Chapter 6

Possible and Definitive Knowledge in Email Communication

6.1 Introduction

In Chapter 5, I presented a model of the knowledge of agents during an email exchange. Here, I will study the same situation under different assumptions. Instead of focusing on common knowledge, I will distinguish between two different kinds of knowledge: potential knowledge and definitive knowledge.

When an agent receives some information via email, it is possible that he read the email and knows its content. However, one cannot be entirely sure of this because he might have overlooked the email, or he may not have received it at all due to some error in the email system. Therefore, I consider the second agent’s knowledge of the email to be potential knowledge. On the other hand, if the agent replies to an email or he forwards it, then he must have read it. In this case I consider the second agent’s knowledge to be definitive knowledge. This is relevant in for example a court case, where someone’s knowledge of an email may be uncertain if it is only known that someone sent it to him, but his knowledge of the email would be absolutely certain if he also replied to it.

The language presented here is related to the logic presented in Chapter 5. There, the language contains propositions about whether an agent was a BCC recipient of an email and common knowledge modalities, which are not present in the language presented here. Another difference between the languages is that in Chapter 5 there is only one type of knowledge while here I distinguish between potential knowledge and definitive knowledge. Also, in Chapter 5 the only email conversations that are considered are those that are actually possible in the sense that no agent sends information he did not receive. In order to enforce this, certain constraints need to be checked on each email conversation before the analysis takes place. Here, I take a much simpler approach. I do not check whether the email conversation is possible in this sense but just analyze whatever information I can get from it. The advantage of this is that it allows me
to check email conversations of which some emails are not available for analysis.

Another important advantage of the current approach is that I give a finite decision procedure. In Chapter 5 the semantics is only defined by epistemic relations on an infinite number of states. It is unclear whether the model checking of that semantics is possible in finite time, and if it is, the procedure is in any case a lot more complex.

For an overview of existing publications related to this chapter, I refer to Section 5.1.2.

6.1.1 Overview

In the next section, I start out with defining the language based on simple messages with a sender and a set of recipients. I also define a semantics that is given by epistemic relations between sets of these messages. In section 6.3 I show that this semantics can be decided without considering all (possibly infinitely many) epistemically related states. Actual emails also have a list of BCC recipients that is only known to the sender and not to the other recipients. In section 6.4 I add this feature to the semantics and show how it fits in the approach of this chapter.

6.2 The Logic of Messages

In this section I will give a language and semantics based on generic messages with a sender and a set of recipients. In the next section I will focus specifically on emails that also have BCC recipients. I do not analyze the content of messages, only their structure in terms of sender, recipients, and whether they are a forward of or a reply to previous messages. Just like in the previous chapter, I will consider the content of a basic message to be some atomic piece of information that I call a note, usually denoted with \( n \).

Let \( \mathcal{A} \) be a set of agents. I consider messages to have one of two forms:

- A basic message containing a note \( n \), represented by a tuple \( (a, n, G) \), where \( a \in \mathcal{A} \) is the sender of the message and \( G \subseteq \mathcal{A} \) is the group of recipients,

- a forward message containing another message, represented by a tuple \( (a, n.m, G) \) where \( a \in \mathcal{A} \) is the sender of the message, \( G \subseteq \mathcal{A} \) is the group of recipients, \( m \) is some other message and \( n \) is a basic note appended to the forward.

I will sometimes leave out the braces from singleton sets, writing for example \((1, n, 2)\) instead of \((1, n, \{2\})\). Given some message \( m \), \( s_m \) denotes its sender and \( r_m \) the set of its recipients. This set of recipients can be used to model both regular and CC recipients of an email. Note that a reply to a message \( m \) can be modeled as \((i, m, G)\) where \( s_m \in G \). A reply to all recipients can be modeled as
(a, m, G \{a\}) where \( a \in r_m \) and \( G = \{s_m\} \cup r_m \). For now, I will assume that the set of recipients is known to the sender and all recipients. In the next section I will also model the BCC recipients of an email.

6.2.1. **Example.** The expression \((1, n, \{2, 3\})\) stands for a message containing note \( n \) from agent 1 to agent 2 and 3. The message \((2, (1, n, \{2, 3\}), \{1, 3\})\) is a reply from agent 2 sent to 1 and 3.

When an agent sends an email to a second agent, the email is usually not read immediately. Sometimes the email is not read at all, for example when it ends up in the spam folder or when the second agent is not very diligent in reading all his emails. Therefore, the first agent cannot be sure that the second agent knows the contents of the email. On the other hand, if the first agent received a reply from the second agent then he is sure the second agent read the email. In the first case, I will say the second agent has *potential* knowledge of the email: he may have read it, but then again he may not. In the second case I will say the second agent has *definitive* knowledge of the email: since he replied on it, he must have read the email. These two kinds of knowledge are reflected in the following definition.

6.2.2. **Definition.** The logic of messages and potential and definitive knowledge \( \mathcal{L}_{PD} \) is defined as follows:

\[
\varphi ::= m \mid \neg \varphi \mid \varphi \land \varphi \mid \hat{K}_a \varphi \mid \bar{K}_a \varphi
\]

Here \( m \) is some message of the form \((b, n, G)\) or \((b, m', G)\) and \( a \in Ag \) is some agent.

The formula \( m \) expresses the fact that message \( m \) was sent. \( \hat{K}_a \varphi \) stands for potential knowledge of agent \( a \), which is achieved when agent \( a \) receives a message that implies \( \varphi \). \( \bar{K}_a \varphi \) stands for definitive knowledge of agent \( a \), which is achieved when agent \( a \) replies to or forwards a message that implies \( \varphi \). I will use the usual abbreviations \( \varphi \lor \psi \) and \( \varphi \rightarrow \psi \). Note that knowledge operators may be nested, for example \( \hat{K}_a \bar{K}_b m \) expresses that agent \( a \) definitively knows that agent \( b \) possibly knows \( m \). This may be the case if agent \( a \) and \( b \) are both recipients of \( m \) and agent \( a \) forwarded \( m \).

6.2.3. **Example.** The formula \( \hat{K}_2(1, n, \{2, 3\}) \) denotes that agent 2 possibly knows that the message \((1, n, \{2, 3\})\) was sent. This is the case whenever this message was sent, because agent 2 is a recipient of it. The formula \( \bar{K}_2(1, n, \{2, 3\}) \) denotes that agent 2 definitely knows about the message, which is the case when he replied to it.
Chapter 6. Possible and Definitive Knowledge in Email Communication

This language is interpreted on a set of messages $M$, which I will sometimes call a state. I do not bother to define an ordering between the messages in $M$. Unlike in the approach presented in Chapter 5, here I do not check whether the set of messages is ‘correct’ in the sense that for instance no agent forwards a message he did not receive. I just take whatever information is in $M$ and see what I can infer from that. This has the advantage that if not all messages are available for analysis, I can still get the most out of the messages that are available.

In order to really get all information from the messages that are available, even if they are forwards of messages that are themselves not in the set $M$, I define a closure operation:

6.2.4. Definition. Given a message $m$ or a set of messages $M$, I define its closure as follows:

\[
Cl(m) := \{ m' \mid m' \text{ is mentioned in } m \},
\]

\[
Cl(M) := \bigcup_{m' \in M} Cl(m').
\]

Just like in the previous chapter, when I say that a message $m'$ is mentioned in another message $m$ I mean that $m$ is $m'$ itself, or a forward of $m'$, or a forward of a forward of $m'$, and so on.

6.2.5. Example. If $M = \{(2, (1, n, \{2, 3\}), \{1, 3\})\}$, then

\[
Cl(M) = \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}.
\]

I will now define the semantics of the language $\mathcal{L}_{PD}$. I start out with the first three clauses.

\[
M \models m \quad \text{iff} \quad m \in Cl(M)
\]

\[
M \models \neg \varphi \quad \text{iff} \quad M \not\models \varphi
\]

\[
M \models \varphi \land \psi \quad \text{iff} \quad M \models \varphi \text{ and } M \models \psi
\]

So I consider $M$ to be evidence for the fact that some message $m$ was sent if $m$ is in the closure of $M$, that is, if some message in $M$ mentions $m$.

For the semantics of potential and definitive knowledge of some agent $a$ I will use the perspective of epistemic logic. For every agent, I will define two relations $\sim^P_a$ and $\sim^D_a$ between states, based on the messages in the states. Then I will say that an agent (potentially or definitively) knows a formula in a certain state if that formula holds in all states related to the original state.

For defining these relations $\sim^P_a$ and $\sim^D_a$ between states, I will not look at all messages in $M$ but only to those that agent $a$ sent or received and those that he sent, respectively.

6.2.6. Definition. For each agent $a$ I define two projections on a set of messages $M$, one for potential knowledge and one for definitive knowledge:

\[
\Pi_a(M) := \{ m \in M \mid a \in \{ s_m \} \cup r_m \},
\]

\[
\Delta_a(M) := \{ m \in M \mid a = s_m \}.
\]
The messages in $\Pi_a(M)$ are exactly those messages for which the fact that they were sent implies that agent $a$ has potential knowledge of this fact. Similarly, the messages in $\Delta_a(M)$ are those messages for which the fact that they were sent implies that agent $a$ has definitive knowledge of that fact.

6.2.7. Example. Let $M = \{(2, n'.(1, n, \{2, 3\}), \{1, 3\})\}$. Then

\[
\begin{align*}
\Delta_1(M) &= \emptyset, \\
\Pi_2(M) &= \{(2, (1, n, \{2, 3\}), \{1, 3\})\}, \\
\Pi_3(M) &= \{(2, (1, n, \{2, 3\}), \{1, 3\})\}, \\
Cl(M) &= \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}, \\
\Delta_1(Cl(M)) &= \{(1, n, \{2, 3\})\}, \\
\Pi_2(Cl(M)) &= \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}, \\
\Pi_3(Cl(M)) &= \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}.
\end{align*}
\]

Note that I should first take the closure of $M$ before taking the projection if I want to consider all messages mentioned in $M$. For example, if I take the projection $\Delta_1$ of $M$, I do not get the original message sent by agent 1. Only if I first take the closure $Cl(M)$ and then the projection $\Delta_1$ do I get the complete result $\{(1, n, \{2, 3\})\}$. This is correct: agent 1 has definitive knowledge of the message $(1, n, \{2, 3\})$ because he sent it.

