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 $\phi$ is common knowledge of a group of agents if everybody knows $\phi$, everybody knows everybody knows $\phi$, 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 $\phi$. 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 $\phi$. 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 $C_\alpha\phi$ which means that it is common knowledge that everybody except agent $\alpha$ knows $\phi$. 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:

\[
\phi ::= p \mid \neg \phi \mid \phi \land \phi \mid K_a \phi \mid C \phi,
\]

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

Intuitively, \( K_a \phi \) means agent \( a \) knows \( \phi \), and \( C \phi \) means all agents commonly know \( \phi \). 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 \( \phi \) is possible, and \( D \phi \) means all agents commonly think \( \phi \) is possible.

We use \( \phi \) and \( \psi \) to represent formulas, \( \Phi \) and \( \Psi \) to represent sets of formulas, \( \top \) and \( \bot \) to denote true and false, respectively. \( \bigvee \Phi \) stands for the disjunction of members in \( \Phi \), 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 \( \phi \).

**Definition 2.2.** A frame is a pair \((W,R)\), where \( W \) is a non-empty 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 \( wR_{a}w' \), we say \( w' \) is a \( \alpha \)-child of \( w \). We say \( R_{a} \) serial if for any \( w \in W \), there is a \( w' \in W \) s.t. \( wR_{a}w' \); we say \( R_{a} \) is transitive if \( wR_{a}u \) and \( uR_{a}v \) imply \( wR_{a}v \); we say \( R_{a} \) is Euclidean if \( wR_{a}u \) and \( wR_{a}v \) imply \( uR_{a}v \). A KD45s 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 \( V : W \rightarrow 2^\mathcal{P} \) 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 \( M,w \not\models \phi \);
- \( M,w \models \phi \land \psi \) iff \( M,w \models \phi \) and \( M,w \models \psi \);
- \( M,w \models K_a \phi \) iff for all v s.t. \( wR_{a}v \), \( M,v \models \phi \);
- \( M,w \models C \phi \) iff for all v s.t. \( wR_{A}v \), \( M,v \models \phi \), where \( R_{A} \) is the transitive closure of the union of \( R_{a} \) for \( a \in \mathcal{A} \).

Consider the context of \( K^C_n \) or KD45Cn. We say \( \phi \) is satisfiable if there is a Kripke model \((M,w)\) s.t. \( M,w \models \phi \).

**Theorem 2.5.** [Halpern and Moses, 1992] The satisfiability problems for \( K^C_n \) and KD45Cn are EXPTIME-complete.

Halpern and Moses presented an algorithm for checking satisfiability in \( K^C_n \). Given a formula \( \phi \in \mathcal{L}_{KC} \), their algorithm first generates a set \( S \) consisting of all subformulas of \( \phi \) 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 \( \phi \) with worlds in \( W \).

We now define a normal form for \( K^C_n \).

**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 \( \phi_0 \land \bigwedge_{a \in \mathcal{A}}(K_a \phi_a \land L_a \psi_a) \land C \mu \land \Delta \lambda \) is a modal term, where \( \phi_0 \) is a propositional term, \( \Psi_a, \Lambda \) are sets of modal terms, and \( \phi_a, \mu, \lambda \) are conjunctions of disjunctions of modal terms.

A formula \( \phi \) is in DNF if it’s a disjunction of modal terms. A formula \( \phi \) 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 \( K^C_n \), 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, i.e., 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 \( \psi \) in each modal atom, we repeat this process. By induction on the modal depth of \( \phi \), we can show that the length of the resulting DNF is at most singly exponential in the length of \( \phi \).

In the rest of the paper, the logic we use is KD45Cn, except that in Section 3.1, we discuss satisfiability for \( K^C_n \).

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 \( m \)th time the children with muddy foreheads will all answer “Yes”.

**Definition 2.8.** A MEP problem is a tuple \((A, P, D, S, I, G)\), where \( A \) is a set of agents, \( P \) is a set of atoms, \( D \) is a set of deterministic actions, \( S \) is a set of sensing actions, \( I \in \mathcal{L}_{KC} \) is the initial KB, and \( G \in \mathcal{L}_{KC} \) is the goal.
Example 1. Assume that there are three children ($A = \{a, b, c\}$) and all of them are muddy. The atoms are: $m_i$, indicating the $i$th child is muddy. The deterministic action-
es 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 $m_a \land m_b \land m_c \land \bigwedge_{i \in A}(K_i \bigwedge_{j \neq i} \neg m_j)$. The goal is $\bigwedge_{i \in A}(m_i \to K_m)$.

