Logics of communication and knowledge  
Sietsma, F.A.G.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 4

Logic of Information Flow on Communication Channels

4.1 Introduction

In this chapter, I present a framework for modeling communication and knowledge that is very general and can be adapted to the natural needs of various situations. The approaches presented in Chapters 3, 5 and 6 are tailored towards specific situations. This is very convenient when modeling exactly such a situation, but if those approaches are not applicable then the approach presented in this chapter will be fit for modeling almost any other situation involving communication and knowledge. Furthermore, in this chapter I also give an explicit treatment of protocols which broadens the perspective to include a great number of issues that come up in practice.

As a running example, consider the following situation. The 1999 ‘National Science Quiz’ of The Netherlands Organisation for Scientific Research (NWO)\(^1\) had the following question:

\textit{Six friends each have one piece of gossip. They start making phone calls. In every call they exchange all pieces of gossip that they know at that point. How many calls at least are needed to ensure that everyone knows all six pieces of gossip?}

To reason about the information flow in such a scenario, I want to take into account the following issues: the messages that the agents possess (e.g. secrets), the knowledge of the agents, the dynamics of the system in terms of information passing (e.g. telephone calls), the underlying communication channels (e.g. the network of landlines) and the protocol the agents follow (e.g. a method to exchange all pieces of gossip). I will combine all these different aspects in an

\(^1\)For a list of references about the problem, cf. [Hurkens, 2000].
approach that is a new combination of Dynamic Epistemic Logic (DEL) and Interpreted Systems (IS).

Interpreted Systems, introduced by [Parikh and Ramanujam, 1985] and [Fagin et al., 1995] independently, are mathematical structures that combine history-based temporal components of a system with epistemic ones (defined in terms of local states of the agents). This framework is convenient when modeling knowledge development based on the given temporal development of a system. In IS, the epistemic structure of a system is generated from the temporal structure in a uniform way. However, the generation of temporal structures is not specified in the framework.

A different perspective on the dynamics of multi-agent systems is provided by DEL [Gerbrandy and Groeneveld, 1997, Baltag and Moss, 2004]. The main focus of DEL is not on the temporal structure of the system but on the epistemic impact of events as the agents perceive them. The development of a system through time is essentially generated by executing action models as discussed in Chapter 3 and 7. The epistemic relations in the initial static model and in the action models are not generated uniformly as in IS. Instead, they are designed by hand. How to obtain a reasonable initial model that fits the scenario to be modeled is not always clear. For real life applications it can be hard to find the correct initial model. Finding the correct action models that correspond to epistemic events can be even harder, as is observed in [Dechesne and Wang, 2007].

Much has been said about the comparison of the two frameworks, based on the observation that certain temporal developments of the system in IS can be generated by sequences of DEL updates on static models (see, e.g., [van Benthem et al., 2009a, Hoshi and Yap, 2009, Hoshi, 2009]). In this chapter, I will demonstrate further benefits of combining the two approaches by presenting a framework where epistemic relations are generated by matching local states and a history of observations as in IS, while keeping the flexibility of explicit actions as in DEL approaches.

The puzzle of the telephone calls was briefly discussed in [van Ditmarsch, 2000, Ch. 6.6] within the original DEL framework. Van Benthem [van Benthem, 2002] raised the research question whether the communication network can be made explicit in DEL. An early proposal to fill in this line of research can be found in [Roelofsen, 2005]. Communication channels in an IS framework made their appearance in [Parikh and Ramanujam, 2003]. In [Pacuit and Parikh, 2007, Apt et al., 2009] the information passing on so-called communication graphs or interaction structures is adressed, where messages are modeled as either atomic propositions or Boolean combinations of atomic propositions. In [Wang et al., 2009] a PDL-style DEL language is developed that allows explicit specification of protocols.

This chapter is organized as follows. I introduce the logic $L^{Ag,N}$ in Section 4.2. Section 4.3 relates the logic to the standard DEL and IS approaches. Section 4.4 introduces a modeling method and illustrates this method by a study of variations
4.2 An Adaptable Logic for Communication, Knowledge and Protocols

In this section I will present a flexible logic that can be adapted to the situation at hand. I will first give the language with its intuitive meaning. Then I will define the states on which this language is to be interpreted, together with its formal semantics.

4.2.1 Language

Let $Ag$ be a finite set of agents, $N$ a finite set of atomic notes and $Act$ a finite set of basic actions. Later on, I will give each action an internal structure that defines its meaning, but for now the actions may be considered to be atomic objects.

I define $net$ to be a hypergraph of agents in $Ag$, representing the communication network. It is a set of subsets of $Ag$, just like in the approach presented in [Apt et al., 2009]. Each subset represents a possible set of recipients of a single message. For example, if $net = \{\{a, b\}, \{a, b, c\}\}$ then the communication network allows for private communication between agent $a$ and $b$ and for group communication between agents $a$, $b$ and $c$. This rules out private communication between $b$ and $c$ or $a$ and $c$.

The set $P_{Ag,N,Act}$ of basic propositions is defined as

$$p := has_a n \mid com(G) \mid past(\bar{\alpha}) \mid future(\bar{\alpha}),$$

where $a \in Ag$, $n \in N$, $G \subseteq Ag$ and $\bar{\alpha} = \alpha_1; \ldots; \alpha_n$ with $\alpha_1, \ldots, \alpha_n \in Act$.

The intended meaning of these propositions is as follows. The proposition $has_a n$ means that agent $a$ possesses note $n$. This is a piece of information that he may send to other agents. The proposition $com(G)$ means that $G$ is a communication channel, so a group message to the group $G$ is in accordance with the communication network. The proposition $past(\bar{\alpha})$ means that the sequence of actions that happened most recently is $\bar{\alpha}$. Finally, the proposition $future(\bar{\alpha})$ means that the sequence of actions $\bar{\alpha}$ could be executed now, in accordance with the current protocol.

Using these propositions, I define the formulas of $\mathcal{L}^{Ag,N}_\tau$ as follows:

$$\varphi ::= \top \mid p \mid \neg \varphi \mid \varphi_1 \land \varphi_2 \mid \langle \pi \rangle \phi \mid C_{G,\varphi},$$

$$\pi ::= \alpha \mid \epsilon \mid \delta \mid \pi_1; \pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^*,$$

where $p \in P_{Ag,N,Act}$, $G \subseteq Ag$, $\alpha \in Act$ and $\epsilon$, $\delta$ are constants for the empty sequence and deadlock, respectively. I define $\Pi$ as the set of all possible protocols $\pi$. 

on the puzzle that was mentioned above. The final section concludes and lists future work.
The intended meaning of the formulas is as follows. The meaning of $\top$ and
the constructs $\neg$ and $\wedge$ is as usual. $C_G \phi$ expresses “the agents in group $G$ have
common knowledge of $\phi$”. A difference between this language and the one pre-
sent in Chapter 3 is that now, $\langle \pi \rangle \phi$ expresses “the protocol $\pi$ can be executed,
and at least one execution of $\pi$ yields a state where $\phi$ holds”. So instead of
expressing that $\phi$ holds in a world considered possible by an agent, this formula
now expresses that $\phi$ holds in a state that is a possible result of the protocol $\pi$.
The protocol $\pi$ is built from actions as the relations in Chapter 3 are built from
the agent’s epistemic relations.

