Reasoning about
Resource-bounded Multi-agent Systems

by Nguyen Hoang Nga, MSc

Thesis submitted to The University of Nottingham
for the degree of Doctor of Philosophy, February 2011

Contents

List of Figures

Abstract

The thesis presents logic-based formalisms for modelling and reasoning about resource-bounded multi-agent systems. In the field of multi-agent system, it is well-known that temporal logics such as CTL and ATL are powerful tools for reasoning about multi-agent systems. However, there is no natural way to utilise these logics for expressing and reasoning about properties of multi-agent systems where actions of agents require resources to be able to perform. This thesis extends logics including Computational Tree Logic (CTL), Coalition Logic (CL) and Alternating-time Temporal Logic (ATL) which have been used to reasoning about multi-agent systems so that the extended ones have the power to specify and to reason about properties of resource-bounded multi-agent systems. While the extension of CTL is adapted for specifying and reasoning about properties of systems of resource-bounded reasoners where the resources are explicitly memory, communication and time, the extensions of CL and ATL are generalised so that any resource-bounded multi-agent system can be modelled, specified and reasoned about. For each of the logics, we describe the range of resource-bounded multi-agent systems they can account for and axiomatisation systems for reasoning which are proved to be sound and complete. Moreover, we also study the satisfiability problem of these logics.

Acknowledgments

I'd like to thank Natasha Alechina who has been my principal supervisor during my PhD research. I am grateful for the great interest she always has in my work, for comment, suggestion and advice she has given me. It is fair to say that this thesis would not exist at all if it were not for her tireless help and encouragement. I am also grateful Brian Logan for not only much useful suggestion and advice but also his general support in both study and life during my last three years.

I also would like to say thanks to Thorsten Altenkirch and Thomas Ågotnes for valuable comments and suggestions on my thesis.

Thanks to Engineering and Physical Sciences Research Council (EPSRC) and the University of Nottingham for funding my research, to the School of Computer Science for providing office space and equipment for my work.

I would also like to thank my friends and my family, without their help and support, I could not have completed this thesis. In particular, I'd like to thank my parents and my brother for their support.

Finally, to my wife Trang, for everything: thank you.

To my wife,

CHAPTER 1

INTRODUCTION

Actions are costly. An action such as “purchase a Ferrari” can be performed only if there are sufficient funds available. The aim of this thesis is to study logic-based formalisms for describing, specifying, reasoning about, and ultimately verifying properties of multi-agent systems where actions of agents are associated with certain costs.

This chapter is devoted to discussing the motivation for establishing such formalisms and the research objectives. At the end of the chapter is an outline of the remainder of the thesis.

1.1 Motivation

In the field of multi-agent systems, logic-based formalisms are powerful tools to specify multi-agent systems and to reason about them. There have been many logics defined and developed either for computational systems in general, such as LTL, CTL [Emerson, 1990] or specially for multi-agent systems including CL [Pauly, 2001, Pauly, 2002] and ATL [Alur et al., 2002]. Most of those formalisms are developed in the setting where agents are provided a number of actions and a system moves from one state to another by the fact that every agent of the system decides to perform an action. While logics such as LTL and CTL allow us to specify properties of multi-agent systems which describe the behaviour of a system as a whole, CL and ATL enable the possibility to specify properties relating to the power of agents or groups of agents.

However, these logics have failed to naturally model the effect of resource bounds on the strategic abilities of individuals or groups of individuals in multi-agent systems. Let us consider memory as a common resource in reasoning systems where it is refered to by other common terminologies such as knowledge and beliefs. It has been captured well by epistemic logics [Hintikka, 1962] which notoriously lead to the problem of logical omniscience [Hintikka, 1978]. This problem means that there is no boundary on the memory which is used to store beliefs of agents described by certain epistemic logics. In order to model beliefs of agents in dynamic systems, there have been several works which extend temporal logics with epistemic modalities such as ATEL [van der Hoek & Wooldridge, 2003]. However, agents specified by ATEL predictably suffer from the logical omniscience problem. There have been other logics proposed for modelling memory bounds of multi-agent systems where there are two common approaches to characterise the effects of memory bounds on agent abilities. The first one restricts strategic abilities by limiting the amount of information which are available to agents about systems states and choices in the past. Examples of logics following this approach are variants of ATL such as ATL_{Ir}, ATL_{ir} [Schobbens, 2004], ATL-R* [Jamroga & van der Hoek, 2004], ATLBM [Ågotnes & Walther, 2009]. Another approach is to interpret formula about beliefs by using syntactic structures [Fagin et al., 1995] such as in $\mathcal{L}_{min}$ and $\mathcal{L}^{\diamond}_{min}$ [Agotnes & Alechina, 2006], SSEL and DSEL [Ågotnes, 2004], BML [Alechina et al., 2006a]. Nevertheless, memory is only one aspect of a long story of resource-bounded multi-agent systems. There are many other resources which can be used by agents and affect significantly their abilities such as processing power, communication bandwidth, time, electrical power, etc. This fact gives rise to the need of a logical framework for modelling and reasoning about effects of bounds of resources used by a system of multiple agents.

This thesis is an effort to provide logic-based formalisms for bounded-resource multi-agent systems. We start by extending BML to the case of multi-agent systems where agents have limited memory and communication. In this approach, the amount of resources available for every agent in a system is recorded in each state. By moving from one state to another, the differences in the amount of resources between those states reflect the cost of action that every agent performs to make that move of the system. In other words, there is no need to attach costs to actions in the model of multi-agent system as they can be inferred from the relation between states and the information of resources encoded in each state. This approach has the following two disadvantages:

1. It can only use to model multi-agent systems where every agent is endowed with an initial amount of resources and while these systems evolve, the amount of resources allocated to every agent reduces gradually, that effectively makes models to shape like trees and increase the number of states in a model.

2. Moreover, using CTL as the background of the formalism prevent us from expressing properties of individual agents and sub-groups of agents.