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

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

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

- $\text{pre} = \bigwedge_i (\neg K_m - K_i \neg m_i)$,
- $\text{eff}_1 = \langle \top, \bigvee_i (m_i \land m_j) \rangle$,
- $\text{eff}_2 = \langle \top, \bigvee_i (\neg m_i \land m_j) \rangle$.

Here $\text{pre}$ says that no child knows if he is muddy, $\text{eff}_1$ 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 $\text{eff}_2$ 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 $\phi \in L_{KC}$ if $\phi \models \text{pre}(a)$. Suppose $a$ is executable wrt $\phi$. The progression of $\phi$ 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 $\bullet$ for $L_{KC}$. We use update for progression wrt deterministic actions and revision for sensing actions.

Definition 2.11. Let $\phi \in L_{KC}$ and $a$ a deterministic action where $\text{eff}(a) = \{\phi_1, \psi_1, \ldots, \phi_m, \psi_m\}$. Suppose $\phi_i, \ldots, \phi_m$ are all the $\phi_i$’s s.t. $\phi_i \models \psi_i$. Then the progression of $\phi$ wrt $a$ is defined as $((\phi \circ \psi_1) \ldots \circ \psi_m).

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

The progression of $\phi$ wrt a sequence of actions (with sensing results for sensing actions) is inductively defined as follows: $\text{prog}(\phi, \epsilon) = \phi$; $\text{prog}(\phi, \langle a; \sigma \rangle) = \text{prog}(\text{prog}(\phi, a), \sigma)$ if $\phi \models \text{pre}(a)$, 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 $K_m$, and then extend it to KD45m.

3.1 $K_m^C$ Satisfiability

The main idea of our algorithm for checking satisfiability in $K_m^C$ is as follows. The input formula is in DNF, and we attempt to build a model for it recursively. When checking a modal term $\delta = \psi_a \land \bigwedge_{\alpha} (K_a \psi_{\alpha}) \land C_{\mu} \land DA$, if $\psi_a$ 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 $\psi_{\alpha}$ and common knowledge $\mu \land C_{\mu}$, and check if the resulting formula is satisfiable.
- If not, we immediately return $\bot$, otherwise we check $\lambda \land \psi_a \land C_{\mu}$ and $DA \land \psi_a \land C_{\mu}$ 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 $\delta$. When checking a CDNF $\phi$, we put it into a DNF $\lor \Delta$, and recursively check each $\delta' \in \Delta$. We call $\delta'$ a derivant of $\phi$.

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 $\delta'$ is derived from a child of a modal term $\delta$, we add an edge from $\delta$ to $\delta'$, mark the edge by $1$ if $\delta'$ is derived from $\psi_a \land C_{\mu}$, and $2$ otherwise. Now, when checking a modal term $\delta$ which already exists, we return its current mark.

A formula $\delta$ 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 $DA$ in each modal term $\delta$, we check whether there is a modal term $\delta''$ reachable from $\delta$. $\delta'$ is a conjunct of $\delta'$ and $\delta''$ is marked by $\top$. If not, we mark $\delta$ 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 $\delta$ be a modal term. For $a \in A$, let $\gamma_a$ denote $a$’s knowledge, i.e., $\gamma_a = \phi_a \land \mu \land C_{\mu}$. We define the set of $\delta$’s children, $Gen^K(\delta)$, as the union of the following:

- $S_1 = \{\gamma_a \land \psi_a | a \in A, \psi_a \in \Psi_a\}$
- $S_{\lambda_1} = \{\gamma_a \land \lambda | a \in A\}$
- $S_{\lambda_2} = \{\gamma_a \land DA \land \psi_a | a \in A\}$, where $\lambda \in \Lambda$.

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

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

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

\begin{itemize}
  \item \textbf{input}: $\phi \in L_{KC}$ is in DNF $\lor \Delta$ \quad \textbf{output}: $\top / \bot$
  \item set $G = (W, R)$ to the empty graph;
  \item if $\text{Sat}^K(\phi, \bot, a, 1) = \bot$ then return $\bot$;
  \item call $\text{Update}$;
  \item foreach $\delta$ in $\Delta$ do
    \begin{itemize}
      \item if $\text{Mark}(\delta) = \top$ then return $\top$;
    \end{itemize}
  \end{itemize}
Function $Sat^K(\phi, \phi', i, n)$

