Coalitions in Epistemic Planning

MSc Thesis *(Afstudeerscriptie)*

written by

Suzanne van Wijk
(born 9 July 1991 in Leiden)

under the supervision of **Dr. Alexandru Baltag**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam.*

Date of the public defense:	**Members of the Thesis Committee:**
3 September 2015	Dr. Alexandru Baltag
	Prof. Dr. Johan van Benthem
	Dr. Roberto Ciuni
	Prof. Dr. Jan van Eijck
	Dr. Jakub Szymanik

Institute for Logic, Language and Computation

Abstract

The aim of this thesis is to augment dynamic epistemic logic and its framework in order to model planning problems where *coalitions* of agents try to reach a given goal. We add an additional *control* relation to the static epistemic models and action models of DEL, similar to choice equivalence in stit logics, thereby enabling us to represent the power of coalitions while keeping the means to talk about specific actions. We then introduce a sound, complete and decidable logic for these augmented models, which can express knowledge, distributed knowledge and both past and future control of coalitions, and we demonstrate how this can be used for coalitional planning.
We then add common knowledge to the logic, in order to model the coordination of agents within a coalition: indeed, common knowledge enables agents to trust the other coalition members to perform the right action. As reduction axioms cannot be found for common knowledge, we show soundness, completeness and decidability for our enriched version of epistemic PDL, where we take as basic programs group indistinguishability relations rather than single agent indistinguishability.
Finally, the thesis proposes a way in which agents can *commit* to certain actions, providing them with a way to communicate their plans and coordinate, thereby greatly improving their possibilities for achieving their goal.

Acknowledgments

The thesis lying before you would not exist in its current form without the help and support of many different people, who helped the process of its creation in many different ways - be it in terms of content or in terms of general support.
First of all, I would like to thank my supervisor, Alexandru Baltag. To start, I am very grateful for your enthusiasm, wonderful ideas and perseverance in the early days of this thesis. You put so many interesting ideas on the table that I am sad we could only work on one of them. Furthermore, thank you for your continuous support, comments and help throughout the writing of this document. Your enthusiasm for logic, and dynamic epistemic logic in particular, is very contagious and kept me engaged in the topic the entire way through.
Moreover, I am grateful to the other members of the thesis committee, Johan van Benthem, Jan van Eijck, Jakub Szymanic and Roberto Ciuni for taking the time to read my thesis. I also have to thank Roberto for his comments and corrections, suggestions for references and for his bringing in a fresh look on things in an earlier stage.
My gratitude also goes to Thomas Bolander. Thomas, thank you for two very interesting meetings in the early stages of this thesis, and the insightful and spot-on remarks that greatly influenced the direction this thesis took.
Then last but not least, a big shout-out to Thomas, Julian and Bastiaan for all the lunches, coffee breaks, late night dinners and rants that kept me sane in those last couple of weeks. I'm glad I never have to know what they would have been like without you.

True knowledge exists in knowing that you know nothing
- Socrates

Contents

1 Introduction

In the field of automated planning, the main goal is to create software for the problem of one or more agents creating a long-term plan to reach their goal. To ensure that it is computationally feasible to solve non-trivial such problems, a number of constraints is put on these problems in Classical Planning as defined by Ghallab et al. [18]. In classical planning, the problems have to be finite, static, fully observable and deterministic. To lift some of these requirements, Bolander and Andersen proposed epistemic planning [10]. This builds on classical planning, but uses Dynamic Epistemic Logic to build a planning problem, thereby lifting the full observability and determinacy constraints.

Dynamic Epistemic Logic (DEL) was created around 2000 by multiple authors. It models the knowledge of agents, and how this knowledge changes when events occur. It is based on the assumption that the world is not fully observable nor fully deterministic, as it deals mainly with what different agents *are* able to distinguish or observe. Gerbrandy laid the ground works with his logic for private announcements in a subgroup [17], where the subgroup learns what is being said, but the others do not. Baltag, Moss and Solecki generalized the existing framework with 'event models' in [5], which turned out to be a crucial addition. It has since then been a grateful research subject, as is witnessed by the extensive literature - see for example an overview from 2008 by Baltag, van Ditmarsch and Moss [6] or van Benthem [26].

In DEL, every agent has their own *indistinguishability* relation, which determines what states of the world look the same to that agent. If all states that are indistinguishable to some agent make the same property true, we can say that this agent *knows* this property. Also dynamically, DEL assumes partial observability, as some of the events that can occur appear the same to an agent, just like states can look the same. This results in indeterminacy, since from then on, the agent should consider it possible that either of those events happened.

In his PhD dissertation, Andersen [2] started the groundwork for multi-agent epistemic planning by generalizing previous work with Bolander and others to include multi-agent models to their epistemic framework. However, even though the planning problems are defined on multi-agent plausibility models, there is still only one acting agent. As it is interesting to look at multi-agent planning with multiple acting agents, we try to approach this from a different direction.

In this thesis we construct a framework that deals with coalitions of agents cooperating in an epistemic planning domain. There are many logics around that express power of coalitions, such as Pauly's coalition logic [24], CTL and CTL*, introduced by Prior [25] and Clark and Emerson [14], ATL, introduced by Alur,

Henziger and Kupferman [1], and multi-agent stit, extended from stit by Belnap et al. [8] and Horty [20]. Stit, short for *seeing to it that*, talks about what an agent or coalition brings about, and originated from Belnap en Perloff [7]. It was continued in many forms in for example Belnap, Perloff and Xu [8], Horty [20] and many others. A lot of work has been done in connecting these logics with each other: Broersen, Herzig and Troquard [11] defined a translation from coalition logic to stit, and Ciuni and Zanardo connected stit and branching-time logics such as CTL and ATL [13]. Also much has been done to connect the above logics of coalition power with epistemic logics: van der Hoek and Wooldridge proposed an epistemic extension of ATL, which they called ATEL [31], which was later extended by Jamroga and van der Hoek [22]. Van Benthem and colleagues link models for DEL and those for epistemic temporal logics in [27], allowing concepts from either type of logic to carry over to the other. In van Benthem and Pacuit [29], stit and dynamic logics of events are connected by embedding stit in matrix games, and the comparison is pushed further by Ciuni and Horty [12]. Van Benthem and Pacuit [29] also hint at how DEL and stit can be combined. It is the latter direction that is followed in this work.

This thesis defines a framework that takes its main components from DEL and stit. We add an *control* relation to model the control a coalition has. In contrast to what van Benthem and Pacuit suggest in [29], we add this relation to the static as well as the action models to allow for memory of control. The same approach is taken in DEL logics of question that use issue relations, such as DEL_Q, proposed by van Benthem and Minica [28]. We define a logic for these models that also has components from DEL and stit, which we call Dynamic Epistemic Coalition Logic (DECL). It takes modalities for knowledge, distributed knowledge, and events from DEL and modalities for control from stit. We show that this logic is sound, complete and decidable. As there are no axioms in standard modal logic to express that a relation is exactly the intersection of other relations, our distributed knowledge modality posed some technical difficulties. By following the lines of the proof in Fagin et al. [16], we avoid these problems and still obtain the desired results.

To make the logic more expressive and useful for our purpose, we add common knowledge to DECL. To show completeness for this logic, we need to extend it further to our version of Epistemic PDL as introduced by van Benthem, van Eijck and Kooi [30]. Our version differs from that of van Benthem and colleagues in the basic programs. Where they take basic programs to be epistemic relations of single agents, we take them to be epistemic relations of *groups* of agents, which is why we call it Group Epistemic PDL. This not only gives us what we want - distributed and common knowledge, completeness and decidability - but it also opens up possibilities for higher levels of group knowledge, such as common distributed knowledge.

Finally, we add a special type of atomic sentences to the language, which allow the agents to commit to certain actions. This makes it possible to model the coordinations of agents in a planning problem fully within the logic, rather than using an external framework for it. As we only add atomic sentences, the completeness and decidability results of DECL with common knowledge carries over immediately.

The rest of this thesis is organized as follows: in chapter 2 we will briefly go over the preliminaries needed for this paper. We will introduce the main concepts from DEL, as well as explain the parts of stit logics that we need, and we give a short introduction to (epistemic) planning.
In chapter 3 we introduce the framework that we will be working with, which takes many concepts from the DEL framework, and adds a *control* relation, which is similar to choice equivalence in stit. In this chapter we also define Dynamic Epistemic Coalition Logic. We will go over some examples of what this language can express and we show that it is sound, complete and decidable. When we apply it to a planning problem, we will see where this logic falls short for that purpose.
In chapter 4 we add common knowledge to dynamic epistemic coalition logic in order to arrive at a logic that is better suited for application to a planning problem. We show that our version of E-PDL, Group Epistemic PDL, which is an extended version of DECL with common knowledge, is sound, complete and decidable. We continue to use this logic to define a planning problem and solution, and again conclude that, although improved, it falls slightly short.
Therefore, in chapter 5 we allow agents to *commit* to actions, making it easier for them to coordinate while planning. We argue that the logic including commitments is still sound, complete and decidable, and we illustrate how committing helps a coalition in creating a solution for a given planning problem.
We conclude the thesis with a summary of the presented logics and what they can or cannot express, before we go on to mention some ideas for future research.

2 Preliminaries

In this chapter we briefly go over definitions and conventions from the fields that are used in this thesis. We will start with an introduction to Dynamic Epistemic Logic (DEL), after which we will go over the main ideas of Seeing To It That logics (stit). To conclude, we will briefly mention automated planning, and epistemic planning in particular.

2.1 Dynamic Epistemic Logic

Dynamic Epistemic Logic describes what agents know about the world and how this changes when they interact with it and each other. Alice might not know whether it is raining, but she will after Bob tells her that it is. There is a lot of literature and research on dynamic epistemic logic, and in this chapter we only go over the basics. If the reader is interested to know more, they can consult for instance [5, 26].

As the name suggest, DEL deals with an ever changing world. This means that it uses two different types of models. The first type of models represents the world as it is at a given time. We call this the static models, and these are the models at which formulas will be evaluated. The second type of model is used to describe the changing of the world. These are called action models, and they consist of one or multiple *events*. These events will change the initial model, either by changing facts about the world, or by changing what agents know about facts of the world. We will introduce both static and action of models, and how we combine them when events occur.
First we will introduce the language of DEL. There are some variants of this that may include common knowledge, distributed knowledge or other modalities, but we will stick to the most basic language.

Definition 2.1 ($\mathcal{L}_{DEL}$)**.** The language $\mathcal{L}_{DEL}$ is formed by the following Backus-Naur form

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid [\sigma]\varphi$$

where p is a propositional letter coming from a finite set P of propositional atoms (denoting 'ontic facts'), i is an 'agent' from a finite set $\mathcal{A}$ of agents and σ an 'epistemic event'or 'epistemic action' from some finite set Σ of 'event names'.
We take $K_i\varphi$ to mean that agent i knows φ, and $[\sigma]\varphi$ to mean that φ holds after execution of σ.

2.1.1 Static models

The static models of DEL are traditionally based on Kripke models, where the states represent varying configurations of the world and the relations between the worlds indicate what configurations the agents consider to 'look the same'. Hence,

if two states are connected for some agent, we say that the agent cannot distinguish between the two. Very often DEL makes use of *pointed* Kripke models, which directly indicate the actual, or real world. This is the state which the modeler, as all-knowing onlooker, knows to be the real one. This is however not required and we will define the static models without it.

A multi-agent epistemic model shows which agents consider which states look the same. Hence, it consists of a nonempty, finite set of worlds, an indistinguishability relation for each agent and a valuation function, that determines which propositional letters are true at what states.

Definition 2.2 (Multi-agent epistemic models)**.** A *multi-agent epistemic model* is a tuple $\mathbf{S} = \langle S, \sim_i, V \rangle_{i \in \mathcal{A}}$ such that:

- S is a nonempty finite set of *states*;
- for each agent $i \in \mathcal{A}, \sim_i \subseteq S \times S$ is an equivalence relation called the *indistinguishability relation*;
- $V : P \to \mathcal{P}(S)$ is a *valuation function*, assigning sets of states to each propositional letter.

As we evaluate formulas at a specific state, the notion of indistinguishability is similar to the notion of 'considering possible', if not the same. Hence, the state where it rains looks the same to Alice as the state where it does not, thus at either of these states she considers the other possible.

Example 2.3. Consider a situation where Alice flipped a coin, but did not look at it yet. She does not know whether it landed heads or tails, and neither does Bob, so they considers both possible. This situation is modeled in Figure 1, where h means that coin came up heads and $\neg h$ that it came up tails.

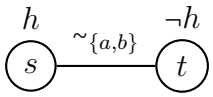

Figure 1: A multi-agent epistemic model

In this figure, and from now on, we leave out the transitive and reflexive relations for the sake of readability, and the reader should remind themselves of the fact that indistinguishability relations are equivalence relations. Hence, at every world, both agents consider that world possible.

As one can see, in this model neither Alice nor Bob can distinguish the two states, and they both consider it possible that the coin landed heads and that it landed tails.

2.1.2 Action Models

In the previous part we showed how DEL models what agents know at a certain point in time. When the agents interact with each other or with the world, this knowledge can change. The simplest example of such an interaction is a public announcement, where one of the agents or another entity announces a certain fact. After such an event, all agents that heard the announcement know the fact, and moreover, every agent knows that every agent knows, etcetera. Another example is an agent turning on the light, therewith not only changing the knowledge of the other agents, but also facts about the world. We call these interactions with the world and other agents*events*, and they are modeled in so-called action models. As the static models, these are Kripke models. Each state in the action model is called an event. An action, and thus an action model, can consist of multiple events because it might be the case that (some of) the agents cannot distinguish between events. For example, if Alice tosses a coin in such a way that Bob cannot see it, she will know whether it landed heads or tails, but Bob will not, which means that we need two events: one where the coin landed heads, and one where it landed tails. Formally:

Definition 2.4 (Action model)**.** An *action model* is a tuple $\mathbf{\Sigma} = \langle \Sigma, \sim_i, \text{pre}, \text{post} \rangle_{i \in \mathcal{A}}$ such that:

- Σ is the nonempty, finite set of event names (also known as 'actions') of the above language $\mathcal{L}_{DEL}$;
- for each agent $\sim_i \subseteq \Sigma \times \Sigma$ is an equivalence relation called the *indistinguishability relation*;
- $\text{pre} : \Sigma \to \mathcal{L}$ is a function assigning a *precondition* to each event;
- $\text{post} : \Sigma \to (P \to \mathcal{L})$ is a function assigning a *postcondition* to each event.

Intuitively, the precondition tells us when an event can happen. For example, we can only walk through a door if it is opened. The postcondition tells us how the event changes the facts of the world. Hence, it tells us that after the event 'switch the light on' is performed, the light is on. Many action models used by for DEL do not have postconditions, as they only change the epistemic states of the agents. Public announcements for example only change what information agents have about the world, but it does not change any facts about the world. However, in this thesis the ontic change is necessary, so we include the postconditions in the action models.

Example 2.5. To continue the previous example, remember that Alice flipped a coin, but neither her nor Bob had seen it yet. Now we will model the action of Alice checking how the coin landed. The action model is depicted in Figure 2, and as one can see, it consists of two events. One event is where Alice checks

the coin to see that it landed heads, and the other to find tails. As Bob is not checking with Alice, he does not know what she finds, so for him the events are indistinguishable. However, the events are distinguishable for Alice, because the moment she sees the coin, she knows which event took place.

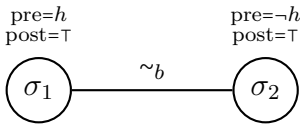

Figure 2: The action model

Note that in this example, the postcondition is $\top$, meaning that no facts about the states changed in the execution of these events, only what agents know about those facts.

2.1.3 Product Update

Now that we know how to model the static world we live in, and the actions that changes this world, we need a notion of *how* the actions change the world. This is done using *product updates*. First the formal definition:

Definition 2.6 (Product update)**.** The *product update* of an epistemic model $\mathbf{S}$ and an action model $\mathbf{\Sigma}$ is a tuple $\mathbf{S} \otimes \mathbf{\Sigma} = \langle S', \sim_i', V' \rangle$, such that:

- $S' = \{(s, \sigma) \in S \times \Sigma; \mathbf{S}, s \vDash \mathrm{pre}(\sigma)\}$;
- $\sim_i' = \{((s, \sigma), (s', \sigma')) \in S' \times S'; s \sim_i s' \text{ and } \sigma \sim_i \sigma'\}$;
- $V'(p) = \{(s, \sigma) \in S'; \mathbf{S}, s \vDash \ \mathrm{post}(\sigma)(p)\}$.

What happens here? First of all, the new set of worlds consists of the Cartesian product of the states and events, leaving out those combinations where the state does not satisfy the precondition of the event. Hence, we try to combine every state with every event, but if the event is not possible in that state, the combination does not get formed. The new indistinguishability relation is such that two states are related in the product if and only if both the old states and the events were related. This means that an agent had to be both unsure about the state, and about the event that happened. The valuation gets adjusted according to the postcondition.

Example 2.7. To illustrate the concept of the product update, consider Alice and Bob and their coin again. We will perform the product update between the two models we defined before, which is shown in Figure 3.
There are a few things to note here. First of all, (s, σ_2) and (t, σ_1) did not form since the preconditions of the events did not match up with the valuation of the

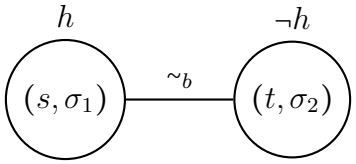

Figure 3: updated model

states. Secondly, as we would expect, Alice now knows the value of the coin whereas Bob still does not, and thirdly, the valuation did not change because the action was not one that changed facts.

2.2 Seeing to it that

In philosophy and computer science, *seeing to it that* logics (stit) are very popular to talk about agency and obligation (see e.g. Belnap et al. [8]). Stit formalizes what an agent chooses to do, or to bring about. Many different versions have been proposed over the years, which is why we will often talk about stit logics.

Formulas from stit logics are often evaluated on *branching time* structures. These consist of a finite set of *moments* that are ordered using a strict partial order with no backwards branching. The idea is that the ordering 'groups' the moments into histories, which represent different ways the world can develop. As in real life, the past is determined, which is represented by the no backwards branching, whereas the future can have multiple outcomes. Intuitively, at every branching point in the structure, the agent can make a choice between the branches at that point. The choice the agent ends up making can influence the way the world looks afterwards. This leads to a relatively intuitive notion of *seeing to it that*: if an agent chooses in such a way that in the next moment φ is true, then he is seeing to it that φ. In the literature, this is often denoted by $[i \ stit \ \varphi]$. The single agent version of stit has been extended to multi-agent stit by Belnap and colleagues [8] and Horty [20], where $[J \ stit \ \varphi]$ is used to say that the agents in J see to it that φ.

As mentioned, over the years many different variants of the original stit logic have been proposed. These include deliberative stit (*dstit*, see Horty and Belnap [21]), where an agent not only sees to it that φ, but also had an alternative that would have resulted in something different, and achievement stit (*astit*), which instead of talking about what an agent is about to bring about, talks about what the agent has already brought about by previous actions (see Belnap et al. [8]). It is this latter one that is related most to the modality $[I]$ that we will introduce in this thesis. As *astit*, it talks about what a coalition has enforced by a prior choice, thereby expressing the power that coalition enforced in some previous moment.

2.3 Planning

Automated planning is a field connected to artificial intelligence that concerns itself with creating long-term plans for agents to achieve some predetermined goal. That which we now call Classical Planning stems from the early 60's and 70's and is defined by Ghallab et al. [18]. They define a planning problem as an initial state, which models the way the world is currently, a transition system, and a set of goal states. The transition system determines which states there are, which actions are available and how the actions change the states. Formally, a transition system as defined by Ghallab et al. looks as follows.

Definition 2.8 (Restricted State-Transition System)**.** Any classical planning domain can be represented as a *restricted state-transition system* $\Sigma = \langle S, A, \gamma \rangle$ where

- S is a finite or recursively enumerable set of states
- A is a finite set of actions
- $\gamma : S \times A \to S$ is a computable, partial state transition function.

Note that the transition function is just defined - from this state with this action, we go to this state. It is a function, and thus determined.

A planning problem is then defined as follows:

Definition 2.9 (Classic Planning Problem)**.** A *classic planning problem* is a tuple $\langle \Sigma, s_0, S_g \rangle$, where

- Σ is a transition system
- s_0 is the initial state
- S_g is the set of goal states

A *solution* to a classic planning problem is a finite sequence of actions, called a plan, such that after this sequence the result is a state in S_g.

Classic planning requires that any planning problem be fully observable, deterministic, finite and static to ensure that planning problems are computationally easier to solve. Another consequence of these restrictions is that a solution is also theoretically easier to construct, as it does not take into account that the world is only partially observable, or that other agents might be acting in it as well.

Bolander and Andersen ([10]) proposed a new method of planning, which they call *epistemic planning*. For their planning problems they lifted the constraints of full observability and determinacy, making it suitable for multi-agent planning in a partially observable world.

2.3.1 Epistemic Planning

Bolander's epistemic planning uses the concepts from DEL, mentioned before, to define a planning problem that does not require that the world be fully observable or deterministic. Instead of a predefined, deterministic transition system as used by Ghallab et al., epistemic planning makes use of *epistemic events* to define how the world changes. Events in DEL are designed to be used in a partially observable framework, and are by definition non-deterministic, which immediately lifts two requirements of classical planning.

Using these events to define how the world changes allows for an agent-dependent view of the world, which ensures that agents can plan for what they know or do not know, and allowing different agents knowing different things about the world. Bolander and Andersen therefore define their state-transition system differently:

Definition 2.10 (Epistemic Planning Domain)**.** Given a finite set P of propositions and a finite set $\mathcal{A}$ of agents, an *epistemic planning domain* on $(P, \mathcal{A})$ is a restricted state-transition system $\Sigma = \langle S, A, \gamma \rangle$ where

- S is a finite or recursively enumerable set of epistemic states
- A is a finite set of actions
- γ is defined by

$$\gamma(s, a) = \begin{cases} s \otimes a & \text{if } s \vDash pre(a) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Aside from the way states transition into one another, an epistemic planning problem as defined by Bolander and Andersen in [10] is similar to a classic planning problem:

Definition 2.11 (Epistemic Planning Problem)**.** An *epistemic planning problem* is a tuple $\langle \Sigma, s_0, \varphi_g \rangle$ where

- Σ is an planning domain
- s_0 is the initial state
- φ_g is the goal formula. The set of goal states consists of those states where φ_g holds

A solution is still a finite sequence of actions such that after execution of all these actions, φ_g holds in the updated model.

The main difference between classical and epistemic planning problems is the way in which actions lead to new states. In a classic planning problem the transition

system Σ determines a partial transition function γ, which defines what state we arrive in after a combination of action and state took place. In an epistemic planning problem, Σ consists of a set of actions and states, and γ is determined by the product update of actions applied to states.

What makes epistemic planning especially interesting is that it allows one to look into *conditional* planning (see for example [3]): situations where an agent does not have all necessary information *yet*, but knows that she will get it after a certain action. She can then conditionalize her plan, to say that if she finds out a, she will do σ, whereas if she finds out $\neg a$ she will do σ', and still be sure to reach her goal because she knows that she will find out either a or $\neg a$. This means that even though an agent is not sure about the world, she can still create a long-term plan in such a way that she is sure to reach her goal.

To formalize this, Andersen and colleagues introduce the concept of a solution to a planning problem [3]. They say a sequence of actions is a *strong* solution if it is the case that every step is executable and the agent knows that after the sequence happened, the goal holds. A sequence of actions is a *weak* solution if every action in the sequence is executable at the right step, and the agent does not know that the goal does not hold after execution. Hence, a strong solution is a sequence of actions such that it is guaranteed to reach the goal, whereas a weak solution is a sequence of actions such that it is possible that it reaches the goal.

3 Power of Coalitions

DEL provides us with a way to talk about what agents know, and how this changes when they interact with each other and the world, and is already used to model a planning problem where one agent plans his course of actions in a world with incomplete information [10, 3]. However, one can conceive of situations where agents cannot perform a task or reach a goal on their own. They might need someone else's knowledge, or they are simply incapable of performing a crucial action themselves. In this chapter we introduce Dynamic Epistemic Coalition Logic, or DECL for short, for exactly these situations. It models what *coalitions* are able to achieve by performing one or more actions. As in coalition logics such as ATL and related logics, Pauly's Coalition Logic or STIT-logics [24, 1, 8], DECL keeps track of what coalitions can achieve, and, like DEL [26], it uses specific actions. Hence, rather than merely stating that a coalition can reach a certain goal, it can also talk about the specific action that brings this about. Combining these properties gives us a way to talk about solutions to planning problems for coalitions.

The models and logic that we will be using are inspired by the fact that, whenever anyone performs an action, the ultimate result is hardly ever determined. Alice might decide that she goes dancing, but whether or not Bob will join her, is up to him rather than her. So when she chooses to go dancing, in fact she chooses to go dancing *independent* of whether Bob goes as well. Another example is the situation where one can perform the action of flipping a coin, but one cannot beforehand decide that one is going to flip the coin and that it will land heads. There are many different factors that can alter the outcome of an action, and our framework models the control of an agent or coalition over the world, by making explicit that which it *cannot* control.

In this framework, performing an action is therefore modeled as choosing a 'group' of events. Then, when all agents, and possibly nature, have chosen an action, the actual event that will happen gets determined. So when Alice decided to go dancing, this included the event where Bob would join her and the one where he would not. Only when he makes his choice is it determined whether they will go together or not. Hence, this framework works with the assumption that we do not always have full control over the consequences of our actions, and that there are other decisions, made by either other agents or some external force like nature, that influence the result of our action.

From now on, whenever we talk about an *event*, we mean a determined, single event. An example is the event of some agent flipping a coin and it landing heads up. The event of Alice and Bob both going dancing. When we say *action*, we mean something an agent can decide to do. It will most often consist of multiple events. Hence, Alice going dancing is an action that consists of two events: the

one where Bob joins her and the one where he does not.

This gives rise to a notion of controllability, or forcing, rather intuitively. If an agent can choose an action in such a way that, no matter what anyone else does, φ holds, we say that this agent can *force* φ. Thus if every event in an action of an agent has the same result, we say that the agent forces that result, as none of the other agents can change it once the first agent makes up their mind.

3.1 Dynamic Epistemic Coalition Logic

In this chapter we introduce Dynamic Epistemic Coalition Logic (DECL), which combines Dynamic Epistemic Logic and components from stit logics to model coalitional planning. In this section, we first give the syntax of DECL, after which we continue with its models and its product update. After this, we discuss some examples, and finally we show that the logic is sound, complete and decidable.

3.1.1 Syntax of DECL

Definition 3.1 (The Language $\mathcal{L}_{DECL}$)**.** Let Σ be a finite set of 'action names', $\mathcal{A}$ be a finite set of 'agents', $0 \notin \mathcal{A}$ be a symbol denoting 'Nature' (seen as a non-agent force that comprises all the influences that are beyond agents' control) and P be a set of propositional letters, denoting 'ontic' (i.e. non-epistemic) facts. . Then $\mathcal{L}_{DECL}$ has the following Backus-Naur form:

$$p \mid \neg\varphi \mid \varphi \wedge \varphi \mid D_I\varphi \mid [J]\varphi \mid [\sigma]\varphi$$

Where $I \subseteq \mathcal{A}$ and $J \subseteq \mathcal{A} \cup \{0\}$ are coalitions and $\sigma \in \Sigma$.
We call the static fragment of $\mathcal{L}_{DECL}$ without the dynamic modality $\mathcal{L}_{DECL^-}$.

We take $[I]\varphi$ to mean that the agents in I have forced φ and $D_I\varphi$ to mean that φ is distributed knowledge among the agents in I, which is traditionally interpreted as a situation where, if all agents in I combine their knowledge, they will all know that φ.
As in DEL, $[\sigma]\varphi$ means that after the event σ happened, φ is the case.

Abbreviations

- We write $K_i\varphi$ for $D_{\{i\}}\varphi$ to mean that agent i knows φ
- We will write $[\sigma_I]\varphi$ for $\bigwedge_{\sigma' \approx_I \sigma} [\sigma']\varphi$, to mean that the agents in I can together enforce φ by each of the agents in I choosing the equivalence class that contains σ.
- We'll write $\Diamond_I\varphi$ for $\bigvee_{\sigma\in\Sigma} [\sigma_I]\varphi$, to say that the agents in I can enforce φ, by choosing their actions wisely.

- $\langle I \rangle$ is the dual of $[I]$.