Therefore, the thesis also presents a converse approach where it does not require the information of available amount of resources in every state. Instead, costs are associated with every action and we extend CL and ATL so that the resulting logic-based formalisms allow expressing the properties about the abilities of individual agents or a sub-group of agents such as the following:

- Given an amount of resources, a sub-group of agents can cooperate to produce a certain result,
- A sub-group of agents can cooperate to maintain some condition until a certain result is produced without spending more than an amount of resources,
- A sub-group of agents can cooperate to maintain some condition forever without spending more than an amount of resources.

Such an approach is much different from those restricting strategic abilities in the fact that it can naturally express properties of resource-bounded multi-agent systems, where, for logics restricting strategic abilities such as ATL_{Ir}, ATL_{ir} and ATLBM, it is harder to determine the limitation of the amount of information about system states and choices in the past to agents from the bounds on memory.

1.2 Research objectives and contribution

The research objectives which are addressed in this thesis are listed below:

1. To extend the logic BML to the case of system of multiple agents which are memory-bounded and communication-bounded.
2. To define computational models for the logics above.
3. To develop a logic-based formalism for resource-bounded multi-agent systems based on the computational models for CL where costs are associated with actions in the models.
4. To develop a logic-based formalism for resource-bounded multi-agent systems based on the computational models for ATL where costs are associated with actions in the models.

1.3 Outline of the thesis

The study of logic-based formalisms for computational models, especially multi-agent systems, has been carried out since the beginning day of Computer Science. In the next chapter, we review some of those formalisms. In Chapter 3, the extension of BML is presented. The resulting logic allows specifying properties of multi-agent systems with bounded memory and communication. After that, in Chapter 4 and 5, we step-by-step introduce two logics which allow us to express properties of abilities of sub-groups of agents under resource bounds. Finally, Chapter 6 completes the thesis by some conclusions and some directions for future work. In the following, we give the summary of the remaining chapters.

Chapter 2 - Background: This chapter provides a literature review of logic-based formalisms which have been established for modelling computational models. We will pay attention to those formalisms which are used especially for the case of multi-agent systems such as normal modal logics, CTL, CL and ATL.

Chapter 3 - Bounded Memory and Communication Logics: We study an extension of BML which allows reasoning about systems of multiple agents under bounds on memory, time and communication. In particular, we present the syntax and semantics of the resulting logics together with related results such as satisfiability problem, soundness and completeness. The chapter is based on the results from [Alechina et al., 2008c] and partly on [Alechina et al., 2008a, Alechina et al., 2008b, Alechina et al., 2009c].

Chapter 4 - Resource-bounded Coalition Logic: This chapter introduces a coalition logic for reasoning about multi-shot games in resource-bounded multi-agent systems. Besides the syntax and semantics of the logical languages, we also look at satisfiability problem, sound and complete results. The chapter is based on [Alechina et al., 2009b] and partly on [Alechina et al., 2009a].

Chapter 5 - Resource-bounded ATL: This chapter studies an extension of ATL for the case of resource-bounded multi-agent systems. We extend ATL to allow reasoning about strategies of resource-bounded multi-agent systems. After introducing the syntax and semantics of the logical language, we study the satisfiability problem and the sound and complete result. The chapter is based on [Alechina et al., 2010a] and [Nguyen, 2010].

Chapter 6 - Conclusion and Future work: finally, this chapter finishes the thesis with some conclusions and points out some directions for the future work.

CHAPTER 2

BACKGROUND

In this chapter, we review the literature of logic-based formalisms for specifying multi-agent systems. The chapter begins with a fundamental logic, namely modal logic, which has been used to express many aspects of agents such as mental states, actions and time. Later in the chapter, we concentrate on temporal logics. The content of this chapter is based on [Blackburn et al., 2002] for modal logic, [Emerson, 1990] for CTL, [Pauly, 2002, Pauly, 2001] for coalition logic, and [Alur et al., 2002, Goranko, 2001, Goranko & van Drimmelen, 2006] for ATL.

2.1 Modal logic

Originally, philosophers developed Modal Logic to study different levels of truth. Apart from *true* and *false*, a fact is possibly true if there is a world on which it is evaluated to be true. In the field of multi-agent systems, modal logic has been used to express many aspects of agents such as beliefs, effects of actions and time. For example, an agent is said to believe a fact if the fact is true in every world that the agent considers possible.

2.1.1 Syntax of modal logic

Formulas in modal logic are defined using the usual logical operators such as negation, conjunction and disjunction together with modalities $\Box$ and $\Diamond$. Given a finite set *Prop* such as $\{p, q, \ldots\}$ of propositional variables, the syntax of Modal Logics is as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \Box\varphi$$

where $p \in Prop$. The cases of other logical operators are defined as follows: $\top \equiv \varphi \vee \neg\varphi$ where $\top$ represents *true*, $\bot \equiv \neg\top$ where $\bot$ represents *false*, $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$,

$\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$ and $\Diamond\varphi \equiv \neg \Box \neg\varphi$.

As mentioned above, an application of modal logics is to characterise many aspects of agents. For example, to express that an agent believes that p is true, we use the formula $\Box p$. In the case of beliefs, the modality $\Box$ is usually renamed by B. Hence, the previous formula becomes Bp. Let us consider another example which expresses the effect of action, we use the formula $\Box p$ to express that after the agent performs an action, p becomes true. If we know the name of the action, such as a, then the formula is often rewritten as $[a]p$.

2.1.2 Semantics of modal logic

Formulas of modal logic are interpreted by means of Kripke models. A Kripke model $M = (S, R, V)$ is composed of a non-empty set S of possible worlds (or we often call states), a binary relation R over the set S and a truth mapping $V : \mathit{Prop} \to 2^S$ which assigns for every propositional variable a subset of possible worlds on which it is true. The truth of a formula in Modal Logic is evaluated at a possible world w in a Kripke model M by induction on the structure of the formula as follows:

- $M, w \vDash p$ iff $w \in V(p)$,
- $M, w \vDash \neg\varphi$ iff $M, w \nvDash \varphi$,
- $M, w \vDash \varphi \vee \psi$ iff $M, w \vDash \varphi$ or $M, w \vDash \psi$,
- $M, w \vDash \Box\varphi$ iff for every $w' \in S$ such that $(w, w') \in R$, $M, w' \vDash \varphi$.