**input**: $\phi$ is in CDNF, $\phi'$ is a modal term, $i \in A$, $n = 1, 2$

if $\phi$ is not a modal term then
transform $\phi$ into DNF $\lor \Delta$;
foreach $\delta$ in $\Delta$
    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 $Mark(\phi)$;
    $W \leftarrow W \cup \{\phi\}$,
    $Mark(\phi) \leftarrow \top$;
$Gen^K(\phi) = S_1 \cup \bigcup_{a \in A} (S_{a1} \cup S_{a2})$;
if one of the following conditions holds:
1. $\phi_0$ 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 $\delta_a \in \bigcup_{a \in A} S_{a1} \cup S_{a2}$ do
    call $Sat^K(\delta_a, \phi, a, 2)$;
return $\top$

Procedure $Update$

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

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

1. We calculate $Gen^K(\phi) = \{\delta_1, \delta_2\}$, where $\delta_1 = \neg p \land p \land Cp$ and $\delta_2 = D-p \land p \land Cp$, and set $W = \{\phi, \delta_1, \delta_2\}$ and $R_a = \{((\phi, \delta_1), 2), (\phi, \delta_2)\}$.
2. We check $\delta_1$ next. Since its propositional part is unsatisfiable, we set $Mark(\delta_1) = \bot$.
3. We turn to $\delta_2$, $Gen^K(\delta_2) = \{\delta_1, \delta_2\}$, $R_a \leftarrow R_a \cup ((\delta_1, \delta_2), (\delta_2, \delta_2)\}_2$, $Mark(\delta_2) = \top$. Thus $Mark(\phi) = \top$.
4. Since $D-p$ is a conjunct of $\phi$ and $\delta_2$ but there isn’t a reachable modal term $\delta'$ s.t. $\neg p$ is a conjunct of $\delta'$ 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 $O(4^{(d+1)l+1})$, where $l$ is the length of $\phi$, $d$ is the modal depth of $\phi$, and $c$ is the depth of nesting of $C$ modalities in $\phi$.

![Figure 1: Procedure of $Check^K(Cp \land D-p)$](image)

**Procedure $Propagate(\delta)$**

**input**: $\delta$ is a modal term

foreach $\delta'$ 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 $\phi$ without $C$. For any modal term $\delta$ generated during the algorithm, $\delta$ corresponds to a substring of $\phi$. Hence the graph has at most $2^d$ nodes. Now consider that $\phi$ contains $C$ modalities. Let $C \nu$ be the conjunction of all subformulas $C\mu$ in $\phi$. The maximal length of $C \nu$ is $cl$. The children generated from $C \nu$ are in the form $\nu_1 \land C \nu$, where $\nu_1$ doesn’t contain common knowledge since $C \nu$ is maximal. For $1 \leq i < d$, we name the children generated from $\nu_i$ as $\nu_{i+1}$. All $\nu_i$ corresponds to a substring of $\nu$, thus the length of $\nu_i$ 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 $\bigwedge_{1 \leq i \leq d} \nu_i \land C \nu$ in a modal term generated during the algorithm is $cl + dcl$. Hence the maximum number of nodes is $2^{(d+1)c+1}$. Let $n$ be the number of nodes. Cycle checking and updating marks requires $O(n^2)$ time. Thus we obtain the complexity result $O(4^{(d+1)c+1})$. 

We proceed to prove soundness and completeness of $Check^K$. For soundness, when $Check^K(\phi)$ returns $\top$, we build a model $M_a$ for $\phi$ by modifying the frame $G = (W, R)$ as follows: remove $\delta \in W$ s.t. $Mark(\delta) = \top$; rename $\delta \in W$ to $w_{\delta}$; and create a valuation map $V$ s.t. $V(w_{\delta})$ satisfies $\phi_0$ in $\delta$. We show that for each $w_{\delta}$ in $M_a$, $w_{\delta}$ satisfies $|\delta|$. To facilitate the proof, we introduce the notion of implicants.

**Definition 3.4.** If the conjunction of a CDNF $\phi$ and another CDNF $\phi'$ is equivalent to a DNF $\lor \Delta$, we call each $\delta \in \Delta$ an implicant of $\phi$, and write $|\delta| = \phi$.