As mentioned above, I will give each action an internal structure. This
internal structure is given for each $\alpha \in Act$ as a tuple of the following form:
\[
\iota(\alpha) := \langle G, \phi, N_1, \ldots N_{|Ag|}, \rho \rangle
\]
Here $G \subseteq Ag$ is the group of agents that can observe $\alpha$. $\phi$ is a formula of $L_{Ag}^N$ that
does not contain any modalities of the form $\langle \pi \rangle$. Moreover, it is the precondition
that should hold in order for $\alpha$ to be executable. I define $Obs(\iota(\alpha)) = G$ and
$Pre(\iota(\alpha)) = \phi$. Additionally, $Pos(\iota(\alpha)) = \langle N_1, \ldots N_{|Ag|}, \rho \rangle$ is the postcondition
that should hold after $\alpha$ has been executed. For every agent $a$, $N_a$ is the set
of notes that get delivered to $a$ by action $\alpha$. Finally, $\rho \in \Pi \cup \{\#\}$ gives the
protocol that the agents are going to follow after execution of $\alpha$. If $\rho = \#$,
then the agents should keep following the current protocol. If $\rho = \pi$ for some
$\pi \in \Pi$ then they should change their protocol to $\pi$. I will assume that an agent
can observe any action by which he receives some note. The converse does not
hold: agents may also observe actions by which no notes are delivered to them.
This happens for example when an agent knows that some other agent receives
a message containing a certain note, but he does not get to know the contents of
the note himself.

Note that by excluding the preconditions of the form $\langle \pi \rangle \phi$ I limit the interde-
pendence of actions. This prevents problems when for example an action would
be mentioned in its own precondition. Even with this constraint I can still express
a lot of useful preconditions. For example, for action $\alpha$,\textit{ future}($\alpha$) is allowed as
a precondition meaning that $\alpha$ can be executed only when it is allowed by the
current protocol.

As usual, I define $\bot$, $\phi \lor \psi$, $\phi \rightarrow \psi$ and $[\pi] \phi$ as the abbreviations of $\neg \top$,
$\neg (\neg \phi \land \neg \psi)$, $\neg \phi \lor \psi$ and $\neg \langle \pi \rangle \neg \phi$ respectively. Moreover, I use the following
additional abbreviations:
\[
\begin{align*}
K_a \phi & := C_{\{a\}} \phi \\
\text{has}_a N & := \bigwedge_{n \in N} \text{has}_a n \\
\text{dhas}_G N & := \bigwedge_{n \in N} \bigvee_{a \in G} \text{has}_a n \\
\text{com}(\text{net}) & := \bigwedge_{G \in \text{net}} \text{com}(G) \land \bigwedge_{G \notin \text{net}} \neg \text{com}(G) \\
\pi^n & := \pi; \pi; \ldots ; \pi \\
\Sigma \Pi' & := \bigcup_{\pi \in \Pi} \pi \text{ where } \Pi' \subset \Pi \text{ is finite.}
\end{align*}
\]
4.2. An Adaptable Logic for Communication, Knowledge and Protocols

Here $K_a \phi$ means that agent $a$ knows $\phi$, $\text{dhas}_G N$ expresses that the messages from $N$ are in distributed possession of the agents in $G$ and $\text{com}(\text{net})$ specifies the communication channels in the network.

By having both the $\text{has}$ and the $K$ operator in the language, I can make the distinction between knowing about a message and knowing about its content. $K_a \text{has}_b n \land \neg \text{has}_a n$ and $K_a \text{has}_b n \land \text{has}_a n$ can express the de dicto and de re reading of knowing that $b$ has a message, respectively. For example, let $n$ be the hiding place of bin Laden, then $K_{\text{CIA}} \text{has}_{\text{Al-Qaeda}} n \land \neg \text{has}_{\text{CIA}} n$ expresses that CIA knows that Al-Qaeda knows the hiding place, which is, however, a secret to CIA.

4.2.2 Semantics

In order to interpret the basic propositions in $P_{\text{Ag,M,Act}}$ I let the finer structure of the basic propositions correspond with a finer structure in the states, replacing the traditional valuation in Kripke structures used in DEL-approaches.

4.2.1. Definition. A state for $\mathcal{L}_t^{\text{Ag,N}}$ is defined as a tuple:

$$ s := \langle \text{net}, \text{N}_I^1, \ldots, \text{N}_I^{|\text{Ag}|}, \bar{\alpha}, \text{N}_I^1, \ldots, \text{N}_I^{|\text{Ag}|}, \pi \rangle. $$

Here $\text{net}$ is the communication graph, $\bar{\alpha}$ is the history of actions that have been executed, for every $a \in \text{Ag}$ $N_a$ gives the set of notes he possesses and $\pi$ gives the protocol the agents are following. I also include for every agent $a \in \text{Ag}$ the set $\text{N}_I^I$ which is the set of notes the agents had in the initial state, which was the state of the systems before the actions in $\bar{\alpha}$ were executed. Given a state $s$, I use $N(s)(a)$ to denote $N_a$, the information set of agent $a$. I use $N_I^I(s)(a)$ to denote $N_a^I$, the initial information set of agent $a$. I use $\text{Net}(s) := \text{net}$ for the communication graph, $H(s) := \alpha$ for the action history and $\text{Prot}(s) := \pi$ for the protocol.

Intuitively, each state represents a past temporal development of the system with its constraint for the future actions. Note that the past is linear ($\bar{\alpha}$ is a single sequence of actions), while the future can be branching (the protocol $\pi$ may allow several possible sequences of actions). From the initial information sets I can construct the initial state of the system before any actions were executed. For a state $s$ as in the previous definition, this is defined as

$$ \text{Init}(s) := \langle \text{net}, \text{N}_I^1, \ldots, \text{N}_I^{|\text{Ag}|}, \epsilon, \text{N}_I^1, \ldots, \text{N}_I^{|\text{Ag}|}, (\Sigma \text{Act})^* \rangle. $$

The initial state has an empty action history, and the information sets of the agents are identical to the initial information sets. Also, no protocol has been set so the protocol is $(\Sigma \text{Act})^*$, which allows all sequences of actions. Note that for any state $s$, the result of executing the history of past actions on $\text{Init}(s)$ should be $s$. 
I will interpret the formulas of $\mathcal{L}_t^{Ag,N}$ on the states defined above. However, in order to give the semantics for future $(\bar{\alpha})$ I need a way to check whether a sequence of actions complies with a certain protocol. Also, in order to give the semantics for $\langle \pi \rangle$ I need to be able to compute the remainder of the protocol after the action has been executed, so I know what the new protocol is. For this purpose I will use the input derivative and the output function (cf. [Brzozowski, 1964, Conway, 1971]).