In the last case, a formula $\Box\varphi$ is true at a possible world w if φ is true at every possible world w' which is related to w by R, that is $(w, w') \in R$. When Model Logic is used to reasoning about beliefs of agents, the relation R specifies worlds that an agent considers possible. For example, from the current world w, an agent considers that w_1, w_2 and w_3 are possible, this situation is described as $(w, w_i) \in R$ for every $i \in \{1, 2, 3\}$. If φ is true in all w_i's, i.e. $M, w_i \vDash \varphi$, then the agent believes φ at the current world w, i.e. $M, w \vDash B\varphi$.

A formula φ is *satisfiable* iff there exists a model M and a possible world w of M such that φ is true at w, that is $M, w \vDash \varphi$. A formula φ is *valid* in a model M iff φ is true at any possible world w of M; in this case, we shall write $M \vDash \varphi$. A formula φ is *valid* in a class of models F iff φ is valid in every model $M \in F$; in this case, we shall write $F \vDash \varphi$. A formula φ is *valid* iff φ is valid in every model M; in this case, we shall write $\vDash \varphi$. In the next sections, we recall how to syntactically generate valid formulas of modal logics and how to determine if a formula is satisfiable.

2.1.3 Axiomatisation systems

In Modal Logic, there is an interesting question: is there any syntactic way to generate all valid formulas with respect to a class of models? It turns out that one way to answer is to define Hilbert-like axiom systems which allow us to reason about those classes of models. Intuitively, all formulas generated by an axiom system define a logic. Given such an axiomatisation system X, a generated formula φ is called *provable* by the system, and we shall denote $\vdash_X \varphi$.

The smallest system is **K**. It contains all propositional tautologies together with the following schema (K)

$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi) \tag{K}$$

as axioms and two following inference rules

Modus Ponens: given $\vdash_{\mathbf{K}} \varphi \to \psi$ and $\vdash_{\mathbf{K}} \varphi$, imply $\vdash_{\mathbf{K}} \psi$,

Generalisation: given $\vdash_{\mathbf{K}} \varphi$, imply $\vdash_{\mathbf{K}} \Box\varphi$.

In the following, we familiarise ourselves with some notions relating to axiomatisation systems. A **K**-proof is a finite sequence of formulas each of which is either an axiom or the result of applying an inference rule upon one or more formulas occurred previously in the sequence. A formula φ is proved in **K** if it is the last element of some **K**-proof, written as $\vdash_{\mathbf{K}} \varphi$. The set of all provable formulas in **K** is called the logic **K**. Sometimes, we call elements of **K** as **K**-theorems. A formula φ is said to be **K**-consistent iff its negation is not provable, i.e. $\nvdash_{\mathbf{K}} \neg\varphi$.

We say that an axiomatisation system is sound with respect to a class of models if formulas provable by the system are valid in the class of models. Conversely, an axiomatisation system is complete with respect to a class of models if any valid formula in the class of models is provable by the system. In particular, **K** is sound and complete with respect to the class of Kripke models (for example, see [Blackburn et al., 2002, p.33, p.193, p.199]). In order to prove the soundness, it is not hard to show that (K) is valid in any Kripke models. Thinking of modality $\Box$ as characterising beliefs, it is intuitive for (K) to be true. Let us consider the following schema:

$$B(\varphi \to \psi) \to (B\varphi \to B\psi) \tag{2.1}$$

The schema (2.1) characterises a property of beliefs where if an agent believes $\varphi \to \psi$ and φ, it also believes ψ is expressed. Whether we are interested in other properties of beliefs about which we would like to reason? For example, it is intuitive to believe in what we already believe. This property can be characterised by the following schema:

$$B\varphi \to BB\varphi \tag{2.2}$$

It is more interesting that there is a correspondence between (2.2) and the class of transitive Kripke models, that is (2.2) is valid in a model iff the model is transitive. Similarly, there are other schemas each of which corresponds to a class of models. In Table 2.1, we list other schemas and their corresponding classes of models. The definition of each class of models in Table 2.1 is defined as

Name	Schema	Corresponding class
T	$\Box\varphi \to \varphi$	Reflexive Kripke models
D	$\Box\varphi \to \Diamond\varphi$	Serial Kripke models
4	$\Box\varphi \to \Box\Box\varphi$	Transitive Kripke models
5	$\Diamond\varphi \to \Box\Diamond\varphi$	Euclidean Kripke models

TABLE 2.1: Schemas and the corresponding classes of models

follows:

- A model $M = (S, R, V)$ is reflexive iff for every $w \in S$, $(w, w) \in R$.
- A model $M = (S, R, V)$ is serial iff for every $w \in S$, $\exists w' \in S$ such that $(w, w') \in R$.
- A model $M = (S, R, V)$ is transitive iff for every $w, w', w'' \in S$, if (w, w') and $(w', w'') \in R$ then $(w, w'') \in R$.
- A model $M = (S, R, V)$ is euclidean iff for every $w, w', w'' \in S$, if (w, w') and $(w, w'') \in R$ then $(w', w'') \in R$.

From the correspondence between schemas and classes of models, it is straightforward to have the soundness and completeness of logics which are extensions of **K** with a subset of schemas (T), (D), (4) and (5). For example, the logic extending **K** with the schema (T) is sound and complete with respect to the class of reflexive models. When a logic is composed of **K** and additional axiom schemas $\Sigma_1, \Sigma_2, \ldots, \Sigma_n$, it is usually given the name $\mathrm{K}\Sigma_1\Sigma_2\ldots\Sigma_n$. Because some of them are so widely used, they have been given special names such as T for KT, S4 for KT4, weak-S5 for KD5 and S5 for KT5. A logic A is smaller than a logic B, written as $A \subseteq B$, iff theorems of A are also theorems of B. It is worth noticing that the more axiom schemas a logic has, the larger the logic is. In this sense, **K** is the smallest modal logic; we also have **K** $\subseteq$ T $\subseteq$ S4 $\subseteq$ weak-S5 $\subseteq$ S5.