3.1.2 Models and Product Update

The models that we use are based mainly on the multi-agent epistemic models and action models of DEL. Besides having an epistemic indistinguishability relation, we also have a *control* relation, or *choice* relation, which is an equivalence relation that models the control an agent can exercise. In our action models, this relation defines a partition on the events, where each equivalence class is a specific action of that agent. In the static models, the control relation keeps track of what agents or a coalition have previously forced. It is important to realize that the control relation is extra, and does not replace the indistinguishability relation. In fact, it complements it, as we assume that if an agent cannot tell two events apart, these two events should be in the same choice equivalence class, and similarly in the static models: if an agent forced something, he must know it. This seems like an intuitive constraint, as we see actions as the choice of an agent. If an agent cannot distinguish between two events, it makes sense that it is impossible for him to choose the one, but not the other, because they look the same to him.

Formally, the action models that we use are defined as follows:

Definition 3.2 (Action Control Model)**.** An *action model* for the language $\mathcal{L}_{DECL}$ is a structure
$\mathbf{\Sigma} = \langle \Sigma, \sim_i, \approx_j, \text{pre}, \text{post} \rangle_{i \in \mathcal{A}, j \in \mathcal{A} \cup \{0\}}$ where

- Σ is the non-empty set of action names of the language $\mathcal{L}_{DECL}$;
- $\sim_i$ is an equivalence relation for each agent $i \in \mathcal{A}$ called the *indistinguishability relation*;
- $\approx_j$ is an equivalence relation for every $j \in \mathcal{A} \cup \{0\}$ such that for all $\sigma \in \Sigma$ we have $\bigcap_{j \in \mathcal{A} \cup \{0\}} [\sigma]_{\approx_j} = \{\sigma\}$;
- $\text{pre} : \Sigma \to \mathcal{L}_{DECL}$ is a function called the *precondition* mapping actions to formulas of $\mathcal{L}$;
- $\text{post} : \Sigma \to (P \to \mathcal{L})$ is a function called the *postcondition*.
- for all $i \in \mathcal{A}$ we require $\sim_i \subseteq \approx_i$

There are some remarks to be made about the action models.

1. The control relation $\approx_i$ is also defined for 0. Here, 0 denotes 'nature', or 'environment', which consists of all external forces that are beyond the control of the agents, but might still influence the current event. Clearly, 0 does not get an epistemic relation.

2. When every agent and 0 has made a choice of action, the combination determines the event. We can make these action models determined, exactly because of 0, as anything that is not directly determined by the agents alone can be explained as being determined by 'nature'.

3. We require that $\sim_i \subseteq \approx_i$ to ensure that agents can only choose what they know or can distinguish.

In this thesis we always work with one big action model. This model contains all possible events that could at some point happen. All these events are partitioned according to the control relations, and at each moment, every agent chooses one of his equivalence classes. The intersection of all equivalence classes is then the event that will actually happen.

In the static models, we have an analogous extra equivalence relation. Instead of showing dynamic control, here it shows control by *past choices*. Hence, we say that if an entire equivalence class satisfies a formula, the agent *has forced* that formula. Note the difference in time when comparing with the action models: in the static models, having forced φ means having made choices in the past, such that φ is now the case, whereas in the action models, forcing φ means the current choice will result in φ, no matter what the other agents choose.

Definition 3.3 (Static Epistemic Control Model)**.** A *static epistemic model* is a structure $\mathbf{S} = \langle S, \sim_i, \approx_j, V \rangle_{i \in \mathcal{A}, j \in \mathcal{A} \cup \{0\}}$ such that

- S is a non-empty set of *states*;
- $\sim_i$ is an equivalence relation for each agent $i \in \mathcal{A}$ called the *indistinguishability relation*;
- $\approx_j$ is an equivalence relation forevery $j \in \mathcal{A} \cup \{0\}$ such that for all $s \in S$ we have $\bigcap_{j \in \mathcal{A} \cup \{0\}} [s]_{\approx_j} = \{s\}$;
- $V : P \to \mathcal{P}(S)$ is the *valuation function* that maps propositional letters to subsets of states.
- for all $i \in \mathcal{A}$ we require $\sim_i \subseteq \approx_i$

This relational way of defining control equivalence is an alternative semantics for stit logics, as proposed first by Kooi and Tamminga [23], and was later shown to be equivalent to the usual semantics on branching-time models by Herzig and Schwarzentruber [19]. Balbiani and colleagues gave another axiomatization of stit based on this new semantics in [4].

Now that we have action control models and static epistemic control models, we have to define the product update. This works almost the same as product updates

with ontic change in DEL, with the addition that two states are $\approx_i$ related if both the original states and the events were. This captures the intuition of keeping track of the history, as it requires not only that an agent or coalition chose an action in a certain way, but it also demands something of how things came to be before they chose that action.

Definition 3.4 (Product Update)**.** Let $\mathbf{S}$ be a static epistemic model and $\mathbf{\Sigma}$ an action model. Then the product update of $\mathbf{S}$ and $\mathbf{\Sigma}$ is $\mathbf{S} \otimes \mathbf{\Sigma} = \langle S', \sim_i', \approx_j', V' \rangle_{i \in \mathcal{A}, j \in \mathcal{A} \cup \{0\}}$ such that

- $S' = \{(s, \sigma) \in S \times \Sigma; S, s \vDash \mathrm{pre}(\sigma)\}$
- $\sim_i' = \{((s, \sigma), (s', \sigma')) \in S' \times S'; s \sim_i s' \text{ and } \sigma \sim_i \sigma'\}$
- $\approx_j' = \{((s, \sigma), (s', \sigma')) \in S' \times S'; s \approx_j s' \text{ and } \sigma \approx_j \sigma'\}$
- $V'(p) = \{(s, \sigma) \in S'; \mathbf{S}, s \vDash \mathrm{post}(\sigma)(p)\}$.

As one can see from the product update, having a control relation in the static models as well allows us to keep track of control throughout multiple numbers of actions. If we do not do this, as van Benthem and Minica suggested in [28], it is almost as if we start on a fresh canvas after each product update, and previous forcing actions are forgotten. By enabling the remembering of control, we in fact also enable coalitions planning multiple steps ahead, while keeping control over what they forced. We will get back to this later in Example 3.12.

Example 3.5. An agent flips a coin. As mentioned before, the agent can only decide to throw it, but not how it lands. Hence the events where the coin lands heads and where it lands tails are connected by the control relation of the agent. This is depicted in Figure 4.

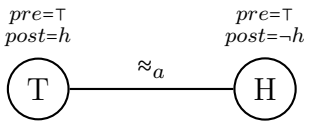

Figure 4: The action control model of flipping a coin

Note that in this example, the agent has complete information about the world and the events - they only do not have full control. Following is an example where one agent neither has full control, nor complete information.

Example 3.6. Alice and Bob want to visit Charlie, but Alice does not know whether to go left or right at the intersection. Bob has been there before, so he knows. Consider the static model and action model in Figures 5 and 6.
Alice knows that Bob knows whether to go left or right, but of course, she cannot *make* him tell her. He is the one who decides whether to do that or not. Given that we assume a cooperative setting, we suppose that Bob will tell her, after which the updated model looks as in Figure 7.

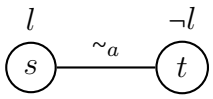

Figure 5: The initial model

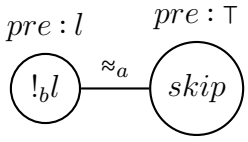

Figure 6: The action model

Figure 7: The updated model

3.1.3 Semantics and Examples

Definition 3.7 (Satisfaction)**.** The *satisfaction* of formulas of DECL on static epistemic models, denoted $\mathbf{S}, s \vDash \varphi$ is defined as follows:

- $\mathbf{S}, s \vDash p$ iff $s \in V(p)$
- $\mathbf{S}, s \vDash \neg\varphi$ iff $\mathbf{S}, s \nvDash \varphi$
- $\mathbf{S}, s \vDash \varphi \vee \psi$ iff $\mathbf{S}, s \vDash \varphi$ or $\mathbf{S}, s \vDash \psi$
- $\mathbf{S}, s \vDash D_I\varphi$ iff $\forall s' \sim_I s : \mathbf{S}, s' \vDash \varphi$
- $\mathbf{S}, s \vDash [I]\varphi$ iff $\forall s' \approx_I s : \mathbf{S}, s' \vDash \varphi$
- $\mathbf{S}, s \vDash [\sigma]\varphi$ iff $(s, \sigma) \in \mathbf{S} \otimes \mathbf{\Sigma}$ implies $\mathbf{S} \otimes \Sigma, (s, \sigma) \vDash \varphi$

Where $\approx_I := \bigcap_{i \in I} \approx_i$ and $\sim_I := \bigcap_{i \in I} \sim_i$.
It is clear that the case where one agent forced something is a special case where $I = \{i\}$.

We now discuss some examples that illustrate what DECL is capable of expressing.

Example 3.8 (A coalition can achieve something the separate agents cannot)**.** Agent a is in the process of stealing a diamond from a vault. She is, however, in a wheelchair, so she called in help from agent b, who has to push her around. Currently, they are in the vault($\neg o$, for *not outside*), while agent a is carrying the diamond (d). The initial model is depicted in Figure 8. They would like to be standing outside the vault while still carrying the diamond, hence the goal formula is $\varphi_g := d \wedge o$. Both agents currently have two actions they can choose from. Agent a can hold on to the diamond (H), or she can drop it (D - for the sake of the example we assume dropping the diamond is a conscious decision), and agent b can either push agent a outside (P) or he can stay where they are (S). The action model is depicted in Figure 9.

Figure 8: The initial model

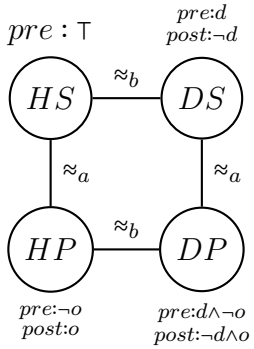

Figure 9: The action model

One can see that there is exactly one event in this action model such that after this event, φ_g holds, and that is if b pushes and a holds on to the diamond (HP). Hence, together they can ensure that they reach their goal: $s \vDash [HP_{\{a,b\}}]\varphi_g$, but each agent on his own cannot: $s \vDash \neg \Diamond_{\{a\}} \varphi_g$ and $s \vDash \neg \Diamond_{\{b\}} \varphi_g$.

Example 3.9 (One agent knows that cooperating can achieve the goal, but the other does not)**.** Suppose $\varphi_g = p$ and currently agent a does not know whether p, but agent b does know. This situation is depicted in Figure 10. The agents can now both decide whether they want to flip the truth value of p, or whether they want to do nothing. This action is shown in Figure 11. One can check that $s \vDash K_b[flip_{\{a,b\}}]\varphi_g \wedge K_b\neg[flip_{\{b\}}]\varphi_g$. Hence, agent b *knows* that he has to cooperate with agent a to be sure to achieve his goal. Furthermore, it is the case that $s \vDash \neg K_a[flip_{\{a,b\}}]\varphi_g$, so agent a does not know that flipping the truth value of p will achieve their goal, since she isn't sure about the initial truth value. Hence, b cannot be sure that she will decide to perform $flip$, and thus if they do not coordinate, he cannot be sure that they reach their goal.

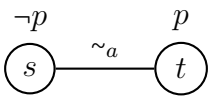

Figure 10: The initial model

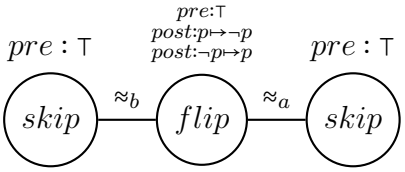

Figure 11: The action model

Example 3.10 (Two agents know different ways of achieving their goal)**.** Consider the initial static model in Figure 12. Agent a knows that p and agent b knows that q. Also consider the action model in Figure 13, where we see that only if they both choose either $!p$ or $!q$ something happens, otherwise the event $skip$ occurs. If p is the case, then the action $!p$ will achieve the goal, but if it isn't it will achieve the opposite. Similarly for q: if it is the case, then action $!q$ will achieve the goal,

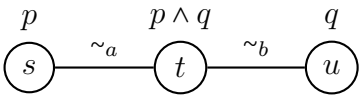

Figure 12: The initial model

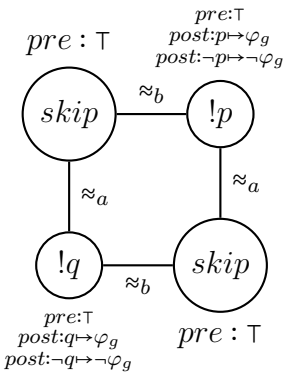

Figure 13: The action model

and if not, it will achieve the opposite. Since a knows p, but not q, she will want to do action $!p$, whereas agent b will want to do action $!q$, as he knows that will achieve the goal. Hence, $t \vDash K_a[!p_{\{a,b\}}]\varphi_g \wedge K_b[!q_{\{a,b\}}]\varphi_g$. However, also note that both agents do not know that the other agent's favorite action will also lead to the goal: $t \vDash \neg K_b[!p_{\{a,b\}}]\varphi_g \wedge \neg K_a[!q_{\{a,b\}}]\varphi_g$. Hence, they both need one another, but they both want to achieve the goal in a different way.

Example 3.11 (Causing)**.** We have been talking about agents *forcing* a certain outcome, but we can express something more. Agent i forcing φ might, after all, have been a coincidence. φ might have been unavoidable, which makes it immediate that i forced it, rather than a result of something agent i did. Thus we would like to express the notion of 'causing', where it is really the actions of an agent that made the world the way it is. Thus we say that agent i *caused* φ if and only if he forced it, *and* he could have avoided it. So we can express that agent i caused φ with

$$[i]\varphi \wedge \langle \mathcal{A} \cup \{0\} \setminus i \rangle \neg \varphi$$

As this thesis is concerned about the power of coalitions, it is interesting to extend this single-agent causing to coalitions. A coalition causing a certain formula entails again that they forced it, but also that if *any* of the members of the coalition had done something else, the result could have been different. This last part means that a coalition caused something if the entire coalition was needed to force it. In the language, we express this by

$$[I]\varphi \wedge \bigwedge_{i \in I} \langle \mathcal{A} \cup \{0\} \setminus \{i\} \rangle \neg \varphi$$

This notion of causing relates more to the *dstit* operator introduced by Horty and Belnap [21] than our earlier notion of forcing did, in the same way that for them *dstit* also means that the agent had another choice, but decided in favor of this particular action.

Example 3.12 (Remembering control)**.** We have previously claimed that including a control relation in the static models allows us to keep track of control throughout multiple actions. In this example we will show what we mean with this.

Suppose a coalition I can perform a joint action σ_I, such that after all events in that action, φ holds. Then surely it is the case that $[\sigma_I]\varphi$. It is natural to say that then, after the joint action happened, the agents in I forced φ, so $[\sigma_I][I]\varphi$. This is however not valid, and we only need a simple counterexample to show this:

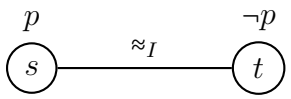

Figure 14: Initial model S

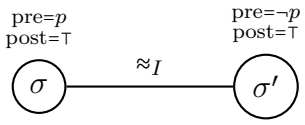

Figure 15: Action model Σ

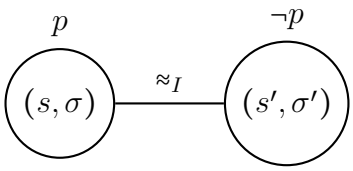

Figure 16: The updated model $S \otimes \Sigma$

As the reader can check, it is the case that $\forall \sigma' \approx_I \sigma : S \otimes \Sigma, (s, \sigma') \vDash p$, and thus that $S, s \vDash \bigwedge_{\sigma' \approx_I \sigma} [\sigma']\varphi$. Hence indeed, $S, s \vDash [\sigma_I]\varphi$. However, $(s', \sigma') \nvDash p$, and thus it is not the case that $\forall \sigma' \approx_I \sigma : \forall (s', \sigma') \approx_I (s, \sigma) : (s', \sigma') \vDash p$, and therefore $\forall \sigma' \approx_I \sigma : (s, \sigma') \nvDash [I]p$, and hence also $\mathbf{S}, s \nvDash [\sigma_I][I]p$.

Hence, we do not have that $\vDash [\sigma_I]\varphi \to [\sigma_I][I]\varphi$. The reason for this is that, even though the coalition I can force φ from the situation as it is *now*, it could very well be that there is some other coalition J that made the situation the way it is now. Hence, we cannot attribute φ completely to I, as J also played a crucial role. One could argue that it is such an intuitive implication that we should add it as a requirement to the model. However, suppose that we do add it as a requirement, so suppose we force the implication $[\sigma_I]\varphi \to [\sigma_I][I]\varphi$. The following would then also be valid:

$$\mathbf{S}, s \vDash [\sigma_i]\varphi \Rightarrow \mathbf{S}, s \vDash [\sigma_i][i]\varphi \Rightarrow \mathbf{S}, s \vDash [\sigma_i]K_i\varphi \Rightarrow {}^1\mathbf{S}, s \vDash K_i[\sigma_i]\varphi$$

Hence, forcing the implication leads to the validity that if an agent has an action with which they can force φ, they know that with this action they can force φ.

[1]Proof: $\mathbf{S}, s \vDash [\sigma_i]K_i\varphi \Rightarrow \mathbf{S}, s \vDash \bigwedge_{\sigma' \approx_i \sigma} [\sigma']K_i\varphi \Rightarrow \forall \sigma' \approx_i \sigma : (S \otimes \Sigma), (s, \sigma') \vDash K_i\varphi \Rightarrow \forall \sigma' \approx_i \sigma : \forall (s', \sigma'') \sim_i (s, \sigma') : (S \otimes \Sigma), (s', \sigma'') \vDash \varphi \Rightarrow \forall \sigma' \approx_i \sigma : \forall s' \sim_i s : (S \otimes \Sigma), (s', \sigma') \vDash \varphi \Rightarrow \forall s' \sim_i s : S, s' \vDash [\sigma_i]\varphi \Rightarrow S, s \vDash K_i[\sigma_i]\varphi$

This is not desirable, as it completely blurs the difference between *having* a strategy and *knowing that* one has a strategy, which was the core of many discussions in coalition logics, see e.g. [22].

The knowledge that $\not\models [\sigma_I]\varphi \to [\sigma_I][I]\varphi$ because the current situation might have been due to some other coalition does give rise to another implication that has a similar intuition, but which *is* a validity: $\models [I][\sigma_I]\varphi \to [\sigma_I][I]\varphi$. This implication says that if, by previous actions, the agents in I forced the fact that they are now in a position where they can force φ by choosing σ_I, then after performing this action, they will have forced φ.
This exactly says what we want: if the agents in I made the world the way it is, and they can now perform an action such that afterwards φ, then clearly after that action, they forced φ. This follows from the fact that not only can they from the way the world is now, force φ, but they forced the way the world is now as well.

3.1.4 Proof System of DECL

Now that we have shown some examples of what DECL can express, we present its proof system, and show that it is sound, complete and decidable.

- All axioms and rules of classical propositional logic
- Necessitation rules for all modalities
- S5 for $[I]$ for all $I \subseteq \mathcal{A} \cup \{0\}$
- S5 for D_I for $I \subseteq \mathcal{A}$
- **Knowledge of (Individual) Control**
  $[I]\varphi \to D_I\varphi$ for $I \subseteq \mathcal{A}$
- **Monotonicity of Control**
  $[I]\varphi \to [J]\varphi$ for $I \subseteq J \subseteq \mathcal{A} \cup \{0\}$
- **Monotonicity of Distributed Knowledge**
  $D_I\varphi \to D_J\varphi$ for $I \subseteq J \subseteq \mathcal{A}$
- **Determinism of Grand Coalition**
  $\varphi \to [\mathcal{A} \cup \{0\}]\varphi$

In addition to the axioms for the static language, we have reduction axioms that will form the basis of reducing the dynamic language to the static language.

Reduction Axioms

- $[\sigma]p \leftrightarrow (\mathrm{pre}(\sigma) \to \mathrm{post}(\sigma)(p))$
- $[\sigma]\neg\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \neg[\sigma]\varphi)$
- $[\sigma](\varphi \wedge \psi) \leftrightarrow [\sigma]\varphi \wedge [\sigma]\psi$
- $[\sigma]D_I\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \bigwedge_{\sigma \sim_I \sigma'} D_I[\sigma']\varphi)$
- $[\sigma][I]\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \bigwedge_{\sigma \approx_I \sigma'}[I][\sigma']\varphi)$

Some remarks about these axioms are in order. First, notice that $[i]\varphi \to K_i\varphi$ is a special case of $[I]\varphi \to D_I\varphi$. It is easily argued that, after one agent forced something, he knows it. One might, however, argue that in the multi-agent case we would prefer something stronger: indeed one can argue that we would like it to be the case that after a coalition forced φ, they commonly know that φ. In response, we point out that this axiom, in fact, emphasizes the meaning of the modality $[I]$, as this in fact *does not* mean that forcing φ was a conscious decision for a coalition of agents. No one agent in the group may intend φ, and it could even be the case that no one is aware that it is being forced, but nonetheless, it is being forced by the actions chosen by the members of the coalition. In Section 4 we add common knowledge to the language of DECL, with which we can express this conscious forcing.

Secondly, both monotonicity axioms imply that a bigger coalition is always more powerful, both in terms of knowledge as in forcing power. In our setting this is a reasonable assumption, as everyone works together. In a setting where agents might try to thwart one another, it could be interesting to allow smaller coalitions to be more powerful.

3.2 Soundness, Completeness and Decidability of DECL

We first show completeness for $\mathcal{L}_{DECL^-}$, the fragment of $\mathcal{L}_{DECL}$ without dynamic modalities. Afterwards we show that the latter can be reduced to the former, implying completeness for the full language.

3.2.1 Preliminaries

Before we can start the proof, we give some definitions and general results that are used later on.

Definition 3.13 (Filtration)**.** Let S be a general Kripke model, and $\Sigma \subseteq \mathcal{L}$ a set of formulas. The relation $\equiv_\Sigma$ on S, defined as

$$s \equiv_\Sigma t \Leftrightarrow \text{ for all } \sigma \in \Sigma : (S, s \vDash \sigma \Leftrightarrow S, t \vDash \sigma)$$

defines an equivalence on S; we denote its equivalence classes with $[s]_\Sigma$, but we will often leave out the subscript if Σ is clear from context. The model $S^f = \langle S^f, \sim_I^f, \approx_J^f, V^f \rangle$ is a *filtration of S through Σ* if

- $S^f = \{[s]; s \in S\}$
- For each $R_\Box^f \in \{\sim_I^f, \approx_J^f; I \subseteq \mathcal{A}, J \subseteq \mathcal{A} \cup \{0\}\}$ the following hold:

  min^f For all $[s], [t] \in S^f$, if $sR_\Box t$, then $[s]R_\Box^f[t]$.

  max^f For all $[s], [t] \in S^f$, if $[s]R_\Box^f[t]$, then for all $\Box\varphi \in \Sigma(S, s \vDash \Box\varphi \to S, t \vDash \varphi)$

- $V^f(p) = \{[s]; s \in V(p)\}$

Lemma 3.14. *Let S^f be a filtration of a general Kripke model S through some Σ. Then for all $[s] \in S^f$ and $\varphi \in \Sigma$, we have $S, s \vDash \varphi \Leftrightarrow S^f, [s] \vDash \varphi$.*

Proof. The proof is by induction on the complexity of the formula.
Base case. Suppose $\varphi = p$ for some $p \in P$.
Then by definition $[s] \in V^f(p) \leftrightarrow s \in V(p)$, and hence $S, s \vDash \varphi \Leftrightarrow S^f, [s] \vDash \varphi$.
Inductive step. The boolean cases are straightforward.
Suppose φ is of the form $\Box\psi$ for some $\Box \in \{\sim_I, \approx_J; I \subseteq \mathcal{A}, J \subseteq \mathcal{A} \cup \{0\}\}$.
For the left-to-right direction, suppose $S, s \vDash \Box\psi$. Now take an arbitrary $[t] \in S^f$ such that $[s]R_\Box^f[t]$. Then by the max^f condition of filtrations and the fact that $S, s \vDash \Box\psi$, we get that $S, t \vDash \psi$. But then by the induction hypothesis $S^f, [t] \vDash \psi$, and thus $S^f, [s] \vDash \Box\psi$.
For the other direction, suppose $S^f, [s] \vDash \Box\psi$. Hence, for all $[t] \in S^f$ such that $[s]R_\Box^f[t]$, we have that $S^f, [t] \vDash \psi$. By the induction hypothesis, we get $S, t \vDash \psi$. Now take an arbitrary $u \in S$ such that $sR_\Box u$. Then, as $sR_\Box u$, we get by the min^f condition of filtrations that $[s]R_\Box^f[u]$, and thus that $S, u \vDash \psi$. Hence $S, s \vDash \Box\psi$. □

Definition 3.15 (Bounded Morphism)**.** Let $S = \langle S, \sim_I, \approx_J, V \rangle_{I \subseteq \mathcal{A}, J \subseteq \mathcal{A} \cup \{0\}}$ and $S' = \langle S', \sim_I', \approx_J', V' \rangle_{I \subseteq \mathcal{A}, J \subseteq \mathcal{A} \cup \{0\}}$ be two regular Kripke models. A mapping $f : S \to S'$ is a *bounded morphism* if the following hold for all $I \subseteq \mathcal{A}$ and $J \subseteq \mathcal{A} \cup \{0\}$:

1. for all $s \in S : s \in V(p)$ if and only if $f(s) \in V'(p)$.
2. (a) for all $s, t \in S$, if $s \sim_I t$ then $f(s) \sim_I' f(t)$
   (b) for all $s, t \in S$, if $s \approx_J t$ then $f(s) \approx_J' f(t)$
3. (a) for all $s \in S$ and $t' \in S'$, if $f(s) \sim_I' t'$, then there exists a $t \in S$ such that $s \sim_I t$ and $f(t) = t'$.
   (b) for all $s \in S$ and $t' \in S'$, if $f(s) \approx_J' t'$, then there exists a $t \in S$ such that $s \approx_J t$ and $f(t) = t'$.

Proposition 3.16. *Let* $S = \langle S, \sim_I, \approx_J, V\rangle_{I\subseteq\mathcal{A}, J\subseteq\mathcal{A}\cup\{0\}}$ *and* $S' = \langle S', \sim'_I, \approx'_J, V'\rangle_{I\subseteq\mathcal{A}, J\subseteq\mathcal{A}\cup\{0\}}$ *be two regular Kripke models, and let* $f : S \to S'$ *be a bounded morphism. Then for any formula* φ *of DECL*$^-$ *and world* $s \in S$*, it is the case that* $S, s \vDash \varphi$ *iff* $S', f(s) \vDash \varphi$*.*

Proof. The proof is by induction on the complexity of φ:
Base case: The base case follows from the definition of bounded morphisms.
Inductive step: The Boolean cases where $\varphi = \neg\psi$ and $\varphi = \psi_1 \wedge \psi_2$ follow immediately, which leaves only the modalities.
Suppose $S, s \vDash D_I\psi$. Take an arbitrary $t' \in S'$ such that $f(s) \sim'_I t'$. According to condition $3(a)$ there exists a $t \in S$ such that $s \sim_I t$ and $f(t) = t'$. By the first consequence we get that $S, t \vDash \psi$, and by the second and the induction hypothesis we get that $S', f(t) \vDash \psi$. Hence it is the case that $S', f(s) \vDash D_I\psi$.
The case for $[I]$ is analogous, which completes the proof. □

3.2.2 Plan of the Proof

We are now ready to start the completeness proof, but before we start we first give a brief explanation of the steps we will take, as this will make it easier for the reader to follow the main argument throughout the proof itself. The completeness proof uses the method introduced by Fagin et al.[15], and consists of three steps.

1. Step 1: Soundness and Completeness for Pseudo-models. First, we create *pseudo-models*. These are structures that look like static epistemic control models, but have separate $\sim$ and $\approx$ relations *for every subset* of agents, instead of just one for every agent. We then show soundness of DECL$^-$ with respect to these pseudo-models and argue that as static epistemic control models are a special case of pseudo-models, the logic is sound with respect to static epistemic control models. Then we define the canonical pseudo-model and prove completeness of DECL$^-$ with respect to this.

   (b) Step 1b: Decidability. In a small detour, we use our version of a Fischer-Ladner closer to filtrate the canonical pseudo-model. Using this, we obtain a finite pseudo-model, with which we prove decidability of DECL$^-$.

2. Step 2: Unraveling. In the second step, we *unravel* the canonical pseudo-model. This means that we create *all* possible histories in the pseudo-model: paths that can be taken when we follow the $\sim$ and $\approx$ relations. These histories are related in such a way that they form a tree.

3. Step 3: Completeness of DECL$^-$. In the third step, we take the tree we just created, and from it define a static epistemic control model. We do this by defining the proper relations, and showing that this newly created structure satisfies the necessary semantic properties. Then we define a bounded morphism between the canonical pseudo-model and the static epistemic control model, which makes completeness with respect to those models immediate.

4. Step 4: Completeness of DECL. In the fourth step, we show how the dynamic language DECL can be reduced to the static language DECL$^-$, thereby showing that the previously obtained results carry over to the dynamic language.