I start out with the output function. This function returns $\epsilon$ if the protocol $\pi$ can be executed by doing no action, and $\delta$ otherwise. It is defined as follows:

$$
o(\epsilon) := \epsilon, \quad o(\delta) := \delta,
$$

$$
o(\alpha) := \delta, \quad o(\pi \cup \pi') := o(\pi) \cup o(\pi'),
$$

$$
o(\pi; \pi') := o(\pi); o(\pi'), \quad o(\pi^*) := \epsilon.
$$

Given a protocol $\pi$ and an action $\alpha$, the remainder of $\pi$ after executing $\alpha$ is the input derivative $\pi \backslash \alpha$ given by:

$$
\begin{align*}
\epsilon \backslash \alpha & := \delta, \quad \delta \backslash \alpha := \delta, \\
\alpha \backslash \alpha & := \epsilon, \quad \beta \backslash \alpha := \delta \quad (\alpha \neq \beta), \\
(\pi \cup \pi') \backslash \alpha & := \pi \backslash \alpha \cup \pi' \backslash \alpha, \\
(\pi; \pi') \backslash \alpha & := ((\pi \backslash \alpha); \pi') \cup (o(\pi); (\pi' \backslash \alpha)), \\
(\pi^*) \backslash \alpha & := \pi \backslash \alpha; \pi^*.
\end{align*}
$$

Let $\pi \backslash (\alpha_0; \alpha_1; \ldots; \alpha_n) = (\pi \backslash \alpha_0) \backslash \alpha_1 \ldots \backslash \alpha_n$. Using these definitions and the axioms of Kleene algebra I can syntactically derive the remaining protocol after executing a sequence of basic actions. For example:

$$
(\alpha \cup (\beta; \gamma))^* \backslash \beta = (\alpha \backslash \beta \cup (\beta; \gamma) \backslash \beta); (\alpha \cup (\beta; \gamma))^* = (\delta \cup (\epsilon; \gamma)); (\alpha \cup (\beta; \gamma))^* = \gamma; (\alpha \cup (\beta; \gamma))^*.
$$

Note that in general it does not hold that $\bar{\beta}; (\pi \backslash \bar{\beta}) = \pi$.

Let $A(\pi)$ be the set of sequences of actions that comply with the protocol $\pi$. It is defined as follows:

$$
A(\delta) = \emptyset \quad A(\epsilon) = \{ \epsilon \} \quad A(\alpha) = \{ \alpha \}
$$

$$
A(\pi; \pi') = \{ \bar{\alpha}; \bar{\beta} \mid \bar{\alpha} \in A(\pi), \bar{\beta} \in A(\pi') \}
$$

$$
A(\pi \cup \pi') = A(\pi) \cup A(\pi')
$$

$$
A(\pi^*) = \{ \bar{\alpha}_1; \ldots; \bar{\alpha}_n \mid \bar{\alpha}_1, \ldots, \bar{\alpha}_n \in A(\pi) \}
$$

In [Conway, 1971], the following is shown:

4.2.2. Lemma. $A(\pi \backslash \bar{\alpha}) = \{ \bar{\beta} \mid \bar{\alpha}; \bar{\beta} \in A(\pi) \}$.

This shows that the input derivative truly computes the remainder of the protocol after executing some basic action.
4.2. An Adaptable Logic for Communication, Knowledge and Protocols

Just like [Cohen and Dam, 2007, Apt et al., 2009], I will give the truth value of $L^{Ag,N}_\ell$ formula on single states instead of pointed Kripke models as is usual in DEL. The interpretation of epistemic formulas depends on a relation $\sim_a^x$ between states, which I will define later.

Given a state $s = \langle net, N^I_1, ..., N^I_{|Ag|}, \overline{\alpha}, N_1, ..., N_{|Ag|}, \pi \rangle$, the semantics of $L^{Ag,N}_\ell$ is defined as follows:

- $s \models has_a(n)$ if and only if $n \in N_a$
- $s \models com(G)$ if and only if $G \in net$
- $s \models past(\beta)$ if and only if $\beta$ is a suffix of $\bar{\alpha}$
- $s \models future(\beta)$ if and only if $\pi \backslash \beta \neq \delta$
- $s \models \neg \varphi$ if and only if $s \not|= \varphi$
- $s \models \varphi_1 \land \varphi_2$ if and only if $s \models \varphi_1$ and $s \models \varphi_2$
- $s \models (\pi)\varphi$ if and only if $\exists s' : s[\pi]s'$ and $s' \models \varphi$
- $s \models C_{G}\varphi$ if and only if $\forall s' : s \sim^x_G s'$ implies $s' \models \varphi$

Here $\sim^x_G$ is the reflexive transitive closure of $\bigcup_{a \in G} \sim_a^x$. As noted above, the relation $\sim_a^x$ is the knowledge relation for agent $a$ and it will be more formally defined later.

The protocols $\pi$ function as state changers. Each protocol describes a transition to a new state in the following way:

- $s[\varepsilon]s'$ if and only if $s = s'$
- $s[\delta]s'$ never
- $s[\beta]s'$ if and only if $s \models Pre(\ell(\beta))$ and $s' = s|_{Pos(\ell(\beta))}$
- $s[\pi_1; \pi_2]s'$ if and only if $\exists s'' : s[\pi_1]s''$ and $s''[\pi_2]s'$
- $s[\pi_1 \cup \pi_2]s'$ if and only if $s[\pi_1]s'$ or $s[\pi_2]s'$
- $s[(\pi_1)^n]s'$ if and only if $\exists n : s[\pi_1; \pi_1; \ldots; \pi_1]s'$

Given $Pos(\ell(\beta)) = \langle N'_1, \ldots, N'_{|Ag|}, \rho \rangle$, $s|_{Pos(\ell(\beta))}$ is the result of executing action $\beta$ at $s$. It is defined as

$$s|_{Pos(\ell(\beta))} = \langle net, N^I_1, \ldots, N^I_{|Ag|}, \overline{\alpha}; \beta, N_1 \cup N'_1, \ldots, N_{|Ag|} \cup N'_{|Ag|}, f(\rho) \rangle,$$

where $f(\rho) = \left\{ \begin{array}{ll} \pi \backslash \beta & \text{if } \rho = \# \\ \pi' & \text{if } \rho = \pi' \end{array} \right.$

So I add the action $\beta$ to the sequence of past actions, I add for each agent $a$ the notes he received by $\beta$ and I change the protocol to a new protocol $\pi'$ if this is prescribed by $\beta$, or to the remainder of the old protocol after executing $\beta$ if no new protocol is dictated.