2.1.4 The satisfiability problem of Modal Logic

The problem of determining whether a formula is satisfiable is called satisfiability problem. A logic is decidable iff its satisfiability problem is decidable and the complexity of its satisfiability problem is the complexity of the logic. It is well-known that **K** it is decidable and the complexity is PSPACE-complete (see, for example, [Blackburn et al., 2002, p.392]).

2.1.5 The logical omniscience problem

As mentioned in Section 2.1.1, one application of modal logic is to describe and reason about beliefs of agents, or more practically, about the contents of the memory of agents. Then, $\Diamond$ is rewritten as B which stands for what an agent believes and a formula of the form $B\varphi$ is interpreted as an agent believes that φ is true, or φ is in its memory. In a model of this modal logic, a set of states to which we have access to from an actual one can be characterised as a freezing moment which gives us all the beliefs of the agent. Each state in the set is considered as a possible world and a belief of the agent must be true at all of its possible worlds. However, axiom (K) gives us the problem of logical omniscience [Hintikka, 1978] where logical consequences of the agent's beliefs are also what the agent beliefs. Only by considering logical tautologies, such amount of beliefs is already unacceptable for an agent in real life. For example, by knowing about all the rules of the chess game, anyone could have *instantly* known about a winning strategy, which is unrealistic.

In order to overcome the problem of logical omniscience, an approach is to interpret formulas about beliefs of agents by using syntactic structure [Fagin et al., 1995] such as in $\mathcal{L}_{min}$ and $\mathcal{L}^{\Diamond}_{min}$ [Agotnes & Alechina, 2006], BML [Alechina et al., 2006a], SSEL and DSEL [Ågotnes, 2004]. Since our first attempt in this thesis is inspired by this approach, let us briefly recall the definition of $\mathcal{L}_{min}$ which is simple enough but still fully describes it.

In $\mathcal{L}_{min}$, belief formulas (such as $B\varphi$) is considered as an atomic, and models for interpreting formulas are syntactic structures. Given a set of propositional variables *Prop*, the syntax of $\mathcal{L}_{min}$ is as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid B\varphi \mid \min(n)$$

where $p \in$ *Prop* and $n \in \mathbb{N}$. The set of formulas of the form $B\varphi$ is denoted as L_B. A model of $\mathcal{L}_{min}$

is a syntactic structure $M = (S, V)$ where S is a non-empty set of states and $V : S \to \wp(Prop \cup L_B)$ is a valuation mapping which determines what is true in a state. Given a $M = (S, V)$, let us denote $V^B(s) = \{\varphi \mid B\varphi \in V(s)\}$; then, the truth of a formula φ of $\mathcal{L}_{min}$ at a state s of M is defined by induction on the structure of φ as follows:

- $M, s \vDash p$ iff $p \in V(s)$,
- $M, s \vDash B\psi$ iff $\psi \in V^B(s)$,
- $M, s \vDash \min(n)$ iff $|V^B(s)| \geq n$,
- $M, s \vDash \neg\psi$ iff $M, s \nvDash \psi$,
- $M, s \vDash \varphi \vee \psi$ iff $M, s \vDash \varphi$ or $M, s \vDash \psi$.

It has been shown in [Agotnes & Alechina, 2006] that $\mathcal{L}_{min}$ is sound and complete, and its satisfiability problem is NP-complete. [Agotnes & Alechina, 2006] also proves that $\mathcal{L}_{min}$ is expressive enough to characterise the *know-at-least* and *only-know* modalities $(\triangle, \triangledown)$ of SSEL [Ågotnes, 2004].

2.2 Computation Tree Logic

Computation Tree Logic (CTL) is a widely used logic for reasoning about concurrent programs. As systems of multiple agents can be seen as a set of concurrent programs, CTL can also be used for reasoning about multi-agent systems. Models of CTL are temporal structures where each state corresponds to a time point and has several possible future states. Each state may be related to others by transitions which correspond to the possible moves of the system. From this point of view, time in CTL is branching to express the non-deterministic nature of systems such as multi-agent systems. In a multi-agent system, each agent may have more than one action to perform at a time. Depending on which action each agent decides to perform, the system may move to different states.

2.2.1 Syntax of CTL

CTL is an extended modal logic of which the modality $\Box$ describes the relationship with states in the distance of one step of time. Moreover, there are also other modalities to express properties overtime which span on more than one step of time. Given a finite set *Prop* = $\{p, q, \ldots\}$, the syntax of CTL is as follows.

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \mathrm{AX}\varphi \mid \mathrm{A}\varphi\mathcal{U}\psi \mid \mathrm{E}\varphi\mathcal{U}\psi$$

We also use other logical operators such as $\wedge$ and $\rightarrow$ which are defined in a similar way as Modal Logics. That is $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$ and $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$. The modality AX is similar to the modality $\Box$ in Modal Logic. Moreover, we also have EX in CTL which is defined like $\Diamond$, that is EX$\varphi \equiv \neg$AX$\neg\varphi$. In the following, we also define other modalities in CTL as abbreviations:

$$\mathrm{AF}\varphi \equiv \mathrm{A}\top\mathcal{U}\varphi$$

$$\mathrm{EF}\varphi \equiv \mathrm{E}\top\mathcal{U}\varphi$$

$$\mathrm{AG}\varphi \equiv \neg\mathrm{EF}\neg\varphi$$

$$\mathrm{EG}\varphi \equiv \neg\mathrm{AF}\neg\varphi$$

Let us now informally explain the meaning of each modality in CTL. The formula AXφ means that whatever move a system makes, φ will be true in the next state. The modality EX is similar to $\Diamond$ in Modal Logic where EXφ is to say that there exists a move by the system so that in the next state, φ is true. The other modalities speak about properties over longer future. The formula A$\varphi\mathcal{U}\psi$ says that for any sequence of moves by the system, φ is true until ψ is true. Likewise, the formula E$\varphi\mathcal{U}\psi$ says that there is a sequence of moves by system where φ is true until ψ is true. The formula AFφ means that for any sequence of moves, φ is eventually true. The formula EFφ means that there is a sequence of moves where φ is eventually true. The formula AGφ means that for any sequence of moves, φ is globally true. The formula EGφ means that there is a sequence of moves where φ is globally true.