3.2.3 The Proof

STEP 1: Soundness and Completeness for Pseudo-models

Definition 3.17 (Pseudo-Model)**.** A *pseudo-model* is a structure $M = \langle S, \sim_I, \approx_I, V \rangle$, where

- $\sim_I$ and $\approx_I$ are equivalence relations;
- $\sim_I \subseteq \approx_I$;
- for $J \subseteq I$:$\sim_I \subseteq \sim_J$ and $\approx_I \subseteq \approx_J$;
- $\approx_{\mathcal{A} \cup \{0\}} = \mathrm{id}$.

It is clear that all epistemic control models are in fact pseudo-models, as they have the same requirements except for also requiring that $\approx_I := \bigcap_{i \in I} \approx_i$ and $\sim_I := \bigcap_{i \in I} \sim_i$.

Proposition 3.18 (Soundness)**.** *All axioms of DECL are valid on pseudo-models.*

Proof. Let $M = \langle S, \sim_I, \approx_I, V \rangle$ be an arbitrary pseudo-model. We will show that all axioms are valid on M. The proof is per axiom.

- All $S5$ axioms follow easily.
- $[I]\varphi \to D_I\varphi$: Suppose $M, s \vDash [I]\varphi$. Take an arbitrary $t \in S$ such that $s \sim_I t$. It follows that $s \approx_I t$, thus we have that $M, t \vDash \varphi$, and hence for any t such that $s \sim_I t$, we get $M, t \vDash \varphi$, thus $M, s \vDash D_I\varphi$.
- $[I]\varphi \to [J]\varphi$ for $I \subseteq J$: Suppose $M, s \vDash [I]\varphi$. Take an arbitrary $t \in S$ such that $s \approx_J t$. From our requirements, it follows that $s \approx_I t$, and thus $M, t \vDash \varphi$. Thus for all t such that $s \approx_J t$ we have $M, t \vDash \varphi$, and hence $M, s \vDash [J]\varphi$.
- $D_I\varphi \to D_J\varphi$ for $I \subseteq J$: Similar as above.
- $\varphi \to [\mathcal{A} \cup \{0\}]\varphi$: Assume $Ms, \vDash \varphi$, and take an arbitrary $t \in S$ such that $s \approx_{\mathcal{A} \cup \{0\}} t$. Then, as $\approx_{\mathcal{A} \cup \{0\}} = \mathrm{id}$, we get that $t = s$, and thus $M, t \vDash \varphi$, and thus for all t such that $s \approx_{\mathcal{A} \cup \{0\}} t : M, t \vDash \varphi$, hence $M, s \vDash [\mathcal{A} \cup \{0\}]\varphi$.

□

Now to prove completeness with respect to the pseudo-models, I will build the canonical structure M^C.

Definition 3.19 (Canonical structure)**.** The canonical structure is a general Kripke model $M^C = \langle S, \sim_I, \approx_I, V \rangle$, such that

- $S = \{s; s \text{ is a maximally consistent set of } DECL \text{ formulas}\}$
- $s \sim_I t$ iff $\forall\varphi(D_I\varphi \in s \Rightarrow \varphi \in t)$
- $s \approx_I t$ iff $\forall\varphi([I]\varphi \in s \Rightarrow \varphi \in t)$
- $V(p) = \{s \in S; p \in s\}$.

Note that the relations can alternatively be defined as follows (see Blackburn *et al.* [9]):

- $s \sim_I t$ iff $\forall\varphi(\varphi \in t \Rightarrow \hat{D}_I\varphi \in s)$
- $s \approx_I t$ iff $\forall\varphi(\varphi \in t \Rightarrow \langle I\rangle\varphi \in s)$

Proposition 3.20. *Let M^C be the canonical structure as described above. Then M^C is a pseudo-model.*

Proof. All semantic properties are treated separately.

- To show that $\sim_I$ is reflexive, suppose $D_I\varphi \in s$. Then by our axioms, $\varphi \in s$. From the definition of $\sim_I$, it follows that id $\subseteq \sim_I$, and thus that $\sim_I$ is reflexive.
- To show that $\sim_I$ is transitive, suppose $s \sim_I t \sim_I s$, and suppose $D_I\varphi \in s$. By axiom 4, we get that $D_ID_I\varphi \in s$. From that we obtain that $D_I\varphi \in t$, and thus $\varphi \in s$. Thus, by definition of $\sim_I$, we have that $s \sim_I s$.
- To show that $\sim_I$ is symmetric, suppose $s \sim_I t$ and $\varphi \in s$. Then by axiom (B), we have $D_I\hat{D}_I\varphi \in s$. Since $s \sim_i t$, we then get that $\hat{D}_I \in t$. By definition, this means that $t \sim_I s$.
- Equivalence for $\approx_I$ follows the same lines as the previous case.
- To show that $\sim_I \subseteq \approx_I$, suppose $s \sim_I s'$, and let $[I]\varphi \in s$. Then since we have the axiom $[I]\varphi \to D_I\varphi$, we get that $D_I\varphi \in s$. But then as $s \sim_I s'$, this means that $\varphi \in s'$, and thus $s \approx_I s'$.
- To show that $\sim_I \subseteq \sim_J$ for $J \subseteq I$, assume $s \sim_I s'$, and let $D_J\varphi \in s$. Then since $D_J\varphi \to D_I\varphi$ is an axiom, we get $D_I\varphi \in s$. But then as $s \sim_I s'$, we get that $\varphi \in s'$, and hence $s \sim_J s'$.
- $\approx_I \subseteq \approx_J$ for $J \subseteq I$ is similar to the $\approx$ case.
- To show that $\approx_{\mathcal{A}\cup\{0\}} =$ id, suppose that $s \approx_{\mathcal{A}\cup\{0\}} s'$, and let $\varphi \in s$. Then by the axiom $\varphi \to [\mathcal{A} \cup \{0\}]\varphi$, we get that $[\mathcal{A} \cup \{0\}]\varphi \in s$. But then, as $s \approx_{\mathcal{A}\cup\{0\}} s'$, we have that $\varphi \in s'$. Hence, $s \subseteq s'$, but as both are maximally consistent sets, and hence s' cannot be strictly bigger than s, it must be the case that $s = s'$, and thus $\approx_{\mathcal{A}\cup\{0\}} =$ id.

□

Hence we showed that the canonical structure is in fact a pseudo-model. As all axioms are valid on pseudo-models, they are also valid on the canonical structure.

Lemma 3.21 (Truth Lemma)**.** $M^C, s \vDash \varphi$ *iff* $\varphi \in s$.

Proof. In [9, p.199], the Truth Lemma is proved for any normal modal logic and any canonical model, hence it also holds for M^C. □

Proposition 3.22. *The logic DECL⁻ is sound and complete with respect to pseudo-models.*

Proof. We showed soundness before.
To show completeness, suppose Γ is a consistent set of formulas from $\mathcal{L}_{DECL}$. By the Lindenbaum Lemma it follows that in the canonical pseudo-model there is a $s \in S$ such that $\Gamma \subseteq s$. From the Truth Lemma it follows that $M^C, s \vDash \Gamma$. Hence Γ is true in the canonical pseudo-model, and since the M^C is a pseudo-model, Γ is satisfiable in pseudo-models. □

STEP 1b: Decidability

In this step we will show that $\mathcal{L}_{DECL}$ is decidable, by using a filtration to create a finite model that satisfies the same formulas as the canonical structure we ended up creating in Step 1.

First we have to define our version of the Fischer-Ladner closure. We call this a suitable set.

Definition 3.23 (Closed set under single negation)**.** Let Γ be a set of formulas. Then Γ is closed under single negation if and only if $\varphi \in \Gamma$ implies that $\sim \varphi \in \Gamma$, where

$$\sim \varphi = \begin{cases} \psi & \text{if } \varphi = \neg\psi \\ \neg\psi & \text{else} \end{cases}$$

Definition 3.24 (A suitable set)**.** Let φ be in the language. Then Σ_φ is *a suitable set for* φ if it is the smallest set such that;

(1) $\varphi \in \Sigma_\varphi$

(2) Σ_φ is closed under subformulas

(3) Σ_φ is closed under single negation

(4) $D_I\varphi \in \Sigma_\varphi$ implies $D_J D_I \varphi \in \Sigma_\varphi$ for $I \subset J$

(5) $[I]\varphi \in \Sigma_\varphi$ implies $[J][I]\varphi \in \Sigma_\varphi$ for $I \subset J$

(6) $[I]\varphi \in \Sigma_\varphi$ implies $D_I[I]\varphi \in \Sigma_\varphi$

(7) $\varphi \in \Sigma_\varphi$ implies $[\mathcal{A} \cup \{0\}]\varphi \in \Sigma_\varphi$ if φ is *not* of the form $[\mathcal{A} \cup \{0\}]\psi$ for some ψ.

Lemma 3.25. *Let Σ_φ be a suitable set for some $\varphi \in \mathcal{L}$. Then Σ_φ is finite.*

Proof. Define $\Sigma^0_\varphi := \{\psi; \psi \text{ is a subformula of } \varphi\}$.
Then as φ is defined by recursion, which ensures that the subformula relation is well-founded, we get that this set is finite.
Now let

$$\begin{aligned}\Sigma^1_\varphi := \Sigma^0_\varphi &\cup \{D_{J_1}D_{J_2}\dots D_{J_m}D_I\theta; D_I\theta \in \Sigma^0_\varphi, I \subset J_m \subset \dots \subset J_2 \subset J_1\} \\ &\cup \{[J_1][J_2]\dots[J_m][I]\theta; [I]\theta \in \Sigma^0_\varphi, I \subset J_m \subset \dots \subset J_2 \subset J_1\} \\ &\cup \{D_{J_1}\dots D_{J_m}D_{I_1}[I_1][I_2]\dots[I_n]\theta; [I_n]\theta \in \Sigma^0_\varphi, I_n \subset \dots I_1 \subset J_m \dots \subset J_1\}\end{aligned}$$

Note that every separate part of Σ^1_φ is finite, as it must be the case that all sequences are finite - they can be as most as long as there are agents), and that thus there can only be finitely many (different) sequences.
The reader can check that Σ^1_φ is closed under constraints (1), (2) and (4) - (7).
Finally, let $\Sigma_\varphi := \Sigma^1_\varphi \cup \{\sim\theta; \theta \in \Sigma^1_\varphi\}$.
Clearly, $|\Sigma_\varphi| \leq 2 \times |\Sigma^1_\varphi|$, so also Σ_φ is finite, and clearly closed under all constraints. □

Definition 3.26. Let $M = \langle S, \sim_I, \approx_I, V\rangle$ be a general Kripke model, and consider a suitable set Σ_φ for some $\varphi \in \mathcal{L}$. Then we define a general Kripke model $M^+ = \langle S^+, \sim^+_I, \approx^+_I, V^+\rangle$ such that

- $S^+ = \{[s]_{\Sigma_\varphi}; s \in S\}$;
- $[s] \sim^+_I [t]$ if and only if $\forall D_I\psi \in \Sigma(M, s \vDash D_I\psi \Leftrightarrow M, t \vDash D_I\psi)$;
- $[s] \approx^+_I [t]$ if and only if $\forall [I]\psi \in \Sigma(M, s \vDash [I]\psi \Leftrightarrow M, t \vDash [I]\psi)$;
- $V^+(p) = \{[s]; s \in V(p)\}$.

Lemma 3.27. *M^+ is a filtration of M through Σ.*

Proof. Clearly, M^+ satisfies the constraints on S^f and V^f, so it is left to show that $\sim^+_I$ and $\approx^+_I$ satisfy min^f and max^f.

- min^f for $\sim^+_I$: Take an arbitrary $[s], [t] \in S^+$ such that $s \sim_I t$, and suppose $D_I\psi \in \Sigma_\varphi$. Then, as $\sim_I$ is an equivalence relation, we get that $M, s \vDash D_I\psi$ if and only if $M, t \vDash D_I\psi$. Hence, by definition, we get $[s] \sim^+_I [t]$.
- max^f for $\sim^+_I$: Take arbitrary $[s], [t]$ such that $[s] \sim^+_I [t]$, and suppose $D_I\psi \in \Sigma_\varphi$ and that $M, s \vDash D_I\psi$. Then again, $M, t \vDash D_I\psi$. But as $\sim_I$ is reflexive, we get that $M, t \vDash \psi$.

- Both min^f and max^f are analogous for $\approx_I^+$.

□

Corollary 3.28. *Let $M = \langle S, \sim_I, \approx_I, V\rangle$ be a general Kripke model, and let M^+ and Σ_φ be as above. Then for all $\sigma \in \Sigma_\varphi$ and $s \in S$, we get that*

$$M, s \vDash \sigma \Leftrightarrow M^+, [s] \vDash \sigma$$

Proof. This follows from Lemma 3.14 and Lemma 3.27. □

Theorem 3.29. *Let $M = \langle S, \sim_I, \approx_I, V\rangle$ be a pseudo-model. Then M^+ constructed as described above is a pseudo-model.*

Proof. To show that M^+ is a pseudo-model we have to show that it satisfies all the semantic properties of pseudo-models.

- It is clear that $\sim_I^+$ and $\approx_I^+$ are equivalence relations.
- $\sim_I^+ \subseteq \approx_I^+$: Take arbitrary $[s], [t] \in S^+$ such that $[s] \sim_I^+ [t]$. This means that $\forall D_I\psi \in \Sigma_\varphi (M, s \vDash D_I\psi \Leftrightarrow M, t \vDash D_I\psi)$.
  Let $[I]\psi \in \Sigma_\varphi$, and suppose that $M, s \vDash [I]\psi$. As $\vdash_{\mathcal{L}_{DECL}} [I]\psi \to D_I[I]\psi$, and since by construction $D_I[I]\psi \in \Sigma_\varphi$, we get that $M, s \vDash D_I[I]\psi$. But then, as $D_I[I]\psi \in \Sigma_\varphi$ and $[s] \sim_I^+ [t]$, we have $M, t \vDash D_I[I]\psi$. As D_I is truthful, it is the case that $M, t \vDash [I]\psi$, and hence $[s] \approx_I^+ [t]$.
- $\sim_I^+ \subseteq \sim_J^+$ for $J \subseteq I$: Take arbitrary $[s], [t] \in S^+$ such that $[s] \sim_I^+ [t]$.
  Clearly, if $J = I$, it is immediate that $[s] \sim_J^+ [t]$, so we focus on the case where $J \subset I$. Suppose $D_J\psi \in \Sigma_\varphi$, and $M, s \vDash D_J\psi$. As $\vdash_{\mathcal{L}_{DECL}} D_J\psi \to D_JD_J\psi$, we can apply the Monotonicity of Distributed Knowledge axiom to get $\vdash_{\mathcal{L}_{DECL}} D_JD_J\psi \to D_ID_J\psi$. Hence, we have that $M, s \vDash D_ID_J\psi$. But then as $[s] \sim_I^+ [t]$ and $D_ID_J\psi \in \Sigma_\varphi$ by construction of Σ_φ, we get that $M, t \vDash D_ID_J\psi$. Again, as D_I is truthful, we obtain $M, t \vDash D_J\psi$, and thus $[s] \sim_J^+ [t]$.
- $\approx_I^+ \subseteq \approx_J^+$ for $J \subseteq I$: this is analogous to the previous case.
- $\approx^+_{\mathcal{A}\cup\{0\}} = \mathrm{id}$: Take arbitrary $[s], [t] \in S^+$ such that $[s] \approx^+_{\mathcal{A}\cup\{0\}} [t]$, and suppose $M, s \vDash \psi$ for some $\psi \in \mathcal{L}_{DECL}$, and let Σ_φ be the suitable set for ψ.
  (a) Suppose ψ is of the form $[\mathcal{A} \cup \{0\}]\theta$, thus $M, s \vDash [\mathcal{A} \cup \{0\}]\theta$. Then as $[s] \approx^+_{\mathcal{A}\cup\{0\}} [t]$ and Σ_φ is closed under subformulas, we get that $M, t \vDash [\mathcal{A} \cup \{0\}]\theta$. Thus we have that for all $\psi \in \mathcal{L}, M, s \vDash \psi$ if and only if $M, t \vDash \psi$. Hence it is the case that $[s] = [t]$.

(b) Now suppose ψ is *not* of the form $[\mathcal{A} \cup \{0\}]\theta$. By the Determinism of Grand Coalition axiom, we get that $M, s \vDash [\mathcal{A} \cup \{0\}]\psi$. By construction of Σ_φ and since $[s] \approx^+_{\mathcal{A}\cup\{0\}} [t]$, we get that $M, t \vDash [\mathcal{A} \cup \{0\}]\psi$. As $[\mathcal{A} \cup \{0\}]$ is truthful, we get that $M, t \vDash \psi$. Thus we have that for all $\psi \in \mathcal{L}, M, s \vDash \psi$ if and only if $M, t \vDash \psi$. Hence it is the case that $[s] = [t]$.

□

Lemma 3.30. *$\mathcal{L}_{DECL}$ has the strong finite model property with respect to pseudo-models.*

Proof. Let φ be a formula of $\mathcal{L}_{DECL}$. Then it is satisfiable if and only if it is satisfied in the canonical pseudo-model M^C by Proposition 3.22. Now let M^+ be the filtration of M^C over the suitable set Σ_φ for φ. Then by Lemma 3.28, φ is satisfied in M^C if and only if it is satisfied in M^+, hence φ is satisfiable iff it is satisfied in M^+. Also, we know that M^+ has at most $2^{|\Sigma_\varphi|}$ states. Hence, every satisfiable formula is satisfied in a model containing at most $2^{|\Sigma_\varphi|}$ states, thus giving $\mathcal{L}_{DECL}$ strong finite model property. □

Theorem 3.31. *The logic $\mathcal{L}_{DECL}$ is decidable.*

Proof. This follows from Lemma 3.30 and Theorem 6.7 in [9, p.340] which states that any normal modal logic that has the strong finite model property with respect to a recursive set of models is decidable. □

STEP 2: Unraveling

Now we will unravel the canonical pseudo-model. For that we first need a few notions.

Definition 3.32 (History)**.** Let $M = \langle S, \sim_I, \approx_J, V\rangle_{I\subseteq\mathcal{A}, J\subseteq\mathcal{A}\cup\{0\}}$ be a general Kripke model and take some $s \in S$. Then a *history with origin s* is a finite sequence $h := (s_0, R_0, s_1, \ldots, R_{n-1}, s_n)$ such that

- for all $k \leq n : s_k \in S$;
- $s_0 = s$;
- for all $k \leq n : R_k \in \{\sim_I; I \subseteq \mathcal{A}\} \cup \{\approx_I; I \subset \mathcal{A} \cup \{0\}\}$;
- for all $k \leq n : s_k R_k s_{k+1}$.

For any history h we write $first(h) = s_0$ and $last(h) = s_n$. These histories form the statespace of the unravelled tree.
For two histories $h = (s_0, R_0, \ldots, R_{n-1}, s_n)$ and $h' = (s'_0, R'_0, \ldots, R'_{m-1}, s'_m)$ we write the concatenation of the two $h + h' := (s_0, R_0, \ldots, R_{n-1}, s_n = s'_0, R'_0, \ldots, R'_{m-1}, s'_m)$.

Note that the R_k's come from the union of all $\sim_I$ relations where $I \subseteq \mathcal{A}$ and all $\approx_I$ relations with $I \subset \mathcal{A} \cup \{0\}$. Hence, we do not unravel the $\approx_{\mathcal{A}\cup\{0\}}$ relation. This is to ensure the semantical property of determinism of the grand coalition when we later go back to the actual models.

Definition 3.33 (Unraveled tree)**.** Let M be a general Kripke model and let $s \in S$. The *unraveling* of M around s is a tuple $\vec{M} = \langle \vec{S}, R_{\sim_I}, R_{\approx_J}, \vec{V} \rangle_{I\subseteq\mathcal{A}, J\subseteq\mathcal{A}\cup\{0\}}$ such that

- $\vec{S} = \{h; \text{first}(h) = s\}$
- $hR_{\sim_I}h'$ iff $h + (\text{last}(h), \sim_I, s') = h'$
- $hR_{\approx_I}h'$ iff $h + (\text{last}(h), \approx_I, s') = h'$
- $\vec{V} : P \to \mathcal{P}(\vec{S})$ such that $\vec{V}(p) := \{h \in \vec{S}; \text{last}(h) \in V(p)\}$.

Now we have defined histories on the canonical pseudo-model. They basically tell us which worlds in M are related by any sequence of relations from a specific world s. These form a tree. Now we will define paths on this tree of histories.

Definition 3.34 ($\mathcal{R}$-path)**.** Let $\vec{M}$ be the unraveling of some general Kripke model M around some world $s \in S$. Let $\mathcal{R} \subseteq \{R_{\sim_I}, R^{-1}_{\sim_I}, R_{\approx_J}, R^{-1}_{\approx_J}; I \subseteq \mathcal{A}, J \subset \mathcal{A}\cup\{0\}\} =: Rel$. An $\mathcal{R}$-*path* from h to h' is a finite sequence $p := (h_0, R_0, h_1, \ldots, R_{n-1}, h_n)$ such that

- for all $k \leq n : h_k \in \vec{S}$;
- $h_0 = h$;
- $h_n = h'$;
- for all $k < n : R_k \in \mathcal{R}$;
- for all $k < n : h_k R_k h_{k+1}$.

If $\mathcal{R}$ is not further specified, we speak of a *path*. For any path p we define again $first(p) = h_0$ and $last(p) = h_n$. Composing paths works the same as composing histories.

Definition 3.35 (Non-redundancy)**.** Let $\mathcal{R} \subseteq Rel$ and p an $\mathcal{R}$-path. We say that p is a non-redundant path if there is no $k < n-1$ such that $h_k = h_{k+2}$ and $R_{k+1} = R_k^{-1}$.

Intuitively, this definition means that a path is non-redundant if it doesn't immediately traverses an edge back.

Lemma 3.36. *Let $\vec{M}$ be the unraveling of a general Kripke model model M around some world $w \in S$. Let $h, h' \in \vec{S}$ be such that $h \neq h'$. Then there is exactly one non-redundant path p from h to h'.*

Lemma 3.37. *Any path p from h to h' contains the unique non-redundant path from h to h'.*

STEP 3: Completeness of DECL$^-$

Now we'll return from the land of trees to the land of models for DECL, and with that show completeness of $\mathcal{L}_{DECL}$ with respect to these models.

Definition 3.38. Let $\vec{M} = \langle \vec{S}, R_{\sim_I}, R_{\approx_J}, \vec{V} \rangle_{I \subseteq \mathcal{A}, J \subseteq \mathcal{A} \cup \{0\}}$ be the unraveling of the canonical pseudo-model M^C around some world $s \in S$. Then define $S = \langle \vec{S}, \sim_i, \approx_j, \vec{V} \rangle_{i \in \mathcal{A}, j \in \mathcal{A} \cup \{0\}}$ to be such that

$$
\begin{aligned}
\sim_i &= \big(\bigcup\{R_{\sim_I}; i \in I \subseteq \mathcal{A}\} \cup \bigcup\{R^{-1}_{\sim_I}; i \in I \subseteq \mathcal{A}\}\big)^* \\
\approx_i &= \big(\bigcup\{R_{\approx_I}; i \in I \subset \mathcal{A} \cup \{0\}\} \cup \bigcup\{R^{-1}_{\approx_I}; I \in I \subset \mathcal{A} \cup \{0\}\} \cup \\
&\qquad \bigcup\{R_{\sim_I}; i \in I \subseteq \mathcal{A}\} \cup \bigcup\{R^{-1}_{\sim_I}; i \in I \subseteq \mathcal{A}\}\big)^* \\
\sim_I &:= \bigcap_{i \in I} \sim_i \\
\approx_I &:= \bigcap_{i \in I} \approx_i
\end{aligned}
$$

Proposition 3.39. *For all $I \subseteq \mathcal{A}, h \sim_I h'$ if and only if the unique non-redundant path from h to $h', p = (h = h_0 S_0 h_1 \ldots S_{n-1} h_n = h')$ is an $\mathcal{R}$-path, with $\mathcal{R} = \{R_{\sim_J}, R^{-1}_{\sim_J}; I \subseteq J \subseteq \mathcal{A}\}$.*

Proof. Suppose $h \sim_I h'$. Then by definition $h \sim_i h'$ for all $i \in I$. Hence, for all $i \in I$ there is an $\mathcal{R}'$-path p' such that $\mathcal{R}' = \{R_{\sim_J}, R^{-1}_{\sim_J}; i \in J \subseteq \mathcal{A}\}$. But then by Proposition 3.37 it must be the case that the unique non-redundant path between h and h' is contained in p'. But then it must be the case that the non-redundant path between h and h' is an $\mathcal{R}^*$-path with $\mathcal{R}^* = \{R_{\sim_J}, R^{-1}_{\sim_J}; I \subseteq J \subseteq \mathcal{A}\}$. □

Proposition 3.40. *For all $I \subset \mathcal{A} \cup \{0\}$ and $h, h' \in \vec{W}, h \approx_I h'$ if and only if the unique non-redundant path from h to $h', p = (h = h_0 S_0 h_1 \ldots S_{n-1} h_n = h')$ is an $\mathcal{R}$-path with $\mathcal{R} = \{R_{\approx_J}, R^{-1}_{\approx_J}, R_{\sim_K}, R^{-1}_{\sim_K}; I \subseteq J \subset \mathcal{A} \cup \{0\}, I \subseteq K \subseteq \mathcal{A}\}$.*

Proof. Suppose $h \approx_I h'$. Then by definition $h \approx_i h'$ for all $i \in I$. Hence, for all $i \in I$ there is an $\mathcal{R}'$-path p' such that $\mathcal{R}' = \{R_{\approx_J}, R^{-1}_{\approx_J}, R_{\sim_K}, R^{-1}_{\sim_K}; i \in J \subset \mathcal{A} \cup \{0\}, i \in K \subseteq \mathcal{A}\}$. But then by Proposition 3.37 it must be the case that the unique non-redundant path between h and h' is contained in p'. But then it must be the case that the non-redundant path between h and h' is an $\mathcal{R}^*$-path with $\mathcal{R}^* = \{R_{\approx_J}, R^{-1}_{\approx_J}, R_{\sim_K}, R^{-1}_{\sim_K}; I \subseteq J \subset \mathcal{A} \cup \{0\}, I \subseteq K \subseteq \mathcal{A}\}$. □

Proposition 3.41. *S is a model for DECL.*

Proof. All relations in S are equivalence relations by definition. It is also immediate from the definition that $\sim_i \subseteq \approx_i$ for all $i \in \mathcal{A}$. It remains to show that $\bigcap_{i \in \mathcal{A} \cup \{0\}} [s]_{\approx_I} = \{s\}$ for all $s \in S$, i.e. we have to show that $h \approx_{\mathcal{A} \cup \{0\}} h' \to h = h'$.
So suppose $h \approx_{\mathcal{A} \cup \{0\}} h'$. Then by definition, $h \approx_i h'$ for all $i \in \mathcal{A} \cup \{0\}$. But for all $i \in \mathcal{A} \cup \{0\}$, we have that

$\approx_i = (\bigcup\{R_{\approx_I}; i \in I \subset \mathcal{A}\cup\{0\}\}\cup\bigcup\{R^{-1}_{\approx_I}; I \in I \subset \mathcal{A}\cup\{0\}\}\cup\bigcup\{R_{\sim_I}; i \in I \subseteq \mathcal{A}\}\cup\bigcup\{R^{-1}_{\sim_I}; i \in I \subseteq \mathcal{A}\})^*$.
However, note that this means that all separate parts are empty, and hence $\bigcap_{i\in\mathcal{A}\cup\{0\}} \approx_i = \mathrm{id}$. Thus $h \approx_{\mathcal{A}\cup\{0\}} h'$ implies $h = h'$. □

So S is in fact a model for DECL. Now we have to show that this model can be mapped into the canonical pseudo-model M^C with a bounded morphism.

Proposition 3.42. *Let S' be the pseudo-model obtained from S by taking $\sim_I := \bigcap_{i\in I} \sim_i$ and $\approx_I := \bigcap_{i\in I} \approx_i$, and let $f : S' \to M^C$ be such that $f(h) = last(h)$. Then f is a bounded morphism.*

Proof. The proof is per condition on bounded morphisms.

1. Take an arbitrary $h \in \vec{S}$. We have to show that $h \in \vec{V}(p)$ iff $f(h) \in V(p)$. We know that $h \in \vec{V}(p)$ if and only if $\mathrm{last}(h) \in V(p)$ since we defined $\vec{V}(p) := \{h \in \vec{S}; \mathrm{last}(h) \in V(p)\}$. Hence, $h \in \vec{V}(p)$ if and only if $f(h) \in V(p)$.