Because one should always take the closure before taking a projection, I will define the following shorthand:

6.2.8. Definition. I define:

\[
\begin{align*}
\Pi^*_a(M) &:= Cl(\Pi_a(Cl(M))), \\
\Delta^*_a(M) &:= Cl(\Delta_a(Cl(M))).
\end{align*}
\]

Now that I have these projections in place, I can continue with defining the relations $\sim^P_a$ and $\sim^D_a$.

6.2.9. Definition. For any two states $M$ and $N$, I define

\[
\begin{align*}
M \sim^P_a N \iff & \quad \Pi^*_a(M) = \Pi^*_a(N), \\
M \sim^D_a N \iff & \quad \Delta^*_a(M) = \Delta^*_a(N).
\end{align*}
\]

With these relations in place, I define the semantics of the knowledge operators as follows:

\[
\begin{align*}
M \models K_a \varphi &\quad \text{iff} \quad N \models \varphi \text{ for all } N \text{ such that } M \sim^P_a N, \\
M \models \overline{K}_a \varphi &\quad \text{iff} \quad N \models \varphi \text{ for all } N \text{ such that } M \sim^D_a N.
\end{align*}
\]

Intuitively, this semantics can be interpreted as follows. $\Pi^*_a(M)$ is the ‘view’ that agent $a$ has on state $M$, when considering his potential knowledge, that is,
assuming that he read every message that was sent to him. On the other hand, \( \Delta^*_a(M) \) is the view of agent \( a \) on state \( M \) if one considers his definitive knowledge, so assuming that he read only the messages which he replied to or forwarded. Now two states look the same to agent \( a \) if his view on them is identical. Therefore, the agent knows something in a certain state if it holds in all states on which he has the same view as on the current state.

Note that the potential knowledge operator and the definitive knowledge operator are not each other’s dual. It is not necessarily the case that if \( M \models \neg \hat{K}_a \varphi \) then also \( M \models \bar{K}_a \varphi \), or vice versa.

6.2.10. Example. Again, let \( M = \{(2, (1, n, \{2, 3\}), \{1, 3\})\} \). Then

\[
\Delta^*_2(M) = \{(1, n, \{2, 3\}), (2, (1, n, \{2, 3\}), \{1, 3\})\}.
\]

Because \((1, n, \{2, 3\}) \in \Delta^*_2(M)\), it holds that \( M \models \bar{K}_2(1, n, \{2, 3\}) \). So in \( M \) agent 2 has definitive knowledge of the message \((1, n, \{2, 3\})\). This is correct because agent 2 sent a forward of this message.

For agent 3 this gives:

\[
\Delta^*_3(M) = \emptyset.
\]

Since \( \Delta^*_3(\emptyset) = \emptyset \), it holds that \( \emptyset \sim^D_3 M \). Because \( \emptyset \not\models (1, n, \{2, 3\}) \), \( M \not\models \bar{K}_3(1, n, \{2, 3\}) \). So agent 3 has no definitive knowledge of the message \((1, n, \{2, 3\})\). This is correct because even though agent 3 should have received the original message and the forward by agent 2, he did not reply to this messages or forward them so it is possible that these messages were lost or he did not read them.

I will not give an axiomatization of these semantics. In fact I believe that a complete axiomatization does not exist in the language that is presented here. A complete axiomatization should express the fact that the knowledge of the agents is limited: an agent does not know about a message \( m \) if he did not receive some message that mentions \( m \). There is no way to express “there is no message that mentions \( m \)” in the language \( \mathcal{L}_{PD} \). If there was only a finite number of possible messages then this might be expressed as the negation of a disjunction of messages mentioning \( m \), but since the number of possible messages is unlimited, this cannot be done. Therefore I am convinced that there is no complete axiomatization of the semantics. However, in the next section I will give a way to do model checking of this semantics.

Even though I will give no complete axiomatization, I can give a number of axioms that are valid on all sets of messages under these semantics. They show that the semantics fit the intuition of email communication and possible and definitive knowledge.
6.3. Model Checking

6.2.11. Theorem. The following axioms hold on all sets of messages:

\[(a,n.m,G) \rightarrow m \quad (6.1)\]
\[m \rightarrow \bar{K}_a m \quad (a = s_m) \quad (6.2)\]
\[m \rightarrow \bar{K}_m \quad (b \in \{s_m \cup r_m\}) \quad (6.3)\]
\[\bar{K}_a \varphi \rightarrow \bar{K}_a \varphi \quad (6.4)\]

Proof. Take some set of messages \(M\).

(6.1): Clearly, if \((a,n.m,G) \in Cl(M)\) then \(m \in Cl(M)\).

(6.2): If \(m \in Cl(M)\) and \(a = s_m\) then \(m \in \Delta_a(M)\), so \(m \in \Delta_a^*(M)\). Let \(N \sim^D_a M\). Then \(\Delta_a^*(N) = \Delta_a^*(M)\) so \(m \in \Delta_a^*(N)\). Then \(m\) is mentioned in some \(m' \in \Delta_a(Cl(N)) \subseteq Cl(N)\), so \(m \in Cl(N)\).

(6.3): The proof is similar to that for (6.2).

(6.4): I will first show that if \(M \sim^P_a N\), then \(M \sim^D_a N\). Suppose \(M \sim^P_a N\). Then \(\Pi_a^*(M) = \Pi_a^*(N)\). Take some \(m \in \Delta_a^*(M)\). Then \(m\) is mentioned in some \(m' \in \Delta_a(Cl(M))\). Then \(a = s_m\), so certainly \(a \in \{s_m\} \cup r_m\). So \(m' \in \Pi_a(Cl(M))\) and \(m' \in \Pi_a^*(M)\). But then \(m' \in \Pi_a^*(N)\). Then \(m'\) is mentioned in some \(m'' \in \Pi_a(Cl(N)) \subseteq Cl(N)\). So \(m'' \in Cl(N)\) and because \(a = s_m\), \(m' \in \Delta_a(Cl(N))\).

So because \(m'\) mentions \(m\), \(m \in \Delta_a^*(N)\). This shows that \(\Delta_a^*(M) \subseteq \Delta_a^*(N)\) and analogously I can prove the converse. So \(M \sim^D_a N\).

Now suppose \(M \models \bar{K}_a \varphi\) and let \(M \sim^P_a N\). Then \(M \sim^D_a N\) so \(N \models \varphi\). Since \(N\) was arbitrary this shows that \(M \models \bar{K}_a \varphi\). \(\square\)

6.3 Model Checking

The semantics given in the previous section are very nice in theory. However, can they also be applied in practice? Can it be decided whether a formula holds given some set of messages? It is not complicated to check formulas without epistemic operators. However, when a formula of the form \(\bar{K}_a \psi\) or \(\bar{K}_a \psi\) needs to be checked in a state \(M\), all states \(M'\) with \(\Pi_a^*(M) = \Pi_a^*(M')\) or \(\Delta_a^*(M) = \Delta_a^*(M')\) have to be checked, respectively. For all we know, there may be infinitely many of these states. In this section I circumvent this problem and I present a way to check formulas with epistemic operators.

6.3.1. Definition. With a literal I mean a message or its negation. If \(l\) is a literal, then its negation \(\bar{l}\) is \(-m\) if \(l = m\) and \(m\) if \(l = \neg m\). I call the disjunction of two literals \(l \lor l'\) a tautology iff it is of the form \(m \lor \neg m'\) (or, equivalently, \(\neg m' \lor m\)), where \(m \in Cl(m')\). I call the disjunction of \(n\) literals \(l_1 \lor ... \lor l_n\) a tautology iff there are two literals \(l_i\) and \(l_j\) occurring in that disjunction such that \(l_i \lor l_j\) is a tautology. I call the conjunction of \(n\) literals \(l_1 \land ... \land l_n\) a contradiction if there are two literals \(l_i\) and \(l_j\) occurring in that conjunction such that \(\bar{l}_i \lor \bar{l}_j\) is a tautology.
It is not hard to see that if \( l_1 \lor \ldots \lor l_n \) is a tautology then for any \( M, M \models l_1 \lor \ldots \lor l_n \). Similarly, if \( l_1 \land \ldots \land l_n \) is a contradiction then for any \( M, M \not\models l_1 \land \ldots \land l_n \).

The general idea of my approach is to define for every formula \( \varphi \) a family \( \mathcal{F}(\varphi) \) of sets of literals. Then I claim that for any model \( M, M \models \varphi \) iff for every \( F \in \mathcal{F}(\varphi) \) there is some \( l \in F \) such that \( M \models l \). One could say that \( \mathcal{F}(\varphi) \) represents a conjunctive normal form of \( \varphi \), using only literals. Because the truth value of literals is easy to check this makes checking the truth value of \( \varphi \) a lot simpler.

So how can any epistemic formula be equivalent to a conjunction of disjunctions of literals? Intuitively, for example the formula \( \hat{K}_a m \) can only be true if there was some message sent or received by agent \( a \) mentioning message \( m \). Therefore the disjunction of all such messages is a condition for the satisfaction of \( \hat{K}_a m \). But because the message sets can contain forwards of forwards of forwards etcetera up to arbitrary depth, there are infinitely many such messages. Therefore, I only consider messages up to a certain depth.

**6.3.2. Definition.** The depth \( \delta(\varphi) \) of a formula \( \varphi \) is defined as follows.

\[
\begin{align*}
\delta((a,n,G)) &= 1 \\
\delta((a,m,G)) &= 1 + \delta(m) \\
\delta(\neg \psi) &= \delta(\psi) \\
\delta(\psi_1 \land \psi_2) &= \max(\delta(\psi_1), \delta(\psi_2)) \\
\delta(\hat{K}_a \psi) &= 1 + \delta(\psi) \\
\delta(\bar{K}_a \psi) &= 1 + \delta(\psi)
\end{align*}
\]

The depth of a set of messages \( M \) is defined as \( \delta(M) := \max\{\delta(m) \mid m \in M\} \). Note that if \( m \in \text{Cl}(m') \) then \( \delta(m) \leq \delta(m') \). This implies that for any \( M, \delta(M) = \delta(\text{Cl}(M)) \).