Now I will define the epistemic relation of an agent $a$ between states. This relation depends on the observational power of the agents, which may vary in different situations. Therefore I represent it as a relation $\sim^a_{obs}$, where $obs$ stands
Chapter 4. Logic of Information Flow on Communication Channels

for the observational power of the agents. A state $s$ is said to be consistent if $\text{Init}(s)[H(s)]s$. It is easy to see that for any $s$, $\text{Init}(s)$ is always consistent. Note that I can actually omit the current information sets $\mathcal{N}(s)$ in the definition of a state, and compute them by applying the actions in $H(s)$ to $\mathcal{N}^I(s)$, thus only generating consistent states. I keep the current information sets in the definition of the state in order to simplify the notation and to evaluate basic propositions more efficiently.

I define that $s \sim_{\text{obs}} s'$ if and only if the following conditions are met:

**consistency** $s$ and $s'$ are consistent.

**local initialization** $\mathcal{N}^I(s)(a) = \mathcal{N}^I(s')(a)$,

**local history** $H(s)_{\text{obs}} = H(s')_{\text{obs}}$, where $\text{obs}$ is the type of observational power of agents.

The type of observational power of the agents defines how the agents observe the history. In other words, it defines their local history $H(s)_{\text{obs}}$. Many definitions of $H(s)_{\text{obs}}$ are possible, giving the agents different observational powers. This is one of the things that make this framework so flexible and allow for adaptation to different situations. Several reasonable definitions are:

1. $H(s)_{a}^{\text{set}} = \{ \alpha \text{ appearing in } H(s) \mid a \in \text{Obs}(\iota(\alpha)) \}$ as in [Apt et al., 2009] and in Chapter 5 and 6. In this set-up, the agents are aware of the actions they can observe but not of the ordering between these actions.

2. $H(s)_{a}^{1\text{st}}$ is the subsequence of $H(s)$ consisting of the first occurrence of each $\alpha \in H(s)_{a}^{\text{set}}$ as in [Baskar et al., 2007]. In this set-up, the agents are aware of the ordering of the first occurrence of the actions they can observe.

3. $H(s)_{a}^{\text{asyn}}$ is the subsequence of $H(s)$ consisting of all the occurrences of each $\alpha \in H(s)_{a}^{\text{set}}$, as in asynchronous systems (cf., e.g., [Shilov and Garanina, 2002]). In this set-up, the agents are aware of all occurrences of the actions they can observe and the ordering between them.

4. $H(s)_{a}^{\tau}$ is the sequence obtained from $H(s)$ by replacing each occurrence of $\alpha \not\in H(s)_{a}^{\text{set}}$ by $\tau$, as in synchronous systems with perfect recall (cf., e.g., [van der Meyden and Shilov, 1999]). In this set-up, the agents are aware of all occurrences of the actions they can observe and they are also aware of the number of actions that have been happened that they cannot observe, and of the order between the actions they can observe and the actions they cannot observe. They do not get to know which actions that they cannot observe have happened.

It is clear from the above definition that $\sim_{\text{obs}}$ is an equivalence relation and the following holds:
4.2.3. **Lemma.** \( \sim^\tau_a \subseteq \sim_{a}^{\text{asyn}} \subseteq \sim_{a}^{1\text{st}} \subseteq \sim_{a}^{\text{set}} \).

So the \( \sim^\tau \) relation is the smallest relation, thereby giving the agents the greatest amount of knowledge, and the \( \sim^{\text{set}} \) relation is the largest, giving the agents only little knowledge.

I call the semantics defined by \( \sim^{\text{obs}}_a \) the *obs-semantics*, and denote the corresponding satisfaction relation as \( \models^{\text{obs}}_a \).

Recall that the agents can always observe the actions that change their information set. This implies the following lemma.

4.2.4. **Lemma.** For any consistent state \( s \), \( s \sim^{\text{obs}}_a s' \) implies \( N(s)(a) = N(s')(a) \), where \( \text{obs} \in \{ \text{set, asyn, 1st, } \tau \} \).

**Proof.** By Lemma 4.2.3, \( s \sim^{\text{obs}}_a s' \) implies \( s \sim^{\text{set}}_a s' \) for all \( \text{obs} \in \{ \text{set, asyn, 1st, } \tau \} \). Therefore I only need to prove the claim for \( \text{obs} = \text{set} \). Suppose \( s \sim^{\text{set}}_a s' \). Then by the definition of \( \sim^{\text{set}}_a \), \( N(\text{Init}(s))(a) = N(\text{Init}(s'))(a) \) and \( H(s)|^\text{set}_a = H(s')|^\text{set}_a \).

So at \( s \) and \( s' \) agent \( a \) initially had the same messages and has observed the same actions. Since agents can always observe the actions that change their information set, this implies that the same message passing actions relevant to \( a \) have happened in \( s \) and \( s' \). Since the actions can only add notes to the information sets of the agents and never delete notes from them, it does not matter how often or in which order those actions have been executed. Therefore the information sets of agent \( a \) in \( s \) and \( s' \) are identical. \( \square \)

By using different semantics in different situations, I can vary the observational power of the agents as is required. By constructing actions that match the situation at hand, I can also vary the exact properties of the communicative events. I will now define some useful basic actions with their internal structure. These actions correspond to communicative events that often come up in practice.

In order to simplify the presentation, I will omit the explicit mentioning of the internal structure map \( \iota \). So I will use \( \text{Obs}(\alpha) \) for \( \text{Obs}(\iota(\alpha)) \) etcetera. Recall that the internal structure of an action \( \alpha \) is a tuple

\[
\iota(\alpha) := (G, \phi, N_1, \ldots, N_{|\text{Ag}|}, \rho)
\]

such that \( N_a = \emptyset \) for \( a \notin \text{Obs}(\alpha) \). The following table lists some basic actions. In Section 4.4 I will use these as building blocks for more complex actions.
Chapter 4. Logic of Information Flow on Communication Channels

In the rightmost column of table I have left out from the postconditions the sets of notes of the agents that do not change, in order to save space.