2.2.2 Semantics of CTL

Semantics of CTL is defined by means of total Kripke structures which are Kripke structures with an additional requirement where each state are related to at least one other in the binary relation of the structure. The binary relation over set of states in a total Kripke structure now has a temporal meaning, when two states are related in the binary relation; this means the system can make a move from one state to another in one step of time. Then, the modalities AX and EX, which corresponds to $\Box$ and $\Diamond$ in Modal Logic, respectively, are used to describe what happens after one step in the future. The other modalities are for the case of longer future, their semantics are defined with the help of the notion of paths in total Kripke structures.

Given a Kripke model $M = (S, R, V)$, a path in M is a (possibly infinite) sequence $\lambda = s_0 s_1 \ldots$ of states such that $(s_i, s_{i+1}) \in R$ for any $i \geq 0$. Given a path $\lambda = s_0 s_1 \ldots$ of M, we denote $\lambda[i] = s_i$ for any $i \geq 0$ and $\lambda[i, j] = s_i s_{i+1} \dots s_j$ for any $j \geq i \geq 0$. Then, the truth of a CTL formula φ at a state $s \in S$ is defined by induction on the structure of φ as follows:

- $M, s \vDash p$ iff $s \in V(p)$,
- $M, s \vDash \neg\varphi$ iff $M, w \nvDash \varphi$,
- $M, s \vDash \varphi \vee \psi$ iff $M, s \vDash \varphi$ or $M, s \vDash \psi$,
- $M, s \vDash$ AXφ iff for every $s' \in S$ such that $(s, s') \in R$, $M, s' \vDash \varphi$,
- $M, s \vDash$ A$\varphi\mathcal{U}\psi$ iff for every path λ of M such that $\lambda[0] = s$, there exists $i \geq 0$ where $M, \lambda[i] \vDash \psi$ and for every $0 \leq j < i$, $M, \lambda[j] \vDash \varphi$,
- $M, s \vDash$ E$\varphi\mathcal{U}\psi$ iff there exists a path λ of M where $\lambda[0] = s$ such that there exists $i \geq 0$ where $M, \lambda[i] \vDash \psi$ and for every $0 \leq j < i$, $M, \lambda[j] \vDash \varphi$.

As we can see from the definition of the semantics for the modality AX, it is just the same for the case of the modality $\Box$ in Modal Logic. However, the last two cases of the semantics are different. The first thing is that we need to make use of the notion paths in the models. Intuitively, a path from one state represents a possible future of a system along time where the system travels from state to state. The formulas A$\varphi\mathcal{U}\psi$ says that for any future of the system starting from a state, the formula ψ will be eventually true and at all the moments before that happens, φ remains true. Meanwhile, the formula E$\varphi\mathcal{U}\psi$ relaxes the condition as comparing to A$\varphi\mathcal{U}\psi$, it requires only the existence of a future. For the other modalities, it is not hard to show the following:

- $M, s \vDash$ EXφ iff there is a state $s' \in S$ such that $(s, s') \in R$, and $M, s' \vDash \varphi$.
- $M, s \vDash$ AFφ iff for every path λ of M such that $\lambda[0] = s$, there exists $i \geq 0$ where $M, \lambda[i] \vDash \varphi$.
- $M, s \vDash$ EFφ iff there exists a path λ of M where $\lambda[0] = s$ such that there exists $i \geq 0$ where $M, \lambda[i] \vDash \psi$.
- $M, s \vDash$ AGφ iff for every path λ of M and $i \geq 0$, $M, \lambda[i] \vDash \varphi$.
- $M, s \vDash$ EGφ iff there exists a path λ of M such that for any $i \geq 0$, $M, \lambda[i] \vDash \psi$.

We say that a formula φ of CTL is satisfiable iff there exists a model M and a state s in M such that $M, s \vDash \varphi$. Similarly, a formula φ of CTL is valid iff for any model M and a state s in M, we have $M, s \vDash \varphi$.

2.2.3 Axiomatisation system for CTL

We present in this section a sound and complete axiomatisation system for CTL. The axiomatisation system allows us to generate any valid formula of CTL. Similar to Modal logic, it has a set of axiom schemas each of which characterises the meaning of a temporal modality and inference rules. The axiom schemas of CTL [Emerson, 1990] are listed as follows:

Ax1. All tautologies of Propositional Logic

Ax2a. $\mathrm{EF}\varphi \leftrightarrow \mathrm{E}\top\mathcal{U}\varphi$

Ax2b. $\mathrm{AG}\varphi \leftrightarrow \neg\mathrm{EF}\neg\varphi$

Ax3a. $\mathrm{AF}\varphi \leftrightarrow \mathrm{A}\top\mathcal{U}\varphi$

Ax4. $\mathrm{EX}(\varphi \vee \psi) \leftrightarrow \mathrm{EX}\varphi \vee \mathrm{EX}\psi$

Ax5. $\mathrm{AX}(\varphi) \leftrightarrow \neg\mathrm{EX}\neg\varphi$

Ax6. $\mathrm{E}\varphi\mathcal{U}\psi \leftrightarrow \psi \vee (\varphi \wedge \mathrm{EXE}\varphi\mathcal{U}\psi)$

Ax7. $\mathrm{A}\varphi\mathcal{U}\psi \leftrightarrow \psi \vee (\varphi \wedge \mathrm{AXA}\varphi\mathcal{U}\psi)$

Ax8. $\mathrm{EX}\top \wedge \mathrm{AX}\top$

Ax9a. $\mathrm{AG}(\theta \to (\psi \wedge \mathrm{EX}\theta)) \to (\theta \to \neg\mathrm{A}\varphi\mathcal{U}\psi)$