I will construct \( \mathcal{F}(\varphi) \) with literals up to a certain depth. I will later show that for any state and formula a bound can be found on the depth of the literals that need to be considered.

**6.3.3. Definition.** Given a message \( m \), let \( \mathcal{M}_A^n(m) \) be the set of all possible messages \( m' \) of depth \( \leq n \) between the agents in \( A \) such that \( m \in \text{Cl}(m') \).

**6.3.4. Definition.** Let \( \varphi \) be a formula with \( \delta(\varphi) \leq n \). I define a family of sets of literals \( \mathcal{F}^n(\varphi) \) as follows. For \( \varphi = m \), let

\[
\mathcal{F}^n(m) := \{\{m\}\}.
\]

For \( \varphi = \neg \psi \), suppose \( \mathcal{F}^n(\psi) = \{F_1, \ldots, F_m\} \). Then

\[
\mathcal{F}^n(\neg \psi) := \{\{l_1, \ldots, l_n\} \mid l_1 \in F_1, \ldots, l_n \in F_n\}.
\]
For $\varphi = \psi_1 \land \psi_2$, let
\[
\mathcal{F}^n(\psi_1 \land \psi_2) := \mathcal{F}^n(\psi_1) \cup \mathcal{F}^n(\psi_2).
\]

For $\varphi = K_a \psi$, let
\[
\mathcal{F}^n(K_a \psi) := \{ \{ m \in F \mid a \in \{ s_m \} \cup r_m \} \cup \{ m' \in M_{Ag}^a(m) \mid m \in F, a \notin \{ s_m \} \cup r_m \} \cup \{ \neg m' \mid \neg m \in F, m' \in Cl(m), a \in \{ s_{m'} \} \cup r_{m'} \} \mid F \in \mathcal{F}^n(\psi) \}.
\]

For $\varphi = \bar{K}_a \psi$, let
\[
\mathcal{F}^n(\bar{K}_a \psi) := \{ \{ m \in F \mid a = s_m \} \cup \{ m' \in M_{Ag}^a(m) \mid m \in F, a \neq s_m, a = s_{m'} \} \cup \{ \neg m' \mid \neg m \in F, m' \in Cl(m), a = s_{m'} \} \mid F \in \mathcal{F}^n(\psi) \}.
\]

I will explain this definition step by step. The definition for $\varphi = m$ is obvious: clearly, $M \models m$ iff there is some $l \in \{ m \}$ such that $M \models l$. For $\varphi = \neg \psi$, note that $M \models \neg \psi$ if $M \not\models \psi$, so if there is $F \in \mathcal{F}^n(\psi)$ such that for any $l \in F$, $M \models \neg l$. But this is exactly the case if there is for any $F' \in \mathcal{F}^n(\neg \psi)$ some $\bar{l} \in F'$ such that $l \in F$ and $M \models \bar{l}$. For $\varphi = \psi_1 \land \psi_2$, note that the necessary condition holds for every $F_1 \in \mathcal{F}^n(\psi_1)$ and for every $F_2 \in \mathcal{F}^n(\psi_2)$ iff it holds for every $F \in \mathcal{F}^n(\psi_1) \cup \mathcal{F}^n(\psi_2)$.

For $\varphi = \bar{K}_a \psi$, I consider every literal in some member of $\mathcal{F}^n(\psi)$ separately. If it is a message $m$ such that $a \in \{ s_m \} \cup \bar{r}_m$ then $m$ is equivalent to $\bar{K}_a m$ so I preserve $m$ in some member of $\mathcal{F}^n(\varphi)$. If it is a message $m$ with $a \notin \{ s_m \} \cup \bar{r}_m$ then agent $a$ has possible knowledge of $m$ if some forward or a forward of a forward etcetera was sent by or to agent $a$. Therefore I replace $m$ by all members of $M_{Ag}^a(m)$ which were sent to or by agent $a$. Note that here I only consider messages of depth $\leq n$. For the case that the literal is the negation of a message $\neg m$, note that agent $a$ knows that $m$ was not sent if there is some message mentioned in $m$ of which he was a sender or a recipient, which was not sent. Therefore I replace $\neg m$ with these messages.

The definition for $\varphi = \bar{K}_a \psi$ is very similar to that for $\bar{K}_a \psi$, only now I only look at messages sent by agent $a$, instead of those sent or received by agent $a$.

The following theorem states that for every model and formula, I can find a number such that the satisfaction of that formula in that model can be decided by looking at a family of sets of literals of depth up to that number:

**6.3.5. Theorem.** For any set of messages $M$ and formula $\varphi$ there is a finite number $n_{M,\varphi} \geq \delta(M)$ such that for every $k \geq n_{M,\varphi}$,

\[M \models \varphi \iff \text{any } F \in \mathcal{F}^k(\varphi) \text{ contains a literal } l \in F \text{ such that } M \models l.\]
Proof. See Section 6.7. □

Now I can check whether a formula \( \varphi \) holds in a state \( M \) by only considering the literals in \( \mathcal{F}^{n_{M, \varphi}}(\varphi) \). However, I have no idea how large \( n_{M, \varphi} \) will be, and I may have to check a very large number of literals. This apparent problem quickly disappears with the following realisation. For any message \( m \) with \( \delta(m) > \delta(M) \), certainly \( M \models \neg m \). So I can remove any \( m \) with \( \delta(m) > \delta(M) \) from any member of \( \mathcal{F}^{n_{M, \varphi}}(\varphi) \). Also, any member of \( \mathcal{F}^{n_{M, \varphi}}(\varphi) \) that contains some literal \( \neg m \) with \( \delta(m) > \delta(M) \) can be removed altogether, because certainly \( M \models \neg m \).

6.3.6. Definition. Given a formula \( \varphi \) and two numbers \( n > k \), I define the restriction of \( \mathcal{F}^n(\varphi) \) to depth \( k \) as follows:

\[
\mathcal{F}^n(\varphi)|_k := \{ l \in F \mid \delta(l) \leq k \mid F \in \mathcal{F}^n(\varphi), F \text{ contains no } \neg m \text{ such that } \delta(m) > k \}.
\]

6.3.7. Theorem. For any state \( M \), formula \( \varphi \) and number \( n > \delta(M) \), there is for every \( F \in \mathcal{F}^n(\varphi) \) some \( l \in F \) such that \( M \models l \), if and only if the same holds for \( \mathcal{F}^n(\varphi)|_{\delta(M)} \).

Proof. Suppose for every \( F \in \mathcal{F}^n(\varphi) \) there is some \( l \in F \) such that \( M \models l \). Take some \( F' \in \mathcal{F}^n(\varphi)|_{\delta(M)} \) and let \( F \) be the set on which \( F' \) is based. Take \( l \in F \) such that \( M \models l \). Because \( M \models l \), either \( l = \neg m \) for some message \( m \) with \( \delta(m) > \delta(M) \) or \( \delta(l) \leq \delta(M) \). In the first case, \( F' \notin \mathcal{F}^n(\varphi) \) by definition so this is not possible. In the second case, \( l \in F' \) so the requirement is satisfied for \( F' \).

Conversely, suppose for every \( F' \in \mathcal{F}^n(\varphi)|_{\delta(M)} \) there is some \( l \in F' \) such that \( M \models l \). Take some \( F \in \mathcal{F}^n(\varphi) \). Suppose there is \( \neg m \in F \) such that \( \delta(m) > \delta(M) \). Then \( M \models \neg m \) so the requirement is satisfied for \( F \). Suppose there is no such \( \neg m \in F \). Then there is \( F' \in \mathcal{F}^n(\varphi)|_{\delta(M)} \) based on \( F \). Then there is some \( l \in F' \) such that \( M \models l \). But \( F' \subseteq F \) so then \( l \in F \) and the requirement is satisfied for \( F \). □

This theorem already reduces the collection of literals that need to be checked to those of depth \( \leq \delta(M) \). Furthermore, checking the truth value of these literals can be optimized in many ways. In many cases a disjunction of all possible messages with a certain sender or recipient will need to be checked, so a data structure that indexes the messages in a state by the agents involved in them might help a lot. All in all, I am convinced that this semantics is a promising basis for an efficient model checker of the language \( \mathcal{L}_{PD} \).

### 6.4 Blind Carbon Copy

In this section I will extend my semantics to an approach specifically tailored to emails. The difference between the earlier messages and emails is that emails
have a set of BCC recipients. These BCC recipients receive the email as well, but this fact is only known to the sender of the email.

Just like in Chapter 5 I define an email to be a construct of the form \( e = m_B \), where \( m \) is a message as defined in the previous section and \( B \subseteq Ag \) is a set of BCC recipients. I will use \( s_e, r_e \) and \( B(e) \) to denote the sender, the set of regular recipients and the set of BCC recipients of an email \( e \). So if \( e = m_B \), then \( s_e = s_m, r_e = r_m \) and \( B(e) = B \). Given an email \( e = m_B \) I will say that \( e \) is based on the message \( m \). I will identify a message without a set of BCC recipients that is a member of a set of emails \( m \in E \) with the same message with an empty set of BCC recipients: \( m_\emptyset \).

Just like in reality, the BCC recipients of a message that is forwarded are not mentioned in the forward. So a forward of an email \( m_B \) is an email of the form \( (i, m, G)_C \). Note that \( B \) is not mentioned in the forward.

I do not change the language with the addition of BCC recipients. This means that the BCC recipients are not mentioned in the logic at all. This differs from the approach presented in Chapter 5, where an extra language construct is introduced in order to make the BCC recipients explicit in the language. However, I will show that it is very well possible to analyze the agents’ knowledge in a situation with BCC recipients without mentioning them explicitly in the language.

Let \( E \) be some set of emails. Just like in the previous section, I will define the closure of the set \( E \). However, this becomes a bit more complicated because I have to take the BCC recipients into account. The following example shows how this complicates matters.