The first group of actions are communicative actions that are done by the agents. These actions must abide by the communication channels and the protocol, which is enforced by having \( \text{com}(\text{Obs}(\alpha)) \land \text{future}(\alpha) \) in the precondition. \( \text{send}_a^G(N) \) is the action that \( a \) sends the set of notes \( N \) to the group \( G \). Apart from respecting the channels and the protocol, the precondition has \( aN \) enforces that agent \( a \) should possess the notes he wants to send. The postcondition of \( \text{send}_a^G(N) \) expresses that the messages in \( N \) get added to the message sets of the agents in \( G \). \( \text{share}_G(N) \) shares the messages from \( N \) within the group \( G \). A precondition is that the messages from \( N \) are already distributed knowledge in the group. \( \text{sendall}_a^G \) differs from \( \text{send}_a^G(N) \) in the fact that \( a \) sends all the notes that he has. Similarly for \( \text{shareall}_G \). \( \text{inform}_a^G(\phi) \) is the group announcement by \( a \) of an arbitrary formula \( \phi \) within \( G \cup \{a\} \). The precondition for this action is that agent \( a \) knows that \( \phi \) holds. Since all agents know that the execution of this action would only be possible if \( \phi \) would hold, all agents who can observe the action know that \( \phi \) holds at the moment it is announced. This way knowledge of \( \phi \) is created among the members of \( G \).

The second group of actions are public announcements that do not respect the channels or the protocol. They model the information that is given to the agents by some external authority. \( \text{exinfo}(\phi) \) models the public announcement of a formula \( \phi \). The only precondition of this announcement is that \( \phi \) should hold. The postcondition is empty. Again, knowledge of \( \phi \) is created by the fact that the agents know that the action can only be done if \( \phi \) holds. \( \text{exprot}(\pi) \) announces the protocol \( \pi \) that the agents are supposed to follow in the future. ts postcondition changes the protocol to \( \pi \) and knowledge of the protocol is created by the fact that all agents observe the announcement.
4.3 Comparison with IS and DEL

The results in this section relate my logic to IS and DEL approaches. Theorem 4.3.1 shows that by the semantics of $\mathcal{L}_{i}^{Ag,N}$, an interpreted system is implicitly generated from a single state. Together with Theorem 4.3.1, Theorem 4.3.3 demonstrates that compared to DEL, my approach models actions in a very powerful and concise manner.

I will compare my approach to IS first. In the following I only consider consistent states.

Given a state $s$ with action history $H(s) = \alpha_1 \alpha_2 \ldots \alpha_n$, I define the history of $s$ as the sequence $his(s) = s_0 s_1 \ldots s_n$ where $s_0 = Init(s)$, $s_n = s$ and for all $1 \leq k \leq n$, $s_k[i]s_k$. Clearly then $s_0 s_1 \ldots s_k = his(s_k)$ for any $k \leq n$.

Given some type of semantics $\text{obs}$, let $ExpT^{\text{obs}}$ be the Interpreted System given by $\{ H, \rightarrow \alpha, \{ R_i | i \in Ag \}, V \}$, where

- $H = \{ his(s) \mid s \text{ is consistent} \}$,
- $\langle s_0 \ldots s_n \rangle \rightarrow \alpha \langle s_0 \ldots s_n s_{n+1} \rangle$ iff $s_n[i]s_{n+1}$,
- $\langle s_0 \ldots s_n \rangle R_i \langle s_0' \ldots s_m' \rangle$ iff $s_n \sim_i s_m'$,
- $V(\langle s_0 \ldots s_n \rangle)(p) = \top$ iff $s_n \models_{\text{obs}} p$, where $p \in P_{Ag,M,Act}$.

This is a straightforward adaptation of my logic to the IS framework. The language $\mathcal{L}_{i}^{Ag,N}$ can be seen as a fragment of Propositional Dynamic Logic (PDL) with basic actions taken from $Act \cup Ag$. Then the $C_G$ operator corresponds to $(\Sigma G)^*$. Let $\models_{\text{PDL}}$ denote the usual semantics of this fragment. The following theorem follows easily:

**4.3.1. Theorem.** For any formula $\varphi \in \mathcal{L}_{i}^{Ag,N}$ and for each consistent $\mathcal{L}_{i}^{Ag,N}$-state $s$:

$$s \models^{\text{obs}} \varphi \text{ iff } ExpT^{\text{obs}}, hist(s) \models_{\text{PDL}} \varphi.$$

This result shows that when I abstract away the inner structure of basic propositions and actions, then the logic can be seen as a PDL language interpreted on ISs that are generated in a particular way in accordance with some constraints.

Next, I will compare my work to standard DEL. Consider the following DEL language $\mathcal{L}_{\text{DEL}}$:

$$\phi := \top | p | \neg \phi | \phi_1 \land \phi_2 | [\mathcal{A}, e][\phi] | C_G \phi$$

Here $p$ is taken from a set of basic propositions $P$, $G \subseteq Ag$ and $\mathcal{A}$ is an action model, as defined in Chapter 2, with $e$ as its designated action. The formula $[\mathcal{A}, e][\phi]$ holds in $\mathcal{M}, w$ for some Kripke model $\mathcal{M}$ and $w \in W^\mathcal{M}$ iff $\phi$ holds in $\mathcal{M} \otimes \mathcal{A}, (w, e)$.

I would like to see if a translation is possible from $\mathcal{L}_{i}^{Ag,N}$ to DEL. Such a translation would go from the actions of $\mathcal{L}_{i}^{Ag,N}$ to the action models of DEL. A protocol $\pi$ would then correspond to a sequence of action models. The first barrier...
in the way of such a translation is the fact that the * operator allows for arbitrarily long sequences of actions, while there is no such operator on modalities of action models in DEL. Therefore, I will consider the star-free fragment of \( L_{Ag,N} \).

However, it turns out that even without the * operator it is not possible to find a translation for all kinds of semantics (set, 1st, etcetera). To see why this is true, recall the following result from [van Benthem et al., 2009a].

4.3.2. Theorem ([van Benthem et al., 2009a]). If we see \([A, e] \) as a basic action modality in the semantics of the PDL language, then for any formula \( \varphi \in L_{DEL} \) and for any model \( M \) and state \( w \in W_M \):

\[
M, w \models_{DEL} \varphi \iff \text{Forest}(M, A), (w) \models_{PDL} \varphi
\]

Here \( A \) is the set of action models and \( \text{Forest}(M, A) \) is the IS generated by executing all possible sequences of action models in \( A \) on \( M \).

Using this theorem, I will now show that the effects of actions in \( L_{Ag,N} \) cannot, in general, be simulated by action models.