Ax9b. $\mathrm{AG}(\theta \to (\psi \wedge \mathrm{EX}\theta)) \to (\theta \to \neg\mathrm{AF}\psi)$

Ax10a. $\mathrm{AG}(\theta \to (\psi \wedge (\varphi \to \mathrm{AX}\theta))) \to (\theta \to \neg\mathrm{E}\varphi\mathcal{U}\psi)$

Ax10b. $\mathrm{AG}(\theta \to (\psi \wedge \mathrm{AX}\theta)) \to (\theta \to \neg\mathrm{EF}\psi)$

Ax11. $\mathrm{AG}(\varphi \to \psi) \to (\mathrm{EX}\varphi \to \mathrm{EX}\psi)$

The two inference rules of CTL [Emerson, 1990] are listed as follows:

- Given $\vdash_{\mathrm{CTL}} \varphi \to \psi$ and $\vdash_{\mathrm{CTL}} \varphi$, imply $\vdash_{\mathrm{CTL}} \psi$
- Given $\vdash_{\mathrm{CTL}} \varphi$, imply $\vdash_{\mathrm{CTL}} \mathrm{AG}\varphi$

The above axiomatisation system for CTL has been proved to be sound and complete with respect to total Kripke structures [Emerson & Halpern, 1982, Emerson, 1990]. Moreover, we have the complexity result of CTL that the satisfiability problem of CTL is EXPTIME-complete [Emerson, 1990]. The model checking problem is to determine whether a given CTL formula is true or not at a given state of a given model. From [Clarke & Emerson, 1982], we have the complexity result that the model checking problem for CTL is in deterministic polynomial time.

2.3 Coalition Logic

CTL logic is a powerful logic for reasoning about the ability of a multi-agent system as a whole. However, it is difficult to use CTL to reason about the ability of individuals or subsets of individuals in a multi-agent system. In this section, we introduce coalition logic [Pauly, 2002] (or CL), which allows us to do so. We first present concepts of Game Frame which are used to defined semantics of CL. Later after that is the syntax and semantics of CL. At the end of the section, we recall some results of axiomatisation system, complexity and model checking for CL.

2.3.1 Game Frames

Coalition logic allows reasoning about Game Frames which explicitly contain individual agents and their abilities at every state of the frames. A Game Frame is defined as follows.

Definition 1. *A Game Frame G is a tuple $(N, \{\Sigma_i \mid i \in N\}, S, T, o)$ in which*

- *N is a non-empty finite set of agents*
- *Σ_i is a non-empty set of actions for each agent $i \in N$*
- *S is a non-empty set of states*
- *$T : S \times N \to \wp(\bigcup_{i \in N} \Sigma_i)$ is a mapping which assigns the available actions for every agent $i \in N$ at a state $s \in S$ and satisfies the following conditions:*
    1. *$T(s, i) \subseteq \Sigma_i$*
    2. *$T(s, i) \neq \emptyset$*
- *$o : S \times \Pi_{i \in N} \Sigma_i \to S$ is a partial mapping which assigns the outcome of a joint action $(a_i)_{i \in N} \in \Pi_{i \in N} \Sigma_i$, where $a_i \in T(s, i)$, at a state $s \in S$.*

Let us discuss the definition of game frames. A game frame $G = (N, \{\Sigma_i \mid i \in N\}, S, T, o)$ contains a non-empty set of states on which the system of multiple agents in N operates. The system moves from one state to another by the fact that each agent decides an action which is available at the starting state. The set of action available to each agent at a state is determined by the function T. The first condition on T, saying that $T(s,i) \subseteq \Sigma_i$, means that actions available to an agent must be among those belonging to the agent. Then, the second condition $T(s,i) \neq \varnothing$ implies that at any state, every agent must be able to perform some action. We can see that this condition is similar to the total condition in models of CTL, so that systems never stuck at a state. After every agent decides an action to perform, the destination state from the current one is determined by the partial output mapping o. For convenience, we denote a joint action for a coalition $C \subseteq N$ as $a_C = (a_i)_{i \in C}$. We also extend the function T for the case of coalitions as follows:

$$T(s,C) = \{a_C = (a_i)_{i \in C} \mid \forall i \in N : a_i \in T(s,i)\}$$

In CL, the property of interest to reason about is the ability of individual agents or groups of individual agents; that is whether there is an action for an individual agent or a joint action for a group of agents to perform so that the systems moves to a state among those of interest without caring about which actions all other agents perform. Formally, we state the property in the following way: at a state s, can a coalition $C \subseteq N$ be effective in achieving a set of states $X \subseteq S$? Given a game frame G, the notion of effectivity for a coalition C can be captured by the *effectivity function* $E_G : S \to \wp(N) \to \wp(\wp(S))$ which is defined as follows:

$$X \in E_G(s)(C) \text{ iff } \exists a_C \in T(s,C) \forall a_{\bar{C}} \in T(s,\bar{C}) : o(s,(a_C, a_{\bar{C}})) \in X$$

In other words, $E_G(s)(C)$ contains all sets of states for which the coalition C is effective at state s. We may notice for the extreme case, when $C = \varnothing$, it is straightforward to see that $E_G(s)(\varnothing)$ defines all sets of states that is unavoidable for the system, that is given $X \in E_G(s)(\varnothing)$, for any joint action a_N, the system will move to a state within X.

Given an effectivity function E_G of a game frame G, we can easily find some trivial properties of E_G such as $S \in E_G(s)(C)$ for any $s \in S$ and $C \subseteq N$. Apart from those, the effectivity structure E_G also has the following properties with given names:

- *Outcome-monotonicity:*

$$\forall s \in S, \forall C \subseteq N, \forall X \subseteq X' \subseteq S : X \in E_G(s)(C) \Rightarrow X' \in E_G(s)(C)$$

- *Coalition-monotonicity:*

$$\forall s \in S, \forall C \subseteq C' \subseteq N, \forall X \subseteq S : X \in E_G(s)(C) \Rightarrow X \in E_G(s)(C')$$

- *C-Regularity:*

$$\forall s \in S, \forall X \subseteq S : X \in E_G(s)(C) \Rightarrow \bar{X} \notin E_G(s)(\bar{C})$$

  E_G is regular if it is C-regular for all $C \subseteq N$.