6.4.1. Example. Suppose Alice sends an email to Bob, with a BCC to Carol. Then Bob does not know that Carol received the message. However, now Carol sends a reply to this email to both Alice and Bob. Then Bob gets to know that Carol received the original email. By sending the reply, Carol revealed her identity as a BCC recipient.

Formalizing this example, let agent 1 be Alice, agent 2 be Bob and agent 3 be Carol. The original email would be formalized as \( (1, n, 2)_3 \) and the reply by Carol as \( (3, (1, n, 2), \{1, 2\}) \). From the second email it can be deduced that 3 was a BCC recipient of the first email. Therefore, the closure of the set \( \{(3, (1, n, 2), \{1, 2\})\} \) should include the message \((1, n, 2)\) with a BCC to agent 3, even though this BCC recipient is not mentioned explicitly.

In order to define the closure, I first compute for each message its BCC recipients, according to some set of emails.

\[
B(m, E) := \{ \ b \in Ag \setminus (\{s_m\} \cup r_m) \mid \\
\exists C : m_C \in E \text{ and } b \in C \text{ or} \\
\exists G : (b, m, G) \text{ is mentioned in some } e \in E \}
\]

So an agent \( b \) is in \( B(m, E) \) if it can be deduced from the set \( E \) that \( b \) was a BCC recipient of \( E \). This is the case if there is some email \( m_C \) in \( E \) that shows that \( b \)
was a BCC recipient because \( b \in C \), or if \( b \) forwarded \( m \) to some other group of agents.

Using this definition I define the closure of a set of emails as follows:

\[
Cl(E) := \{ m_{B(m,E)} \mid \exists e \in E : m \text{ is mentioned in } e \} 
\]

So I take any message that is mentioned in some email in \( E \), and add the BCC recipients that can be deduced from \( E \).

Now that I have defined the closure of a set of emails, I should also define the projections for the agent’s knowledge. In order to simplify the definitions, I first define a new notion of union that takes BCC recipients into account:

\[
E \cup^* E' := \{ m_B \in E \mid \neg \exists B' : m_{B'} \in E' \} \cup \\
\{ m_{B'} \in E' \mid \neg \exists B : m_B \in E \} \cup \\
\{ m_{B \cup B'} \mid m_B \in E, m_{B'} \in E' \} 
\]

This notion of union is designed to make sure that if a message occurs in both \( E \) and \( E' \) with different BCC recipients, the BCC recipients are joined in one set instead of including the message twice.

I continue with the projection for potential knowledge. In this definition I carefully make out which BCC recipients of each email are visible to the agent. If the agent is the sender of the email, all BCC recipients are visible to him. If he is a regular recipient and not a BCC recipient, then none are visible. If he is a BCC recipient himself, then he only knows that he himself is a BCC recipient and he does not know the identity of any other BCC recipients.

\[
\Pi_a(E) := \{ m_B \in E \mid a = s_m \} \cup^* \\
\{ m_\emptyset \mid \exists B : m_B \in E, a \in r_m \} \cup^* \\
\{ m_{(a)} \mid \exists B : m_B \in E, a \in B \} 
\]

Note that in this definition I ignore what the agent can deduce about the BCC recipients of an email by looking at forwards sent by those BCC recipients. That is why, after applying a projection, I will always take the closure of the result.

Now I turn to the projection for definitive knowledge. This is quite simple: since I only look at emails where the agent is the sender, all BCC recipients are visible to him so they are all preserved by the projection.

\[
\Delta_a(E) := \{ m_B \in E \mid a = s_m \} 
\]

Again, I define a shorthand for taking the projection and the closure:

\[
\Pi^*_a(E) := Cl(\Pi_a(Cl(E))), \\
\Delta^*_a(E) := Cl(\Delta_a(Cl(E))).
\]
Note that if one views a message as an email with an empty set of BCC recipients, then the new definitions for closure and projections coincide with the ones given in Section 6.2.

The semantics of the language on sets of emails is defined in the same way as for sets of messages. I define that $E \sim_P i E'$ iff $\Pi^*_i(E) = \Pi^*_i(E')$, and similarly for $\sim_D$ and $\Delta^*_i$. Then the semantics for sets of emails is given by:

\[
\begin{align*}
E \models m & \iff \exists B : m_B \in \text{Cl}(E) \\
E \models \neg \varphi & \iff E \not\models \varphi \\
E \models \varphi \land \psi & \iff E \models \varphi \text{ and } E \models \psi \\
E \models \hat{K}_a \varphi & \iff E' \models \varphi \text{ for all } E' \text{ such that } E \sim_P a E' \\
E \models \bar{K}_a \varphi & \iff E' \models \varphi \text{ for all } E' \text{ such that } E \sim_D a E'
\end{align*}
\]

The following example shows how this semantics works out.

6.4.2. Example. Suppose agent 1 sends an email to agent 2, with a BCC to 3 and 4. Then agent 3 forwards this email to agent 2. I formalize this as follows:

\[
\begin{align*}
E &= \{(1, n, 2)_{\{3, 4\}}, (3, (1, n, 2), 2)\} \\
\text{Cl}(E) &= E.
\end{align*}
\]

In order to analyze the knowledge of agent 3, I compute the projections $\Pi^*_3(E)$ and $\Delta^*_3(E)$:

\[
\begin{align*}
\Pi^*_3(E) &= \{(1, n, 2)_3, (3, (1, n, 2), 2)\} \\
\Delta^*_3(E) &= \{(1, n, 2)_3, (3, (1, n, 2), 2)\}.
\end{align*}
\]

Because $(1, n, 2)_3 \in \Pi^*_3(E)$, it holds that $E \models \hat{K}_3(1, n, 2)$. This was to be expected: agent 3 possibly knows about the email $(1, n, 2)$ because he received a BCC of it.

Because $(3, (1, n, 2), 2) \in \Delta_3(\text{Cl}(E))$ it holds that $(1, n, 2)_3 \in \Delta^*_3(E)$ and $E \models \hat{K}_E(1, n, 2)$. Intuitively speaking, agent 3 definitively knows about $(1, n, 2)$ because he sent a forward of it.

Now I consider the knowledge of agent 4 about agent 3’s knowledge:

\[
\begin{align*}
\Pi^*_4(E) &= \{(1, n, 2)_{\{3, 4\}}\} \\
\Pi^*_3(\Pi^*_4(E)) &= \emptyset.
\end{align*}
\]

Because $(1, n, 2) \not\in \Pi^*_3(\Pi^*_4(E))$, it holds that $E \models \neg \hat{K}_4 \hat{K}_3(1, n, 2)$. So agent 4 does not know that 3 knows about the first email. This is because agent 4 does not know that 3 was also a BCC recipient. However, agent 1 does know this, as is shown by the following projections:

\[
\begin{align*}
\Pi^*_1(E) &= \{(1, n, 2)_{\{3, 4\}}\} \\
\Pi^*_3(\Pi^*_1(E)) &= \{(1, n, 2)_3\} \\
\Delta^*_3(\Pi^*_1(E)) &= \emptyset.
\end{align*}
\]
Because agent 1 is the sender of the first email, agent 3 is preserved as a BCC recipient in the projection $\Pi_1^*(E)$. Then when I take the potential knowledge projection for agent 3 the original message is again preserved so $(1, n, 2)_{(3)} \in \Pi_3^*(\Pi_1^*(E))$. Therefore, $E \models K_1 \bar{K}_3(1, n, 2)$.

However, the forward by agent 3 is not in $\Pi_1^*(E)$, nor is any other email sent by agent 3, so when I take the definitive knowledge projection for agent 3 then the result is the empty set: $\Delta_3^*(\Pi_1^*(E)) = \emptyset$. Therefore, $(1, n, 2) \notin \Delta_3^*(\Pi_1^*(E))$ and $E \models \neg K_1 K_3(1, n, 2)$: agent 1 does not know that agent 3 definitively knows about the original message, because he did not receive agent 3’s forward. This means that agent 1 cannot be entirely sure that his email actually reached agent 3. Agent 2, on the other hand, did receive agent 3’s forward. Let me consider the projections for agent 2:

$$
\Pi_2(Cl(E)) = \{(1, n, 2), (3, (1, n, 2), 2), (3, (1, n, 2), 2)\},
$$

$$
\Pi_2^*(E) = \{(1, n, 2), (3, (1, n, 2), 2), (3, (1, n, 2), 2)\},
$$

$$
\Pi_3(Cl(\Pi_2(Cl(E)))) = \{(1, n, 2), (3, (1, n, 2), 2), (3, (1, n, 2), 2)\},
$$

$$
\Pi_3^*(\Pi_2^*(E)) = \{(1, n, 2), (3, (1, n, 2), 2), (3, (1, n, 2), 2)\},
$$

$$
\Delta_3^*(\Pi_2^*(E)) = \{(1, n, 2), (3, (1, n, 2), 2), (3, (1, n, 2), 2)\}.
$$

When I take the projection $\Pi_2(Cl(E))$, then initially no BCC recipients of $(1, n, 2)$ are preserved because as a regular recipient, agent 2 does not know the identity of the BCC recipients. However, because agent 3 forwarded the email $(1, n, 2)$, agent 2 knows that agent 3 was a BCC recipient. This is reflected by the fact that in the closure $Cl(\Pi_2(Cl(E)))$, agent 3 is a BCC recipient of $(1, n, 2)$. This shows exactly why it is important to apply the closure after applying a projection.

Because 3 is a BCC recipient of $(1, n, 2)$ in $\Pi_2^*(E)$, the message $(1, n, 2)$ is preserved in $\Pi_3^*(\Pi_2^*(E))$, and because of that

$$
E \models K_2 K_3(1, n, 2).
$$

Something even stronger can be said: because

$$
(3, (1, n, 2), 2) \in \Pi_2^*(E)
$$

it also holds that

$$
(1, n, 2)_{(3)} \in \Delta_3^*(\Pi_2^*(E)),
$$

which means that

$$
E \models \bar{K}_2 K_3(1, n, 2).
$$

Intuitively, agent 2 knows that agent 3 definitely knows about the first message because he received the forward by agent 3.
6.5 Model Checking with BCC

Now that I have extended the semantics with BCC, I can ask again the question of whether it is possible to do model checking of the semantics in finite time. I think this is certainly possible.