4.3.3. Theorem. There is no DEL-model \( M \) such that for all consistent \( L_{Ag,N} \)-states \( s \) there is some \( w \in W_M \) that satisfies for all formulas \( \varphi \in L_{Ag,N} \):

\[
s \models \varphi \iff M, w \models_{DEL} \varphi.
\]

Proof. Suppose there was such \( M \). Then by Theorem 4.3.1 and 4.3.2,

\[
(ExpT^{obs}, \text{hist}(s)) \sim (\text{Forest}(M, A), (w)),
\]

where \( \sim \) is the bisimulation for transitions labeled with \( \text{Act} \cup Ag \). In [van Benthem et al., 2009a] it is shown that any model of the form \( \text{Forest}(M, A) \) must satisfy the property of perfect recall. This property states that if the agents cannot distinguish two sequences of actions \( \bar{\alpha}; \alpha \) and \( \bar{\beta}; \beta \) then they cannot distinguish \( \bar{\alpha} \) and \( \bar{\beta} \). But \( ExpT^{obs} \) does not satisfy this property for \( obs \in \{set, 1st, asyn\} \). For example, if \( \gamma \) is some action that \( b \) cannot observe then \( send^a_b(N); \gamma \sim_{b}^{obs} send^a_b(N) \), but \( send^a_b(N) \not\sim_{b}^{obs} \gamma \). So the set-, 1st- and asyn-semantics cannot be translated to a DEL model.

\[\square\]

4.4 Applications

4.4.1 Common Knowledge

This framework gives an interesting perspective on common knowledge. It may not be surprising that common knowledge cannot be reached without public communication [Halpern and Moses, 1990]. I first focus on asynchronous semantics.
4.4. Applications

One might think that achieving common knowledge becomes easier if the agents can publicly agree on a common protocol before the communication is limited to non-public communication. However, in the case of asynchronous semantics common knowledge still cannot be achieved, even if the agents can publicly agree on a protocol. Recall that I say an action $\alpha$ respects the communication channels if $\text{Pre}(\alpha) \models \text{com}(\text{Obs}(\alpha))$.

4.4.1. Theorem. For any state $s$ with $\text{Ag} \not\in \text{Net}(s)$, any protocol $\pi$ containing only actions that respect the communication channels, any $\varphi \in \mathcal{L}_{\text{Ag}}^N$ and any sequence of actions $\bar{\alpha}$:

$$s \models^{\text{async}} \langle \text{exprot}(\pi) \rangle (C\text{Ag}\varphi \rightarrow \neg \langle \bar{\alpha} \rangle C\text{Ag}\varphi)$$

Proof. Let $s[\text{exprot}(\pi)]t$ and suppose $t \models^{\text{async}} \neg C\text{Ag}\varphi$. Towards a contradiction, let $\bar{\alpha}$ be the minimal sequence of actions such that $t \models^{\text{async}} \langle \bar{\alpha} \rangle C\text{Ag}\varphi$. Let $\bar{\alpha} = \bar{\beta}; \alpha$, $t[\bar{\beta}]u$ and $u[\alpha]\bar{v}$. Since $\text{Ag} \not\in \text{Net}(s)$ and $\alpha$ respects the communication channel, $\text{Obs}(\alpha) \neq \text{Ag}$ so there exists $a \not\in \text{Obs}(\alpha)$. Then $H(u)|^a_{\text{async}} = H(v)|^a_{\text{async}}$ so $u \sim^a_{\text{async}} v$. Since $\bar{\alpha}$ was minimal, $u \not\models^{\text{async}} C\text{Ag}\varphi$. But then $u \models^{\text{async}} \neg K_a C\text{Ag}\varphi$ so $v \not\models^{\text{async}} C\text{Ag}\varphi$. This contradicts my assumption, so there cannot be such $\bar{\alpha}$. So $s \models^{\text{async}} \langle \text{exprot}(\pi) \rangle (\neg C\text{Ag}\varphi \rightarrow \neg \langle \bar{\alpha} \rangle C\text{Ag}\varphi)$.

Essentially, even if the agents agree on a protocol beforehand, the agents that cannot observe the final action of the protocol will never know whether this final action has been executed and thus common knowledge is never established. This is because in the asynchronous semantics, there is no sense of time. If there would be some kind of clock and the agents would agree to do an action on every “tick”, the agents would be able to establish common knowledge. This is exactly what I try to achieve with the $\tau$-semantics. Here every agent observes a “tick” the moment some action is executed. This way, they can agree on a protocol and know when it is finished. I will show examples of how this can result in common knowledge in the discussion of the telephone call scenario.

Here I will first investigate what happens in $\tau$-semantics if the agents cannot publicly agree on a protocol beforehand. I will show that in this case they cannot reach common knowledge of basic formulas. I start out with a lemma stating that actions preserve the agent’s relations.

4.4.2. Lemma. For any two states $s$ and $t$ and any action $\alpha$, if $s \sim^\tau_1 t$ and there are $s'$, $t'$ such that $s[\alpha]\bar{s}'$ and $t[\alpha]\bar{t}'$ then $s' \sim^\tau_1 t'$.

Proof. Suppose $s \sim^\tau_1 t$. Then $H(s)|^\tau_1 = H(t)|^\tau_1$. Suppose $i \in \text{Obs}(\alpha)$. Then $H(s')|^\tau_1 = (H(s)|^\tau_1; \alpha) = (H(t)|^\tau_1; \alpha) = H(t')|^\tau_1$. Suppose $i \not\in \text{Obs}(\alpha)$. Then $H(s')|^\tau_1 = (H(s)|^\tau_1; \tau) = (H(t)|^\tau_1; \tau) = H(t')|^\tau_1$. So $s' \sim^\tau_1 t'$.
This result may seem counter-intuitive, since for example a public announcement action may give the agents new information and thus destroy their epistemic relations. However, in my framework I model the new knowledge introduced by communicative actions by the fact that these actions would not be possible in states that do not satisfy the precondition of the action. In this lemma I assume that there are $s'$, $t'$ such that $s[\alpha]s'$ and $t[\alpha]t'$. This means that $s$ and $t$ both satisfy the preconditions of $\alpha$, so essentially no knowledge that distinguishes $s$ and $t$ is introduced by $\alpha$.

Let $L_{\text{bool}}$ be the following fragment of $L_i^{Ag,N}$:

\[ \phi ::= \text{has}_i m \mid \text{com}(G) \mid \neg \phi \mid \phi_1 \land \phi_2 \]

It is trivial to show that any action that does not change the agents’ message sets or the protocol does not change the truth value of these basic formulas:

**4.4.3. Lemma.** Let $\alpha$ be an action that does not change the agents’ message sets or the protocol. For any $\phi \in L_{\text{bool}}$ and any state $s$: $s \models \phi \leftrightarrow \langle \alpha \rangle \phi$.

Combining the properties of the actions from the previous lemma, I call an action $\text{dummy}(G)$ to be a *dummy action* for a group of agents $G$ if it has the precondition $\text{com}(G) \land \text{future}(\text{dummy}(G))$, it does not change the message sets of the agents or the protocol and $\text{Obs}(\text{dummy}(G)) = G$. An example of dummy action is $\text{inform}^t_G(\top)$. One could see it as “idle talk”.