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

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

**Proof.** We prove by structural induction on $\psi$. Consider $\psi$ in the following forms:

- A propositional literal. Then $\psi$ is a conjunct of the propositional term of $\delta$. By the definition of $V(w_\delta)$, we have $M_\delta, w_\delta \models \psi$.
- $\psi_1 \land \psi_2$. Then $\delta \models \psi_1$ and $\delta \models \psi_2$. By induction, $M_\delta, w_\delta \models \psi_1$ and $M_\delta, w_\delta \models \psi_2$. So $M_\delta, w_\delta \models \psi_1 \land \psi_2$.
- $\psi_1 \lor \psi_2$. Then $\delta \models \psi_1$ or $\delta \models \psi_2$. By induction, $M_\delta, w_\delta \models \psi_1$ or $M_\delta, w_\delta \models \psi_2$. So $M_\delta, w_\delta \models \psi_1 \lor \psi_2$.

**3.2 KD45$^C_n$ Satisfiability**

The main idea of our algorithm for checking satisfiability in KD45$^C_n$ is as follows. We require the input formula $\phi$ to be in a certain normal form. We apply to $\phi$ the algorithm for checking satisfiability in $K_0^C$. When the algorithm returns $\top$ and a model for $\phi$, 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 $\phi$. To motivate our normal form, let’s consider two examples.

**Example 2.** Let $A = \{a, b\}$, $P = \{p\}$, and $\phi = DK_0 p$. Consider $M = (W, R, V)$ where $W = \{(0, 1, 2), R_0 = \{(0, 1), (1, 2)\}, R_0 = 0, V(0) = V(1) = 0, V(2) = \{p\}$. Then $(M, 0)$ is a model for $\phi$ in $K_0^C$, but it can’t be made a model for $\phi$ in KD45$^C_n$ by adding edges. Now suppose we have a modality $D_0 \phi$ meaning $\phi$ 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 $R_0 = \{(0, 1)\}$. Then $(M', 0)$ is a model for $\phi = D_0 K_0 p$ in $K_0^C$, and it can be made a model for $\phi$ in KD45$^C_n$ by adding edges.

**Example 3.** Let $\phi = L_0 (p \land K_0 \neg p)$. Then $\phi$ is satisfiable in $K_0^C$ but not in KD45$^C_n$, thus $K_0^C$ algorithm obtains the wrong answer. Note that $\phi$ entails $K_0 \neg p$ in KD45$^C_n$. If we add $K_0 \neg p$ to $\phi$, $K_0^C$ algorithm could answer $false$ correctly.

In the following, we introduce our normal form.

**Definition 3.8.** The semantics of the subscripted common knowledge modality $C_a$ where $a \in A$ is defined as follows:

- $M, w \models C_a \phi$ iff for all $u, v$ s.t. $wR_n u$ and $uR_v v$ where $b \neq a$, we have $M, v \models \phi$, where $R_n$ is the reflexive transitive closure of the union of $R_a$ for $a \in A$.

We let $D_0 \phi$ abbreviate for $\neg C_0 \phi$. We use $L_{KCS}$ to refer to $KCS$ where the $C_a$ 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 $\phi_1 \land \phi_2 \land \ldots \land \phi_n$ is in SDNF (resp. normal SDNF) if it’s a disjunction of modal $S$-terms (resp. normal S-terms).

A formula $\phi$ 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 $\Psi_a$ is not empty. Due to seriality, we can always replace an empty $\Psi_a$ with the set $\{\top\}$ in KD45$^C_n$.

**Proposition 3.10.** The following hold in KD45$^C_n$:

- $C\phi \equiv \bigwedge_{a \in A} (K_a \phi \land C_a K_a \phi)$;
- $D\phi \equiv \bigvee_{a \in A} (L_a \phi \lor D_a L_a \phi)$.

**Proof.** We only prove Item 1. The $\Rightarrow$ direction is obvious. For the $\Leftarrow$ direction, suppose $M, w \models \bigwedge_{a \in A} (K_a \phi \land C_a K_a \phi)$. Let $v$ be reachable from $w$ and $p$ a shortest path from $w$ to $v$. If $p$ is of length 1, since $M, w \models \bigwedge_{a \in A} K_a \phi$, $w$ must be marked $\top$. Therefore, by Lemma 3.5, $M, w \models \phi$.
$M, v \models \phi$. 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 $M, w \models C_aK_a\phi$, $M, v \models \phi$. 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$_n^C$:
- $K_a(K_a\phi \land \phi^+ \lor \psi) \equiv K_a(\phi \land K_a\phi \land \phi^+ \lor \psi)$;
- $K_a(C_b\phi \land \phi^+ \lor \psi) \equiv K_a(\phi \land C_b\phi \land \phi^+ \lor \psi)$;
- $L_a(K_a\phi \land \phi^+ \lor \psi) \equiv K_a\phi \land L_a(K_a\phi \land \phi^+ \lor \psi)$;
- $L_a(C_b\phi \land \phi^+ \lor \psi) \equiv K_a\phi \land L_a(C_b\phi \land \phi^+ \lor \psi)$.