When a message $m$ has to be sent with a set of BCC recipients $B$, this can be done as an email $m_B$. But another option is for the sender of $m$ to first send the message $m$, and then send a forward $(s_m, m, b)$ for every $b \in B$. This is the simulation I already mentioned in Chapter 5. I will make this formal as follows.

6.5.1. Definition. Given a message $m$, let $\beta(m)$ be the message constructed from $m$ by replacing all occurrences in $m$ of some message $(b, m', G)$ where $b \not\in \{s_m\} \cup r_m$ by the message $(b, (s_m, m', b), G)$. Similarly, for a formula $\varphi$, $\beta(\varphi)$ is constructed by replacing all occurrences of messages $m$ in $\varphi$ by $\beta(m)$.

So if some agent forwarded a message of which he was not the sender or a regular recipient, in which case he must have been a BCC recipient, then I replace the forward by a forward of a forward by the sender of the first message. Using this transformation $\beta$ I can transform a set of emails to a set of messages as follows:

6.5.2. Definition. Given a set of emails $E$, I construct $\beta(E)$ by replacing each email $m_B$ with the messages in

$$\{m\} \cup \{(s_m, m, b) \mid b \in B\}$$

and subsequently replacing every message $m$ in the result by $\beta(m)$.

This transformation can be interchanged with the application of the projection.

6.5.3. Lemma (22). For any set of emails $E$ and any agent $a$,

$$\beta(\Pi_a(E)) = \Pi_a(\beta(E))$$

Similarly for $\Delta_a$.

Proof. Take some $m \in \beta(\Pi_a(E))$.

Suppose $m = \beta(m^1)$ for some $m^1_B \in \Pi_a(E)$. Then there is $m^2_C \in \Pi_a(Cl(E))$ mentioning $m^1$. Then $a \in \{s_{m^2} \cup r_{m^2} \cup C\}$ and $m^2$ is mentioned in some $m^3_D \in E$. Then $\beta(m^1) \in \beta(E)$ and $\beta(m^2)$ is mentioned in $\beta(m^3)$ so $\beta(m^3) \in Cl(\beta(E))$. Suppose $a \in s_{m^2} \cup r_{m^2}$. Then $\beta(m^2) \in \Pi_a(Cl(\beta(E)))$ and because $\beta(m^1)$ is mentioned in $\beta(m^2)$ then $m \in \Pi_a(\beta(E))$. Suppose $a \in C$. Then $a \in B(m^2, E)$. Then either there is some set $C'$ such that $a \in C'$ and $m^2_{C'} \in E$ or there is some group $G$ such that $(a, m^2, G) \in E$. Suppose the first case. Then $(s_{\beta(m^2)}, \beta(m^2), a) \in \beta(E)$.
so \((s_{\beta(m^2)}, \beta(m^2), a) \in \Pi_a(Cl(\beta(E)))\) and \(\beta(m^1) \in \Pi_a^*(\beta(E))\). Suppose the second case. Then \((a, (s_{m^2}, \beta(m^2), a), G) \in \beta(E)\) so \((a, (s_{m^2}, \beta(m^2), a), G) \in \Pi_a(Cl(\beta(E)))\) and \(\beta(m^1) \in \Pi_a^*(\beta(E))\).

Suppose \(m = (s_{m'}, \beta(m'), b)\) for some \(m'_B \in \Pi_a(E)\) with \(b \in B\). Then \(b \in B(m', \Pi_a(Cl(E)))\). Suppose there is \(C\) such that \(b \in C\) and \(m'_C \in \Pi_a(Cl)(E)\). Then \(a' \in \{s_{m'}\} \cup \{b\}\) and \(b \in B(m', E)\). Suppose there is \(D\) with \(b \in D\) and \(m'_D \in E\). Then \((s_{m'}, \beta(m'), b) \in \beta(E)\). Since \(a \in \{s_{m'}\} \cup \{b\}\) then \((s_{m'}, \beta(m'), b) \in \Pi_a^*(\beta(E))\). Suppose there is no such \(D\). Then \((b, m', G)\) is mentioned in \(Cl(E)\) for some group \(G\). Then \((b, (s_{m'}, \beta(m'), b), G) \in Cl(\beta(E))\) and because \(a \in \{s_{m'}\} \cup \{b\}\) then \((b, (s_{m'}, \beta(m'), b), G) \in \Pi_a(Cl(\beta(E)))\) so \((s_{m'}, \beta(m'), b) \in \Pi_a^*(\beta(E))\). Now suppose there is no such \(C\). Then there is \(G'\) such that \((b, m', G')\) is mentioned in \(\Pi_a(Cl(E))\). By a similar reasoning as above then \((b, (s_{m'}, \beta(m'), b), G') \in \Pi_a(Cl(\beta(E)))\) so \(m \in \Pi_a^*(\beta(E))\).

For the converse, take some \(m \in \Pi_a(\beta(E))\). Then there is some \(m' \in \Pi_a(Cl(\beta(E)))\) mentioning \(m\). Then \(a \in \{s_{m'}\} \cup r_{m'}\) and \(m'\) is mentioned in some \(m'' \in \beta(E)\). Suppose \(m'' = \beta(m^1)\) for some \(m'_B \in E\). Then there is some \(m^2\) mentioned in \(m''\) such that \(m' = \beta(m^2)\). Then \(a \in \{s_{m'}\} \cup r_{m^2}\) so \(m''_B \in \Pi_a(Cl(E))\) for some \(C\). Then there is some \(m^3\) mentioned in \(m^2\) such that \(m = \beta(m^3)\). So then there is some \(D\) such that \(m''_D \in \Pi_a(E)\) and \(m \in \beta(\Pi_a^*(E))\).

Now suppose \(m'' = (s_{m^3}, \beta(m^3), b)\) for some \(m'_B \in E\) with \(b \in B\). Then there is \(m^2\) mentioned in \(m^1\) such that \(m' = \beta(m^2)\). Because \(a \in \{s_{m'}\} \cup r_{m'}\) then there is some \(C\) such that \(m''_B \in \Pi_a(Cl(E))\). Then there is some \(m^3\) mentioned in \(m^2\) such that \(m = \beta(m^3)\). Then \(m_3 \in \Pi_a^*(E)\) so \(m \in \beta(\Pi_a^*(E))\).

In Chapter 5, two differences between the original email \(m_B\) and the simulation with forwards are mentioned. The first one is that every agent in \(B\) receives a forward of \(m\) instead of \(m\) itself. This syntactic difference is preserved when the agents in \(B\) forward the message or the forward of the message. However, it does not influence the agent’s knowledge about \(m\) or about each other’s knowledge of \(m\).

The second difference is that when an agent is a BCC recipient, and he does not reveal this fact to others by sending a forward, then he knows that the other agents do not know he received the message. This is because the BCC recipients are not included in forwards of the original message. On the other hand, if the sender of the message sent a separate forward to the former BCC recipient then the sender may forward this forward to other agents, thereby informing them that the former BCC recipient knows about the message. In other words, the BCC feature makes the fact that these agents receive the message a secret, while a separate forward does not.

This may seem contradictory to Lemma 6.5.3 because it seems that that result implies that the transformation \(\beta\) does not influence the knowledge relations. This apparent contradiction is caused by the fact that it is possible that there are two sets of emails \(E\) and \(E'\) such that \(\Pi_a(\beta(E)) = \Pi_a(\beta(E'))\) while \(\Pi_a(E) \neq \Pi_a(E')\).
Then, clearly $\beta(E) \sim_a \beta(E')$ while $E \not\sim_a E'$. The following example shows how this can occur.

6.5.4. Example. Consider the following sets of emails:

$$E_1 : \{(1, n, 2)_3\}$$
$$E_2 : \{(1, n, 2), (1, (1, n, 2), 3), (1, (1, (1, n, 2), 3), 2)\}$$

Then $E_1 \not\sim_3 E_2$. Note that $E_2 \models \hat{K}_2\hat{K}_3(1, n, 2)$ while $E_1 \not\models \hat{K}_2\hat{K}_3(1, n, 2)$. In fact, there is no $E'$ such that $E_1 \sim_3 E'$ and $E' \models \hat{K}_2\hat{K}_3(1, n, 2)$, so $E_1 \models \hat{K}_3\neg\hat{K}_2\hat{K}_3(1, n, 2)$.

Now look at the transformed sets of emails:

$$\beta(E_1) = \{(1, n, 2), (1, (1, n, 2), 3)\}$$
$$\beta(E_2) = \{(1, n, 2), (1, (1, n, 2), 3), (1, (1, (1, n, 2), 3), 2)\}$$

I have $\beta(E_1) \sim_3 \beta(E_2)$. However, $\beta(E_2) \models \hat{K}_2\hat{K}_3(1, n, 2)$ so

$$\beta(E_3) \not\models \hat{K}_3\neg\hat{K}_2\hat{K}_3(1, n, 2).$$

This shows that even though the $\beta$ transformation gives a good simulation of a set of emails without using BCC, it is not perfect. In other words, BCC really adds something new from an epistemic perspective. Therefore, for deciding the model checking problem with BCC it is not enough to simply translate the sets of emails to sets of messages and handle the model checking as in Section 6.3.

A better way to solve the model checking problem would be to adapt the definition of $F^n(\varphi)$ from the previous section for the case with BCC recipients. This new definition of $F^n(\varphi)$ will have the same function as for the semantics without BCC. However, now the sets in $F^n(\varphi)$ will not only contain literals, but also constructs of the form $m_j$ and negations of these constructs. Here $m$ is a message and $j$ is a single agent. The satisfaction of these constructs in a state is defined as follows:

$$E \models m_b \text{ iff there is } B \subseteq Ag : m_B \in E, b \in B.$$

Note that I do not want to extend the logic with this new construct $m_j$. I only use it to decide the truth value of the formulas.

I continue with the new definition of $F^n(\varphi)$.