- *C-Maximality:*

$$\forall s \in S, \forall C \subseteq N, \forall X \subseteq S : X \notin E_G(s)(C) \Rightarrow \bar{X} \in E_G(s)(\bar{C})$$

  E_G is maximal if it is C-maximal for all $C \subseteq N$.

- *Superadditivity:*

$$\forall s \in S, \forall C_1, C_2 \subseteq N \text{ and } C_1 \cap C_2 = \varnothing, \forall X_1, X_2 \subseteq S :$$
$$X_1 \in E_G(s)(C_1) \text{ and } X_2 \in E_G(s)(C_2) \Rightarrow X_1 \cap X_2 \in E_G(s)(C_1 \cup C_2)$$

It turns out that studying properties of effectivity functions originated from game frames is useful for defining axiomatisation systems of CL. Given an effectivity function in general, i.e. an arbitrary function $E : S \to \wp(N) \to \wp(\wp(S))$, we call it *playable* iff it has the following properties:

1. $\forall C \subseteq N$: $\varnothing \notin E(C)$,
2. $\forall C \subseteq N$: $S \in E(C)$,
3. N-maximality,
4. Outcome-monotonicity,
5. Super-additivity.

We have the following result about the correspondence between playable effectivity function and game frames in [Pauly, 2002].

Lemma 1. [1] *An effectivity function E is playable iff it is the effectivity function of some game frame.*

[1] Strictly speaking, Lemma 1 fails due to a flaw in Pauly's proof. In fact, the definition of playable effectivity function needs at least one more requirement which we shall call N-determinacy and present in Section 4.4.4.

What Lemma 1 tells us is twofold. Firstly, playable effectivity functions and game frames are interchangeable. As we will see later when defining semantics of CL, we use playable effectivity functions to define models of CL instead of game frames. Secondly, the lemma also reveals axiom schemas for the axiomatisation system of CL.

2.3.2 Syntax and semantics

CL contains modalities which have the form of $[C]$ where $C \subseteq N$. A formula of the form $[C]\varphi$ expresses the property that the coalition C has a joint action to force φ true in the next state, independently on the decisions of other agents out of C. Given a finite set *Prop* of propositional variables and a finite set N of agents, the syntax of CL is as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid [C]\varphi$$

where $p \in \textit{Prop}$ and $C \subseteq N$. The other logical operations are defined as usual: $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$, $\varphi \to \psi \equiv \neg\varphi \vee \psi$, $\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$.

Formulas of CL are interpreted by means of effectivity models. An effectivity model M is a tuple (S, E, V) where:

- S is a non-empty set of states,
- $E : S \to \wp(N) \to \wp(\wp(S))$ is a playable effectivity function,
- $V : \textit{Prop} \to \wp(S)$ is a mapping which assigns truth of propositional variables at each state in S.

Given a effectivity model $M = (S, E, V)$, the truth of a formula in CL is defined at a state of the model by induction on the structure of the formula as follows:

- $M, s \models p$ iff $s \in V(p)$,
- $M, s \models \neg\varphi$ iff $M, s \not\models \varphi$,
- $M, s \models \varphi \vee \psi$ iff $M, s \models \varphi$ or $M, s \models \psi$,
- $M, s \models [C]\varphi$ iff $\{s' \mid M, s' \models \varphi\} \in E(s)(C)$.

In the definition of semantics for CL, the last case says that the formula $[C]\varphi$ is true at a state s iff it is effective for the coalition C to force the system to go to a state where φ is true. Using the effectivity function E in the model, this requirement is equivalent to the condition $\{s' \mid M, s' \vDash \varphi\} \in E(s)(C)$. Because of Lemma 1, we can also define semantics of CL by means of game models. A game model M is a pair (G, V) where $G = (N, \{\Sigma_i\}_{i \in N}, S, T, o)$ is a game frame and $V : Prop \to \wp(S)$ is a mapping which assigns the truth of propositional variables at each state in S. Then, the last case of the semantics is defined as follows:

- $M, s \vDash [C]\varphi$ iff $\exists a_C \in T(s, C)$ such that $\forall a_{\bar{C}} \in T(s, \bar{C})$: $M, o(s, (a_C, a_{\bar{C}})) \vDash \varphi$

As usual, a formula φ in CL is satisfiable iff there exists an effectivity model M and a state s such that $M, s \vDash \varphi$. φ is valid in a model M iff $M, s \vDash \varphi$ for any state s of M. Then, φ is valid iff it is valid in any effectivity model.

2.3.3 Axiomatisation system

Given a set N of agent, the axiomatisation system allows us to generate valid formulas of Coalition Logic. This axiomatisation is defined by the following axiom schemas and inference rules:

Axioms

$(\bot)$ $\neg[C]\bot$

$(\top)$ $[C]\top$

(N) $\neg[\varnothing]\varphi \to [N]\neg\varphi$

(M) $[C](\varphi \wedge \psi) \to [C]\varphi$

(S) $[C_1]\varphi_1 \wedge [C_2]\varphi_2 \to [C_1 \cup C_2](\varphi_1 \wedge \varphi_2)$ where $C_1 \cap C_2 = \varnothing$

Inference rules

Modus Ponens Given $\vdash_{\text{CL}} \varphi \to \psi$ and $\vdash_{\text{CL}} \varphi$, imply $\vdash_{\text{CL}} \psi$

Equivalence Given $\vdash_{\text{CL}} \varphi \leftrightarrow \psi$, imply $\vdash_{\text{CL}} [C]\varphi \leftrightarrow [C]\psi$

A formula φ is a theorem of CL iff $\vdash_{\text{CL}} \varphi$. φ is consistent iff $\neg\varphi$ is not a theorem of CL. We have the result from [Pauly, 2002] that the above axiomatisation system of CL is sound and complete with respect to effectivity models. The proof is done by constructing a canonical effectivity model of φ and showing that the effectivity function embodied in the canonical model is playable. Moreover, also from [Pauly, 2002], we have the complexity result of CL that the satisfiability problem for CL is PSPACE-hard.