Here $\phi \in L_{KCS}$, $\phi^+$ is a modal $S$-term and $\psi$ is in SDNF.

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

For example, $L_aK_a\phi \equiv L_a(K_a\phi \land L_aK_a\phi)$ and $K_aK_aK_a\phi \equiv K_a(\phi \land K_a(\phi \land K_a\phi))$.

**Proposition 3.13.** Let $\phi$ be in unfolded SDNF. Given any formula $\psi$ and agent $a$, if $\phi \models K_a\psi$ in KD45$_n$, then we also have $\phi \models K_a\psi$ in KD45$_n^C$.

Now we define the set of children of a modal $S$-term. Note that due to transitivity, $K_a\phi$ entails $K_a\phi$ in KD45$_n^C$.

**Definition 3.14.** Let $\delta = \phi_0 \land \bigwedge_{\alpha \in A}(K_a\phi_\alpha \land L_a\Psi_\alpha \land C_a\mu_\alpha \land D_\alpha L_a\phi_\alpha)$ be a modal $S$-term. For $a \in A$, we let $\gamma_a$ denote the knowledge of $a$, i.e., $\gamma_a = \phi_a \land K_a\phi_a \land C_a\mu_a \land \bigwedge_{b \neq a}(\mu_b \land C_b\mu_b)$. We define the set of $\delta$‘s children, written $Gen^{ KD45}_n(\delta)$, as the union of the following sets:
- $S_{b_1} = \{\gamma_a \land \psi_a \mid a \in A, \psi_a \in \Psi_a\}$,
- $S_{b_2} = \{\gamma_a \land \lambda \mid a \in A, a \neq b\}$,
- $S_{b_3} = \{\gamma_a \land D_\alpha L_a\phi_\alpha \mid a \in A\}$, where $b \in A, \lambda \in \Lambda_b$.

**Proposition 3.15.** Let $\delta$ be a satisfiable modal $S$-term. Then the following hold for $Gen^{ KD45}_n(\delta)$:
- for all $\delta' \in S_1$, $\delta'$ is satisfiable;
- for all $a \in A$ and $\lambda \in \Lambda_b$, there is $\delta' \in S_{b_1} \cup S_{b_2} \cup S_{b_3}$ s.t. $\delta'$ is satisfiable.

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

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

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

When Check$^{ KD45}_n(\phi)$ returns $\top$, we first build a model $M_\delta$ for $\phi$ 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 $\Psi_a$ non-empty. The normal form we use and the definition of $Gen^{ KD45}_n$ ensure that the resulting KD45$_n^C$ model $M_\delta$ still satisfies $\phi$.

**Lemma 3.16.** Let $a, b \in A$. When Check$^{ KD45}_n(\phi)$ returns $\top$, for all $w_1, w_2 \in M_\phi$ s.t. $w_3R_aw_2$, we have:
- when $\delta \models K_a\phi$, $\delta' \models K_a\phi$;
- when $\delta \models C_b\phi$, $\delta' \models C_b\phi$;
- when $\delta \models \neg C_b\phi$, either $\delta \models \neg C_b\phi$ or $\delta \models K_aC_b\phi$.

**Proof.** We prove by induction on the order ($w_3, w_2$) is added to $R_a$. The induction cases are straightforward. We prove the base case where ($w_3, w_2$) is in $R_a$ before edges are added. Item 1 and Item 3 follow from the definition of Gen$^{ KD45}_n$. From $\delta \models K_a\phi$ we can infer that $\delta$ is an implicant of $K_a\phi$, $K_aK_a\phi$ or $L_aK_a\phi$. Since $\delta$ is in unfolded SDNF, we have $\delta \models K_a\phi$. From $\delta' \models C_b\phi$ we can infer that $\delta$ is an implicant of $C_b\phi$, $K_aC_b\phi$ or $L_aC_b\phi$. If $\delta$ is an implicant of $L_aC_b\phi$, we have $\delta \models K_aC_b\phi$ by unfolding rules.

**Lemma 3.17.** When Check$^{ KD45}_n(\phi)$ returns $\top$, for all $w \in M_\phi$, we have $M_\delta$, $w_3 \models \psi$ if $\delta \models \psi$.