2. (a) We have to show that for all $h, h' \in \vec{S}$, if $h \sim_I h'$, then $f(h) \sim_I f(h')$. So take arbitrary $I \subseteq \mathcal{A}$ and $h, h' \in \vec{S}$ such that $h \sim_I h'$. This means that the non-redundant path p is an $\mathcal{R}$-path such that $\mathcal{R} = \{R_{\sim_J}, R^{-1}_{\sim_J}; I \subseteq J \subseteq \mathcal{A}\}$. Thus $p = (h = h_0 R_0 h_1 \dots R_{n-1} h_n = h')$ where for all $k < n, R_k \in \mathcal{R}$. Hence by definition of the $R_{\sim_J}$'s, we get that for all $k < n$, $\mathrm{last}(h_k) \sim_I last(h_{k+1})$. but then as $\sim_I$ is transitive, we get $\mathrm{last}(h) \sim_I last(h')$, and hence $f(h) \sim_I f(h')$.

   (b) We have to show that for all $h, h' \in \vec{S}$, if $h \approx_I h'$, then $f(h) \approx_I f(h')$.
   Take arbitrary $I \subseteq \mathcal{A}\cup\{0\}$ and $h, h' \in \vec{S}$ such that $h \approx_I h'$.
   Suppose $I \subset \mathcal{A}\cup\{0\}$. Then this means that the non-redundant path p is an $\mathcal{R}$-path such that $\mathcal{R} = \{R_{\approx_J}, R^{-1}_{\approx_J}, R_{\sim_K}, R^{-1}_{\sim_K}; J \subset \mathcal{A}\cup\{0\}, I \subseteq \mathcal{A}\}$. Hence, $p = (h = h_0 R_0 h_1 \dots R_{n-1} h_n = h')$ where for all $k < n, R_k \in \mathcal{R}$. Thus for all $k < n$, $last(h_k) \sim_I last(h_{k+1})$ or $last(h_k) \approx_I last(h_{k+1})$. Note that as $\sim_i \subseteq \approx_i$ for all $i \in \mathcal{A}$, we get that $last(h_k) \sim_I last(h_{k+1})$ implies $last(h_k) \approx_I last(h_{k+1})$, and thus we get that for all $k < n, last(h_k) \approx_I last(h_{k+1})$. But then as $\approx_I$ is transitive, we get $last(h) \approx_I last(h')$, and hence $f(h) \approx_I f(h')$.
   Now suppose $I = \mathcal{A}\cup\{0\}$. Then $h \approx_{\mathcal{A}\cup\{0\}} h'$ means that $h = h'$, and hence $last(h) = last(h')$, and thus $last(h) \approx_{\mathcal{A}\cup\{0\}} last(h')$, and thus $f(h) \approx_{\mathcal{A}\cup\{0\}} f(h')$.

3. (a) We have to show that for all $h \in \vec{S}$ and $t' \in S$, if $f(h) \sim_I t'$, then there exists a $h' \in \vec{S}$ such that $h \sim_I h'$ and $f(h') = t'$.
   Take arbitrary $I \subseteq \mathcal{A}, h \in \vec{S}$ and $t \in S$ such that $f(h) \sim_I t$. This means that $last(h) \sim_I t$. We have to show that there is a $h' \in \vec{S}$ such that $h \sim_I h'$ and $f(h') = t$.

Let $h' = h + (last(h), \sim_I, v)$. Then $h \sim_I h'$ by definition and $last(h') = t$, and thus $f(h') = t$.

(b) The $\approx$ case is analogous to the $\sim$ case.

□

Theorem 3.43 (Completeness for DECL⁻)**.** *$\mathcal{L}_{DECL^-}$ is weakly complete with respect to DECL models: For any DECL⁻-consistent formula φ there is a DECL M such that there is a s such that $M, s \vDash \varphi$.*

Proof. Let φ be an DECL⁻-consistent formula. By Lindenbaum's Lemma, $\{\varphi\}$ can be extended to a maximal consistent set Φ. By definition, $\Phi \in S^C$, where S^C is the set of states in the canonical structure M^C.
Now let $\vec{M}$ be the unraveling of M^C around $s := \Phi$, and let S be the generated DECL model. Note that the history $(s) \in \vec{S}$. Define $f : S \to M^C$ to be such that $f(h) = last(h)$. By Lemma 3.42, this is a bounded morphism. By Lemma 3.16, we have that $S, (s) \vDash \varphi$ if and only if $M^C, s \vDash \varphi$ as $last(s) = s$. But as $\varphi \in \Phi$, we have that $M^C, s \vDash \varphi$, and hence $S, (s) \vDash \varphi$. □

STEP 4: Completeness for DECL

Now that we have completeness for $\mathcal{L}_{DECL^-}$, we will show that $\mathcal{L}_{DECL}$ can be reduced to that language. We will show this in two steps, the first one being that any formula of the form $[\sigma]\varphi$ can be rewritten as a formula without the dynamic modality, and from there we will show that hence, any formula of $\mathcal{L}_{DECL}$ can be rewritten as a provably equivalent formula of $\mathcal{L}_{DECL^-}$.

Lemma 3.44. *The reduction axioms are valid on all DECL models.*

Proof. Take an arbitrary DECL model S. We will show that each reduction axiom is valid on S.

- $[\sigma]p \leftrightarrow (\mathrm{pre}(\sigma) \to \mathrm{post}(\sigma)(p))$
  Proof:
  $$\begin{aligned} S, s \vDash [\sigma]p &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow S \otimes \Sigma, (s, \sigma) \vDash p \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \to \mathrm{post}(\sigma)(p) \end{aligned}$$

- $[\sigma]\neg\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \neg[\sigma]\varphi)$
  Proof:
  $$\begin{aligned} S, s \vDash [\sigma]\neg\varphi &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow S \otimes \Sigma, (s, \sigma) \vDash \neg\varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow S \otimes \Sigma, (s, \sigma) \nvDash \varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow S, s \nvDash [\sigma]\varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \to \neg[\sigma]\varphi \end{aligned}$$

- $[\sigma](\varphi \wedge \psi) \leftrightarrow [\sigma]\varphi \wedge [\sigma]\psi$
  Proof:
  $$\begin{aligned} S, s \vDash [\sigma](\varphi \wedge \psi) &\Leftrightarrow S \otimes \Sigma, (s, \sigma) \vDash \varphi \wedge \psi \\ &\Leftrightarrow S \otimes \Sigma, (s, \sigma) \vDash \varphi \text{ and } S \otimes \Sigma, (s, \sigma) \vDash \psi \\ &\Leftrightarrow S, s \vDash [\sigma]\varphi \wedge [\sigma]\psi \end{aligned}$$

- $[\sigma]D_I\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \bigwedge_{\sigma \sim_I \sigma'} D_I[\sigma']\varphi)$
  Proof:
  $$\begin{aligned} S, s \vDash [\sigma]D_I\varphi &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow S \otimes \Sigma, (s, \sigma) \vDash D_I\varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow \forall (s', \sigma') \sim_I (s, \sigma) : S \otimes \Sigma, (s', \sigma') \vDash \varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow \forall (s', \sigma') \sim_I (s, \sigma) : S, s' \vDash [\sigma']\varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow \forall \sigma' \sim_I \sigma : S, s \vDash D_I[\sigma']\varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \to \textstyle\bigwedge_{\sigma' \sim_I \sigma} D_I[\sigma']\varphi \end{aligned}$$

- $[\sigma][I]\varphi \leftrightarrow (\mathrm{pre}(\sigma) \textstyle\bigwedge_{\sigma \approx_I \sigma'} [I][\sigma']\varphi)$
  Proof:
  $$\begin{aligned} S, s \vDash [\sigma][I]\varphi &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow S \otimes \Sigma, (s, \sigma) \vDash [I]\varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow \forall (s', \sigma') \approx_I (s, \sigma) : S \otimes \Sigma, (s', \sigma) \vDash \varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow \forall \sigma' \approx_I \sigma \forall s' \approx_I s : S, s' \vDash [\sigma']\varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \Rightarrow \forall \sigma' \approx_I \sigma : S, s \vDash [I][\sigma']\varphi \\ &\Leftrightarrow S, s \vDash \mathrm{pre}(\sigma) \to \textstyle\bigwedge_{\sigma' \approx_I \sigma} [I][\sigma']\varphi \end{aligned}$$

□

Proposition 3.45. *Let φ be a formula of $\mathcal{L}_{DECL}$ of the form $[\sigma]\psi$. Then there exists a formula $\varphi' \in \mathcal{L}_{DECL^-}$ such that $\vdash_{\mathcal{L}_{DECL}} \varphi \leftrightarrow \varphi'$.*

Proof. We will show this by induction on the complexity of ψ.
Base case Let $\varphi = [\sigma]\psi$, with $\psi = p$ for some $p \in P$, and $\sigma \in \Sigma$. Then by the reduction axiom, we get that $[\sigma](p) \leftrightarrow (\mathrm{pre}(\sigma) \to \mathrm{post}(\sigma)(p)($. Hence let $\varphi' := \mathrm{pre}(\sigma) \to \mathrm{post}(\sigma)(p)$. Then $\varphi' \in \mathcal{L}_{DECL^-}$, and $\vdash [\sigma]\varphi \leftrightarrow \varphi'$.
Inductive step Suppose that for $[\sigma]\psi \in \mathcal{L}_{DECL}$ there exists a $\psi' \in \mathcal{L}_{DECL^-}$ such that $[\sigma]\psi \leftrightarrow \psi'$. We want to show that for $\varphi = [\sigma]\chi$ where χ has a complexity of one more than ψ, we have that there is a $\varphi' \in \mathcal{L}_{DECL^-}$ such that $\vdash_{\mathcal{L}} \varphi \leftrightarrow \varphi'$.

- Suppose $\varphi = [\sigma]\neg\psi$. Then by the reduction axiom, we get that $\vdash \varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \neg[\sigma]\psi)$. Then by the induction hypothesis,$\vdash \varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \neg\psi')$. Note that $\mathrm{pre}(\sigma) \to \neg\psi =: \varphi'$ is in fact a formula of $\mathcal{L}_{DECL^-}$. This concludes this case.

- Suppose $\varphi = [\sigma](\psi_1 \wedge \psi_2)$. Then by the reduction axiom we have $\vdash \varphi \leftrightarrow [\sigma]\psi_1 \wedge [\sigma]\psi_2$. By the induction hypothesis, we get $\vdash \varphi \leftrightarrow \psi_1' \wedge \psi_2'$, so let $\varphi' := \psi_1' \wedge \psi_2'$, which is a formula of $\mathcal{L}_{DECL^-}$, which concludes this case.

- Suppose $\varphi = [\sigma]D_I\psi$. Then by the reduction axiom, we get that $\vdash \varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \bigwedge\limits_{\sigma \sim_I \sigma'} D_I[\sigma']\psi)$. By the induction hypothesis we get that $\vdash \varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \bigwedge\limits_{\sigma \sim_I \sigma'} D_I\psi_{\sigma'})$. Hence if we define $\varphi' := \bigwedge\limits_{\sigma \sim_I \sigma'} D_I\psi_{\sigma'}$, we can conclude this case.

- Suppose $\varphi = [\sigma][I]\psi$. Then by the appropriate reduction axiom, we get $\vdash \varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \bigwedge\limits_{\sigma \approx_I \sigma'} [I][\sigma']\psi)$. Thus by the induction hypothesis, we get

that $\vdash \varphi \leftrightarrow (\mathrm{pre}(\sigma) \rightarrow \bigwedge_{\sigma \approx_I \sigma'} [I]\psi')$, which concludes this case, and hence the proof.

□

Proposition 3.46. *For any formula φ of $\mathcal{L}_{DECL}$, there exists a formula φ' of $\mathcal{L}_{DECL^-}$ such that $\vdash_{\mathcal{L}_{DECL}} \varphi \leftrightarrow \varphi'$.*

Proof. The proof is by induction on the complexity of φ.
Base case Suppose $\varphi = p$ for some $p \in P$, and define $\varphi' = p$. Then clearly $\varphi' \in \mathcal{L}_{DECL^-}$ and $\vdash \varphi \leftrightarrow \varphi'$.
Inductive step Suppose that any $\psi \in \mathcal{L}_{DECL}$ is such that there exists a $\psi' \in \mathcal{L}_{DECL^-}$ such that $\vdash \psi \leftrightarrow \psi'$. We have to show that for any formula φ of the form $\neg\psi, \psi_1 \wedge \psi_2, D_I\psi, [I]\psi$ or $[\sigma]\psi$, there is a formula $\varphi' \in \mathcal{L}_{DECL^-}$ such that $\vdash \varphi \leftrightarrow \varphi'$.
The Boolean, D_I and $[I]$ cases are trivial.
Suppose φ is of the form $[\sigma]\psi$. Then by Proposition 3.45, we get that there is a $\varphi' \in \mathcal{L}_{DECL^-}$ such that $\vdash \varphi \leftrightarrow \varphi'$, which concludes the proof. □

Theorem 3.47 (Completeness of $DECL$)**.** *For any formula φ, if $\vdash_{DECL} \varphi$, then there is a $DECL$-model $\mathbf{S}$ such that there is a $s \in S$ such that $\mathbf{S}, s \vDash \varphi$.*

Proof. This follows from Theorem 3.43 and Proposition 3.46. □

3.3 Planning with DECL

We introduced a dynamic epistemic logic that can talk about the power of coalitions. In this section we will discuss how to use DECL to define and solve specific planning issues. We have talked about agents' *goal* before in several examples, and we have hinted at what agents would probably decide to do in several situations. We will now formalize this intuition by introducing *planning problems* and how to reach a solution to them.

Planning problems

When we as human beings try to plan for a certain goal, the information we have at that moment is some, possibly incomplete, information about the world as it is now, the way we would like the world to be, and the things we can do to change the world. We follow the approach by Andersen and Bolander [10] when we take these three things as the main ingredients for formal planning problems:

Definition 3.48 (Planning Problem)**.** A *planning problem* is a tuple $\mathbb{P} = \langle (S, s), \Sigma, \mathcal{A}, \varphi_g \rangle$ where

- (S, s) is a pointed epistemic model called the *initial state*;
- Σ is an action model containing the *available actions*;

- $\mathcal{A}$ is a set of *agents*
- φ_g is a formula in $\mathcal{L}_{DECL}$ called the *goal*

Throughout the thesis we have assumed that the full set of agents has the same goal, and we will keep to this assumption. Thus, the idea of a planning problem is that the agents in $\mathcal{A}$ will try to achieve φ_g, using the actions available to them in Σ, from the initial model (S, s), of which they might not have full information.

Depending on the information the agents have, they will want to perform a sequence of events such that after this sequence, their goal is true. Thus a *solution* to a planning problem is a sequence of n events $\sigma_1, \dots, \sigma_n$ such that after this sequence, their goal is true:

$$(((S \otimes \Sigma) \dots) \otimes \Sigma), (((s, \sigma_1), \dots), \sigma_n) \vDash \varphi_g$$

In this thesis we are mainly concerned with *single-step* solutions, as they on their own convey the power of the logic very well. A single-step solution is a solution that consists of only one joint action, rather than a sequence.

Note that a solution is defined on a *pointed* model. This means that since we are often working with incomplete information, agents might not have full knowledge of the world or of what state they are in. Because of this, we need a notion of solution that deals with what agents know - and with how their knowledge differs.

Leading up to this we have given some examples where some agents knew how to reach the goal, while others did not, or where agents knew of different ways to reach their goal. This has given us some intuition about what makes a solution a solution - if one agent knows that some joint action will lead to the goal, then surely that would be a good way to go. Hence we need that at least one agent knows that a certain sequence of actions is a solution to ensure that the agents reach their goal. If such a solution does not exist, the next best thing is to decide on the sequence of actions of which no agent knows that it does *not* reach the goal: a sequence that might achieve what they want.

Formally, we say:

Definition 3.49 (Single-step Solution)**.** For a planning problem $\mathbb{P} = \langle (S, s), \Sigma, \mathcal{A}, \varphi_g \rangle$ and joint action σ_I such that for all $\sigma, \sigma' \in \sigma_I$ we have that $\sigma \sim_I \sigma'$, we say that σ_I is a
strong single-step solution if $\mathbf{S}, s \vDash \bigvee_{i \in I} K_i[\sigma_I]\varphi_g$
weak single-step solution if $\mathbf{S}, s \vDash \bigwedge_{i \in I} \neg K_i \neg [\sigma_I]\varphi_g$
for $I \subseteq \mathcal{A}$

The notion of a single-step solution can easily be extended to that of a k-step solution for some $k \in \mathbb{N}$. A solution is then a *sequence* of actions, rather than a single action. The conditions can be kept the same, except for then requiring that after the entire sequence has happened, we need the goal to hold.

As we define it, a solution is an action for which one agent knows that it will achieve the goal. The other agents, however, might not be aware of this, or they might even have other ideas about which joint action is the right one. In our setting, we do not have a way for agents to communicate. Hence, because the agents do need *some* way to communicate, as they need to coordinate on their actions, we will for now assume that they are, outside of our framework, in a situation where they can discuss and coordinate their possibilities. We suppose that they have all means and actions to reach a joint conclusion. With this added assumption, it is clear why one agent knowing that an action leads to the goal is enough: he can tell the other agents, who then also know. If there are multiple actions that are solutions, the agents can in this external way discuss which to choose.

Finding a plan

Iit is trivial to check whether any of the available joint actions is a strong or weak single-step solution for a given planning problem by computing the result of applying every event. It is more interesting to check whether a sequence of actions is a k-step solution of either kind. It would most certainly involve the notion of a *planning tree* as mentioned in [3], where one iteratively applies the available actions before using model checking to see whether any sequence of joint actions satisfies the conditions to be considered a solution.

4 Common Knowledge in Epistemic Planning

In the previous chapter we introduced a logic that talks about the power of coalitions in a dynamic setting. We could express that some joint action of agents would lead them to their goal, and we could express knowledge about this fact. What we could not express was a way in which every agent is sure of the course to take and what his or her colleagues would do. Hence, we assumed that the agents had some external way in which they could coordinate their choice, rather than us modeling this part. In this chapter we extend the logic we introduced in the previous chapter in order to be able to model agents that can coordinate and make a deliberate decision.

4.1 Dynamic Epistemic Coalition Logic with Common Knowledge

For agents to be able to coordinate, it is important that they can rely on other agents' information. It might be that Alice knows that only one option leads to the goal, but if she does not know that Bob knows this, she still cannot be sure that together they will reach it, as he might think another action is better. Maybe she knows that Bob knows, but then Bob doesn't know that she knows that he knows, in which case he cannot be entirely sure about the choice that Alice will make. This clearly can go on forever, so what we would like to be able to express is common knowledge: all agents know, that all agents know, that ... that φ. Therefore we add common knowledge to dynamic epistemic coalition logic to obtain Dynamic Epistemic Coalition Logic with Common Knowledge, or DECL-C.

4.1.1 Syntax and Semantics of DECL-C

Definition 4.1 ($\mathcal{L}_{DECL\text{-}C}$)**.** The language $\mathcal{L}_{DECL\text{-}C}$ is given by the following Backus-Naur form:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid D_I\varphi \mid C_I\varphi \mid [J]\varphi \mid [\sigma]\varphi$$

where $I \subseteq \mathcal{A}$ and $J \subseteq \mathcal{A} \cup \{0\}$ are coalitions, $\sigma \in \Sigma$ for some given Σ and p is a propositional letter.

The formulas of DECL-C are evaluated on the same models and in the same way as the formulas of DECL. The only thing we have to define is when common knowledge is satisfied.

Definition 4.2 (Satisfaction)**.** The *satisfaction* of these formulas on static epistemic models, denoted $\mathbf{S}, s \vDash \varphi$ is defined as follows:

- $\mathbf{S}, s \vDash C_I\varphi$ iff $\forall t$ s.t. $(s,t) \in (\bigcup_{i\in I} \sim_i)^* : \mathbf{S}, t \vDash \varphi$

Where $(\bigcup_{i \in I} \sim_i)^*$ is the reflexive, transitive closure of $\bigcup_{i \in I} \sim_i$.

Example 4.3 (It is common knowledge that two agents can only reach the goal together)**.** Consider the initial model in Figure 17 and the action model in Figure 18. Both agents have complete information about the world, but they do not have full control: it takes both of them to flip the truth value of p. However, their goal is p, and currently it is $\neg p$.

Figure 17: The static model S

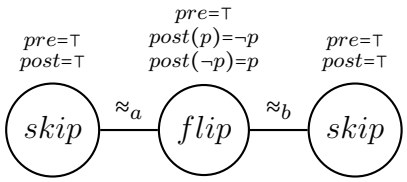

Figure 18: The action model Σ

One can see that not only is it the case that $S, s \vDash C_I[flip_{\{a,b\}}]p$, but also that $S, s \vDash C_I \neg \Diamond_{\{a\}} p \wedge C_I \neg \Diamond_{\{b\}} p$. Hence, it is common knowledge that if they work together, they can reach their goal by both choosing $flip$, but also that neither of them has any way to reach the same result on his or her own. Hence, it is common knowledge that it would make sense for them to both choose $flip$, and thus they can both trust the other agent to help out.

Example 4.4 (There are two joint actions for which it is common knowledge that they achieve the goal)**.** Suppose two agents, Annie and Bernadette, want to go out some evening, but before they can make a decision on whether to go dancing or to go to a bar, Annie's phone loses power. Now each of them has to choose whether to go dancing or to go to the bar. They prefer to be out together, so $\varphi_g = (D_a \wedge D_b) \vee (B_a \wedge B_b)$. The situation can be represented as in Figures 19 and 20. One can check that it is the case that $S, s \vDash C_I[BaBb_{\{a,b\}}]\varphi_g \wedge C_I[DaDb_{\{a,b\}}]\varphi_g$. This means that although there are two sure ways to reach the goal, they cannot be sure of which one the other will pick, and hence they cannot count on the other agent to make the same choice without further communication.

4.2 Group Epistemic PDL

It is well known within the DEL literature that it is not possible to derive a reduction axiom for C_I like we did for all modalities in DECL. Without a reduction axiom, we cannot show that the dynamic language reduces to the static language, which means that we cannot prove completeness for the entire language. However, if we extend the language further, we *are* able to obtain reduction axioms. Thus for this completeness proof, we create our own version of epistemic propositional dynamic logic (E-PDL) as presented first by van Benthem et al. in [30]. Thus we will present this extended language, argue why it does everything DECL-C does, and proceed to prove that it is sound, complete and decidable.

Figure 19: The static model S

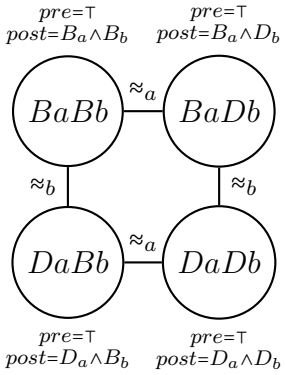

Figure 20: The action model Σ

4.2.1 Syntax and Semantics of GE-PDL

E-PDL is basically PDL where the programs are reinterpreted as the epistemic relations in the model. The way more complex programs are constructed from the basic programs reflect sequences and deeper levels of knowledge. Thus, using the Kleene star we can represent common knowledge. The E-PDL presented by van Benthem et al. [30] uses the single epistemic relations as basic programs. However, as we are concerned with group knowledge and control, our basic programs are in fact epistemic *group* relations, and the language we will use is Group Epistemic PDL, or GE-PDL.

Definition 4.5 (Language)**.** GE-PDL is formed as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_\pi\varphi \mid [J]\varphi \mid [\sigma]\varphi$$

$$\pi ::= I \mid \pi + \pi' \mid \pi;\pi' \mid \pi^* \mid ?\varphi$$

where $I \subseteq \mathcal{A}$ and $J \subseteq \mathcal{A} \cup \{0\}$ are coalitions.

There are some remarks to be made about this language.
First, to prevent confusion with notation we will use K_π where van Benthem and colleagues use $[\pi]$, and where they use $\pi_1 \cup \pi_2$, as is the usual notation in PDL, we write $\pi_1 + \pi_2$, because $\sim_{I\cup J}$ is something very different than $\sim_I \cup \sim_J$ in our framework.

Definition 4.6 (Satisfaction)**.** The *satisfaction* of these formulas on static epistemic models, denoted $M, s \vDash \varphi$ is defined as follows:

- $M, s \vDash p$ iff $s \in V(p)$
- $M, s \vDash \neg\varphi$ iff $M, s \nvDash \varphi$
- $M, s \vDash \varphi \vee \psi$ iff $M, s \vDash \varphi$ or $M, s \vDash \psi$

- $M, s \vDash K_\pi \varphi$ iff $\forall s' \sim_\pi s : M, s' \vDash \varphi$
- $M, s \vDash [I]\varphi$ iff $\forall s' \approx_I s : M, s' \vDash \varphi$
- $M, s \vDash [\sigma]\varphi$ iff $(s, \sigma) \in M \otimes \mathbf{\Sigma}$ implies $M \otimes \Sigma, (s, \sigma) \vDash \varphi$

Where $\sim_\pi$ is defined inductively as follows:

- $\sim_I := \bigcap_{i \in I} \sim_i$
- $\sim_{\pi_1;\pi_2} := \sim_{\pi_1} ; \sim_{\pi_2}$
- $\sim_{\pi_1+\pi_2} := \sim_{\pi_1} \cup \sim_{\pi_2}$
- $\sim_{\pi^*} := (\sim_\pi)^*$
- $s \sim_{?\theta} t$ iff $s = t$ and $M, s \vDash \theta$

To get some intuition behind this, we will show how to 'translate' formulas of DECL to formulas of GE-PDL.

- $D_I \varphi$ is written as $K_I \varphi$
- $[I]\varphi$ is still written as $[I]\varphi$
- $C_I \varphi$ is written as $K_{(\sum I)^*} \varphi$

Where $\sum I = i_1 + i_2 + \ldots + i_k$ for a fixed enumeration $i_1, \ldots, i_k$ of I.

So we can express all modalities of DECL into modalities of GE-PDL. However, the latter can also sequences of modalities of the former into one modality. For example $K_{I;J}$ is equivalent to $K_I K_J \varphi$, which expresses that it is distributed knowledge between the agents in I that it is distributed knowledge between the agents in J that φ.

By taking epistemic group relations as basic programs, rather than the singular epistemic relations, we can express 'Common Distributed Knowledge'. In this language this can be expressed as $K_{(I+J)^*}$, meaning to say that it is common distributed knowledge between the two groups that φ. Hence, in both groups I and J it is distributed knowledge that it is distributed knowledge in the other group that it is distributed knowledge in the other group ... that φ. Note that this is something very different from $K_{(I \cup J)^*}$ as this denotes common distributed knowledge in the single group $I \cup J$, which is equivalent to the usual notion of distributed knowledge $K_{I \cup J} = D_{I \cup J}$ as the distributed knowledge relation is already transitive and reflexive. Both of these modalities are different from $K_{(\Sigma(I \cup J))^*} \varphi$, which merely denotes common knowledge $C_{I \cup J} \varphi$ in the big group $I \cup J$.

The interesting thing about this common distributed knowledge $K_{(I+J)*}$ between two groups is that it can be converted into full common knowledge only by communications performed within the two distinct groups, whereas to similarly convert simple distributed knowledge $K_{I\cup J}\varphi$ into common knowledge we need public announcements to the entire group $I \cup J$.

Example 4.7 (Common Distributed Knowledge)**.** To see an example of common distributed knowledge consider the static model in Figure 21.

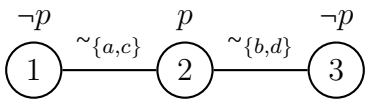

Figure 21: Static model

One can see that in this model it is the case that $K_{\{a,b\}}p$ and that $K_{\{c,d\}}p$. However, it is also the case that $K_{\{c,d\}}K_{\{a,b\}}p$ and $K_{\{a,b\}}K_{\{c,d\}}p$. We can continue this argument an arbitrary number of times, so we reach $K_{(\{a,b\}+\{c,d\})^*}p$ - it is common distributed knowledge between a, b and c, d that p. It is also immediately clear that if only a and b share their information, and c and d share theirs in a semi-public way where everyone can see that *some* message is being sent, but is not necessarily aware of the contents of the message, we achieve common knowledge of p for the entire set of agents.

Example 4.8. Suppose two countries are at war - a very special kind of war where everything happens only after representatives for each party have agreed on it. Also these representatives are very special, as it is common knowledge that a representative will announce to his side what he agreed upon with the other representative. Now say a is the representative of one side of the war, whereas b is the representative of the other side. Together they decide that the battle will happen tomorrow at dawn. Now this fact is common knowledge between a and b, and thus common distributed knowledge between the two sides of the war. Hence, after both a and b go back to their respective armies, and announce their decision, it will be common knowledge between *both* camps that the battle will happen tomorrow since it is common knowledge that both a and b will announce the decision.