6.5.5. Definition. Let $\varphi$ be a formula with $\delta(\varphi) \leq n$. I define a family of sets of literals $F^n(\varphi)$ as follows. For $\varphi = m$, let

$$F^n(m) := \{\{m\}\}$$

For $\varphi = \neg \psi$, suppose $F^n(\psi) = \{F_1, ..., F_n\}$. Then

$$F^n(\neg \psi) := \{\{l_1, ..., l_n\} \mid l_1 \in F_1, ..., l_n \in F_n\},$$
where \( l \) is given by \(-m\) if \( l = m \) and \( m \) if \( l = -m \). For \( \varphi = \psi_1 \land \psi_2 \), let

\[
\mathcal{F}^n(\psi_1 \land \psi_2) := \mathcal{F}^n(\psi_1) \cup \mathcal{F}^n(\psi_2).
\]

For \( \varphi = K_a\psi \), let

\[
\mathcal{F}^n(K_a\psi) := \left\{ \bigcup_{l \in F} F^n_{K_a}(l) \mid F \in \mathcal{F}^n(\psi) \right\},
\]

where \( F^n_{K_a}(l) \) is given by

\[
\begin{align*}
\{m\} & \quad \text{if} \quad l = m, a \in \{s_m\} \cup r_m, \\
\{m' \in M^a_\phi(m) \mid a \in \{s_{m'}\} \cup r_{m'}\} & \quad \text{if} \quad l = m, a \notin \{s_m\} \cup r_m, \\
\{m'_a \mid m' \in M^a_\phi(m)\} & \quad \text{if} \quad l = -m, \\
\{-m' \mid m' \in \text{Cl}(m), a \in \{s_{m'}\} \cup r_{m'}\} & \quad \text{if} \quad l = m_b, a \in \{s_m\} \cup \{b\}, \\
\{m_b\} & \quad \text{if} \quad l = -m_b, a \in \{s_m\} \cup \{b\}, \\
\{-m_b\} & \quad \text{if} \quad l = m, a \notin \{s_m\} \cup \{b\}. \\
\end{align*}
\]

For \( \varphi = K_a\psi \), let

\[
\mathcal{F}^n(K_a\psi) := \left\{ \bigcup_{l \in F} F^n_{K_a}(l) \mid F \in \mathcal{F}^n(\psi) \right\},
\]

where \( F^n_{K_a}(l) \) is given by

\[
\begin{align*}
\{m\} & \quad \text{if} \quad l = m, a = s_m, \\
\{m' \in M^a_\phi(m) \mid a = s_{m'}\} & \quad \text{if} \quad l = m, a \neq s_m, \\
\{-m' \mid m' \in \text{Cl}(m), a = s_{m'}\} & \quad \text{if} \quad l = -m, \\
\{m_b\} & \quad \text{if} \quad l = m_b, a = s_m, \\
\{m' \in M^a_\phi((b, m, G)) \mid a = s_{m'}\} & \quad \text{if} \quad l = m_b, a \neq s_m, \\
\{-m_b\} & \quad \text{if} \quad l = -m_b, a = s_m, \\
\{-m_b\} & \quad \text{if} \quad l = -m_b, a \neq s_m. \\
\end{align*}
\]

The first three clauses of this definition are identical to the definition for the semantics without BCC. The difference is in the knowledge operators. Suppose \( \varphi = K_a\psi \). Again, I consider each literal in some member of \( \mathcal{F}^n(\psi) \) separately.

If \( l = m \) and \( a \in \{s_m\} \cup r_m \) then \( m \) implies \( K_a m \) so I preserve \( m \).

If \( l = m \) and \( a \notin \{s_m\} \cup r_m \) then \( a \) potentially knows \( m \) iff he sent or received some message in \( M^a_\phi(m) \), or if he was a BCC recipient of such a message.

If \( l = -m \) then \( a \) potentially knows \( m \) iff there is some message in \( \text{Cl}(m) \) of which he was the sender or a recipient which was not sent.
If \( l = m_b \) or \( l = \neg m_b \) and \( a \in \{s_m\} \cup \{b\} \) then \( a \) certainly knows whether \( b \) was a BCC recipient of \( m \) so I preserve \( m_b \) or \( \neg m_b \).

If \( l = m_b \) and \( a \not\in \{s_m\} \cup \{b\} \) then \( a \) knows that \( b \) was a BCC recipient of \( m \) if \( a \) has received a forward \((b,m,G)\) of \( m \) by \( b \) or \( a \) is the sender, recipient or BCC recipient of some message in \( M^*_G((b,m,G)) \) for such a \((b,m,G)\).

If \( l = \neg m_b \) and \( a \not\in \{s_m\} \cup \{b\} \) then \( a \) knows \( b \) was not a BCC recipient of \( m \) if \( a \) knows that \( m \) was not sent, which is the case when some message in \( Cl(m) \) of which \( a \) is a sender or a recipient was not sent.

For the case that \( \varphi = K_a \psi \), I also consider each literal separately.

If \( l = m \) and \( a = s_m \) I preserve \( m \). If \( a \not= s_m \) then \( a \) has definitive knowledge of \( m \) if he is the sender of some message in \( M^*_G(m) \).

If \( l = \neg m \) then \( a \) has definitive knowledge of \( l \) if \( a \) is the sender of some message in \( Cl(m) \) that was not sent.

If \( l = m_b \) and \( a = s_m \) then I preserve \( m_b \). If \( a \not= s_m \) then \( a \) definitively knows that \( b \) was a BCC recipient if he sent some message in \( M^*_G((b,m,G)) \), for some group of agents \( G \).

If \( l = \neg m_b \) and \( a = s_m \) then I preserve \( \neg m_b \). If \( a \not= s_m \) then \( a \) definitively knows \( b \) was not a BCC recipient of \( m \) if he definitively knows that \( m \) was not sent, which is the case if he was the sender of some message in \( Cl(m) \) that was not sent.

I am convinced that the equivalent of Theorem 6.3.5 and 6.3.7 also hold for the case with BCC recipients.

**6.5.6. Conjecture.** For any set of messages \( M \) and formula \( \varphi \) there is a finite number \( n_{M,\varphi} \geq \delta(M) \) such that for every \( k \geq n_{M,\varphi} \), \( M \models \varphi \) iff any \( F \in F^k \varphi \) contains a literal \( l \in F \) such that \( M \models l \).

**6.5.7. Conjecture.** For any state \( M \), formula \( \varphi \) and number \( n > \delta(M) \), there is for every \( F \in F^n(\varphi) \) some \( l \in F \) such that \( M \models l \), if and only if the same holds for \( F^n(\varphi)|\delta(M) \).

This would give a way to decide the semantics for the case with BCC recipients.

### 6.6 Conclusion

I have presented a logic that reasons about the knowledge of agents after a certain collection of messages or emails have been sent. Specifically I have focussed on the difference between having received a message and having replied to it. In the first case, it is not sure that the recipient has received the email in good order and also read it. In the second case it is. I have given a semantics based on the epistemic logic perspective, that is based on relations between states given by sets of messages or emails. The difference between messages and emails is that
the first only have a public list of recipients, while the second also have a secret list of BCC recipients.

Since the number of related states may be infinite, this perspective does not immediately give a way to decide the truth value of the formulas in finite time. Therefore I presented a way to decide each formula by looking at the truth value of certain literals. This decision procedure is proved correct for the case of messages. I also give a definition of this procedure for emails.

All in all I have presented a strong basis for a formal model checker that can be applied to sets of messages or emails in order to analyze who knows what in any situation where messages or emails are sent.
6.7 Proof of Theorem 6.3.5

I first state some facts that I will implicitly use throughout this section. I omit their proof, but they follow easily from the definition of closure and the semantics. For any two sets of messages $M$ and $N$ and any agent $a \in Ag$, the following hold:

- $Cl(Cl(M)) = Cl(M)$,
- $Cl(M \cup N) = Cl(M) \cup Cl(N)$,
- If $N \subseteq M$ then $Cl(N) \subseteq Cl(M)$,
- If $N \subseteq Cl(M)$ then $Cl(N) \subseteq Cl(M)$,
- $M \sim^P_a Cl(M)$ and $M \sim^D_a Cl(M)$,
- If $M \sim^P_a N$ and $M \models \bar{K}_a\varphi$ then $N \models \bar{K}_a\varphi$,
- If $M \sim^D_a N$ and $M \models \bar{K}_a\varphi$ then $N \models \bar{K}_a\varphi$.

6.7.1. Lemma. For any set of messages $M$, $\Pi^*_a(M) \subseteq Cl(M)$. Similarly for $\Delta^*_a$.

Proof. Suppose $m \in \Pi^*_a(M) = Cl(\Pi_a(Cl(M)))$. Then there is $m' \in \Pi_a(Cl(M))$ that mentions $m$. Then $m' \in Cl(M)$, so because $m'$ mentions $m$,

$$m \in Cl(Cl(M)) = Cl(M).$$

So $\Pi^*_a(M) \subseteq Cl(M)$. □

6.7.2. Lemma. For any two sets of messages $M$ and $N$, $M \sim^P_a N$ iff

$$\Pi_a(Cl(M) \setminus Cl(N)) = \emptyset$$

and

$$\Pi_a(Cl(N) \setminus Cl(M)) = \emptyset.$$ 

Similarly for $\sim^D_a$ and $\Delta_a$.

Proof. Take two sets of messages $M$ and $N$ and suppose $M \sim^P_a N$. For the sake of contradiction, suppose one of the sets mentioned above is non-empty. Without loss of generality, suppose there is some $m \in \Pi_a(Cl(M) \setminus Cl(N))$. Then $a \in \{s_m \cup r_m\}$ and $m \in Cl(M)$ and $m \notin Cl(N)$. Then $m \in \Pi_a(Cl(M))$ so $m \in \Pi^*_a(M)$. But because $M \sim^P_a N$, $\Pi^*_a(M) = \Pi^*_a(N)$ so then $m \in \Pi^*_a(N)$. But by Lemma 6.7.1 $\Pi^*_a(N) \subseteq Cl(N)$, so $m \in Cl(N)$. But I already knew that $m \notin Cl(N)$. This is a contradiction, so such $m$ cannot exist and these sets must be empty.
For the converse I use contraposition. Suppose \( M \not\sim^P_a N \). Then \( \Pi_a^*(M) \neq \Pi_a^*(N) \). Without loss of generality, take \( m \in \Pi_a^*(M) \setminus \Pi_a^*(N) \). Then there is \( m' \in \Pi_a^*(\text{Cl}(M)) \) that mentions \( m \). Then \( a \in \{s_{m'} \cup r_m\} \) and \( m' \in \text{Cl}(M) \).