**4.4.4. Theorem.** Let $A$ be a set of basic actions respecting the communication channels such that for any agent $a$ there is a dummy action $\text{dummy}(G)$ such that $a \not\in G \subseteq \text{Ag}$. Let $s$ be a state such that $\text{Ag} \not\in \text{Net}(s)$ and it is common knowledge at $s$ that the protocol is $\pi = (\Sigma A)^*$ (any action in $A$ is allowed). Then for any $\phi \in L_{\text{bool}}$ and any sequence of actions $\bar{\alpha}$,

\[ s \models^t \neg C_{\text{Ag}} \phi \rightarrow \neg \langle \bar{\alpha} \rangle C_{\text{Ag}} \phi \]

**Proof.** Suppose towards a contradiction that $s \models \neg C_I \phi$ and there is a minimal sequence $\bar{\alpha}$ such that $s \models^t \langle \bar{\alpha} \rangle C_{\text{Ag}} \phi$. Let $\bar{\alpha} = \bar{\beta}; \alpha$ and let $a \not\in \text{Obs}(\alpha)$. Such a always exists since $\text{Ag} \not\in \text{Net}(s)$. Let $\text{dummy}(G)$ be a dummy action such that $a \not\in G$. Let $s[\bar{\beta}]u$. Since $\bar{\alpha}$ is minimal, $u \models^t \neg C_{\text{Ag}} \phi$, so there is a $\sim_{\text{Ag}}$-path from $u$ to a world $t$ such that $t \not\models^t \phi$. Since it is common knowledge that any action in $A$ is possible, $\text{dummy}(G)$ can be executed at any world on the path from $u$ to $t$. By lemma 4.4.2 $\text{dummy}(G)$ preserves the relations between states so there are states $u'$, $t'$ such that $u[\text{dummy}(G)]u'$, $t[\text{dummy}(G)]t'$ and $u' \sim_{\text{Ag}} t'$. Also, since $t \not\models^t \phi$ and by lemma 4.4.3, $t' \not\models^t \phi$. So $u'$ not $\models^t C_{\text{Ag}} \phi$. This means that if $\text{dummy}(G)$ would be executed in state $u'$, then $C_{\text{Ag}} \phi$ would not hold in the resulting state.

Let $u[\text{dummy}(G)]u'$ and $u[\alpha]v$. Because $a \not\in G$, $a$ cannot see the difference between executing $\text{dummy}(G)$ and $\alpha$: $H(u')|_a^\tau = (H(u)|_a^{\tau}; \tau) = H(v)|_a^{\tau}$ so $u' \sim_a v$. 

---

**Chapter 4. Logic of Information Flow on Communication Channels**
4.4. Applications

But I just showed that $u' \not \models^\tau C_{Ag}\phi$, so then $v' \not \models^\tau C_{Ag}\phi$. But this contradicts my assumption that $\beta; \alpha$ induced common knowledge of $\phi$. \qed

Before turning to the specific scenario of the telephone calls, I propose the following general modeling method:

1. Select a set of suitable actions $Act$ with internal structures to model the communicative events in the scenario.

2. Design a single state as the real world to model the initial setting, i.e., $\langle net, N_1, ..., N_{|Ag|}, \alpha, N_1, ..., N_{|Ag|}, (\Sigma A)^* \rangle$ where $net$ models the communication network and $N_a$ models the information possessed by agent $a$.

3. Translate the informal assumptions of the scenario into formulas $\varphi$ and protocols $\pi$ in $\mathcal{L}_{Ag,N}^\iota$.

4. Use $exinfo(\varphi)$ and $exprot(\pi)$ to make the assumptions and the protocol common knowledge.

I will demonstrate how I can use this method to model the telephone call scenario. Let me first recall the scenario: in a group of people, each person has one secret. They can make private telephone calls amongst themselves in order to communicate these secrets. The original puzzle concerns the minimal number of telephone calls needed to ensure everyone gets to know all secrets.

I start out by selecting a set of suitable actions that fit the scenario. I define them as follows.

\begin{align*}
call^a_b &:= shareall_{\{a,b\}} \\
message^a_b &:= sendall_{\{b\}}
\end{align*}

Here $\text{call}^a_b$ is the call between agents $a$ and $b$ in which they share all the notes (or secrets) they possess. Later on I will also be interested in what happens if the agents can only leave voicemail messages instead of making two-way calls. For this purpose I use $\text{message}^a_b$, where agent $a$ sends all secrets he possesses to agent $b$. Let $A = \bigcup_{a,b \in Ag} \text{call}^a_b \cup \bigcup_{a,b \in Ag} \text{message}^a_b$.

Next, I define the information sets of the agents. For every agent $a$, I define his set of notes as $N_a = \{s_a\}$, where $s_a$ is his secret. Let $S$ be the set of all secrets. The communication network allows for pairwise communication between the agents. I define it as $Net = \{\{a, b\} | a, b \in A\}$. Then the initial state is

$s_I := \langle Net, \{s_1\}, ..., \{s_{|Ag|}\}, \varepsilon, \{s_1\}, ..., \{s_{|Ag|}\}, (\Sigma A)^* \rangle$.

I want to vary the communicative powers of the agents in different situations. Therefore I will define different protocols that restrict the actions the agents can execute. I define $\pi_{call} := (\bigcup_{a,b \in Ag} \text{call}^a_b)^*$, $\pi_{mail} := (\bigcup_{a,b \in Ag} \text{message}^a_b)^*$ as the protocols where the agents can only make telephone calls or only send voicemails, respectively.
In order to reason about the number of calls the agents need to make to reach their goal, I will use the following abbreviations:

\[ \diamondsuit_n \phi := \left( \bigcup_{k \leq n} (\Sigma A)^k \right) \phi \]

\[ \diamondsuit_{\text{min}(n)} \phi := \diamondsuit_n \phi \land \neg \diamondsuit_{n-1} \phi \]

\[ \diamondsuit_n \phi \] expresses that a state where \( \phi \) holds can be reached by sequentially executing at most \( n \) actions from \( A \). \( \diamondsuit_{\text{min}(n)} \phi \) expresses that \( n \) is the minimal such number. Note that \( A \) does not contain any actions that change the protocol, therefore the formulas express whether the agents can achieve \( \phi \) with the current protocol. Note that the temporal operator \( \diamondsuit \) (sometimes called \( F \)) of IS approaches (e.g. [Pacuit and Parikh, 2007]) can be defined by \( \langle (\Sigma A)^* \rangle \) while \( \diamondsuit_n \) serves as a generalization of the arbitrary announcement that is added to DEL in [Agotnes et al., 2009].