4.2.2 Proof System of GE-PDL

Before we can show that GE-PDL is sound and complete, we list the axioms that we use. They partly overlap with the axioms for DECL, but we add axioms for programs.

Axioms

Inference rules

- All axioms and rules of classical propositional logic
- Necessitation rules for all modalities

Axioms for DECL

- S5 for $[I]$ for all $I \subseteq \mathcal{A} \cup \{0\}$
- S5 for K_I for all basic programs I
- **Knowledge of (Individual) Control**
  $[I]\varphi \to K_I\varphi$ for $I \subseteq \mathcal{A}$
- **Monotonicity of control**
  $[I]\varphi \to [J]\varphi$ for $I \subseteq J \subseteq \mathcal{A} \cup \{0\}$
- **Monotonicity of distributed knowledge**
  $K_I\varphi \to K_J\varphi$ for $I \subseteq J \subseteq \mathcal{A}$
- **Determinism of grand coalition**
  $\varphi \to [\mathcal{A} \cup \{0\}]\varphi$

Axioms for programs

- $K_\pi(\varphi \to \psi) \to (K_\pi\varphi \to K_\pi\psi)$
- $K_{\pi_1;\pi_2}\varphi \leftrightarrow K_{\pi_1}K_{\pi_2}\varphi$
- $K_{\pi_1+\pi_2}\varphi \leftrightarrow K_{\pi_1}\varphi \wedge K_{\pi_2}\varphi$
- $K_{\pi^*}\varphi \leftrightarrow \varphi \wedge K_\pi K_{\pi^*}\varphi$
- $K_{\pi^*}(\varphi \to K_\pi\varphi) \to (\varphi \to K_{\pi^*}\varphi)$
- $K_{?\theta}\psi \leftrightarrow (\theta \to \psi)$

Reduction Axioms

Let N be the number of events in Σ, and suppose $\sigma_1, \ldots, \sigma_N$ is an enumeration of all events without repetitions, and let $\sigma = \sigma_n$ be any of these events (for some arbitrary $n \leq N$). Then we have the following reduction axioms.

- $[\sigma]p \leftrightarrow (\mathrm{pre}(\sigma) \to \mathrm{post}(\sigma)(p))$
- $[\sigma]\neg\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \neg[\sigma]\varphi)$
- $[\sigma](\varphi \wedge \psi) \leftrightarrow [\sigma]\varphi \wedge [\sigma]\psi$
- $[\sigma]K_\pi\varphi \leftrightarrow (\bigwedge_{m=0}^{N-1}[T_{nm}(\pi)][\sigma_m]\varphi)$

- $[\sigma][I]\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \bigwedge_{\sigma \approx_I \sigma'} [I][\sigma']\varphi)$

Where T_{nm} is a program transformer, which is defined as follows:

Definition 4.9 (T_{nm} program transformers)**.**

$$T_{nm}(I) = \begin{cases} ?pre(\sigma_n); I & \text{if } \sigma_n \sim_I \sigma_m \\ ?\bot & \text{otherwise} \end{cases}$$

$$T_{nm}(\pi_1;\pi_2) = \bigcup_{k=0}^{N-1} (T_{nk}(\pi_1); T_{km}(\pi_2))$$

$$T_{nm}(\pi_1 + \pi_2) = T_{nm}(\pi_1) \cup T_{nm}(\pi_2)$$

$$T_{nm}(?\theta) = \begin{cases} ?pre(\sigma_n) \wedge [\sigma_n]\theta & \text{if } n = m \\ ?\bot & \text{otherwise} \end{cases}$$

$$T_{nm}(\pi^*) = K_{nmN}(\pi)$$

Where $K_{nmN}(\pi)$ is given by Definition 5.10:

Definition 4.10. $K_{nmk}(\pi)$ is defined by recursion on k as follows:

$$K_{nm0}(\pi) = \begin{cases} ?(p \vee \neg p) \cup T_{nm}(\pi) & \text{if } n = m \\ T_{nm} & \text{otherwise} \end{cases}$$

$$K_{nm(k+1)}(\pi) = \begin{cases} (K_{kkk}(\pi))^* & \text{if } n = k = m \\ (K_{kkk}(\pi))^*; K_{kmk}(\pi) & \text{if } n = k \neq m \\ K_{nkk}(\pi); (K_{kkk}(\pi))^* & \text{if } n \neq k = m \\ K_{nmk}(\pi) \cup (K_{nkk}(\pi); (K_{kkk}(\pi))^*; K_{kmk}(\pi)) & \text{otherwise} \end{cases}$$

4.3 Soundness, Completeness and Decidability of GE-PDL

We will first show completeness for the static language, after which we will continue with completeness for the dynamic language. Before we start the proof, we will give a brief explanation of the steps we will take, as this will make it easier for the reader to follow the main argument.

4.3.1 Preliminaries

Again, before we start the proof we will state some definitions and general results that we will use during the proof.

Definition 4.11 (Filtration)**.** Let M be a general Kripke model, and $\Sigma \subseteq \mathcal{L}$ a set of formulas. The relation $\equiv_\Sigma$ on S, defined as

$$s \equiv_\Sigma t \Leftrightarrow \text{ for all } \sigma \in \Sigma : (M, s \vDash \sigma \Leftrightarrow M, t \vDash \sigma)$$

defines an equivalence on S; we denote its equivalence classes with $[s]_\Sigma$, but we will often leave out the subscript if Σ is clear from context. The model $M^f = \langle S^f, \sim^f_\pi, \approx^f_I, V^f \rangle$ is a *filtration of S through* Σ if

- $S^f = \{[s]; s \in S\}$
- For each $R^f_\Box \in \{\sim^f_\pi, \approx^f_J; \pi \text{ a program}, J \subseteq \mathcal{A} \cup \{0\}\}$ the following hold:

  min^f For all $[s], [t] \in S^f$, if $sR_\Box t$, then $[s]R^f_\Box[t]$.

  max^f For all $[s], [t] \in S^f$, if $[s]R^f_\Box[t]$, then for all $\Box\varphi \in \Sigma(S, s \vDash \Box\varphi \to S, t \vDash \varphi)$

- $V^f(p) = \{[s]; s \in V(p)\}$

Lemma 4.12. *Let S^f be a filtration of S through some Σ. Then for all $[s] \in S^f$ and $\varphi \in \Sigma$, we have $S, s \vDash \varphi \Leftrightarrow S^f, [s] \vDash \varphi$.*

Proof. The proof is by induction on the complexity of the formula.
Base case. Suppose $\varphi = p$ for some $p \in P$. Then by definition $[s] \in V^f(p) \leftrightarrow s \in V(p)$, and hence $S, s \vDash \varphi \Leftrightarrow S^f, [s] \vDash \varphi$.
Inductive step. The boolean cases are straightforward.
Suppose φ is of the form $\Box\psi$ for some $\Box \in \{\sim_\pi, \approx_J; \pi \text{ a program}, J \subseteq \mathcal{A} \cup \{0\}\}$.
For the left-to-right direction, suppose $S, s \vDash \Box\psi$. Now take an arbitrary $[t] \in S^f$ such that $[s]R^f_\Box[t]$. Then by the max^f condition of filtrations and the fact that $S, s \vDash \Box\psi$, we get that $S, t \vDash \psi$. But then by the induction hypothesis $S^f, [t] \vDash \psi$, and thus $S^f, [s] \vDash \Box\psi$.
For the other direction, suppose $S^f, [s] \vDash \Box\psi$. Hence, for all $[t] \in S^f$ such that $[s]R^f_\Box[t]$, we have that $S^f, [t] \vDash \psi$. By the induction hypothesis, we get $S, t \vDash \psi$. Now take an arbitrary $u \in S$ such that $sR_\Box u$. Then, as $sR_\Box u$, we get by the min^f condition of filtrations that $[s]R^f_\Box[u]$, and thus that $S, u \vDash \psi$. Hence $S, s \vDash \Box\psi$. □

Definition 4.13 (Bounded Morphism)**.** Let $M = \langle S, \sim_\pi, \approx_I, V\rangle_{I \subseteq \mathcal{A} \cup \{0\}}$ and $M' = \langle S', \sim'_\pi, \approx'_I, V'\rangle_{I \subseteq \mathcal{A} \cup \{0\}}$ be two general Kripke models. A mapping $f : M \to M'$ is a *bounded morphism* if the following hold for all programs π and $I \subseteq \mathcal{A} \cup \{0\}$:

1. for all $s \in S : s \in V(p)$ if and only if $f(s) \in V'(p)$.
2. (a) for all $s, t \in S$, if $s \sim_\pi t$ then $f(s) \sim'_\pi f(t)$
   (b) for all $s, t \in S$, if $s \approx_I t$ then $f(s) \approx'_I f(t)$
3. (a) for all $s \in S$ and $t' \in S'$, if $f(s) \sim'_\pi t'$, then there exists a $t \in S$ such that $s \sim_\pi t$ and $f(t) = t'$.
   (b) for all $s \in S$ and $t' \in S'$, if $f(s) \approx'_I t'$, then there exists a $t \in S$ such that $s \approx_I t$ and $f(t) = t'$.

Proposition 4.14. *Let $M = \langle S, \sim_\pi, \approx_I, V\rangle_{I \subseteq \mathcal{A} \cup \{0\}}$ and $M' = \langle S', \sim'_\pi, \approx'_I, V'\rangle_{I \subseteq \mathcal{A} \cup \{0\}}$ be two general Kripke models, and let $f : M \to M'$ be a bounded morphism. Then for any formula φ of $DECL - C^-$ and world $s \in S$, it is the case that $M, s \vDash \varphi$ iff $M', f(s) \vDash \varphi$.*

Proof. The proof is by induction on the complexity of φ:
Base case: The base case follows from the definition of bounded morphisms.
Inductive step: Suppose that for all $s \in S$ and formulas ψ with a lower complexity that φ it is the case that $M, s \vDash \psi$ if and only if $M', f(s) \vDash \psi$. The Boolean cases where $\varphi = \neg\psi$ and $\varphi = \psi_1 \wedge \psi_2$ follow immediately, which leaves only the modalities.

- For the left-to-right direction, suppose $M, s \vDash K_\pi \psi$. Take an arbitrary $t' \in S'$ such that $f(s) \sim_\pi t'$. According to condition $3(a)$ there exists a $t \in S$ such that $s \sim_\pi t$ and $f(t) = t'$. By the first consequence we get that $M, t \vDash \psi$, and by the second we get that $M', f(t) \vDash \psi$. Hence it is the case that $M', f(s) \vDash K_\pi \psi$.
  For the right-to-left direction, suppose $M', f(s) \vDash K_\pi \psi$. Take $t \in S$ such that $s \sim_\pi t$. Then by condition 2(a) it is the case that $f(s) \sim'_\pi f(t)$, and thus that $M', f(t) \vDash \psi$. Hence, $M, t \vDash \psi$, and thus $M, s \vDash K_\pi \psi$.

- The case for $[I]$ is analogous.

□

4.3.2 Plan of the Proof

The completeness proof consists of three steps, which are similar to the ones taken in the previous chapter.

1. Step 1: Soundness and Completeness for Pseudo-Models. First, we create *pseudo-models*. These are structures that look like static epistemic control models, but have separate $\sim$ relations for every possible program and separate $\approx$ relations for every subset of agents, instead of just one for every agent. We then show soundness with respect to these pseudo-models and argue that as static epistemic control models are a special case of pseudo-models, the logic is sound with respect to static epistemic control models. Then we define the canonical pseudo-model and prove completeness with respect to this.

   (b) Step 1b: Decidability. After this, we use a filtration of the canonical pseudo-model to obtain a finite pseudo-model with some additional nice properties, hence proving decidability.

2. Step 2: Unraveling. In the second step, we *unravel* the finite pseudo-model that was the result of the filtration. This means that we create *all* possible histories in the pseudo-model: paths that can be taken when we follow the $\sim$ and $\approx$ relations. These histories are related in such a way that they form a tree.

3. Step 3: Completeness for GE-PDL$^-$. In the third step, we take the tree we just created, and from there define a static epistemic control model. We

do this by defining the proper relations, and showing that this newly created structure satisfies the necessary semantic properties. Then we define a bounded morphism between the filtered pseudo-model and the static epistemic control model, which makes completeness with respect to those models immediate.

4. Step 4: Completeness for GE-PDL. In the last step we show that the dynamic language can be reduced to the static language using the reduction axioms and a translation mechanism, hence showing that also the dynamic language is complete.

4.3.3 The Proof

STEP 1: Soundness and Completeness for Pseudo-Models

Definition 4.15 (Pseudo-model)**.** A pseudo-model is a structure $M = \langle S, \sim_\pi, \approx_I, V \rangle$ where

- $\approx_I$ and $\sim_I$ (for I a basic program) are equivalence relations
- $\sim_I \subseteq \approx_I$ for I a basic program
- For $J \subseteq I$:$\sim_I \subseteq \sim_J$ and $\approx_I \subseteq \approx_J$
- $\approx_{\mathcal{A}\cup\{0\}}$=id
- We define by induction:
  - $\sim_{?\theta}$ iff $s = t$ and $s \vDash \varphi$
  - $\sim_{\pi_1;\pi_2} = \sim_{\pi_1} ; \sim_{\pi_2}$
  - $\sim_{\pi_1+\pi_2} = \sim_{\pi_1} \cup \sim_{\pi_2}$
  - $(\sim_\pi)^* \subseteq \sim_{\pi^*}$

Proposition 4.16. *All axioms are valid on pseudo-models*

Proof. We will show this per axiom

- S5 for K_I and $[I]$ follows easily, as $\sim_I$ and $\approx_I$ are equivalence relations
- $[I]\varphi \to [J]\varphi$ for $I \subseteq J$: Suppose $M, s \vDash [I]\varphi$. Take an arbitrary $t \in S$ such that $s \approx_J t$. From our requirements, it follows that $s \approx_I t$, and thus $M, t \vDash \varphi$. Thus for all t such that $s \approx_J t$ we have $M, t \vDash \varphi$, and hence $M, s \vDash [J]\varphi$.
- $K_I\varphi \to K_J\varphi$ for $I \subseteq J$: Similar as above.
- $\varphi \to [\mathcal{A} \cup \{0\}]\varphi$: Assume $Ms, \vDash \varphi$, and take an arbitrary $t \in S$ such that $s \approx_{\mathcal{A}\cup\{0\}} t$. Then, as $\approx_{\mathcal{A}\cup\{0\}}$= id, we get that $t = s$, and thus $M, t \vDash \varphi$, and thus for all t such that $s \approx_{\mathcal{A}\cup\{0\}} t : M, t \vDash \varphi$, hence $M, s \vDash [\mathcal{A} \cup \{0\}]\varphi$.

- $K_\pi(\varphi \to \psi) \to (K_\pi\varphi \to K_\pi\psi)$: Suppose $M, s \vDash K_\pi(\varphi \to \psi)$ and $M, s \vDash K_\pi\varphi$, and take an arbitrary $t \in S$ such that $s \sim_\pi t$. Then $M, t \vDash \varphi \to \psi$ and $M, t \vDash \varphi$, and thus $M, t \vDash \psi$. Hence, $M, s \vDash K_\pi\psi$.

- $K_{\pi_1;\pi_2}\varphi \leftrightarrow K_{\pi_1}K_{\pi_2}\varphi$: $M, s \vDash K_{\pi_1;\pi_2}\varphi$ iff for all $t \in S$ such that $s \sim_{\pi_1;\pi_2} t$ it is the case that $M, t \vDash \varphi$. This is the case iff for all t such that $s \sim_{\pi_1}; \sim_{\pi_2} t$, we have $M, t \vDash \varphi$. This happens if and only if for all t such that there is a u such that $s \sim_{\pi_1} u$ and $u \sim_{\pi_2} t$, we have $M, t \vDash \varphi$. This is true iff for all u such that $s \sim_{\pi_1} u$ and all t such that $u \sim_{\pi_2} t$, it is the case that $M, t \vDash \varphi$. Which happens if and only if $M, s \vDash K_{\pi_1}K_{\pi_2}\varphi$.

- $K_{\pi_1+\pi_2}\varphi \leftrightarrow K_{\pi_1}\varphi \wedge K_{\pi_2}\varphi$:

$$\begin{aligned} M, s \vDash K_{\pi_1+\pi_2}\varphi &\Leftrightarrow \forall t \text{ s.t. } s \sim_{\pi_1+\pi_2} t : M, t \vDash \varphi \\ &\Leftrightarrow \forall t \text{ s.t. } s \sim_{\pi_1} \cup \sim_{\pi_2} t : M, t \vDash \varphi \\ &\Leftrightarrow \forall t \text{ s.t. } s \sim_{\pi_1} t \text{ or } s \sim_{\pi_2} t : M, t \vDash \varphi \\ &\Leftrightarrow \forall t \text{ s.t. } s \sim_{\pi_1} t : M, t \vDash \varphi \text{ and } \forall t \text{ s.t. } s \sim_{\pi_2} t : M, t \vDash \varphi \\ &\Leftrightarrow M, s \vDash K_{\pi_1}\varphi \wedge K_{\pi_2}\varphi \end{aligned}$$

- $K_{\pi^*}\varphi \to \varphi \wedge K_\pi K_{\pi^*}\varphi$: Suppose $M, s \vDash K_{\pi^*}\varphi$, and take an arbitrary $t \in M$ such that $s \sim_\pi; (\sim_\pi)^* t$. Then since $\sim_{\pi^*} = (\sim_\pi)^* = \sim_\pi; (\sim_\pi)^*$, we get that it is also the case that $s \sim_{\pi^*} t$, but then since $M, s \vDash K_{\pi^*}\varphi$, we have that $M, t \vDash \varphi$, and hence $M, s \vDash K_\pi K_{\pi^*}\varphi$. Note that as $(\sim_\pi)^*$ is the reflexive, transitive closure, it is immediate that $M, s \vDash \varphi$.

- $K_{\pi^*}(\varphi \to K_\pi\varphi) \to (\varphi \to K_{\pi^*}\varphi)$: Suppose $M, s \vDash K_{\pi^*}(\varphi \to K_\pi\varphi)$ and $M, s \vDash \varphi$, and take an arbitrary t such that $s \sim_{\pi^*} t$. Then $s(\sim_\pi)^* t$, and thus there is a sequence of $u_1 \dots u_n$ such that $s \sim_\pi u_1 \sim_\pi \cdots \sim_\pi u_n \sim_\pi t$. As $s \sim_{\pi^*} s$ and $M, s \vDash \varphi$, we have that $s \vDash K_\pi\varphi$, and thus $M, u_1 \vDash \varphi$. But then as $s \sim_\pi u_1$, also $s \sim_{\pi^*} u_1$, and thus $M, u_1 \vDash K_\pi\varphi$. Repeating this argument eventually gives us that $M, t \vDash \varphi$, and thus $M, s \vDash K_{\pi^*}\varphi$.

- $K_{?\theta}\psi \leftrightarrow (\theta \to \psi)$:

$$\begin{aligned} M, s \vDash K_{?\theta}\psi &\Leftrightarrow \forall t \text{ s.t. } s \sim_{?\theta} t : M, t \vDash \psi \\ &\Leftrightarrow \forall t \text{ s.t. } t = s \text{ and } M, s \vDash \theta : M, t \vDash \psi \\ &\Leftrightarrow \forall t\big((t = s \wedge M, t \vDash \theta) \to M, t \vDash \psi\big) \\ &\Leftrightarrow M, s \vDash \theta \to M, s \vDash \psi \\ &\Leftrightarrow M, s \vDash \theta \to \psi \end{aligned}$$

□

Definition 4.17 (Canonical Structure)**.** Let $M^C = \langle S^C, \sim_\pi, \approx_I, V^C \rangle$ be such that

- $S^C = \{s; s \text{ is a maximally consistent set}\}$
- $s \sim_\pi t$ if and only if $\forall\varphi(K_\pi\varphi \in s \to \varphi \in t)$
- $s \approx_I t$ if and only if $\forall\varphi([I]\varphi \in s \to \varphi \in t)$.
- $s \in V^C(p)$ if and only if $p \in s$.

Lemma 4.18 (Truth Lemma)**.** *For all $\varphi \in \mathcal{L}_{GE\text{-}PDL^-}$ we have that in the canonical structure M^C, $s \vDash \varphi$ if and only if $\varphi \in s$.*

Proposition 4.19. *The canonical structure is a pseudo-model*

Proof. We will prove this per property of pseudo-models.

- To show that $\sim_I$ is reflexive, suppose $K_I\varphi \in s$. Then by our axioms, $\varphi \in s$. From the definition of $\sim_I$, it follows that $\mathrm{id} \subseteq \sim_I$, and thus that $\sim_I$ is reflexive.
- To show that $\sim_I$ is transitive, suppose $s \sim_I t \sim_I s$, and suppose $K_I\varphi \in s$. By axiom 4, we get that $K_IK_I\varphi \in s$. From that we obtain that $K_I\varphi \in t$, and thus $\varphi \in s$. Thus, by definition of $\sim_I$, we have that $s \sim_I s$.
- To show that $\sim_I$ is symmetric, suppose $s \sim_I t$ and $\varphi \in s$. Then by axiom (B), we have $K_I\hat{K}_I\varphi \in s$. Since $s \sim_i t$, we then get that $\hat{K}_I \in t$. By definition, this means that $t \sim_I s$.
- Equivalence for $\approx_I$ follows the same lines as the previous case.
- To show that $\sim_I \subseteq \approx_I$, suppose $s \sim_I s'$, and let $[I]\varphi \in s$. Then since we have the axiom $[I]\varphi \to K_I\varphi$, we get that $K_I\varphi \in s$. But then as $s \sim_I s'$, this means that $\varphi \in s'$, and thus $s \approx_I s'$.
- To show that $\sim_I \subseteq \sim_J$ for $J \subseteq I$, assume $s \sim_I s'$, and let $K_J\varphi \in s$. Then since $K_J\varphi \to K_I\varphi$ is an axiom, we get $K_I\varphi \in s$. But then as $s \sim_I s'$, we get that $\varphi \in s'$, and hence $s \sim_J s'$.
- $\approx_I \subseteq \approx_J$ for $J \subseteq I$ is similar to the $\approx$ case.
- To show that $\approx_{\mathcal{A}\cup\{0\}} = \mathrm{id}$, suppose that $s \approx_{\mathcal{A}\cup\{0\}} s'$, and let $\varphi \in s$. Then by the axiom $\varphi \to [\mathcal{A} \cup \{0\}]\varphi$, we get that $[\mathcal{A} \cup \{0\}]\varphi \in s$. But then, as $s \approx_{\mathcal{A}\cup\{0\}} s'$, we have that $\varphi \in s'$. Hence, $s \subseteq s'$, but as both are maximally consistent sets, and hence s' cannot be strictly bigger than s, it must be the case that $s = s'$, and thus $\approx_{\mathcal{A}\cup\{0\}} = \mathrm{id}$.
- $\sim_{\pi_1;\pi_2} = \sim_{\pi_1}; \sim_{\pi_2}$.
  For the right-to-left direction, suppose $s \sim_{\pi_1}; \sim_{\pi_2} t$. Then there is a u such that $s \sim_{\pi_1} u$ and $u \sim_{\pi_2} t$. Thus, by definition, there is a u such that $\forall\varphi(K_{\pi_1}\varphi \in s \to \varphi \in u)$ and $\forall\psi(K_{\pi_2}\psi \in u \to \psi \in t)$. Now suppose $K_{\pi_1;\pi_2}\varphi \in s$. Then

$K_{\pi_1}K_{\pi_2}\varphi \in s$. Hence $K_{\pi_2}\varphi \in u$, and thus $\varphi \in t$. This means that $s \sim_{\pi_1;\pi_2} t$.
For the left-to-right direction, suppose $s \sim_{\pi_1;\pi_2} t$. Then $\forall\varphi(K_{\pi_1;\pi_2}\varphi \in s \rightarrow \varphi \in t)$. We can rewrite this as $\forall\varphi(K_{\pi_1}K_{\pi_2}\varphi \in s \rightarrow \varphi \in t)$. Now we have to construct a w such that $s \sim_{\pi_1} w$ and $w \sim_{\pi_2} t$.
Let $\{\varphi; K_{\pi_1}\varphi \in s\} \cup \{\hat{K}_{\pi_2}\psi; \psi \in t\} \subseteq w'$. If we can show that w' is consistent, then we know that there is a maximally consistent set w that contains w', which ensures that $s \sim_{\pi_1} w$ and $w \sim_{\pi_2} t$, which means that $s \sim_{\pi_1};\sim_{\pi_2} t$. Hence, showing that w' is consistent completes this part of the proof.
Suppose towards a contradiction that w' is inconsistent. Hence, suppose there are $\varphi_1, \ldots, \varphi_n$ and $\psi_1, \ldots, \psi_m$ such that $K_{\pi_1}\varphi_1, \ldots, K_{\pi_1}\varphi_n \in s$, and $\psi_1, \ldots, \psi_n \in t$, and $\vdash (\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \hat{K}_{\pi_2}\psi_1 \wedge \cdots \wedge \hat{K}_{\pi_2}\psi_n) \rightarrow \bot$.
Now take $\varphi := \varphi_1 \wedge \cdots \wedge \varphi_n$. We claim that $K_{\pi_1}\varphi \in s$, since it is the case that $\vdash K_{\pi_1}\varphi \leftrightarrow K_{\pi_1}\varphi_1 \wedge \cdots \wedge K_{\pi_1}\varphi_n$. Since for all i we have that $K_{\pi_1}\varphi_i \in s$, it is surely the case that the conjunction is in s, and thus also $K_{\pi_1}\varphi$.
Now let $\psi := \psi_1 \wedge \cdots \wedge \psi_m$. We claim that $\psi \in t$. For this it is enough to note that as each $\psi_i \in t$, surely also the conjunction is.
Now since
$\vdash \varphi \rightarrow (\varphi_1 \wedge \cdots \wedge \varphi_n)$
and
$\vdash \hat{K}_{\pi_2}\psi \rightarrow (\hat{K}_{\pi_2}\psi_1 \wedge \cdots \wedge \hat{K}_{\pi_2}\psi_n)$
it is also the case that
$\vdash \varphi \wedge \hat{K}_{\pi_2}\psi \rightarrow (\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \hat{K}_{\pi_2}\psi_1 \wedge \cdots \wedge \hat{K}_{\pi_2}\psi_n)$.
But then as
$\vdash (\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \hat{K}_{\pi_2}\psi_1 \wedge \cdots \wedge \hat{K}_{\pi_2}\psi_n) \rightarrow \bot$
we get that
$\vdash (\varphi \wedge \hat{K}_{\pi_2}\psi) \rightarrow \bot$.
This rewrites to
$\vdash \varphi \rightarrow K_{\pi_2}\neg\psi$.
But then by necessitation and the (K) axiom, we get
$\vdash K_{\pi_1}\varphi \rightarrow K_{\pi_1}K_{\pi_2}\neg\psi$.
Then as we just showed that $K_{\pi_1}\varphi \in s$, it is also the case that $K_{\pi_1}K_{\pi_2}\neg\psi \in s$. But then also $K_{\pi_1;\pi_2}\neg\psi \in s$. But then as $s \sim_{\pi_1;\pi_2} t$, it must be the case that $\neg\psi \in t$, but we showed that $\psi \in t$, thus we have a contradiction. Hence, w' is consistent.

- $\sim_{\pi_1+\pi_2}=\sim_{\pi_1} \cup \sim_{\pi_2}$: Take arbitrary $s,t \in S$. Then $s \sim_{\pi_1+\pi_2} t$ if and only if $\forall\varphi(K_{\pi_1+\pi_2}\varphi \in s \rightarrow \varphi \in t)$. This is the case if and only if $\forall\varphi(K_{\pi_1}\varphi \wedge K_{\pi_2}\varphi \in s \rightarrow \varphi \in t)$, which happens if and only if $\forall\varphi((K_{\pi_1}\varphi \in s \rightarrow \varphi \in t) \vee (K_{\pi_2}\varphi \in s \rightarrow \varphi \in t))$. Which is the case if and only if $s \sim_{\pi_1} t$ or $s \sim_{\pi_2} t$, which is equivalent to $s \sim_{\pi_1} \cup \sim_{\pi_2} t$.

- $(\sim_\pi)^* \subseteq \sim_{\pi^*}$: Suppose $s(\sim_\pi)^*t$. Then there is a sequence $u_1, \ldots, u_n$ such that $s \sim_\pi u_1 \sim_\pi \cdots \sim_\pi u_n \sim_\pi t$. Now suppose $K_{\pi^*}\varphi \in s$. Then by since $\vdash K_{\pi^*}\varphi \rightarrow K_\pi K_{\pi^*}\varphi$, we have that $K_\pi K_{\pi^*}\varphi \in s$. This means that $K_{\pi^*}\varphi \in u_1$, but that

in turn means that $K_\pi K_{\pi^*}\varphi \in u_1$, which gives us that $K_{\pi^*}\varphi \in u_2$. Repeating this argument eventually gives us that $K_{\pi^*}\varphi \in t$. Then since $\vdash K_{\pi^*}\varphi \to \varphi$, it is the case that $\varphi \in t$, and thus $s \sim_{\pi^*} t$.