Suppose \( m' \in \text{Cl}(N) \). Then \( m' \in \Pi_a^*(\text{Cl}(N)) \) so because \( m' \) mentions \( m \), \( m \in \Pi_a^*(N) \). This contradicts my assumption, so I conclude that \( m' \not\in \text{Cl}(N) \). So then \( m' \in \text{Cl}(M) \setminus \text{Cl}(N) \). Then because \( a \in \{s_{m'} \cup r_m\}, m' \in \Pi_a^*(\text{Cl}(M) \setminus \text{Cl}(N)) \). So \( \Pi_a^*(\text{Cl}(M) \setminus \text{Cl}(N)) \neq \emptyset \).

6.7.3. Lemma. For any set of messages \( M \) and any message \( m \in \text{Cl}(M) \), \( M \models \bar{K}_a m \) iff \( m \in \Pi_a^*(M) \). Similarly for \( \bar{K}_a \) and \( \Delta_a^* \).

Proof. Suppose \( m \in \Pi_a^*(M) \). Then for any \( M' \) such that \( M \sim^P_a M' \), \( m \in \Pi_a^*(M') \subseteq \text{Cl}(M') \) so \( M' \models m \). So \( M \models \bar{K}_a m \). Conversely, suppose \( M \models \bar{K}_a m \). Let \( M' = \text{Cl}(M) \setminus \{m' \in \text{Cl}(M) \mid m' \text{ mentions } m\} \). Clearly, \( M' \not\models m \) so \( M \not\sim^P_a M' \).

Note that \( \text{Cl}(M') \setminus \text{Cl}(M) = \emptyset \) and \( \text{Cl}(M) \setminus \text{Cl}(M') = \{m' \in \text{Cl}(M) \mid m' \text{ mentions } m\} \). So then by Lemma 6.7.2, there is \( m' \in \Pi_a^*(\text{Cl}(M) \setminus \text{Cl}(M')) \). Then \( m' \text{ mentions } m \) and \( a \in \{s_{m'} \cup r_m\} \). Then \( m' \in \Pi_a^*(M) \) and \( m \in \Pi_a^*(M) \).

6.7.4. Lemma. For any set of messages \( M \) and message \( m \), either \( M \models \bar{K}_a \neg m \) or \( M \not\sim^D_a M \cup \{m\} \). Similarly for \( \bar{K}_a \) and \( \sim^D_a \).

Proof. Suppose \( M \not\sim^D_a M \cup \{m\} \). Then by Lemma 6.7.2 either \( \Pi_a^*(\text{Cl}(M \cup \{m\}) \setminus \text{Cl}(M)) \neq \emptyset \) or \( \Pi_a^*(\text{Cl}(M \cup \{m\}) \setminus \text{Cl}(M)) \neq \emptyset \). Clearly, \( \text{Cl}(M) \setminus \text{Cl}(M) \cup \{m\}) = \emptyset \). So I can take some \( m' \in \Pi_a^*(\text{Cl}(M \cup \{m\}) \setminus \text{Cl}(M)) \). Then \( m' \in \text{Cl}(M \cup \{m\}) \) and \( m' \not\in \text{Cl}(M) \). So \( m' \in \text{Cl}(\{m\}) \).

Take some \( M' \) such that \( M \sim^P_a M' \). Suppose \( m \in \text{Cl}(M') \). Then because \( m' \in \text{Cl}(\{m\}) \), \( m' \in \text{Cl}(M') \) and because \( a \in \{s_{m'} \cup r_m\} \), \( m' \in \Pi_a^*(\text{Cl}(M')) \). Then also \( m' \in \Pi_a^*(M') \). But \( m' \not\in \text{Cl}(M') \). \( M \sim^P_a M' \) so \( \Pi_a^*(M') = \Pi_a^*(M) \) and \( m' \in \Pi_a^*(M) \). But by Lemma 6.7.1 \( \Pi_a^*(M) \subseteq \text{Cl}(M) \), so \( m' \in \text{Cl}(M) \). But we already saw that \( m' \not\in \text{Cl}(M) \). This is a contradiction so \( m \not\in \text{Cl}(M') \) and \( M' \not\models m \). But \( M' \) was chosen arbitrarily, so \( M \models \bar{K}_a \neg m \).

The proof for \( \bar{K}_a \) and \( \sim^D_a \) is analogous.

6.7.5. Lemma. For any set of messages \( M \) and message \( m \), either \( M \models \bar{K}_a m \) or \( M \not\sim_a^P \text{Cl}(M) \setminus \{m' \in \text{Cl}(M) \mid m' \text{ mentions } m\} \).

Proof. Let \( N = \{m' \in \text{Cl}(M) \mid m' \text{ mentions } m\} \). Suppose \( M \not\sim_a^P \text{Cl}(M) \setminus N \). Then by Lemma 6.7.2 either \( \Pi_a^*(\text{Cl}(M) \setminus \text{Cl}(M) \setminus N)) \neq \emptyset \) or \( \Pi_a^*(\text{Cl}(M) \setminus N) \setminus \text{Cl}(M)) \neq \emptyset \). \( \text{Cl}(M) \setminus N \subseteq \text{Cl}(M) \) so \( \text{Cl}(\text{Cl}(M) \setminus N) \subseteq \text{Cl}(M) \) so
\[6.7. \text{Proof of Theorem 6.3.5}\]

Let \(M, K\) be sets of messages such that \(M \models K\). Then \(\Pi_a(M) = \Pi_a(K)\) so \(m \in \Pi_a(M)\). By Lemma 6.7.5, the fact that \(M \models K\) implies that \(M \models K\) if \(m \in -\).

Suppose \(l_1, \ldots, l_n\) be literals such that \(l_1 \lor \ldots \lor l_n\) is not a tautology. Let \(M\) be a set of messages such that \(M \models K\). Then \(\Pi_a(M) \subseteq \Pi_a(K)\), so \(m \in \Pi_a(M)\) and \(M \models m\). But \(M\) was chosen arbitrarily, so \(M \models K\).

6.7.6. Lemma. Let \(l_1, \ldots, l_n\) be literals such that \(l_1 \lor \ldots \lor l_n\) is not a tautology. Let \(M\) be a set of messages such that \(M \models K\). Then \(\Pi_a(M) \subseteq \Pi_a(K)\), so \(m \in \Pi_a(M)\) and \(M \models m\). But \(M\) was chosen arbitrarily, so \(M \models K\).

Proof. I will give a proof with induction on the number of literals \(n\). If \(n = 1\) then the result becomes trivial. Suppose the result holds for \(n\) and take literals \(l_1, \ldots, l_{n+1}\) and a set of messages \(M\) such that \(M \models K\) or \(l_{n+1}\). If \(M \models K\) then \(K\) or \(l_{n+1}\) follows by induction hypothesis. Suppose otherwise.

Then there is some \(M\) such that \(M \models P\) and \(M' \models \neg l_1 \land \ldots \land \neg l_n\). Then because \(M \models K\) or \(l_{n+1}\), it must be the case that \(M' \models l_{n+1}\). We claim that \(M \models K\).

Suppose \(l_{n+1} = m\) for some message \(m\). Let \(N = \{m' \in C(M') \mid m' \models \text{mentions } m\}\). By Lemma 6.7.5, the fact that \(M' \models \neg K\) implies that \(M' \models P\) \(C(M') \setminus N\). Clearly, \(C(M') \setminus N \models l_{n+1}\) implies that \(m' \models l_{n+1}\). Suppose there is some \(l_a\) such that \(C(M') \setminus N \models l_a\). I already know that \(M' \models l_a\), so then it must be the case that \(l_a = m'\) and \(m' \models \). But then \(m'\) mentions \(m\) and \(l_1 \lor \ldots \lor l_n\) is a tautology.

Suppose \(l_{n+1} = m\) for some message \(m\). By Lemma 6.7.4 the fact that \(M' \models \neg K\) implies that \(M' \models P\). Clearly, \(M' \cup \{m\} \models l_{n+1}\). Suppose there is some \(l_a\) such that \(M' \cup \{m\} \models l_a\). I already know that \(M' \models l_a\) so then it must be the case that \(l_a = m'\) for some message \(m' \in C(m)\). But then \(l_1 \lor \ldots \lor l_n\) is a tautology. So \(M' \cup \{m\} \models l_a\) for any \(l_a\). I assumed that \(M \models K\) or \(l_{n+1}\), so this is a contradiction.

I conclude that \(M \models K\).

6.7.7. Lemma. Let \(M, M'\) be sets of messages and let \(l_1, \ldots, l_n\) be literals such that \(M \models P\), \(M' \models l_1 \land \ldots \land l_n\). Then there is \(M''\) such that \(M \models P\), \(M'' \models l_1 \land \ldots \land l_n\). Similar for \(P\).