**Proof.** We only prove the cases whose proofs are different from those for $K^C_{n_1}$. Consider $\psi$ in the following forms:
- $K_a\psi$. By Lemma 3.16, for all $w_4$, s.t. $w_3R_4w_2$, $\delta' \models \psi$. By induction, $M_\delta, w_4 \models \psi$. So $M_\delta, w_3 \models K_a\psi$.
- $C_b\psi$. By Lemma 3.16, for all $w_5$, s.t. $w_3R_5w_2$, $\delta'' \models C_b\psi$. Then by the definition of Gen$^{ KD45}_n$, for all $w_5$, s.t. $w_5R_6w_2$, where $b \neq a$, we have $\delta'' \models \psi$. By induction, $M_\delta, w_5 \models \psi$. So $M_\delta, w_3 \models C_b\psi$. □

**Theorem 3.18.** Check$^{ KD45}_n(\phi)$ returns $\top$ iff $\phi$ 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$^{ KD45}_n(\phi)$ is $O(n^{(d+1)(c+d+n+l)})$, 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 $\phi \land \phi'$ is satisfiable, $\phi \land \phi' \models \phi \land \phi'$, while update satisfies the distribution property that when both $\phi_1$ and $\phi_2$ are satisfiable, $(\phi_1 \lor \phi_2) \land \phi' \models \phi_1 \land \phi' \lor \phi_2 \land \phi'$. We illustrate our main ideas with three examples.

**Example 4.** Revise $K_a(\neg p \land K_a\neg p)$ with $K_a\phi$. When we recursively revise $\neg p \land K_a\neg p$, we cannot simply revise it with $p$, which will give us the incorrect result $K_a(p \land K_a\neg p) \land C\phi$;
we will have to revise it with p and carry Cp, which gives us the correct result \( K_a(p \land K_a p) \land Cp \).

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

**Example 6.** Revise \( L_a(p \land q) \land L_a(\neg p \land q) \) with \( K_a p \land L_a r \). 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 \( L_a(p \land q) \land K_a p \land L_a r \). However, consider revising \( L_a(p \land \neg q) \land L_a(\neg p \land q) \) with \( K_a q \land L_a r \). Since all old possibilities are inconsistent with new knowledge, we revise all of them with new knowledge. The result is \( L_a(p \land q) \land L_a(\neg p \land q) \land K_a q \land L_a r \). The same idea applies to the revision of common possibilities.

Motivated by Example 4, we define a “revision with carry” operator \( \alpha \circ \gamma \) where \( \gamma \) is the carry. The difference between \( \alpha' \) and \( \gamma \) is that the revision result must entail \( \alpha' \) while the result only need to be consistent with \( \gamma \).

**Definition 4.1.** Let \( \circ \) be a revision operator. The revision of \( \alpha \) with \( \alpha' \) and carry \( \gamma \), written \( \phi \circ_\gamma \phi' \), is defined as follows:

1. \( \phi \circ_\gamma \phi' = \phi \circ \phi', \) if \( \phi \land \phi' \) is satisfiable;
2. \( \phi \circ_\gamma \phi' = \phi \circ (\phi' \lor \gamma) \), otherwise.

Similarly, we can define the “update with carry” operator.

Motivated by Example 6, we define the \( \ast \) operator to restrict attention to consistent pairs of formulas if possible.

**Definition 4.2.** \( \Phi \ast \Phi' = \{ (\phi, \phi') \mid \phi \in \Phi, \phi' \in \Phi', \phi \land \phi' \) is satisfiable \}, if there are \( \Phi \) and \( \Phi' \) s.t. \( \phi \land \phi' \) is satisfiable;

2. \( \{ (\phi, \phi') \mid \phi \in \Phi, \phi' \in \Phi' \}, \) 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 \( \phi \) and \( \phi' \) be in unfolded normal SDNF. The revision of \( \phi \) with \( \phi' \), written \( \phi \circ \phi' \), is defined recursively:

1. When \( \phi \) and \( \phi' \) are propositional formulas, \( \phi \circ \phi' = \phi \circ_\gamma \phi' \), where \( \circ_\gamma \) is Satoh [1988]’s revision operator;
2. When \( \phi = \bigvee \Delta \) and \( \phi' = \bigvee \Delta' \), \( \phi \circ \phi' = \bigvee (\delta \circ \delta' \mid (\delta, \delta') \in \Delta \ast \Delta') \);
3. When \( \phi \) and \( \phi' \) are normal S-terms and \( \phi \land \phi' \) is satisfiable, \( \phi \circ \phi' \) is \( \phi \land \phi' \) converted to a normal S-term;
4. Otherwise, \( \phi \) and \( \phi' \) are normal S-terms, and \( \phi \land \phi' \) is not satisfiable. Let \( \phi = \phi_0 \land \bigwedge_{a \in A}(K_a \phi_a \land L_a \Psi_a \land C_a \mu_a \land D_a \Lambda_a) \), and \( \phi' = \phi'_0 \land \bigwedge_{a \in A}(K_a \phi'_a \land L_a \Psi'_a \land C_a \mu'_a \land D_a \Lambda'_a) \). Then \( \phi \circ \phi' = \phi''_0 \land \bigwedge_{a \in A}(K_a \phi''_a \land L_a \Psi''_a \land C_a \mu''_a \land D_a \Lambda''_a) \) where:
   a. \( \phi''_0 = \phi_0 \land \phi'_0; \)
   b. If \( C_a \mu_a \land D_a \Lambda_a \) is satisfiable, \( \mu''_a = (\mu_a \circ_1 (\mu'_a \land \Lambda'_a)) \lor \forall \psi_a \in \Psi_a (\phi_a \land \phi'_a \land \Lambda_{a \neq b \neq a} \mu''_b) \) where \( \gamma_1 = C_a \mu'_a \land D_a \Lambda'_a \), otherwise \( \mu''_a = \mu'_a; \)
   c. \( \phi''_a = (\phi_a \land \phi'_a \land \Lambda_{a \neq b \neq a} \mu''_b) \lor \forall \psi_a \in \Psi_a (\phi_a \land \phi'_a \land \Lambda_{a \neq b \neq a} \mu''_b); \)
   d. \( \Psi''_a = \{ \phi \land \phi' \mid \phi, \phi' \in \Psi; \phi''_a \land \Lambda_{a \neq b \neq a} \mu''_b \} \cup \Psi_a \) where \( \gamma = C_a \mu''_a \land D_a \Lambda''_a \);
   e. \( \Lambda''_a = \{ \lambda \mid \lambda \in \Lambda_a \land D_a \Lambda_a \land \bigwedge_{a \in A} (K_a \phi''_a \land L_a \Psi''_a \land C_a \mu''_a \land D_a \Lambda''_a) \) 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:

1. A disjunction \( \bigvee \Delta \) of normal S-terms is d-sat if each \( \delta \in \Delta \) is d-sat;
2. A normal S-term \( \phi_0 \land \bigwedge_{a \in A}(K_a \phi_a \land L_a \Psi_a \land C_a \mu_a \land D_a \Lambda_a) \) is d-sat if it is satisfiable and each disjunction in each \( \phi_a \) or \( \mu_a \) is d-sat.

**Proposition 4.5.** An unfolded normal S-term without D modalities \( \phi = \phi_0 \land \bigwedge_{a \in A}(K_a \phi_a \land L_a \Psi_a \land C_a \mu_a \land D_a \Lambda_a) \) is satisfiable if the following hold:

1. \( \phi_0 \) is propositionally satisfiable;
2. For all \( a \in A \) and for all \( \psi_a \in \Psi_a, \phi_0 \land \bigwedge_{a \in A}(K_a \phi_a \land L_a \Psi_a \land C_a \mu_a \land D_a \Lambda_a) \) is satisfiable.

**Proof.** We construct a model \( (M, w) \) for \( \phi \). By Item 1, we create a world \( w \) where \( V(w) \) satisfies \( \phi_0 \). By Item 2, for each \( a \in A \) and \( \psi_a \in \Psi_a \), there is a KD45\( C_a \) model \( (M_{\psi_a}, w_{\psi_a}) \) satisfying \( \phi_0 \land \bigwedge_{a \in A}(K_a \phi_a \land L_a \Psi_a \land C_a \mu_a \land D_a \Lambda_a) \). We add each \( (M_{\psi_a}, w_{\psi_a}) \) to \( M \) and let \( w = \bigcup_{a \in A}w_{\psi_a} \). Then \( (M, w) \) is a KD45\( C_n \) model for \( \phi \) after calculating the transitive and Euclidean closures.

**Proposition 4.6.** Let \( \phi \) and \( \phi' \) be two d-sat unfolded normal SDNFs. Then \( \phi \circ \phi' \) is a d-sat normal SDNF, and \( \phi \circ \phi' \) is unsatisfiable. Moreover, when \( \phi \land \phi' \) is satisfiable, \( \phi \circ \phi' \) is unsatisfiable.