- $s \sim_{?\theta} t$ if and only if $s = t$ and $s \vDash \theta$:
  For the left-to-right direction suppose $s \sim_{?\theta} t$. Then by construction $\forall\psi(K_{?\theta}\psi \to \psi \in t)$. Then since $\vdash K_{?\theta}\varphi \to (\theta \to \varphi)$, this means $\forall\psi((\theta \to \psi) \in s \to \psi \in t)$. It is trivial that $\theta \to \theta \in s$, which means that $\theta \in t$. Now take an arbitrary $\psi \in s$. Then $\theta \to \psi \in s$, and hence $\psi \in t$. Thus $s \subseteq t$, but as they are both maximally consistent sets, we get that $s = t$. Since $\theta \in t$, we get by the Truth Lemma that $t \vDash \theta$, and thus we get that $s = t$ and $s \vDash \theta$.
  For the right-to-left direction suppose $s = t$ and $s \vDash \theta$, and suppose $\theta \to \psi \in s$ for some ψ. Then clearly $\psi \in s$, and as $s = t$, we get that $\psi \in t$, so for any ψ it is the case that $(\theta \to \psi \in s \to \psi \in t)$, so for any ψ we have $(K_{?\theta}\psi \in s \to \psi \in t)$, and hence $s \sim_{?\theta} t$.

□

Proposition 4.20. *$\mathcal{L}_{GE\text{-}PDL^-}$ is complete with respect to pseudo-models.*

STEP 1b: Decidability

In this step we will show that the logic is decidable. We will do this using a filtration.

Definition 4.21 (Closed set under single negation)**.** Let Γ be a set of formulas. Then Γ is closed under single negation if and only if $\varphi \in \Gamma$ implies that $\sim \varphi \in \Gamma$, where

$$\sim \varphi = \begin{cases} \psi & \text{if } \varphi = \neg\psi \\ \neg\psi & \text{else} \end{cases}$$

Definition 4.22 (A suitable set)**.** Let φ be in the language. Then Σ_φ is *a suitable set for* φ if it is the smallest set such that;

(1) $\psi \in \Sigma_\varphi$

(2) Σ_φ is closed under subformulas

(3) Σ_φ is closed under single negation

(4) $K_I\psi \in \Sigma_\varphi$ implies $K_J K_I\psi \in \Sigma_\varphi$ for $I \subset J$

(5) $[I]\psi \in \Sigma_\varphi$ implies $[J][I]\psi \in \Sigma_\varphi$ for $I \subset J$

(6) $[I]\psi \in \Sigma_\varphi$ implies $K_I[I]\psi \in \Sigma_\varphi$

(7) $\psi \in \Sigma_\varphi$ implies $[\mathcal{A} \cup \{0\}]\psi \in \Sigma_\varphi$ if ψ is *not* of the form $[\mathcal{A} \cup \{0\}]\theta$ for some θ.

(8) $K_{\pi_1;\pi_2}\psi \in \Sigma_\varphi$ implies $K_{\pi_1}K_{\pi_2}\psi \in \Sigma_\varphi$.

(9) $K_{\pi_1+\pi_2}\psi \in \Sigma_\varphi$ implies $K_{\pi_1}\psi \wedge K_{\pi_2}\psi \in \Sigma_\varphi$

(10) $K_{\pi^*}\psi \in \Sigma_\varphi$ implies $K_\pi K_{\pi^*}\psi \in \Sigma_\varphi$

(11) $K_{?\theta}\psi \in \Sigma_\varphi$ implies $\theta \to \psi \in \Sigma_\varphi$.

Lemma 4.23. *Let Σ_φ be a suitable set for some $\varphi \in \mathcal{L}_{GE\text{-}PDL^-}$. Then Σ_φ is finite.*

Proof. Define $\Sigma_\varphi^{0_b} := \{\psi; \psi \text{ is a subformula of } \varphi\}$. Note that this is finite, since φ is defined by recursion and thus the subformula relation is well founded.
We will now recursively make sure that the set is closed under conditions (8)–(11). Let

$$\begin{aligned}\Sigma_\varphi^{i_a} := \Sigma_\varphi^{(i-1)_b} &\cup \{K_{\pi_1}K_{\pi_2}\psi; K_{\pi_1;\pi_2}\psi \in \Sigma_\varphi^{(i-1)_b}\}\\ &\cup \{K_{\pi_1}\psi \wedge K_{\pi_2}\psi; K_{\pi_1+\pi_2}\psi \in \Sigma_\varphi^{(i-1)_b}\}\\ &\cup \{K_\pi K_{\pi^*}\psi; K_{\pi*}\psi \in \Sigma_\varphi^{(i-1)_b}\}\\ &\cup \{\theta \to \psi; K_{?\theta}\psi \in \Sigma_\varphi^{(i-1)_b}\}\end{aligned}$$

and define $\Sigma_\varphi^{i_b} := \{\psi; \psi \text{ is a subformula of some } \theta \text{ such that } \theta \in \Sigma_\varphi^{i_a}\}$.
We keep recursively closing the set until we arrive at $\Sigma_\varphi^{n_b}$ such that $\Sigma_\varphi^{n_b} = \Sigma_\varphi^{(n-1)_b}$. Let $\Sigma_\varphi^n = \Sigma_\varphi^{n_b}$, and note that the only programs that feature in it are basic programs, and that it is closed under conditions (1), (2) and (8)–(11). We now argue that there exists an n for which $\Sigma_\varphi^{n_b} = \Sigma_\varphi^{(n-1)_b}$, and that Σ_φ^n is finite.
First, there is an n such that $\Sigma_\varphi^{n_b} = \Sigma_\varphi^{(n-1)_b}$, because programs are defined recursively, which means that with the above procedure, all programs will eventually be reduced to basic programs.
Secondly, Σ_φ^n is finite, because at each step i we add at most $2 \times |\Sigma_\varphi^{(i-1)_b}|$ formulas, namely the new formula in step $(i-1)_a$ and its subformula in step $(i-1)_b$. Hence, Σ_φ^n is finite.
Now we will make sure Σ_φ is closed under conditions (3) – (7) as well: Define

$$\begin{aligned}\Sigma_\varphi^{n+1} := \Sigma_\varphi^n &\cup \{D_{J_1}D_{J_2}\dots D_{J_m}D_I\theta; D_I\theta \in \Sigma_\varphi^n, I \subset J_m \subset \dots \subset J_2 \subset J_1\}\\ &\cup \{[J_1][J_2]\dots[J_m][I]\theta; [I]\theta \in \Sigma_\varphi^n, I \subset J_m \subset \dots \subset J_2 \subset J_1\}\\ &\cup \{D_{J_1}\dots D_{J_m}D_{I_1}[I_1][I_2]\dots[I_n]\theta; [I_n]\theta \in \Sigma_\varphi^n, I_n \subset \dots I_1 \subset J_m \dots \subset J_1\}\end{aligned}$$

And finally let $\Sigma_\varphi := \Sigma_\varphi^{n+1} \cup \{\sim\theta; \theta \in \Sigma_\varphi^{n+1}\}$.
The proof that this is finite follows the same lines as the proof of Lemma 3.25.
□

Definition 4.24. Let $M = \langle S, \sim_\pi, \approx_I, V\rangle$ be a general Kripke model, and consider a suitable set Σ_φ for some $\varphi \in \mathcal{L}_{GE\text{-}PDL^-}$. Then we define a general Kripke model $M^+ = \langle S^+, \sim_\pi^+, \approx_I^+, V^+\rangle$ such that

- $S^+ = \{[s]; s \in S\}$;
- $[s] \sim_I^+ [t]$ if and only if $\forall K_I\psi \in \Sigma(M, s \vDash K_I\psi \Leftrightarrow M, t \vDash K_I\psi)$;
- $[s] \approx_I^+ [t]$ if and only if $\forall [I]\psi \in \Sigma(M, s \vDash [I]\psi \Leftrightarrow M, t \vDash [I]\psi)$;
- We define by induction
  - $[s] \sim_{?\theta}^+ [t]$ if and only if $[s] = [t]$ and $M, s \vDash \theta$
  - $\sim_{\pi_1;\pi_2}^+ = \sim_{\pi_1}^+ ; \sim_{\pi_2}^+$
  - $\sim_{\pi_1+\pi_2}^+ = \sim_{\pi_1}^+ \cup \sim_{\pi_2}^+$
  - $\sim_{\pi^*}^+ = (\sim_{\pi}^+)^*$
- $V^+(p) = \{[s]; s \in V(p)\}$.

Lemma 4.25. *M^+ is a filtration of M through Σ_φ.*

Proof. Clearly, M^+ satisfies the constraints on S^f and V^f, so it is left to show that $\sim_\pi^+$ and $\approx_I^+$ satisfy min^f and max^f.

- We will show min^f for $\sim_\pi^+$ by induction. Hence, we will show that for all $[s], [t] \in S^+$, if $s \sim_\pi t$, then $[s] \sim_\pi^+ [t]$.
  Base case Take arbitrary $[s], [t] \in S^+$ such that $s \sim_I t$, and suppose $K_I\psi \in \Sigma_\varphi$. Then, as $\sim_I$ is an equivalence relation, we get that $M, s \vDash K_I\psi$ if and only if $M, t \vDash K_I\psi$. Hence, by definition, we get $[s] \sim_I^+ [t]$.
  Inductive step
  - Take an arbitrary$[s], [t] \in S^+$ such that $s \sim_{\pi_1;\pi_2} t$. Then by definition $s \sim_{\pi_1}; \sim_{\pi_2} t$, so there is a u such that $s \sim_{\pi_1} u \sim_{\pi_2} t$. By the induction hypothesis, min^f holds for π_1 and π_2, so $[s] \sim_{\pi_1}^+ [u] \sim_{\pi_2}^+ [t]$. This means that $[s] \sim_{\pi_1}^+ ; \sim_{\pi_2}^+ [t]$, and thus by definition $[s] \sim_{\pi_1;\pi_2}^+ [t]$.
  - Take arbitrary $[s], [t] \in S^+$ such that $s \sim_{\pi_1+\pi_2} t$. Then by definition $s \sim_{\pi_1} \cup \sim_{\pi_2} t$. By the induction hypothesis, for both π_1 and π_2 we have that min^f holds, so we know that $[s] \sim_{\pi_1}^+ [t]$ or $[s] \sim_{\pi_2}^+ [t]$. But this means that $[s] \sim_{\pi_1}^+ \cup \sim_{\pi_2}^+ [t]$, and hence $[s] \sim_{\pi_1+\pi_2}^+ [t]$.
  - Take arbitrary $[s], [t] \in S^+$ such that $s \sim_{?\theta} t$. Then by definition, $s = t$ and $M, s \vDash \theta$. But this means that $[s] = [t]$ and $M, s \vDash \theta$, and thus $[s] \sim_{?\theta} [t]$.
  - Take arbitrary $[s], [t] \in S^+$ such that $s \sim_{\pi^*} t$. Define $A := \{[u]; [s] \sim_{\pi^*}^+ [u]\}$ and define $Form(s)$ to be the conjunction of formulas in Σ_φ that are true in M, s, and define σ_A to be the disjuction of all $Form(a)$ such that $[a] \in A$. Then $M, s \vDash \sigma_A$ if and only if $[s] \in A$.

Claim 1:$M, s \vDash K_{\pi^*}\sigma_A$.
To see this, note that $M, s \vDash K_{\pi^*}(\sigma_A \to K_\pi K_{\pi^*}\sigma_A) \to (\sigma_A \to K_{\pi^*}\sigma_A)$. Also we will prove the following claim. Together with the fact that $M, s \vDash \sigma_A$ since $[s] \sim^+_{\pi^*} [s]$, and thus $[s] \in A$, Claim 2 implies Claim 1.

Claim 2 $M, s \vDash K_{\pi^*}(\sigma_A \to K_\pi K_{\pi^*}\sigma_A)$
Take an arbitrary t such that $s \sim_{\pi^*} x$ and suppose $M, x \vDash \sigma_A$. We have to show that $M, x \vDash K_\pi K_{\pi^*}\sigma_A$, so take an arbitrary y such that $x \sim_\pi y$. Then since $x \in A$, we get that $[s] \sim^+_{\pi^*} [t]$, and thus $[s](\sim^+_\pi)^*[t]$, which means that there is an n such that $[s](\sim^+_\pi)^n[t]$. By the induction hypothesis, we have that $[x] \sim^+_\pi [y]$, and thus $[s](\sim^+_\pi)^{n+1}[y]$, and thus $[s] \sim^+_{\pi^*} [y]$. This means that $y \in \sigma_A$, and thus $M, y \vDash \sigma_A$, and therefore $M, t \vDash K_\pi\sigma_A$. However, since $[s] \sim^+_{\pi^*} [y]$, we can do repeat the same procedure to obtain that $M, y \vDash K_\pi\sigma_A$. By repetition, we get that $M, x \vDash K_\pi K_\pi \dots K_\pi\sigma_A$, and therefore $M, x \vDash K_\pi K_{\pi^*}\sigma_A$, which proves Claim 2, and thus we proved Claim 1. But this gives us what we want, as this implies that $M, t \vDash \sigma_A$ since $s \sim_{\pi^*} t$, and thus $[t] \in A$, and therefore $[s] \sim^+_{\pi^*} [t]$.

- We will show max^f for $\sim^+_\pi$ by induction. Hence, we will show that for all $[s], [t] \in S^+$, if $[s] \sim^+_\pi [t]$, then for all $\Box\varphi \in \Sigma_\varphi(M, s \vDash \Box\varphi \to M, t \vDash \varphi)$
  Base case Take arbitrary $[s], [t]$ such that $[s] \sim^+_I [t]$, and suppose $K_I\psi \in \Sigma_\varphi$ and that $M, s \vDash K_I\psi$. Then $M, t \vDash K_I\psi$. But as $\sim_I$ is reflexive, we get that $M, t \vDash \psi$.
  Inductive step
  - Take arbitrary $[s], [t]$ such that $[s] \sim^+_{\pi_1;\pi_2} [t]$, and suppose $K_{\pi_1;\pi_2}\psi \in \Sigma_\varphi$ and that $M, s \vDash K_{\pi_1;\pi_2}\psi$. As $[s] \sim^+_{\pi_1;\pi_2} [t]$, we get that $[s] \sim^+_{\pi_1}; \sim^+_{\pi_2} [t]$, and thus that there is a $[u]$ such that $[s] \sim^+_{\pi_1} [u] \sim^+_{\pi_2} [t]$. Also note that as $M, s \vDash K_{\pi_1;\pi_2}\psi$, also $M, s \vDash K_{\pi_1}K_{\pi_2}\psi$. As max^f holds for $\sim_{\pi_1}$ and Σ_φ is a suitable set, we get that $M, u \vDash K_{\pi_2}\psi$, and thus, as max^f holds for $\sim_{\pi_2}$, we get that $M, t \vDash \psi$.
  - Take arbitrary $[s], [t]$ such that $[s] \sim^+_{\pi_1+\pi_2} [t]$ and suppose $K_{\pi_1+\pi_2}\psi \in \Sigma_\varphi$ and that $M, s \vDash K_{\pi_1+\pi_2}\psi$. As $[s] \sim^+_{\pi_1+\pi_2} [t]$, we get that $[s] \sim^+_{\pi_1} \cup \sim^+_{\pi_2} [t]$, hence $[s] \sim^+_{\pi_1} [t]$ or $[s] \sim^+_{\pi_2} [t]$. As $M, s \vDash K_{\pi_1+\pi_2}\psi$, also $M, s \vDash K_{\pi_1}\psi \wedge K_{\pi_2}\psi$. Then since max^f holds for both $\sim^+_{\pi_1}$ and $\sim^+_{\pi_2}$ and Σ_φ is suitable, we get that $M, t \vDash \psi$.
  - Take arbitrary $[s], [t]$ such that $[s] \sim^+_{?\theta} [t]$, and suppose $K_{?\theta}\psi \in \Sigma_\varphi$ and $M, s \vDash K_{?\theta}\psi$. Since $[s] \sim^+_{?\theta} [t]$, we have that $[s] = [t]$ and $M, s \vDash \theta$. As $M, s \vDash K_{?\theta}\psi$, it is also the case that $M, s \vDash \theta \to \psi$. But then $M, s \vDash \psi$. Then since $[s] = [t]$ we know that $\forall\sigma \in \Sigma_\varphi(M, s \vDash \sigma \Leftrightarrow M, t \vDash \sigma)$. This combined with the fact that Σ_φ is a suitable set, we know that $M, t \vDash \psi$.

– Take arbitrary $[s],[t]$ such that $[s] \sim^{+}_{\pi^*} [t]$ and suppose $M, s \vDash K_{\pi^*}\psi$ for some $K_{\pi^*}\psi \in \Sigma_\varphi$. As $[s](\sim^{+}_{\pi})^*[t]$, there is a sequence $[s] = [u_1], [u_2], \ldots, [u_n] = [t]$ such that for all $k < n, [u_k] \sim_\pi [u_{k+1}]$. Note that as Σ_φ is suitable, we have that $K_\pi K_{\pi^*}\psi \in \Sigma_\varphi$, and as $M, s \vDash K_{\pi^*}\psi$, we also have that $M, s \vDash K_\pi K_{\pi^*}\psi$. Together, this means that $M, u_1 \vDash K_{\pi^*}\psi$. By repeating this argument, we eventually obtain that $M, t \vDash K_{\pi^*}\psi$. But then since $\vdash K_{\pi^*}\psi \to \psi$, we have that $M, t \vDash \psi$.

- Both min^f and max^f for $\approx^{+}_I$ are analogous to the basecase of $\sim^{+}_\pi$.

□

Corollary 4.26. *Let $M = \langle S, \sim_I, \approx_I, V\rangle$ be a general Kripke model, and let M^+ and Σ_φ be as above. Then for all $\sigma \in \Sigma_\varphi$ and $s \in S$, we get that*

$$M, s \vDash \sigma \Leftrightarrow M^+, [s] \vDash \sigma$$

Proof. This follows from Lemma 4.12 and Lemma 4.25 □

Definition 4.27 (Regular Pseudo-Model)**.** A general Kripke model $M = \langle S, \sim_\pi, \approx_I, V\rangle$ is a *regular pseudo-model* if it satisfies all conditions on pseudo-models and in addition satisfies the following constraint:

- $\sim_{\pi^*} = (\sim_\pi)^*$

Theorem 4.28. *Let $M = \langle S, \sim_I, \approx_I, V\rangle$ be a pseudo-model. Then M^+ as described above is a regular pseudo-model.*

Proof. To show that M^+ is a pseudo-model we have to show that it satisfies all the semantic properties of pseudo-models.

- It is clear that $\sim^{+}_I$ and $\approx^{+}_I$ are equivalence relations.

- $\sim^{+}_I \subseteq \approx^{+}_I$: Take arbitrary $[s],[t] \in S^+$ such that $[s] \sim^{+}_I [t]$. This means that $\forall D_I\psi \in \Sigma_\varphi (M, s \vDash D_I\psi \Leftrightarrow M, t \vDash D_I\psi)$.
  Let $[I]\psi \in \Sigma_\varphi$, and suppose that $M, s \vDash [I]\psi$. As $\vdash [I]\psi \to D_I[I]\psi$, we get that $M, s \vDash D_I[I]\psi$. But then, as $D_I[I]\psi \in \Sigma_\varphi$ and $[s] \sim^{+}_I [t]$, we have $M, t \vDash D_I[I]\psi$. As D_I is truthful, it is the case that $M, t \vDash [I]\psi$, and hence $[s] \approx^{+}_I [t]$.

- $\sim^{+}_I \subseteq \sim^{+}_J$ for $J \subseteq I$: Take arbitrary $[s],[t] \in S^+$ such that $[s] \sim^{+}_I [t]$.
  Clearly, if $J = I$, it is immediate that $[s] \sim^{+}_J [t]$, so let's focus on the case where $J \subset I$. Suppose $D_J\psi \in \Sigma_\varphi$, and $M, s \vDash D_J\psi$. As $\vdash D_J\psi \to D_J D_J\psi$, we can apply the Monotonicity of Distributed Knowledge axiom to get $\vdash D_J D_J\psi \to D_I D_J\psi$. Hence, we have that $M, s \vDash D_I D_J\psi$. But then as $[s] \sim^{+}_I [t]$ and $D_I D_J\psi \in \Sigma_\varphi$ by construction of Σ_φ, we get that $M, t \vDash D_I D_J\psi$. Again, as D_I is truthful, we obtain $M, t \vDash D_J\psi$, and thus $[s] \sim^{+}_J [t]$.

- $\approx^+_I \subseteq \approx^+_J$ for $J \subseteq I$: this is analogous to the previous case.
- $\approx^+_{\mathcal{A}\cup\{0\}} = \mathrm{id}$: Take arbitrary $[s], [t] \in S^+$ such that $[s] \approx^+_{\mathcal{A}\cup\{0\}} [t]$, and suppose $M, s \vDash \psi$ for some $\psi \in \mathcal{L}$, and let Σ_φ be the suitable set for ψ.
  (a) Suppose ψ is of the form $[\mathcal{A} \cup \{0\}]\theta$, thus $M, s \vDash [\mathcal{A} \cup \{0\}]\theta$. Then as $[s] \approx^+_{\mathcal{A}\cup\{0\}} [t]$ and Σ_φ is closed under subformulas, we get that $M, t \vDash [\mathcal{A} \cup \{0\}]\theta$. Thus we have that for all $\psi \in \mathcal{L}, M, s \vDash \psi$ if and only if $M, t \vDash \psi$. Hence it is the case that $[s] = [t]$.
  (b) Now suppose ψ is *not* of the form $[\mathcal{A} \cup \{0\}]\theta$. Then by the Determinism of Grand Coalition axiom, we get that $M, s \vDash [\mathcal{A} \cup \{0\}]\psi$. By construction of Σ_φ and since $[s] \approx^+_{\mathcal{A}\cup\{0\}} [t]$, we get that $M, t \vDash [\mathcal{A} \cup \{0\}]\psi$. As $[\mathcal{A} \cup \{0\}]$ is truthful, we get that $M, t \vDash \psi$. Thus we have that for all $\psi \in \mathcal{L}, M, s \vDash \psi$ if and only if $M, t \vDash \psi$. Hence it is the case that $[s] = [t]$.
- That $\sim_{\pi_1;\pi_2} = \sim_{\pi_1}; \sim_{\pi_2}$ and $\sim_{\pi_1+\pi_2} = \sim_{\pi_1} \cup \sim_{\pi_2}$ and $\sim_{\pi^*} = (\sim_\pi)^*$ follows immediately from the definition.
- That $[s] \sim^+_{?\theta} [t]$ iff $[s] = [t]$ and $M^+, [s] \vDash \theta$ follows from the definition and Corollary 4.26.

□

Lemma 4.29. *$\mathcal{L}_{GE\text{-}PDL^-}$ has the strong finite model property with respect to pseudo-models.*

Proof. Let φ be a formula of $\mathcal{L}_{GE\text{-}PDL^-}$. Then it is satisfiable if and only if it is satisfied in the canonical structure M^C by Proposition 4.20. Now let M^+ be the filtration of M^C over the suitable set Σ_φ for φ. Then by Corollary 4.26, φ is satisfied in M^C if and only if it is satisfied in M^+, hence φ is satisfiable iff it is satisfied in M^+. Also, we know that M^+ has at most $2^{|\Sigma_\varphi|}$ states. Hence, every satisfiable formula is satisfied in a model containing at most $2^{|\Sigma_\varphi|}$ states, thus giving GE-PDL$^-$ strong finite model property. □

Theorem 4.30. *The logic $\mathcal{L}_{GE\text{-}PDL^-}$ is decidable.*

Proof. This follows from Lemma 4.29 and Theorem 6.7 in [9, p.340] which states that any normal modal logic that has the strong finite model property with respect to a recursive set of models is decidable. □

STEP 2: Unraveling

Now we will partially unravel the regular pseudo-model that we constructed at the end of step 1. For that we need a few notions.

Definition 4.31 (History)**.** Let $M = \langle S, \sim_\pi, \approx_I, V\rangle_{I\subseteq\mathcal{A}\cup\{0\}}$ be a general Kripke model and take some $s \in S$. Then a *history with origin* s is a finite sequence $h := (s_0, R_0, s_1, \ldots, R_{n-1}, s_n)$ such that

- for all $k \leq n : s_k \in S$;
- $s_0 = s$;
- for all $k < n : R_k \in \{\sim_I; I \text{ is a basic program}\} \cup \{\approx_I; I \subset \mathcal{A} \cup \{0\}\}$;
- for all $k \leq n : s_k R_k s_{k+1}$.

For any history h we write $first(h) = s_0$ and $last(h) = s_n$. These histories will form the state space of the unraveled tree.
For two histories $h = (s_0, R_0, \ldots, R_{n-1}, s_n)$ and $h' = (s'_0, R'_0, \ldots, R'_{m-1}, s'_m)$ we write the concatenation of the two $h + h' := (s_0, R_0, \ldots, R_{n-1}, s_n = s'_0, R'_0, \ldots, R'_{m-1}, s'_m)$.

Note that the R_k's come from the union of all $\sim_I$ relations where $I \subseteq \mathcal{A}$ and all $\approx_I$ relations with $I \subset \mathcal{A} \cup \{0\}$. Hence, we do not unravel the $\approx_{\mathcal{A}\cup\{0\}}$ relation. This is to ensure the semantical property of determinism of the grand coalition when we later go back to the actual models. We also do not unravel the more complex programs, as we do not need them.

Definition 4.32 (Unraveled tree)**.** Let M be a general Kripke model and let $s \in S$. The *unraveling* of M around s is a general Kripke model $\vec{M} = \langle \vec{S}, R_{\sim_\pi}, R_{\approx_I}, \vec{V}\rangle_{I\subseteq\mathcal{A}\cup\{0\}}$ such that

- $\vec{S} = \{h; \text{first}(h) = s\}$
- $hR_{\sim_I}h'$ iff $h + (last(h), \sim_I, s) = h'$
- $hR_{\approx_I}h'$ iff $h + (last(h), \approx_I, s) = h'$
- $\vec{V} : P \to \mathcal{P}(\vec{S})$ such that $\vec{V}(p) := \{h \in \vec{S}; last(h) \in V(p)\}$.

Now we have defined histories on the canonical pseudo-model. They basically tell us which worlds in M^C are related by any sequence of relations from a specific state s. These form a tree. Now we will define paths on this tree of histories.

Definition 4.33 ($\mathcal{R}$-path)**.** Let $\vec{M}$ be the unraveling of a general Kripke model M around some world $s \in S$. Let $\mathcal{R} \subseteq \{R_{\sim_I}, R^{-1}_{\sim_I}, R_{\approx_J}, R^{-1}_{\approx_J}; I \subseteq \mathcal{A}, J \subseteq \mathcal{A} \cup \{0\}\} =: Rel$. An $\mathcal{R}$-*path* from h to h' is a finite sequence $p := (h_0, R_0, h_1, \ldots, R_{n-1}, h_n)$ such that

- for all $k \leq n : h_k \in \vec{S}$;
- $h_0 = h$;
- $h_n = h'$;

- for all $k < n : R_k \in \mathcal{R}$;
- for all $k < n : h_k R_k h_{k+1}$.

If $\mathcal{R}$ is not further specified, we speak of a *path*. For any path p we define again $first(p) = h_0$ and $last(p) = h_n$. Composing paths works the same as composing histories.

Definition 4.34 (Non-redundancy)**.** Let $\mathcal{R} \subseteq Rel$ and p an $\mathcal{R}$-path. We say that p is a non-redundant path if there is no $k < n-1$ such that $h_k = h_{k+2}$ and $R_{k+1} = R_k^{-1}$.

Intuitively, this definition means that a path is non-redundant if it doesn't immediately traverses an edge back.

Lemma 4.35. *Let $\vec{M}$ be the unraveling of a general Kripke model M around some world $s \in S$. Let $h, h' \in \vec{S}$ be such that $h \neq h'$. Then there is exactly one non-redundant path p from h to h'.*

Lemma 4.36. *Any path p from h to h' contains the unique non-redundant path from h to h'.*

STEP 3: Completeness for GE-PDL$^-$

Now we'll return from the land of trees to the land of models for DECL, and with that show completeness of $\mathcal{L}_{GE\text{-}PDL^-}$ with respect to models for DECL.