Then the following result states that exactly \( 2|Ag| - 4 \) calls are necessary to make sure every agent knows all secrets:

**4.4.5. Lemma.** For any \( \text{obs} \in \{ \text{set}, 1st, asyn, \tau \} \),

\[ s_I \models_{\text{obs}} \langle \text{exprot}(\pi_{\text{call}}) \rangle \diamondsuit_{\text{min}(2|Ag| - 4)} \bigwedge_{a \in Ag} \text{has}_a S. \]

A proof of this proposition is given in [Hurkens, 2000]. The protocol given there is the following: pick a group of four agents 1 ... 4 and let 4 be their informant. Let agent 4 call all other agents, then let the four agents communicate all their secrets within their group and let all other agents call agent 4 again. In my framework this can be expressed as follows:

\[ \text{call}\{1, 2, 3, 4\}, \ldots; \text{call}\{1, 2, 3, 4\}; \text{call}\{1, 2, 3, 4\}, \text{call}\{1, 2, 3, 4\}; \text{call}\{1, 2, 3, 4\}, \ldots; \text{call}\{1, 2, 3, 4\} \]

Now I turn to the question that arises when the agents cannot make direct telephone calls, but they can only leave voicemail messages. This means that any agent can tell the secrets he knows to another agent, but he cannot in the same call also learn the secrets the other agent knows. How many voicemail messages would the agents need in this case?

The agents could use \( \text{message}_b^a \) to mimic each \( \text{call}_b^n \), which gives

\[ s_I \models_{\text{obs}} \langle \text{exprot}(\pi_{\text{mail}}) \rangle \diamondsuit_{\leq 4|Ag|-8} \bigwedge_{a \in Ag} \text{has}_a S. \]

However, they can do much better, as the following lemma shows.

**4.4.6. Lemma.** For any \( \text{obs} \in \{ \text{set}, 1st, asyn, \tau \} \),

\[ s_I \models_{\text{obs}} \langle \text{exprot}(\pi_{\text{mail}}) \rangle \diamondsuit_{\text{min}(2|Ag|-2)} \bigwedge_{a \in Ag} \text{has}_a S. \]

4.4. Applications

**Proof.** Consider the following protocol:

\[ \text{message}_1^2; \text{message}_2^3; \ldots; \text{message}_{|Ag|-1}^{|Ag|}; \text{message}_1^{|Ag|}; \text{message}_2^{|Ag|}; \ldots; \text{message}_{|Ag|-1}^{|Ag|}. \]

Clearly, this results in all agents knowing all secrets. The length of this protocol is \(2|Ag|-2\). I claim that this protocol is minimal. To see why this claim holds, first observe that there has to be one agent who is the first to learn all secrets. For this agent to exist all other agents will first have to make at least one call to reveal their secret to someone else. This is already \(|Ag|-1\) calls. The moment that agent learns all secrets, since he is the first, all other agents do not know all secrets. So each of them has to receive at least one more call in order to learn all secrets. This also takes \(|Ag| - 1\) calls which brings the total number of calls to \(2|Ag|-2\). \(\Box\)

As the above results show, it is possible to make sure all agents know all secrets. However, in these results the secrets are not common knowledge yet, since the agents do not know that everyone knows all secrets. I will investigate whether common knowledge of all secrets can be established. I will assume that prior to the start of the protocol, the distribution of the secrets is common knowledge. For this purpose I use the following abbreviation:

\[ \text{SecDis}_Ag := \bigwedge_{a \in Ag} (\text{has}_a s_a \wedge \bigwedge_{b \neq a} \neg \text{has}_b s_a) \]

If there are only three agents, then achieving common knowledge of all secrets is possible by making telephone calls:

**4.4.7. Lemma.** If \(|Ag| \leq 3\) then for some \(n \in \mathbb{N}\):

\[ s_I \models ^\tau \langle \text{exinfo}(\text{SecDis}_Ag); \text{exprot}(\pi_{\text{call}}) \rangle \otimes^n C_{Ag} \bigwedge_{a \in Ag} \text{has}_a S. \]

**Proof.** For \(|Ag| < 3\) the proof is trivial. Suppose \(|Ag| = 3\), say \(Ag = \{1, 2, 3\}\). A protocol that results in the desired property is \(\text{call}^1_2; \text{call}^2_3; \text{call}^2_1\). After execution of this protocol all agents know all secrets, and agent 2 knows this. Also, since agent 1 learned the secret of agent 3 from agent 2, he knows that agent 2 and 3 must have communicated after the last time he spoke to agent 2, so agent 3 must know the secret of agent 1. Regarding agent 3, he knows agent 2 has all secrets the moment he communicated with agent 2, and he observed a \(\tau\) when agent 2 called agent 1 after that. Since there are only three agents, agent 3 can deduce that agent 1 and 2 communicated so he knows agent 1 knows all secrets. Since all agents can reason about each other’s knowledge, it is common knowledge that all agents have all secrets. \(\Box\)
I do not extend this result to the case with more than three agents. If there are more than three agents, agents that are not participating in the phone call will never know which of the other agents are calling, which makes it much harder to establish common knowledge.

Now imagine a situation where the agents are beforehand allowed to publicly announce a specific protocol they are going to follow which is more complex than just the set of actions they can choose from. Then, in the $\tau$-semantics, it is possible to reach common knowledge:

4.4.8. Proposition. There is a protocol $\pi$ of call actions such that

$$s_I \models^{\tau} \langle \text{exinfo}(\text{SecDis}_{Ag}) \rangle \langle \text{exprot}(\pi) \rangle \diamond \leq n C_{Ag} \wedge \bigwedge_{a \in Ag} \text{has}_a S$$

Proof. Let $\pi$ be the protocol given in the proof of proposition 4.4.5. Since each agent observes a $\tau$ at every communicative action, they can all count the number of communicative actions that have been executed and they all know when the protocol has been executed. So at that moment, it will be common knowledge that everyone has all secrets. □

This shows the use of the ability to communicate about the future protocol and not only about the past and present. There are many more situations where announcing the protocol is very important, for example in the puzzle of 100 prisoners and a light bulb [Dehaye et al., 2003] and in many situations in distributed computing.

4.5 Conclusion

In this chapter I proposed an expressive framework that combines properties from dynamic epistemic logic and interpreted systems. The framework is very flexible and it can be adapted to almost any situation that concerns communication and knowledge. I specifically include the communication network in my set-up, which allows for reasoning about the network and about the agents’ knowledge of the network. I showed how this framework can be used to model communication by applying it to the example with the telephone calls mentioned in the introduction of this chapter.

The framework is very flexible in modeling different observational powers of agents and various communicative actions. For example, the communicative action that is used in [Pacuit and Parikh, 2007], “$a$ gets $b$’s information without $b$ noticing this”, can be modeled as $\alpha = \text{download}_b^a$ with $\text{Obs}(\alpha) = \{a\}$, $\text{Pre}(\alpha) = \text{com}(\{a, b\})$ and a postcondition containing $N_a := N_a \cup N_b$. Because of the freedom in the design of the actions and observational powers, this framework can facilitate the comparison of different approaches with different assumptions.