**Proof.** The conjunct property follows from Definition 4.3 Item 3 directly. Now we assume that \( \phi \land \phi' \) is unsatisfiable.

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

1. \( md(\phi') = 0 \). Then \( \phi \circ_\gamma \phi' \) is unsatisfiable by Satoh’s revision.
\[ md(\phi') > 0. \]
\[ \phi = \bigvee \Delta \text{ and } \phi' = \bigvee \Delta'. \]
By Definition 4.3 Item 2, \( \phi \circ \phi' = \bigvee \{ \delta \circ \delta' \mid (\delta, \delta') \in \Delta \times \Delta' \}. \)
For each \( (\delta, \delta') \in \Delta \times \Delta' \), since higher-order subformulas in \( \delta' \) 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 \( \text{md}(\phi) = n + 1 \), we consider \( \phi \) and \( \phi' \) in two cases:

- \( \phi \) and \( \phi' \) are normal S-terms. Let \( \phi'' = \phi \circ \phi' \). By Items (a), (d), and (e), we have \( \phi'' = (\phi, \phi') \models \phi_0 \models \Lambda_{a \in A} L_a \Psi_a \models \Lambda_{a \in A} D_a \mu_a \). So \( \text{md}(\phi'' \circ \phi'') = \text{md}(\phi) + \text{md}(\phi') \).

- Otherwise, we follow the proof in the base case.

Let \( \gamma \) be a modal S-term s.t. \( \phi \circ \gamma \) is d-sat and \( \phi'' = \phi \circ \phi' \). Now we prove that \( \phi' \circ \phi' \) and \( \phi'' \circ \phi'' \) are d-sat and \( \phi \circ \phi' \models \phi' \).

\[ \text{md}(\phi'') = 0. \]
Then \( \phi \circ \phi' \) is d-sat by Satoh’s revision.

\[ \text{md}(\phi'') > 0. \]
Let \( \phi = \bigvee \Delta \) and \( \phi' = \bigvee \Delta' \). We have \( \phi \circ \phi' = \bigvee \{ \delta \circ \delta' \mid (\delta, \delta') \in \Delta \times \Delta' \} \). For each \( (\delta, \delta') \in \Delta \times \Delta' \), let \( \phi = \phi_0 \) and \( \delta = \delta_0 \models \Lambda_{a \in A} \Psi_a \models \Lambda_{a \in A} D_a \mu_a \). Again by Satoh’s revision, \( \delta \circ \delta' \) is d-sat.

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 exists a new common knowledge and knowledge, we only keep them and revise them.

Definition 4.7. Let \( \phi \) and \( \phi' \) be in unfolded normal SDNF. The update of \( \phi \) with \( \phi' \), written \( \phi \circ \phi' \), is defined recursively:

1. When \( \phi \) and \( \phi' \) are propositional formulas, \( \phi \circ \phi' = \phi \circ_w \phi' \), where \( \circ_w \) is Winslett [1988]’s update operator;
2. When \( \phi = \bigvee \Delta, \phi \circ \phi' = \bigvee \delta \Delta \circ \delta' \);
3. When \( \phi \) is an S-term, and \( \phi' = \bigvee \Delta', \phi \circ \phi' = \bigvee \{ \phi \circ \phi' \mid (\phi, \phi') \in \phi \times \phi' \};
4. Otherwise, \( \phi \) and \( \phi' \) are normal S-terms, \( \phi \circ \phi' = \phi'' \) have the form as in Definition 4.3, and:

(a) \( \phi'' = \phi_0 \circ \phi_0' \);
(b) If \( C_a \mu_a \wedge \Lambda_{b \neq a} (K_b \phi_b' \wedge L_b \Psi_b') \) is satisfiable, \( \mu'' = (\mu_0 \circ_{1 \neq 0} \mu_0') \wedge \Lambda_{a \in A} (K_a \phi_a' \wedge \Lambda_{a \in A} C_a \mu_a') \wedge \Lambda_{a \in A} D_a \mu_a' \).
(c) \( \phi'' = \phi_0 \circ (\phi_0 \circ \Lambda_{b \neq a} \mu_b' \wedge \Lambda_{a \in A} \mu_a') \wedge \Lambda_{a \in A} \Lambda_{a \in A} C_a \mu_a' \wedge \Lambda_{a \in A} \mu_a' \wedge \Lambda_{a \in A} D_a \mu_a' \).
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 multiagent 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