Definition 4.37. Let $\vec{M} = \langle \vec{S}, R_{\sim_I}, R_{\approx_J}, \vec{V} \rangle_{I \subseteq \mathcal{A}, J \subseteq \mathcal{A} \cup \{0\}}$ be the unraveling of M^+ around some world $[s] \in S^+$. Then define $S = \langle \vec{S}, \sim_i, \approx_j, \vec{V} \rangle_{i \in \mathcal{A}, j \in \mathcal{A} \cup \{0\}}$ to be such that

$$\begin{aligned}
\sim_i &= \big(\bigcup\{R_{\sim_I}; i \in I \subseteq \mathcal{A}\} \cup \bigcup\{R_{\sim_I}^{-1}; i \in I \subseteq \mathcal{A}\}\big)^* \\
\approx_i &= \big(\bigcup\{R_{\approx_I}; i \in I \subset \mathcal{A} \cup \{0\}\} \cup \bigcup\{R_{\approx_I}^{-1}; I \in I \subset \mathcal{A} \cup \{0\}\} \cup \\
&\qquad \bigcup\{R_{\sim_I}; i \in I \subseteq \mathcal{A}\} \cup \bigcup\{R_{\sim_I}^{-1}; i \in I \subseteq \mathcal{A}\big)^*
\end{aligned}$$

Then, as we want this to be an actual model for DECL, we define

- $\approx_I := \bigcap_{i \in I} \approx_i$
- $\sim_I := \bigcap_{i \in I} \sim_i$
- $\sim_{\pi_1;\pi_2} := \sim_{\pi_1}; \sim_{\pi_2}$
- $\sim_{\pi_1+\pi_2} := \sim_{\pi_1} \cup \sim_{\pi_2}$
- $\sim_{\pi^*} := (\sim_\pi)^*$
- $h \sim_{?\theta} h'$ if and only if $h = h'$ and $last(h) \vDash_{M^+} \theta$

Proposition 4.38. *For all $I \subseteq \mathcal{A}, h \sim_I h'$ if and only if the unique non-redundant path from h to $h', p = (h = h_0 S_0 h_1 \dots S_{n-1} h_n = h')$ is an $\mathcal{R}$-path, with $\mathcal{R} = \{R_{\sim_J}, R^{-1}_{\sim_J}; I \subseteq J \subseteq \mathcal{A}\}$.*

Proof. Suppose $h \sim_I h'$. Then by definition $h \sim_i h'$ for all $i \in I$. Hence, for all $i \in I$ there is an $\mathcal{R}'$-path p' such that $\mathcal{R}' = \{R_{\sim_J}, R^{-1}_{\sim_J}; i \in J \subseteq \mathcal{A}\}$. But then by Proposition 4.36 it must be the case that the unique non-redundant path between h and h' is contained in p'. But then it must be the case that the non-redundant path between h and h' is an $\mathcal{R}^*$-path with $\mathcal{R}^* = \{R_{\sim_J}, R^{-1}_{\sim_J}; I \subseteq J \subseteq \mathcal{A}\}$. □

Proposition 4.39. *For all $I \subset \mathcal{A} \cup \{0\}$ and $h, h' \in \vec{W}, h \approx_I h'$ if and only if the unique non-redundant path from h to $h', p = (h = h_0 S_0 h_1 \dots S_{n-1} h_n = h')$ is an $\mathcal{R}$-path with $\mathcal{R} = \{R_{\approx_J}, R^{-1}_{\approx_J}, R_{\sim_K}, R^{-1}_{\sim_K}; I \subseteq J \subset \mathcal{A} \cup \{0\}, I \subseteq K \subseteq \mathcal{A}\}$.*

Proof. Suppose $h \approx_I h'$. Then by definition $h \approx_i h'$ for all $i \in I$. Hence, for all $i \in I$ there is an $\mathcal{R}'$-path p' such that $\mathcal{R}' = \{R_{\approx_J}, R^{-1}_{\approx_J}, R_{\sim_K}, R^{-1}_{\sim_K}; i \in J \subset \mathcal{A} \cup \{0\}, i \in K \subseteq \mathcal{A}\}$. But then by Proposition 4.36 it must be the case that the unique non-redundant path between h and h' is contained in p'. But then it must be the case that the non-redundant path between h and h' is an $\mathcal{R}^*$-path with $\mathcal{R}^* = \{R_{\approx_J}, R^{-1}_{\approx_J}, R_{\sim_K}, R^{-1}_{\sim_K}; J \subset \mathcal{A} \cup \{0\}, I \subseteq K \subseteq \mathcal{A}\}$. □

Proposition 4.40. *Let $f : S \to M^+$ be such that $f(h) = last(h)$. Then f is a bounded morphism.*

Proof. The proof is per condition of bounded morphisms.

1. We have to show that for any $h \in \vec{S}$ that $h \in \vec{V}(p)$ iff $f(h) \in V^C(p)$.
   Take an arbitrary $h \in \vec{S}$. then $h \in \vec{V}(p)$ if and only if $last(h) \in V^C(p)$ since we defined $\vec{V}(p) := \{h \in \vec{S}; last(h) \in V^C(p)\}$. Hence, $h \in \vec{V}(p)$ if and only if $f(h) \in V^C(p)$.

2. (a) We have to show that $h \sim_\pi h'$ implies $f(h) \sim_\pi f(h')$. We will show this by induction on the complexity of programs.
   Base case Take arbitrary $I \subseteq \mathcal{A}$ and $h, h' \in \vec{S}$ such that $h \sim_I h'$. This means that the non-redundant path p^* is an $\mathcal{R}$-path such that $\mathcal{R} = \{R_{\sim_J}, R^{-1}_{\sim_J}; I \subseteq J \subseteq \mathcal{A}\}$. Thus $p^* = (h = h_0 R_0 h_1 \dots R_{n-1} h_n = h')$ where for all $k < n, R_k \in \mathcal{R}$. Hence by definition of the $R_{\sim_J}$'s, we get that for all $k < n$, $last(h_k) \sim_I last(h_{k+1})$. but then as $\sim_I$ is transitive, we get $last(h) \sim_I last(h')$, and hence $f(h) \sim_I f(h')$.
   Inductive step

   - Take arbitrary $h, h' \in \vec{S}$ such that $h \sim_{\pi_1;\pi_2} h'$. Then by construction $h \sim_{\pi_1}; \sim_{\pi_2} h'$, and thus there is an $l \in \vec{S}$ such that $h \sim_{\pi_1} l \sim_{\pi_2} h'$. Then by the induction hypothesis, we get $f(h) \sim^+_{\pi_1} f(l) \sim^+_{\pi_2} f(h')$, and hence $f(h) \sim^+_{\pi_1}; \sim^+_{\pi_2} f(h')$, and thus by construction $f(h) \sim^+_{\pi_1;\pi_2} f(h')$.

- Take arbitrary $h, h' \in \vec{S}$ such that $h \sim_{\pi_1+\pi_2} h'$. Then by construction $h \sim_{\pi_1} \cup \sim_{\pi_2} h'$. Thus we have that $h \sim_{\pi_1} h'$ or $h \sim_{\pi_2} h'$. Then by the induction hypothesis we have $f(h) \sim^+_{\pi_1} f(h')$ or $f(h) \sim^+_{\pi_2} (h')$. This gives us $f(h) \sim^+_{\pi_1} \cup \sim^+_{\pi_2} (h')$, and thus by construction $f(h) \sim^+_{\pi_1+\pi_2} (h')$.
- Take arbitrary $h, h' \in \vec{S}$ such that $h \sim_{\pi^*} h'$. By construction this means that $h(\sim_\pi)^* h'$, and thus that there is a sequence $l_1, \ldots, l_n$ such that $h \sim_\pi l_1 \sim_\pi \cdots \sim_\pi l_n \sim_\pi h'$. Then by the induction hypothesis, this means that $f(h) \sim^+_\pi f(l_1) \sim^+_\pi \cdots \sim^+_\pi f(l_n) \sim^+_\pi f(h')$, and thus we get $f(h)(\sim^+_\pi)^* f(h')$, which by construction means that $f(h) \sim^+_{\pi^*} f(h')$.
- Take arbitrary $h, h' \in \vec{S}$ such that $h \sim_{?\theta} h'$. Then by construction $h = h'$ and $last(h) \vDash_{M^+} \theta$. Since $h = h'$, also $f(h) = f(h')$, and as $f(h) = last(h)$, we have that $f(h) \vDash \theta$, and thus $f(h) \sim_{?\theta} f(h')$.

(b) We have to show that $h \approx_\pi h'$ implies $f(h) \approx_\pi f(h')$Take arbitrary $I \subseteq \mathcal{A} \cup \{0\}$ and $h, h' \in \vec{S}$ such that $h \approx_I h'$.
Suppose $I \subset \mathcal{A} \cup \{0\}$. Then this means that the non-redundant path p is an $\mathcal{R}$-path such that $\mathcal{R} = \{R_{\approx_J}, R^{-1}_{\approx_J}, R_{\sim_K}, R^{-1}_{\sim_K}; I \subseteq J \subset \mathcal{A} \cup \{0\}, I \subseteq K \subseteq \mathcal{A}\}$. Hence, $p = (h = h_0 R_0 h_1 \ldots R_{n-1} h_n = h')$ where for all $k < n, R_k \in \mathcal{R}$. Thus for all $k < n, last(h_k) \sim_I last(h_{k+1})$ or $last(h_k) \approx_I last(h_{k+1})$. Note that as $\sim_i \subseteq \approx_i$ for all $i \in \mathcal{A}$, we get that $last(h_k) \sim_I last(h_{k+1})$ implies $last(h_k) \approx_I last(h_{k+1})$, and thus we get that for all $k < n, last(h_k) \approx_I last(h_{k+1})$. But then as $\approx_I$ is transitive, we get $last(h) \approx_I last(h')$, and hence $f(h) \approx_I f(h')$.
Now suppose $I = \mathcal{A} \cup \{0\}$. Then $h \approx_{\mathcal{A}\cup\{0\}} h'$ means that $h = h'$, and hence $last(h) = last(h')$, and thus $last(h) \approx_{\mathcal{A}\cup\{0\}} last(h')$, and thus $f(h) \approx_{\mathcal{A}\cup\{0\}} f(h')$.

3. (a) We have to show that $f(h) \sim_\pi t'$ implies that there is a $h' \in \vec{S}$ such that $f(h') = t'$ and $h \sim_I h'$. We will show this by induction on the complexity of programs.
Base case Take arbitrary $I \subseteq \mathcal{A}, h \in \vec{S}$ and $t \in S^+$ such that $f(h) \sim^+_I t$. This means that $last(h) \sim^+_I t$. We have to show that there is a $h' \in \vec{S}$ such that $h \sim_I h'$ and $f(h') = t$.
Let $h' = h + (last(h), \sim_I, t)$. Then $h \sim_I h'$ by definition and $last(h') = t$, and thus $f(h') = t$.
Inductive step

- Take arbitrary $h \in \vec{S}$ and $t' \in S^+$ such that $f(h) \sim^+_{\pi_1;\pi_2} t'$. Then by definition $f(h) \sim^+_{\pi_1}; \sim^+_{\pi_2} t'$. Hence there exists a $u' \in S^+$ such that $f(h) \sim^+_{\pi_1} u' \sim^+_{\pi_2} t'$. Then by two applications of the induction hypothesis there is a $u \in \vec{S}$ such that $h \sim_{\pi_1} u$ and $f(u) = u'$ and there is a $t \in \vec{S}$ such that $u \sim_{\pi_2} t$ and $f(t) = t'$. Thus $h \sim_{\pi_1} u \sim_{\pi_2} t$,

and thus $h \sim_{\pi_1}; \sim_{\pi_2} t$, and thus $h \sim_{\pi_1;\pi_2} t$.

- Take arbitrary $h \in \vec{S}$ and $t' \in S^+$ such that $f(h) \sim^+_{\pi_1+\pi_2} t'$. Then by definition $f(h) \sim^+_{\pi_1} \cup \sim^+_{\pi_2} t'$, and thus $f(h) \sim^+_{\pi_1} t'$ or $f(h) \sim^+_{\pi_2} t'$. Then by the induction hypothesis there is a $t \in \vec{S}$ such that $h \sim_{\pi_1} t$ and $f(t) = t'$ or there is a $t \in \vec{S}$ such that $h \sim_{\pi_2} t$ and $f(t) = t'$. Hence, there is a $t \in \vec{S}$ such that $h \sim_{\pi_1} t$ or $h \sim_{\pi_2} t$ and $f(t) = t'$. Thus $h \sim_{\pi_1} \cup \sim_{\pi_2} t$, and thus $h \sim_{\pi_1+\pi_2} t$.
- Take arbitrary $h \in \vec{S}$ and $t' \in S^+$ such that $f(h) \sim^+_{\pi^*} t'$. Then by definition $f(h)(\sim^+_\pi)^* t'$, and thus there are $u'_1, \ldots, u'_n \in S^+$ such that $f(h) \sim^+_\pi u'_1 \sim^+_\pi \cdots \sim^+_\pi u'_n \sim^+_\pi t'$. But then by the induction hypothesis, there are $u_1, \ldots, u_n, t \in \vec{S}$ such that $h \sim_\pi u_1 \sim_\pi \cdots \sim_\pi u_n \sim_\pi t$ such that for all $i \leq n : f(u_i) = u'_i$ and $f(t) = t'$. Hence there is a t such that $h(\sim_\pi)^* t$ and $f(t) = t'$, and thus there is a t such that $h \sim_{\pi^*} t$ and $f(t) = t'$.
- Take arbitrary $h \in \vec{S}$ and $t' \in S^+$ such that $f(h) \sim^+_{?\theta} t'$. Then by definition $f(h) = t'$ and $f(h) \vDash \theta$. Let $t = h$. Then it is immediate that $h = t$ and $last(h) = f(h) \vDash_{M^+} \theta$, and thus that $h \sim_\theta t$. Also $f(t) = f(h) = t'$, so there is a t such that $h \sim_{?\theta} t$ and $f(t) = t'$.

(b) The $\approx$ case is analogous to the base case of the $\sim$ case.

□

Proposition 4.41. *S is a model for DECL-C.*

Proof. All relations in S are equivalence relations by definition. It is also immediate from the definition that $\sim_i \subseteq \approx_i$ for all $i \in \mathcal{A}$, and that $\sim_{\pi_1;\pi_2} = \sim_{\pi_1}; \sim_{\pi_2}$ and $\sim_{\pi_1+\pi_2} = \sim_{\pi_1} \cup \sim_{\pi_2}$ and $\sim_{\pi^*} = (\sim_\pi)^*$ follows immediately from the construction. It remains to show that $\bigcap_{i \in \mathcal{A} \cup \{0\}} [s]_{\approx_I} = \{s\}$ for all $s \in S$ and that $h \sim_{?\theta} h'$ if and only if $h = h'$ and $h \vDash \theta$.

- $\bigcap_{i \in \mathcal{A} \cup \{0\}} [s]_{\approx_I} = \{s\}$ for all $s \in S$: We have to show that $h \approx_{\mathcal{A} \cup \{0\}} h'$ implies $h = h'$.
  So suppose $h \approx_{\mathcal{A} \cup \{0\}} h'$. Then by definition, $h \approx_i h'$ for all $i \in \mathcal{A} \cup \{0\}$. But for all $i \in \mathcal{A} \cup \{0\}$, we have that
  $\approx_i = (\bigcup\{R_{\approx_I}; i \in I \subset \mathcal{A} \cup \{0\}\} \cup \bigcup\{R^{-1}_{\approx_I}; I \in I \subset \mathcal{A} \cup \{0\}\} \cup \bigcup\{R_{\sim_I}; i \in I \subseteq \mathcal{A}\} \cup \bigcup\{R^{-1}_{\sim_I}; i \in I \subseteq \mathcal{A}\})^*$.
  However, note that this means that all seperate parts are empty, and hence $\bigcap_{i \in \mathcal{A} \cup \{0\}} \approx_i = \mathrm{id}$. Thus $h \approx_{\mathcal{A} \cup \{0\}} h'$ implies $h = h'$.

- $h \sim_{?\theta} h'$ iff $h = h'$ and $h \vDash \theta$. We have that $h \sim_{?\theta} h'$ iff $h = h'$ and $M^+, last(h) \vDash \theta$. Since $f : h \mapsto last(h)$ is a Bounded Morphism, we know that for all σ, we have that $M, h \vDash \sigma$ if and only if $M^+, f(h) \vDash \sigma$. This means that $h = h'$ and $M^+, last(h) \vDash \theta$ if and only if $h = h'$ and $M, h \vDash \theta$. Thus $h \sim_{?\theta} h'$ if and only if $h = h'$ and $M, h \vDash \theta$.

□

Theorem 4.42 (Completeness for GE-PDL⁻)**.** $\mathcal{L}_{GE\text{-}PDL^-}$ *is weakly complete with respect to DECL-C models.*

Proof. Let φ be an GE-PDL⁻ -consistent formula. By Lindenbaum's Lemma, $\{\varphi\}$ can be extended to a maximal consistent set Φ. By definition, $\Phi \in S^C$, where S^C is the set of states in the canonical pseudo-model. Let M^+ be the filtration of M^C over Σ_φ.
Now let $\vec{M}$ be the unraveling of M^+ around $[s]$ such that $\Phi \in [s]$, and let S be the generated DECL-C model. Note that the history $([s]) \in \vec{V}$. Define $f : S \to M^C$ to be such that $f(h) = last(h)$. By Lemma 4.40, this is a bounded morphism. By Lemma 4.14, we have that $S, ([s]) \vDash \varphi$ if and only if $M^+, [s] \vDash \varphi$. But as $\varphi \in \Phi$ and by Lemma 4.26, we have that $M^+, [s] \vDash \varphi$, and hence $S, ([s]) \vDash \varphi$. □

Step 4: Completeness for GE-PDL

We showed that the static language GE-PDL⁻ is complete. We will now show that the dynamic language can be reduced to the static language, thereby showing completeness for the dynamic language as well.

Lemma 4.43. *The reduction axioms are valid on models for DECL-C.*

Proof. Showing that the reduction axioms for $[\sigma]p, [\sigma]\neg\varphi, [\sigma](\varphi \wedge \psi)$ and $[\sigma][I]\varphi$ are valid follows the exact same lines as it did in the proof of completeness for DECL. The proof of the reduction axiom for $[\sigma]K_\pi\varphi$ is analogous to the same proof in [30, Thm. 48]. □

The reduction axioms we defined before give rise to a translation function from formulas from GE-PDL to formulas from GE-PDL⁻, thereby showing that the former can be reduced to the latter.

Definition 4.44 (Translation)**.** The function t takes a formula of GE-PDL and yields a formula of GE-PDL⁻.

$$
\begin{array}{ll}
t(\top) & = \top \\
t(p) & = p \\
t(\neg\varphi) & = \neg t(\varphi) \\
t(\varphi_1 \wedge \varphi_2) & = t(\varphi_1) \wedge t(\varphi_2) \\
t(K_\pi\varphi) & = K_{r(\pi)} t(\varphi) \\
t([I]\varphi) & = [I] t(\varphi) \\
t([\sigma]\top) & = \top \\
t([\sigma]p) & = t(pre(\sigma)) \to t(post(\sigma)(p)) \\
t([\sigma]\neg\varphi) & = t(pre(\sigma)) \to \neg t([\sigma]\varphi) \\
t([\sigma](\varphi_1 \wedge \varphi_2)) & = t([\sigma]\varphi_1) \wedge t([\sigma]\varphi_2) \\
t([\sigma_n]K_\pi\varphi) & = \bigwedge_{m=0}^{N-1} [T_{nm}(r(\pi))] t([\sigma_m]\varphi) \\
t([\sigma][I]\varphi) & = t(pre(\sigma)) \to \bigwedge_{\sigma' \approx_I \sigma} [I] t([\sigma']\varphi) \\
t([\sigma][\sigma']\varphi) & = t([\sigma] t([\sigma']\varphi))
\end{array}
$$

Where the function r is defined as follows:

$$\begin{array}{ll} r(I) & = I \\ r(?\theta) & = ?t(\theta) \\ r(\pi_1;\pi_2) & = r(\pi_1);r(\pi_2) \\ r(\pi_1+\pi_2) & = r(\pi_1)\cup r(\pi_2) \\ r(\pi^*) & = (r(\pi))* \end{array}$$

Theorem 4.45 (Completeness of GE-PDL)**.** *For any φ in the language $\mathcal{L}_{GE-PDL}$ we have that*

$$\vDash \varphi \textit{ iff } \vdash \varphi$$

Proof. By Theorem 4.42, the static language GE-PDL$^-$ is complete, and with the translation procedure above we showed that any formula from GE-PDL is equivalent to a formula of GE-PDL$^-$. □

4.4 Planning with DECL-C

With the extended language that includes common knowledge, we have the possibility to model more of the planning process, and exclude the idea that the agents have some external way of communicating. The main advantage of common knowledge is that if a joint action is commonly known to be the one that the agents are performing, everyone can rely on everyone to perform their part of that action. Hence, we can define what it means for an action to be a solution, without assuming that this condition is only a prerequisite for some agent telling the others that this joint action is the one that we are performing, as it is in planning with DECL.

In Example 4.3 we touched upon a possible criterion for an action being a good choice: it was common knowledge between the agents that it would achieve the goal. However, in Example 4.4 we saw that this might not be enough. If there are multiple joint actions the agents can perform such that they achieve the goal, the agents still do not know which one to choose. Thus we say that a joint action σ_I is a *strong salient solution* if it is common knowledge between the agents in I that σ_I will achieve the goal, and that there is no different action that has the same property.

It being common knowledge that σ_I is the only action such that it will achieve the goal is a very strict condition on it being a solution, and it will often be the case that there does not exists such a joint action. There are multiple other, less strong, constraints one can define for solutions. We call the one that we will highlight here a *weak salient solution*, and we define this as the single joint action σ_I such that it is common knowledge that it *might* reach the goal whereas all the other joint actions will definitely *not* reach the goal. One can see this as the 'last resort' - if

we cannot be sure to reach the goal, we should at least perform the action that will not exclude it. Of course, there are many other levels of solutions one can define, but for the purpose of this thesis we will stick to these two.

Definition 4.46 (Single-step Solution)**.** For a single-step planning problem $\mathbb{P} = \langle (M, s), \Sigma, \mathcal{A}, \varphi_g \rangle$ and joint action σ_I such that for all $\sigma, \sigma' \in \sigma_I$ we have that $\sigma \sim_I \sigma'$, we say that σ_I is a :
strong salient solution if $M, s \vDash C_I([\sigma_I]\varphi_g \wedge \bigwedge_{\sigma' \notin \sigma_I} \neg[\sigma'_I]\varphi_g)$
weak salient solution if $M, s \vDash C_I(\neg[\sigma_I]\neg\varphi_g \wedge \bigwedge_{\sigma' \notin \sigma_I} [\sigma'_I]\neg\varphi_g)$
for $I \subseteq \mathcal{A}$

Again, extending to k-step solutions and searching the tree for a solution is trivial. Also finding a solution follows the same steps as it does for planning without common knowledge.

Both strong and weak salient solutions depend on there being exactly one such solution, but this is a situation that will not occur very often. It is therefore not very likely that agents will find a solution with these constraints, but we need them to ensure that everyone is aware of the one plan the agents will use to reach their goal. In the next chapter we extend the logic further in order to make it easier for the agents to reach a situation that makes these constraints true.

5 Committing to actions

In the previous section, we introduced a logic for coalitions in epistemic planning, and we defined a planning problem and two solution concepts. These concepts depended on there being exactly one salient solution. However, it might not always be the case that there is such a joint action of which it is common knowledge that it will reach the goal, or that it is the only one that satisfies that constraint. Especially if the latter is the case, the agents will need some way to talk about which of the possible joint actions they will take, and hence, agents have to be able to talk about what they will do themselves. Hence, there needs to be some form of coordination. For this purpose we introduce *commitments*.

In this chapter we will extend the logic to allow for commitments, we argue that also the extended logic is sound, complete and decidable. We then go on to explain how committing to actions work and give some examples. We show how to use commitments to express the concept of *responsibility*, and finally we will give an idea of how committing can be used for planning.

Intuitively, committing in the sense of this thesis is very comparable to committing as humans do it. There is number of actions available, and we communicate to other agents (or humans) that we will restrict ourselves to only a subset of those actions. The result of an agent committing is that other agents will also be limited in their actions. By iteratively committing, agents might be able to coordinate which joint action they will perform.

To facilitate this committing to a subset of actions, we introduce a new type of atomic sentences: $commit_i(\Gamma)$, saying that agent i is committed to do an action in Γ, where $\Gamma \subseteq \Sigma$. We extend the valuation of the static model to also range over these atomic sentences, thereby making the commitment of an agent state-dependent. We write $\mathbb{C}$ for the set of all atomic sentences of the form $commit_i(\Gamma)$ for some $i \in \mathcal{A}$ and $\Gamma \subseteq \Sigma$.

The models we use for this extension of the language are similar to the models we used previously, but we have to augment both the static and the action models with a structure that refers to commitments. For the static models, this means that the valuation is now a function from both propositional letters and the new atomic sentences to the power set of the state space. Thus:

Definition 5.1 (Augmented Static Model)**.** A *static epistemic model* is a structure $M = \langle S, \sim_i, \approx_j, V \rangle_{i \in \mathcal{A}, j \in \mathcal{A} \cup \{0\}}$ such that

- S is a non-empty set of *states*;
- $\sim_i$ is an equivalence relation for each agent $i \in \mathcal{A}$ called the *indistinguishability relation*;

- $\approx_j$ is an equivalence relation for every $j \in \mathcal{A} \cup \{0\}$ such that for all $s \in S$ we have $\bigcap_{j \in \mathcal{A} \cup \{0\}} [s]_{\approx_j} = \{s\}$;
- $V : P \cup \mathbb{C} \to \mathcal{P}(S)$ is the *valuation function* that maps propositional letters and *commit* sentences to subsets of states.
- for all $i \in \mathcal{A}$ we require $\sim_i \subseteq \approx_i$

5.1 DECL with Common Knowledge and Commitments

We adjust the language and satisfaction to enable commitments from agents by incorporating the new atomic sentences.

5.1.1 Syntax and Semantics of DECL-CC

Definition 5.2 (The language $\mathcal{L}_{DECL\text{-}CC}$)**.** The language $\mathcal{L}_{DECL\text{-}CC}$ is defined by the following Backus-Naur form:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid D_I\varphi \mid C_I\varphi \mid [J]\varphi \mid [\sigma]\varphi \mid commit_i(\Gamma) \mid [!\Gamma_i]\varphi$$

where $I \subseteq \mathcal{A}, J \subseteq \mathcal{A} \cup \{0\}$ and $i \in \mathcal{A}$ and $\Gamma \subseteq \Sigma$ for Σ a given action model.

Here, $D_I\varphi, C_I\varphi, [I]\varphi$ and $[\sigma]\varphi$ have the same meaning as before. $commit_i(\Gamma)$ is the atomic sentence meaning that agent i is committed to the set of actions Γ. We use $!\Gamma_i$ as a notation for the action where agent i commits to Γ, and thus $[!\Gamma_i]\varphi$ says that φ holds after the event of agent i committing to Γ.

Abbreviations

- We will write $commit_I(\Gamma)$ for $C_I(\bigwedge_{i \in I} commit_i(\Gamma_i))$ such that $(\bigcap_{i \in I} \Gamma_i) = \Gamma$. This assumes that a group of agents can only commit to a set of actions if it is common knowledge within that group that everyone committed to his or her part of that set of actions.

5.1.2 Committing Actions

As said before, we want agents to iteratively commit to actions. Hence, we need a way in which they can do this. In fact, by introducing these new atomic sentences, this becomes very simple. We just add events to the action model that might have some preconditions, depending on the requirements that we will set to committing to actions, and that have a postcondition that changes the truth value of the extra atomic sentences.
In our framework, agents can commit to multiple subsets of actions. If Alice first commits to paying her taxes, and then to have dinner with Bob, she basically commits to do both. Hence, we define

$$Commit_i^s := \bigcap \{\Gamma; S, s \vDash commit_i(\Gamma)\}$$

Hence, $Commit_i^s$ is the set of actions that is compatible with all the commitments agent i made at state s. Note that when i did not make a commitment yet, $Commit_i^s = \Sigma$. This is an intuitive consequence, because if an agent has not committed to doing a specific action, she can still do anything she wants.

We can now talk about the commitments of agents and groups of agents, but this is only interesting if agents can actually commit to something. For this we add *committing* actions to the action model. As we assume throughout this thesis that agents have the same goal and are willing to cooperate to reach this, we make these committing actions analogous to public announcements: after committing to a set of actions, this becomes common knowledge within the entire set of agents. We therefore denote the action of agent i committing to Γ as $!\Gamma_i$. Just as for public announcement, we assume the conventions that all actions of the form $!\Gamma_i$, such that Γ is a subset of the given Σ and i an agent, are well formed actions, although it is not necessarily the case that all of them are available.