Proof. First note that because \(M \models P\), \(M' \models l_1 \land \ldots \land l_n\), for any \(l_a\) I have that \(M \models \neg K\) or \(l_a\). Let \(M' = \{m \in \{l_1, \ldots, l_n\} \mid M \models m\}\). For any \(m \in M'\), \(M \models \neg K\) or \(m\) so then by repeated application of Lemma 6.7.4 I get that \(M \models P\) or \(M \cup M'\). Let \(M' = \{m \in C(M \cup M') \mid m\) mentions \(m'\) such that \(\neg m' \in C(M')\) or \(\neg m' \in C(M)\).
\{l_1, \ldots, l_n\}\}. For any \(\neg m' \in \{l_1, \ldots, l_n\}\) it holds that \(M \not\models K, m'\) so then \(M \cup M^+ \not\models K, m'\). Then by repeated application of Lemma 6.7.4 I get that \(M \sim^P_a \text{Cl}(M \cup M^+) \setminus M^+\). Clearly, every \(l_n\) of the form \(l_a = \neg m\) is satisfied in \(\text{Cl}(M \cup M^+) \setminus M^+\). Now take some \(l_a\) of the form \(l_a = m\). Clearly, \(m \in \text{Cl}(M \cup M^+)\). Suppose \(\text{Cl}(M \cup M^+) \setminus M^+ \not\models m\). Then \(m \in M^+\), so \(m\) mentions some \(m'\) such that \(\neg m' \in \{l_1, \ldots, l_n\}\). But then \(l_1 \wedge \ldots \wedge l_n\) is a contradiction which is not possible because \(M' \models l_1 \wedge \ldots \wedge l_n\). So \(\text{Cl}(M \cup M^+) \setminus M^- \models l_1 \wedge \ldots \wedge l_n\). It is not hard to see that \(\delta(\text{Cl}(M \cup M^+) \setminus M^-) \leq \max(\delta(M), \delta(l_1), \ldots, \delta(l_n))\). \(\square\)

6.7.8. COROLLARY. Let \(M\) be a set of messages and \(l_1, \ldots, l_n\) be literals. Suppose that for any \(M' \sim^P_a M\) with \(\delta(M') \leq \max(\delta(M), \delta(l_1), \ldots, \delta(l_n))\), \(M' \models l_1 \vee \ldots \vee l_n\). Then for any \(M''\) such that \(M \sim^P_a M'', M'' \models l_1 \vee \ldots \vee l_n\).

6.7.9. THEOREM. For any set of messages \(M\) and formula \(\varphi\) there is a finite number \(n_{M, \varphi} \geq \delta(M)\) such that for every \(k \geq n_{M, \varphi}\),

\[ M \models \varphi \text{ iff any } F \in \mathcal{F}^k\varphi \text{ contains a literal } l \in F \text{ such that } M \models l. \]

PROOF. I will give a proof with structural induction on \(\varphi\).

Suppose \(\varphi = m\). Let \(n_{M, \varphi} = \max(\delta(M), \delta(m))\). Then for any \(k \geq n_{M, \varphi}\), \(\mathcal{F}^k(\varphi) = \{\{m\}\}\) and the desired result follows immediately.

Suppose \(\varphi = \neg \psi\). Let \(n_{M, \varphi} = n_{M, \psi}\) and take some \(k \geq n_{M, \varphi}\). Suppose \(M \models \neg \psi\). Then there is \(F\) in \(\mathcal{F}^k(\psi)\) such that for every \(l \in F\), \(M \models \bar{\bar{l}}\). Then for every \(F' \in \mathcal{F}^k(\neg \psi)\) there is \(\bar{\bar{l}} \in F'\) such that \(l \in F\) and \(M \models \bar{\bar{l}}\). For the converse I will use contraposition. Suppose that \(M \models \psi\). Then for every \(F \in \mathcal{F}^k(\psi)\) there is some \(l \in F\) such that \(M \models l\). Let \(F' \in \mathcal{F}^k(\neg \psi)\) be the set containing the negation of exactly these literals. Then there is no \(\bar{\bar{l}} \in F'\) such that \(M \models \bar{\bar{l}}\). So then it does not hold that every \(F' \in \mathcal{F}^k(\neg \psi)\) contains some \(l' \in F'\) such that \(M \models l'\).

Suppose \(\varphi = \psi_1 \wedge \psi_2\). Let \(n_{M, \varphi} = \max(n_{M, \psi_1}, n_{M, \psi_2})\). The result follows by definition and induction hypothesis.

Suppose \(\varphi = \hat{K}_a \psi\). Construct \(n_{M, \varphi}\) as follows. If \(M \models \hat{K}_a \psi\) then \(n_{M, \varphi} = \max(\delta(\psi), n_{M, \psi})\). Otherwise, let \(k_1\) be the minimal number such that \(k_1 = n_{M_1, \psi}\) for some state \(M_1\) such that \(M_1 \models \neg \psi\) and \(M \sim^P_a M_1\). Let \(n_{M, \varphi} = \max(\delta(\psi), n_{M_1, \psi}, k_1)\). Take some \(k \geq n_{M, \varphi}\).

Suppose \(M \models \hat{K}_a \psi\). Take some \(F \in \mathcal{F}^k(\hat{K}_a \psi)\). Then there is some \(F' \in \mathcal{F}^k(\psi)\) on which \(F\) is based. Suppose \(F' = \{l_1, \ldots, l_n\}\). Let \(M\) be the collection of sets of messages \(M'\) such that \(M \sim^P_a M'\) and \(\delta(M') \leq k\). This collection is finite. For any \(M' \in M\), \(M' \models \psi\) and by induction hypothesis, \(M' \models l_1 \vee \ldots \vee l_n\). Note that \(\max(\delta(M), \delta(l_1), \ldots, \delta(l_n)) \leq k\). So by Corollary 6.7.8, \(M \models \hat{K}_a (l_1 \vee \ldots \vee l_n)\).

Then by Lemma 6.7.6, \(M \models \hat{K}_a l_j \vee \ldots \vee PK_a l_n\). Take some \(l_j\) such that \(M \models \hat{K}_a l_j\). I claim that \(M \models l\) for some \(l \in F\) based on \(l_j\).
Suppose \( l_j = m \) and \( a \in \{ s_m \} \cup r_m \). Then let \( l = m \) and I am done.

Suppose \( l_j = m \) and \( a \notin \{ s_m \} \cup r_m \). Because \( M \models \text{\( \hat{K}_a m \)} \), I have by Lemma 6.7.3 that \( m \in \Pi^*_{\alpha}(M) \). So there must be some \( m'' \in \Pi^*_{\alpha}(\text{Cl}(M)) \) mentioning \( m \). Then \( a \notin \{ s_m'' \} \cup r_m'' \) so \( m'' \neq m \) and there must be \( b, G \) such that \( m'' = (b, m', G) \), where \( m'' \) mentions \( m \) and \( a \in \{ b \} \cup G \). Also, \( m'' \in \text{Cl}(M) \) so \( M \models m'' \). Clearly, \( m'' \in F' \). I let \( l = m'' \) and I am done.

Suppose \( l_j = \neg m \). Let \( M' = M \cup \{ m \} \). Then because \( M \models \text{\( \neg \hat{K}_a m \)} \), \( M \notin \text{\( a \)} M' \).

Note that \( \text{Cl}(M) \setminus \text{Cl}(M') = \emptyset \) so then by Lemma 6.7.2 there is \( m' \in \Pi_{\alpha}(\text{Cl}(M') \setminus \text{Cl}(M)) \). But if \( m' \in \text{Cl}(M \cup \{ m \}) \setminus \text{Cl}(M) \) then \( m' \in \text{Cl}(\{ m \}) \). So \( m' \) is mentioned in \( m \). Also, if \( m' \in \Pi_{\alpha}(\text{Cl}(M') \setminus \text{Cl}(M)) \) then \( a \in \{ s_m \} \cup r_m \) so \( \neg m' \in F' \). But \( m' \notin \text{Cl}(M) \) so \( M \models \neg m' \).

Since \( F \) was chosen arbitrarily from \( F^k(\text{\( \hat{K}_a \psi \)}) \), this proves the desired result.

Now, suppose that for any \( F \in F^k(\text{\( \hat{K}_a \psi \)}) \), there is \( l \in F \) such that \( M \models l \).

For the sake of contradiction, suppose \( M \models \neg \text{\( \hat{K}_a \psi \)} \). Then by construction of \( n_{M,\psi} \) there is some \( M_1 \) such that \( M_1 \models \neg \psi \), \( M \sim^F \text{\( \hat{K}_a \)} M_1 \) and \( n_{M,\psi} \leq n_{M,\psi} \).

I claim that for any \( F' \in F^k(\psi) \), there is \( l' \in F' \) such that \( M_1 \models l' \). Take such \( F' \). Let \( F \in F^k(\psi) \) be the set based on \( F' \) and take \( l \in F \) such that \( M \models l \). Let \( l' \in F' \) be the literal on which \( l \) is based. I claim that \( M_1 \models l' \).

Suppose \( l = l' = m \) and \( a \in \{ s_m \} \cup r_m \). Then \( m \in \Pi^*_{\alpha}(M) \) and by Lemma 6.7.3, \( M \models \text{\( \hat{K}_a m \)} \) so \( M_1 \models m \).

Suppose \( l = (j, m', G) \), \( l' = m \), \( m' \) mentions \( m \), \( a \notin \{ s_m \} \cup r_m \) and \( a \in \{ b \} \cup G \). Then \( (b, m', G) \in \Pi^*_{\alpha}(M) \), so again by Lemma 6.7.3 \( M_1 \models (b, m', G) \). But since \( m \in \text{Cl}(m') \) and \( m' \in \text{Cl}((b, m', G)) \), then \( M_1 \models m \).

Suppose \( l = \neg m' \), \( l' = \neg m \), \( m \) mentions \( m' \) and \( a \in \{ s_m \} \cup r_m \). For the sake of contradiction suppose \( M_1 \models m \). Then because \( m' \in \text{Cl}(m) \), \( M_1 \models m' \). But \( a \in \{ s_m \} \cup r_m \) so then \( m' \in \Pi^*_{\alpha}(M_1) \) and by Lemma 6.7.3 \( M \models m' \). But this contradicts my assumption that \( M \models l \). So then it must be the case that \( M_1 \models \neg m \).

Suppose \( n_{M_1,\psi} \leq k \). Then we can apply the induction hypothesis to derive that \( M_1 \models \psi \), which is a contradiction with my earlier claim. So \( n_{M_1,\psi} > k \geq n_{M,\psi} \).

But this contradicts the construction of \( M_1 \). We conclude that our assumption that \( M \models \neg \text{\( \hat{K}_a \psi \)} \) was false, so \( M \models \text{\( \hat{K}_a \psi \)} \).

**Suppose** \( \psi = \text{\( \hat{K}_a \psi \)} \). The proof is analogous to that for \( \text{\( K_a \psi \)} \). \( \square \)