. Each agent can move to a neighboring room and tell the secret to others in the room. The goal is to let all agents except agent 1 believe that the secret is false, while agent 1 believes that it is true.

Prisoners-and-Lightbulb: PL(n). This domain is adapted from a puzzle in [van Ditmarsch and Kooi, 2015]. There are n prisoners in the prison. Every day one of them is interrogated in a room furnished with a light bulb. The goal is to let one of them know that all the prisoners have been interrogated.

Our experiments were run on a Windows machine with 3.50GHz CPU and 8GB RAM. The results are shown in Table 1. The 2nd-4th columns indicate the number of agents, the

Domain	$\mathcal{A}$	$\mathcal{P}$	$\lvert\mathcal{S}+\mathcal{D}\rvert$	MEPC
CC(2)	2	16	6+16	44.831-33.830(5/10165)
CC(3)	2	20	14+16	1953.0-1703.0(5/35351)
MC(2,2)	2	3	0+2	0.014-0.001(2/3)
MC(3,3)	3	4	0+2	0.101-0.013(3/4)
MC(4,2)	4	5	0+2	0.201-0.027(2/3)
MC(4,3)	4	5	0+2	2.603-0.435(3/4)
MC(4,4)	4	5	0+2	5.101-0.895(4/5)
MC(5,5)	5	6	0+2	108.682-20.187(5/6)
PA(2)	2	1	1+3	0.011-0.001(4/11)
PA(3)	3	1	1+4	0.013-0.001(5/15)
PA(4)	4	1	1+5	0.019-0.003(6/20)
PA(5)	5	1	1+6	0.027-0.008(7/25)
SC(2)	2	5	0+13	0.103-0.018(0/42)
SC(3)	3	10	0+25	9.198-1.597(9/2610)
SC(4)	4	17	0+41	738.566-61.571(10/70951)
PL(2)	2	12	0+6	0.061-0.002(5/21)
PL(3)	3	15	0+9	0.416-0.050(7/93)
PL(4)	4	18	0+12	4.040-0.443(9/385)
PL(5)	5	21	0+15	30.944-3.582(11/1493)

Table 1: Experimental Results

number of atoms, and the number of sensing and deterministic actions. In the last column, A-B(C/D) indicates A seconds of total time, B seconds spent on satisfiability solving, depth C of solution tree (C=0 means the problem is unsolvable), and D nodes searched. The results show that our planner is capable of solving these problems of planning with common knowledge. However, our planner doesn't scale well, due to the exponential time complexity of the satisfiability solving algorithm and the naive search method we use.

6 Conclusion

In this paper, we have extended an existing framework for multi-agent epistemic planning with the capability to deal with general common knowledge. We propose a novel normal form for multi-agent KD45 with common knowledge which makes use of the subscripted common knowledge operator and unfolds knowledge in a certain way. We propose satisfiability solving, revision and update algorithms for this normal form. We implemented a planner MEPC, and it is capable of solving several domains involving typical usage of common knowledge. Despite the current limitations of our work, we have made a significant first step towards multi-agent epistemic planning with common knowledge. In the future, we are interested in extending our work to handle common knowledge of a subset of agents. Proposing more efficient algorithms for reasoning about common knowledge is another important future work.

Acknowledgments

We acknowledge support from the Natural Science Foundation of China under Grant Nos. 61572535 and 61463044.

References

[Bolander and Andersen, 2011] Thomas Bolander and Mikkel Birkegaard Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.

[del Cerro *et al.*, 2001] Luis Fariñas del Cerro, David Fauthoux, Olivier Gasquet, Andreas Herzig, Dominique Longin, and Fabio Massacci. LoTREC : The generic tableau prover for modal and description logics. In *Proceedings of the First International Joint Conference on Automated Reasoning, (IJCAR 2001)*, pages 453–458, 2001.

[Fagin *et al.*, 1995] Ronald Fagin, Joseph Y Halpern, Moshe Y Vardi, and Yoram Moses. *Reasoning about knowledge*. MIT Press, 1995.

[Hales *et al.*, 2012] James Hales, Tim French, and Rowan Davies. Refinement quantified logics of knowledge and belief for multiple agentsc. In *Proceedings of the Ninth Conference on Advances in Modal Logic, (AiML 2012)*, pages 317–338, 2012.

[Halpern and Moses, 1992] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(2):319–379, 1992.

[Huang *et al.*, 2017] Xiao Huang, Biqing Fang, Hai Wan, and Yongmei Liu. A general multi-agent epistemic planner based on higher-order belief change. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, (IJCAI 2017)*, pages 1093–1101, 2017.

[Kominis and Geffner, 2015] Filippos Kominis and Hector Geffner. Beliefs in multiagent planning: From one agent to many. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, (ICAPS 2015)*, pages 147–155, 2015.

[Muise *et al.*, 2015] Christian J. Muise, Vaishak Belle, Paolo Felli, Sheila A. McIlraith, Tim Miller, Adrian R. Pearce, and Liz Sonenberg. Planning over multi-agent epistemic states: A classical planning approach. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, (AAAI 2015)*, pages 3327–3334, 2015.

[Satoh, 1988] Ken Satoh. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the Second International Conference on Fifth Generation Computer Systems, (FGCS 1988)*, pages 455–462, 1988.

[van Ditmarsch and Kooi, 2015] Hans van Ditmarsch and Barteld Kooi. *One Hundred Prisoners and a Light Bulb*. Springer, 2015.

[Van Ditmarsch *et al.*, 2007] Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. *Dynamic epistemic logic*, volume 337. Springer, 2007.

[Winslett, 1988] Marianne Winslett. Reasoning about action using a possible models approach. In *Proceedings of the Seventh National Conference on Artificial Intelligence, (AAAI 1988)*, pages 89–93, 1988.