All committing actions have an analogous postcondition, which ensures that the valuation function gets updated appropriately:

$$post(!\Gamma_i)(commit_i(\Gamma)) = \top$$

Note that we do not change the truth value of previous commitments. Together with the fact that $Commit_i^s$ is the intersection of all commitments that are valid at state s, defining the postcondition in this way ensures that an agent can only *narrow down* his commitments, as the intersection will at most get smaller. This is a choice we made in this framework, as we wanted to model agents that are cooperating and trying to reach a salient action to reach their goal. In other frameworks it might be that one wants to make a different decision about this.

How committing actions work is shown in the following example.

Example 5.3. Alice and Bob want to spend the evening together, but they don't know where to go. Both of them can either go to the boxing or the dancing, but their goal is to go together. They have just raised the issue of where to go, so no one has committed to anything. The model is shown in Figure 22.
The action model is shown in Figure 23. Note that Alice cannot decide for Bob where he is going to, and vice versa. Since $Commit_a = Commit_b = \Sigma$ in the initial model, neither of them knows what the other one will do (they might not even know it of themselves!).
It is clear that without any communication, Alice and Bob can only guess at what the other person will do, and hence have only a limited chance at reaching their goal. However, if Alice commits to either one of her possibilities, Bob knows what she will do. So we add the action of Alice committing to going dancing. It is shown in Figure 24, and it is a single event, like public announcement, meaning that after it happened, it is common knowledge that Alice committed.

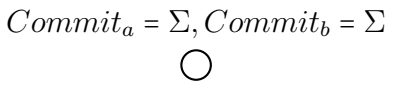

Figure 22: The initial model

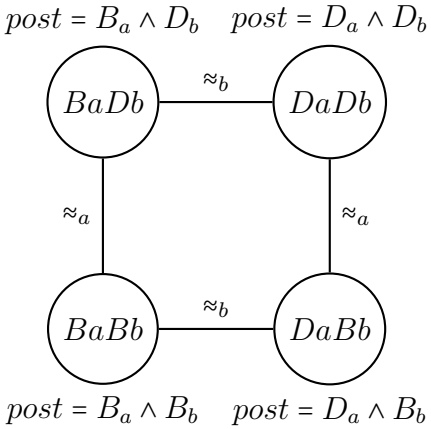

Figure 23: The action model

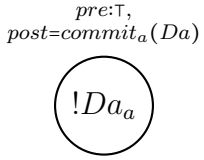

Figure 24: The action of Alice committing to going dancing

As Alice hadn't committed to anything before, and thus it was the case that $Commit_a = \Sigma$, the result of committing in the initial state is as follows:

Figure 25: The static model after committing

Hence, after Alice publicly committed to go dancing, Bob knows she did so, and that, if he wants to spend time with her, he should go dancing as well.

In this example, we argued that as Alice publicly committed to dancing, Bob knows that, in order to spend time with Alice, he should go dancing. However, there are currently no mechanisms in the framework that assure that Alice will actually *stick* to her commitments. There are multiple ways of ensuring this, some more strict than others. As this thesis deals with agents that want to cooperate, and are therefore not interested in lying to one another, we decided on the following construction:

Definition 5.4 (Augmented action model)**.** Let Σ be the given action model. We then define the *augmented action model* as $\mathbf{\Sigma}^+ = \langle \Sigma^+, \sim_i, \approx_i, pre, post \rangle$ where

- $\Sigma^+ = \Sigma \cup \{!\Gamma_i; \Gamma \subseteq \Sigma \text{ s.t. } \Gamma \text{ is closed under } \approx_i, i \in \mathcal{A}\}$

- $\sim_i^{\Sigma^+} = \sim_i^{\Sigma} \cup \{(!\Gamma_j, !\Gamma_j); j \in \mathcal{A} \cup \{0\}, \Gamma \subseteq \Sigma \text{ closed under } \approx_j\}$
- $\approx_i^{\Sigma^+} = \approx_i^{\Sigma} \cup \{(!\Gamma_i, !\Gamma_i); \Gamma \subseteq \Sigma \text{ closed under } \approx_i\} \cup \{(!\Gamma_j, !\Gamma'_j); j \in \mathcal{A}, j \neq i, \Gamma, \Gamma' \subseteq \Sigma \text{ closed under } \approx_j\}$
- $pre_{\Sigma^+}(\sigma) = \begin{cases} pre_\Sigma(\sigma) \wedge \bigwedge\limits_{\substack{\Gamma \subseteq \Sigma \\ \sigma \notin \Gamma \\ i \in \mathcal{A} \cup \{0\}}} \neg commit_i(\Gamma) & \text{iff } \sigma \in \Sigma \\ \top & \text{otherwise} \end{cases}$
- $post_{\Sigma^+}(\sigma) = post_\Sigma(\sigma)$
  $post_{\Sigma^+}(!\Gamma_i)(commit_i(\Gamma)) = \top$
  $post_{\Sigma^+}(!\Gamma_i)(q) = q$ for every atomic sentence $q \neq commit_i(\Gamma)$

Here, $\sim_i^{\Sigma}$ and $\approx_i^{\Sigma}, pre_\Sigma(\sigma)$ and $post_\Sigma(\sigma)$ are the epistemic relations, control relations, preconditions and postconditions in the original action model Σ.

Intuitively, we add possible committing actions to the action state space, and we put an extra constraint on the previously existing actions in the form of an extended precondition. This extra constraint says that none of the agents can currently be committed to a set of actions that excludes σ. If this were the case, this agent would be violating at least one of his commitments, which we do not wish to allow in our framework.

In the augmented action model, we require that the committing actions available to a certain agent are closed under his control relation. This means that an agent can only commit to an action, or a set of actions, he can decide to do. Consider Example 5.3, where Alice commits to going dancing. She cannot reasonably commit to both of them going dancing, as she cannot control what Bob chooses to do. Hence, the action of her committing to exactly the event where they both go dancing is not available to her. Instead, we require that she, and every other agent, commits to an action or a set of actions she can in fact choose to perform.

Furthermore, we leave the existing relations from the old action model intact, and add indistinguishability relations such that all committing actions are always public and control relations such that (a) every agent has full control over her own committing actions, and (b) every agent can force another agent to commit to *something*, without being able to force that agent to commit to a specific set of events.

Example 5.5 (Alice and Bob going dancing, continued)**.** The augmented action model in the example of Alice and Bob going dancing or to the boxing is depicted in Figure 26. As one can see, the old actions now have a precondition, and there are committing actions for both Alice and Bob for either of their possibilities.

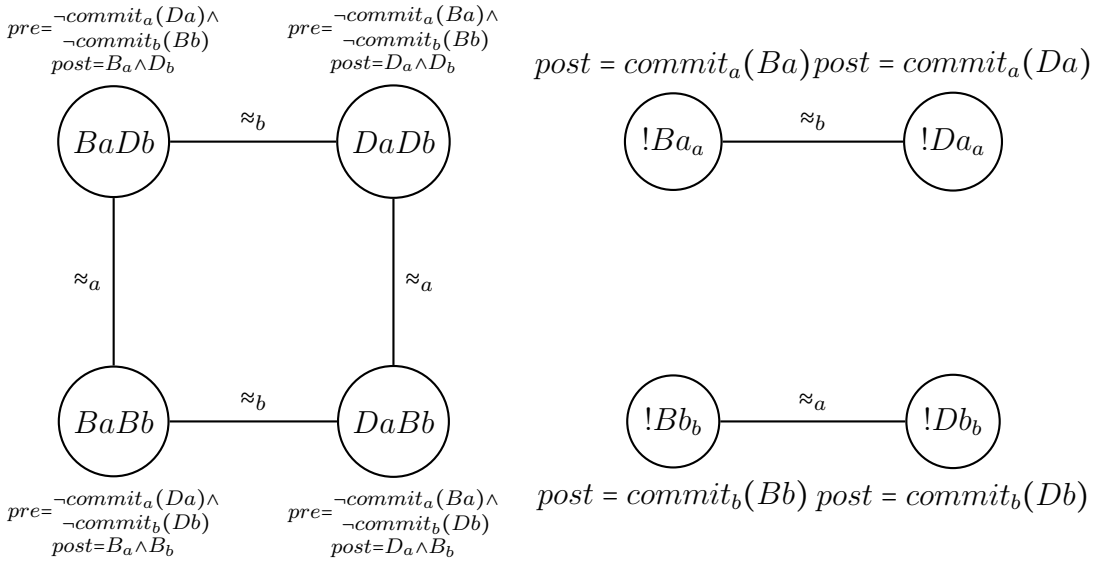

Figure 26: The action model

We stipulated that Alice committed to go dancing. Hence, the event $!Da_a$ took place, and the resulting model is depicted in Figure 25. Hence, everyone knows that Alice is committed to going dancing, and hence that the events where she goes to see the boxing cannot happen anymore, ensuring that Bob can make an informed decision about what to do.

The attentive reader might have noticed that it is possible for an agent to commit in such a way that all events are excluded: if Alice had previously committed to seeing the boxing with Charlie, her committing to dancing ensured that $Commit_a = \varnothing$, as the action where she goes to the boxing does not overlap with the action of her going dancing. In case this happens, and an agent commits in such a way that there is no action left possible, as all states will falsify the second part of the precondition of *every* action, we assume the responsible agent can, from then on, no longer perform actions. In some way, she will then have lost her right to participate in the planning. Her commitments are no longer part of the preconditions of events and she will not be able to make choices. We believe that this is a reasonable assumption, as in properly stated situations an agent will never be forced to commit in such a way, and as they all want to cooperate to reach a common goal, they will not deliberately exclude themselves from participating.

5.1.3 Semantics of DECL-CC

Definition 5.6 (Satisfaction)**.** Satisfaction of the new atomic sentences and the added committing actions on augmented static models is as follows:

- $M, s \vDash [\sigma]\varphi$ iff $M \otimes \Sigma^+, (s, \sigma) \vDash \varphi$

- $M, s \vDash commit_i(\Gamma)$ if and only if $s \in V(commit_i(\Gamma))$
- $M, s \vDash [!\Gamma_i]\varphi$ iff $M \otimes \Sigma^+, (s, !\Gamma_i) \vDash \varphi$

Where $i \in \mathcal{A}$ and $\Gamma \subseteq \Sigma$.

5.2 Group Epistemic PDL with Commitments

As also DECL-CC uses the common knowledge operator, we have to augment the group epistemic PDL we introduced in the previous chapter with commitments in order to obtain a completeness result. In this section we therefore present Group Epistemic PDL with commitments, GE-PDLc. It has the same base as GE-PDL, but as DECL-CC it has an additional type of atomic sentence and additional type actions.

Definition 5.7 (Language $\mathcal{L}_{GE-PDLc}$)**.** The language $\mathcal{L}_{GE-PDLc}$ is formed as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_\pi\varphi \mid [J]\varphi \mid [\sigma]\varphi \mid commit_i(\Gamma) \mid [!\Gamma_i]\varphi$$
$$\pi ::= I \mid \pi + \pi' \mid \pi;\pi' \mid \pi^* \mid ?\varphi$$

where $I \subseteq \mathcal{A}, i \in \mathcal{A} \cup \{0\}$ and $J \subseteq \mathcal{A} \cup \{0\}$ are coalitions, $[\sigma]$ is an event in Σ, and ρ is of the form Γ for some $\Gamma \subseteq \Sigma$.

Definition 5.8 (Satisfaction)**.** The *satisfaction* of these formulas on static epistemic models, denoted $M, s \vDash \varphi$ is defined as follows:

- $M, s \vDash p$ iff $s \in V(p)$
- $M, s \vDash \neg\varphi$ iff $M, s \nvDash \varphi$
- $M, s \vDash \varphi \vee \psi$ iff $M, s \vDash \varphi$ or $M, s \vDash \psi$
- $M, s \vDash K_\pi\varphi$ iff $\forall s' \sim_\pi s : M, s' \vDash \varphi$
- $M, s \vDash [I]\varphi$ iff $\forall s' \approx_I s : M, s' \vDash \varphi$
- $M, s \vDash [\sigma]\varphi$ iff $(s, \sigma) \in M \otimes \Sigma^+$ implies $M \otimes \Sigma^+, (s, \sigma) \vDash \varphi$
- $M, s \vDash commit_i(\Gamma)$ if and only if $s \in V(commit_i(\Gamma))$
- $M, s \vDash [!\Gamma]\varphi$ iff $M \otimes \Sigma^+, (s, !\Gamma) \vDash \varphi$

Where $\Gamma \subseteq \Sigma$ and $\sim_\pi$ is defined inductively as follows:

- $\sim_I := \bigcap_{i \in I} \sim_i$
- $\sim_{\pi_1;\pi_2} := \sim_{\pi_1} ; \sim_{\pi_2}$
- $\sim_{\pi_1+\pi_2} := \sim_{\pi_1} \cup \sim_{\pi_2}$
- $\sim_{\pi^*} := (\sim_\pi)^*$
- $s \sim_{?\theta} t$ iff $s = t$ and $M, s \vDash \theta$

5.2.1 Proof System of GE-PDLc

The axioms for GE-PDLc are the same as for GE-PDL in the previous chapter, but now they apply to the new atomic sentences and the committing events as well. Also, all preconditions, postconditions and indistinguishability relations come from the augmented action model Σ^+. For the reader's convenience, all axioms are listed again below.

Inference rules

- All axioms and rules of classical propositional logic
- Necessitation rules for all modalities

Axioms for DECL

- S5 for $[I]$ for all $I \subseteq \mathcal{A} \cup \{0\}$
- S5 for K_I for all basic programs I
- **Knowledge of (Individual) Control**
  $[I]\varphi \to K_I\varphi$ for $I \subseteq \mathcal{A}$
- **Monotonicity of control**
  $[I]\varphi \to [J]\varphi$ for $I \subseteq J \subseteq \mathcal{A} \cup \{0\}$
- **Monotonicity of distributed knowledge**
  $K_I\varphi \to K_J\varphi$ for $I \subseteq J \subseteq \mathcal{A}$
- **Determinism of grand coalition**
  $\varphi \to [\mathcal{A} \cup \{0\}]\varphi$

For $\varphi \in \mathcal{L}_{GE-PDLc}$

Axioms for programs

- $K_\pi(\varphi \to \psi) \to (K_\pi\varphi \to K_\pi\psi)$
- $K_{\pi_1;\pi_2}\varphi \leftrightarrow K_{\pi_1}K_{\pi_2}\varphi$
- $K_{\pi_1+\pi_2}\varphi \leftrightarrow K_{\pi_1}\varphi \wedge K_{\pi_2}\varphi$
- $K_{\pi^*}\varphi \leftrightarrow \varphi \wedge K_\pi K_{\pi^*}\varphi$
- $K_{\pi^*}(\varphi \to K_\pi\varphi) \to (\varphi \to K_{\pi^*}\varphi)$
- $K_{?\theta}\psi \leftrightarrow (\theta \to \psi)$

For $\varphi \in \mathcal{L}_{GE-PDLc}$

Reduction Axioms

Let N be the number of events in Σ, and suppose $\sigma_1, \ldots, \sigma_N$ is an enumeration of all events without repetitions, and let $\sigma = \sigma_n$ be any of these events (for some arbitrary $n \leq N$). Then we have the following reduction axioms.

- $[\sigma]p \leftrightarrow (\mathrm{pre}(\sigma) \to \mathrm{post}(\sigma)(p))$
- $[\sigma]\neg\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \neg[\sigma]\varphi)$
- $[\sigma](\varphi \wedge \psi) \leftrightarrow [\sigma]\varphi \wedge [\sigma]\psi$
- $[\sigma]K_\pi\varphi \leftrightarrow (\bigwedge_{m=0}^{N-1}[T_{nm}(\pi)][\sigma_m]\varphi)$
- $[\sigma][I]\varphi \leftrightarrow (\mathrm{pre}(\sigma) \to \bigwedge_{\sigma \approx_I \sigma'}[I][\sigma']\varphi)$

Where p is any atomic sentence in the language $\mathcal{L}_{GE\text{-}PDLc}, \sigma \in \Sigma^+$, the preconditions, postconditions, epistemic relations $\sim_i$ and control relations $\approx_i$ are taken from the augmented model Σ^+ and T_{nm} is a program transformer, which is defined as follows:

Definition 5.9 (T_{nm} program transformers)**.**

$$T_{nm}(I) = \begin{cases} ?pre(\sigma_n); I & \text{if } \sigma_n \sim_I \sigma_m \\ ?\bot & \text{otherwise} \end{cases}$$

$$T_{nm}(\pi_1;\pi_2) = \bigcup_{k=0}^{N-1}(T_{nk}(\pi_1);T_{km}(\pi_2))$$

$$T_{nm}(\pi_1 + \pi_2) = T_{nm}(\pi_1) \cup T_{nm}(\pi_2)$$

$$T_{nm}(?\theta) = \begin{cases} ?pre(\sigma_n) \wedge [\sigma_n]\theta & \text{if } n = m \\ ?\bot & \text{otherwise} \end{cases}$$

$$T_{nm}(\pi^*) = K_{nmN}(\pi)$$

Where $K_{nmN}(\pi)$ is given by Definition 5.10:

Definition 5.10. $K_{nmk}(\pi)$ is defined by recursion on k as follows:

$$K_{nm0}(\pi) = \begin{cases} ?(p \vee \neg p) \cup T_{nm}(\pi) & \text{if } n = m \\ T_{nm} & \text{otherwise} \end{cases}$$

$$K_{nm(k+1)}(\pi) = \begin{cases} (K_{kkk}(\pi))^* & \text{if } n = k = m \\ (K_{kkk}(\pi))^*; K_{kmk}(\pi) & \text{if } n = k \neq m \\ K_{nkk}(\pi); (K_{kkk}(\pi))^* & \text{if } n \neq k = m \\ K_{nmk}(\pi) \cup (K_{nkk}(\pi); (K_{kkk}(\pi))^*; K_{kmk}(\pi)) & \text{otherwise} \end{cases}$$

5.2.2 Soundness, Completeness and Decidability of GE-PDLc

Since the proof system is the same as in the previous chapter, with merely an extended language and different action models, the proof of the previous chapter carries over without any additional work.
Clearly, for the soundness, completeness and decidability of the static language, it is enough to ensure that the valuation of all types of models includes the new atomic sentences. As these are just atomic sentences, without any special properties, all valuations are easily extended.
To see that the result carries over to the dynamic language as well, note that GE-PDL does not have any constrictions on the type of actions within the action model. This means that the proof that GE-PDL can be reduced to GE-PDL$^-$ carries over immediately to GE-PDLc and its static component.
Hence, GE-PDLc is sound, complete and decidable with respect to augmented static models.

5.3 Responsibility

Before introducing commitments, we could express that agents had a way to deliberately force something. We could say that they commonly knew that if all of them chose to do their part of some action, this action would have a certain result. What we could not talk about was whether they were planning to perform this action or not. For example, suppose that Alice and Bob commonly know that Alice buying food and Bob cooking is the only way for them to have dinner together today. This knowledge does not, in any way, imply that they are planning to have dinner together. Similarly, even though the European leaders might commonly know that removing Greece from the Euro zone will lead to economic distress, this does not mean that they are planning to perform this action. To say exactly that, we need commitments. Together with deliberate forcing, commitments form *responsibility*.

$$resp_I(\varphi) ::= \bigvee_{\sigma \in \Sigma} C_I(\bigwedge_{i \in I} commit_i(\sigma_i) \wedge [\sigma_I]\varphi \wedge \bigvee_{\sigma' \notin \sigma_I} [\sigma'_I]\neg\varphi)$$

This formula says that a coalition I is *responsible* for φ if it is common knowledge between them that each of them is committed to do their part of an action of which they know that it results in φ, whereas they could also have chosen a different joint action that would have resulted in $\neg\varphi$.

We believe that this is a reasonable way to formalize responsibility, as it not only involves agents knowing the result of their joint action, but in addition, while there is another joint action that would have had a different result, the coalition is committed to doing the former. They are knowingly and deliberately committed to forcing φ.

5.4 Committing strategically

With DECL-CC we cannot only express common knowledge, which greatly improves the possibilities for planning, we can let agents commit to actions. By doing so, we limit the events that are possible in a certain state, as all events now have a precondition that they cannot be excluded by the current commitments of *any* agent. This allows for coordination between agents, since they can communicate what they are planning to do, which means that a coalition has more means to ensure they are all on the same track. Say, for example, that in the big action model there are two joint actions for which it is common knowledge that they will achieve the goal. Then if one of the agents commits in such a way that one of the joint actions is no longer an option, the other agents will know to choose the other one. This advantage is most interesting when looking at multi-step planning, as the agents can then include committing in the planning, but this is not a necessity, because the commitments can already be encoded in the initial model.

The planning problems and solutions for DECL-CC can be formulated in exactly the same way as those for DECL-C, but the added value of being able to commit to actions is that the agents now have more means to arrive at a situation where they can reach a solution. Thus, rather than redefining what makes a sequence of actions a solution, it is interesting to think about *strategic* commitments to make in order to make the planning run as smoothly as possible. A number of examples of ways in which agents can commit in a strategic way is listed below.

1. Commit to an action only if it is common knowledge that this specific action reaches the goal.
   This seems like a good thing to do if there are multiple joint actions for which it is common knowledge that they achieve the goal, since it will immediately make clear which one the agent wants to do. This will ensure that the agents will not perform actions that do not match up to reach the goal, but cooperate and perform a joint action that reaches the goal, just like Bob will know to go dancing after Alice announces that she is.
   If there is no joint action for which it is common knowledge that it will achieve the goal, this of course only prohibits the agents to come to some sort of understanding in another way. Hence, using this constraint as a conditional strategy is a reasonable idea: *if* there are multiple actions for which it is common knowledge that they achieve the goal, commit to one of them. Otherwise a more conservative committing strategy might be in order.

2. Exclude only one action at a time.
   This suggestion for strategic committing means that every agent is fairly conservative in committing, and always commits to a set of actions in such a way that they only not commit to one of them. This would prevent that

one agent reduces the options so much that perhaps a solution is excluded as well. Of course, after one agent committed, another agent can commit, so that they iteratively narrow down their options more and more. This might lead to a situation where the possible joint actions are limited so much that eventually there is one joint action left for which it is either common knowledge that it *will* lead to the goal, or for which it is common knowledge that it *might* lead to the goal.

These ways of committing in a strategic way could be listed in 'rules' that agents should adhere to, thereby providing a more limited search space, making any algorithm based on it faster and more efficient.

In conclusion, allowing agents to commit to actions makes it easier for them to reach a point where it is common knowledge between all agents what the best course of action for them is. It gives agents an opportunity to exclude one possibility, in order to make another the salient solution. This greatly strengthens the notion of planning problem and solution as presented in the previous chapter, as it increases the probability of the agents finding a solution that works for all them, without running the risk of one half of the agents doing one action, and the other half doing another, or no agent knowing anything.

6 Conclusion

In this thesis we introduced a way to incorporate coalitions of agents in existing frameworks for epistemic planning. We introduced new Kripke models that, besides the usual indistinguishability relation, consisted of a control relation that shows what agents are able to control by choosing particular actions to perform.

We defined three versions of a sound, complete and decidable logic for these new models, and showed how these logics can be used to formalize the creation of plans for coalitions. They formalize what is needed for a coalition to be able to reach their goal, and what is needed for them to actually decide on a plan, rather than all agents doing what is best for them individually. While doing this we maintained the property of DEL that it works with specific actions, and the property of stit that it can deal with coalition power, making it suitable to formalize planning in a multi-agent, cooperative system.

Of course, the work done in this thesis does not provide a full, all-encompassing solution to every problem encountered in epistemic planning. We do hope, however, that it will spark new research and that it will inspire others to consider cooperative, multi-agent automated planning in this light, as we believe that this framework captures some of the most important aspects of that planning.

Unfortunately, time is limited, and we did not have a chance to fully develop every interesting idea we had while writing this thesis. Hence, we will give some ideas about ways in which the framework can be enhanced, or how one might be able to take it into a different direction.

First of all, the framework as it stands now is, in principle, able to deal with k-step planning: given a natural number k, we can look k steps ahead and check whether any sequence of k or less actions is a solution. What it cannot do yet is to check whether there is any sequence of actions that is a solution, independent of a natural number k. It would be interesting to look into defining a new modality $\Diamond_I^* \varphi_g$, which would exactly say that there is some sequence of joint actions for I such that after this sequence of actions, φ_g holds, and incorporate this in the planning. This would require that we look a bit more into the construction of the search tree, as we cannot check an infinite number of sequences. One can find ideas on how to restrict this tree in [3].

Another thing that would be interesting to look at in the future is plausibility relations. In this thesis we only used purely epistemic relations - either an agent knows something or they don't. It would in fact be very interesting to see how the framework can be extended when we add beliefs to the mix. Knowledge is very straightforward: if an agent knows a certain joint action will reach the goal, he

can announce it, making it common knowledge that it will. If an agent *believes* a joint action will reach the goal, the process of convincing the other agents becomes a lot more involved. Agents might have different believes, making it more difficult for them to agree to a certain joint action, because neither of them *knows* that the action reaches the goal. Hence adding belief probably implies that we also need a way to formalize the process leading up to the agents making a decision on what plan to perform.

References

[1] Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 2002.

[2] Mikkel Birkegaard Andersen. Towards theory-of-mind agents using automated planning and dynamic epistemic logic, 2014.

[3] Mikkel Birkegaard Andersen, Thomas Bolander, and Martin Holm Jensen. Conditional epistemic planning. In *Lecture Notes in Artificial Intelligence 7519*, pages 94–106, 2012.

[4] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Alternative axiomatics and complexity for deliberative stit theories. *Journal of Philosophical Logic*, 54:387–406, 2008.

[5] Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements, common knowledge and private suspicions. In *TARK proceedings*, pages 43–56, 1998.

[6] Alexandru Baltag, Hans van Ditmarsch, and Lawrence Moss. Epistemic logic and information update. In *Handbook of the Philosophy of Information*. Elsevier Science Publishers, 2008.

[7] Nuel Belnap and Michael Perloff. Seeing to it that: a canonical form for agentives. *Theoria*, 54:175–199, 1988.

[8] Nuel Belnap, Michael Perloff, and Ming Xu. *Facing the future: Agents and choice in our indeterminist world*. Oxford University Press, 2001.

[9] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 2001.

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

[11] Jan Broersen, Andreas Herzig, and Nicolas Troquard. From coalition logic to stit. *Electronic Notes in Theoretical Computer Science*, 157, 2006.

[12] Roberto Ciuni and John Horty. Stit logics, games, knowledge and freedom. In *Johan van Benthem on Logic and Information Dynamics*, Outstanding Contributions to Logic 5. 2014.

[13] Roberto Ciuni and Alberto Zanardo. Completeness of a branching-time logic with possible choices. *Studia Logica*, 96:393–420, 2010.

[14] Edmund Clark and Ernest Emerson. Using branching-time temporal logic to synthesize synchronisation skeletons. *Science of Computer Programming*, 2:241–266, 1982.

[15] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[16] Ronald Fagin, Joseph Y. Halpern, and Moshe Y. Vardi. What can machines know? on the properties of knowledge in distributed systems. *Journal of the ACM*, 39, 1992.

[17] Jelle Gerbrandy. Bisimulations on planet kripke, 1999A.

[18] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice*. Morgan Kaufmann, 2004.

[19] Andreas Herzig and Francois Schwarzentruber. Properties of logics of individual and group agency. In *Advances in Modal Logic*, volume 7, pages 133–149, 2008.

[20] John F. Horty. *Agency and Deontic Logic*. Oxford University Press, 2001.

[21] John F. Horty and Nuel Belnap. The deliberative stit: A study of action, omission, ability and obligation. *Journal of Philosophical Logic*, 24:583–644, 1995.

[22] Wojciech Jamroga and Wiebe van der Hoek. Agents that know how to play. *Fundamenta Informaticae*, 63:185–219, 2004.

[23] Barteld Kooi and Allard Tamminga. Moral conflicts between groups of agents. *Journal of Philosophical Logic*, 37:1–21, 2007.

[24] Marc Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12:149–166, 2002.

[25] Arthur Prior. *Past, present and future*. Clarendon University Press, 1967.

[26] Johan van Benthem. *Logical Dynamics of Information*. Cambridge University Press, 2010.

[27] Johan van Benthem, Jelle Gerbrandy, Tomohiro Hoshi, and Eric Pacuit. Merging frameworks for interaction. *Journal of Philosophical Logic*, 38, 2009.

[28] Johan van Benthem and Stefan Minica. Toward a dynamic logic of questions. *The journal of Philosophical Logic*, 2012.

[29] Johan van Benthem and Eric Pacuit. Connecting logics of choice and change. In *Nuel Belnap on Indeterminism and Free Action*. Springer, 2014.

[30] Johan van Benthem, Jan van Eijck, and Barteld Kooi. Logics of communication and change. *Information and Computation*, 204, 2006.

[31] Wiebe van der Hoek and Michael Wooldridge. Tractable multiagent planning for epistemic goals. In *Proceedings of AAMAS*, 2002.