2.4 Alternating-time Temporal Logic

Alternating-time Temporal Logic (ATL) [Alur et al., 2002] is an extension of CTL for modelling multi-agent systems. The quantifiers over future paths in CTL are parametrised by sets of agents in ATL for expressing the selective quantification over possible futures as a result of the interaction between a coalition and its complement. In other words, ATL allows us to model the existence of strategies for coalitions of agents in a system to achieve certain goals in short (one step computations) and long (many step computations) futures. When a system contains only one agent, ATL is equivalent to CTL. Moreover, CL can also be seen as a fragment of ATL as CL only expresses the existence of strategies for coalitions to achieve goals in one step futures.

Semantics of ATL is initially defined by means of Alternating-time Transition Systems (ATS) which is later generalised to Concurrent Game Structures (CGS). It is worth noticing that both ATSs and CGSs are equivalent to Game Frames and Effectivity Models which were used in the previous section to define the semantics of CL. In this section, we briefly review ATL by looking at the definition of CGS, then, the syntax and semantics of ATL. Finally, we recall some essential results of ATL including the axiomatisation system for ATL, its complexity for the satisfiability problem and the model-checking problem.

2.4.1 Concurrent Game Structures

A Concurrent Game Structure describes a multi-agent system where the system transits from one state to another as the result of performing a joint action by all the agents in the systems. In this section, we review the concept of CGSs together with related notions.

Definition 2. *A Concurrent Game Structure S is a tuple $(n, Q, Prop, V, d, \sigma)$ where*

- *n is the number of agents in the system. For convenience, the agents are identified with the number $1, \ldots, n$ and we denote the set of agent by $N = \{1, \ldots, n\}$.*
- *Q is a non-empty set of states.*
- *Prop is the set of propositional variables.*

- *V : Prop $\rightarrow \wp(Q)$ is a mapping which assigns the truth of propositional variables at each state in Q.*

- *$d : Q \times N \rightarrow \mathbb{N}$ is a mapping which assigns the number of actions available to each agent at every state of Q. It is required that $d(q,i) \geq 1$ for all $q \in Q$ and $i \in N$. For convenience, the actions available to an agent i at a state q is identified with the numbers $1,\ldots,d(q,i)$. Then, we write $D(q)$ to denote the set $\Pi_{i=1,n}\{1,\ldots,d(q,i)\}$ of all joint actions of agents in N.*

- *Given a state $q \in Q$ and $a \in D(q)$, $\sigma(q,a)$ defines the result state when agents in N perform the joint action a at q.*

CGSs have a requirement on the mapping d which defines the number of actions available for every agent that $d(q,i) \geq 1$ where $i \in N$ and $q \in Q$. It is similar to the requirement of Game Frames on the mapping T which determines the set of actions available for every agent at a state where the returning sets by the mapping T are non-empty. This means from any state in the system, it is always possible to transit to another one by some joint action of every agent.

Given a CGS $\mathcal{S}$, we denote the set of available actions for each agent at a state as $D(q,i) = \{1,\ldots,d(q,i)\}$. Similarly, we denote the set of joint actions of a coalition $A \subseteq N$ at a state as $D(q,A) = \Pi_{i\in A}D(q,i)$. When $A = N$, we simply write $D(q)$ to denote the set of available joint actions for all agents in the system.

We define that a state $q \in Q$ is a successor of another state $q' \in Q$ iff there exists a joint action $a \in D(q)$ such that $\sigma(q,a) = q'$. Then, a computation in $\mathcal{S}$ is an infinite sequence $\lambda = q_0q_1\ldots$ of states in Q where for every $i \geq 0$ we have that q_{i+1} is the successor of q_i. For convenience, we denote $\lambda[i] = q_i$, $\lambda[i,j] = q_i\ldots q_j$ for any $j \geq i \geq 0$.

Given an joint action $a_A \in D(q,a)$ of a coalition A at state $q \in Q$, we define the set of possible outcomes by the action a_A as follows:

$$\mathit{out}(q,a_A) = \{q' \in Q \mid \exists a_{\bar{A}} : \sigma(q,(a_A,a_{\bar{A}}) = q'\}$$

The last concept we mention in this section is strategies. It is essential to define the semantics of ATL. Given a CGS $\mathcal{S}$, we define a *strategy* F_A for a coalition $A \subseteq N$ as a mapping from the set Q^+ of finite and non-empty sequences of states to an joint action for A such that $F_A(\lambda q) \in D(q,A)$. When the coalition A follows the strategy F_A starting from some state q_0, agents in A decide to perform the joint action $F_A(q_0)$. Because agents out of A may have many actions available to perform, each joint action by the complement coalition of A can transit the system to a different state q_1. From each state q_1, agents in A decide to perform the joint action $F_A(q_0q_1)$ and joint actions available to the complement coalition of A once again transit the system to different states q_2. This happens again and again to form a set of possible computations of the strategy F_A. Formally, we define the set of possible computations of a strategy F_A, starting from a state $q_0 \in Q$, as follows:

$$\text{out}(q_0, F_A) = \{q_0q_1 \ldots \mid \forall i \geq 0 : q_{i+1} \in \text{out}(q_i, F_A(q_0 \ldots q_i))\}$$

2.4.2 Syntax and Semantics of ATL

ATL is an extension of CTL to express properties of coalitional abilities in short and long time futures. Given a finite set N of agents and a set *Prop* of propositional variables, the syntax of ATL is defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \langle\langle A\rangle\rangle\bigcirc\varphi \mid \langle\langle A\rangle\rangle\Box\varphi \mid \langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$$

where $p \in$ *Prop* and $A \subseteq N$. As usual, other logical operators are defined as abbreviations: $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$ and $\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

As we mentioned in the previous section, the semantics of ATL is defined by means of CGSs. Given a CGS $\mathcal{S} = (n, Q, \mathit{Prop}, V, d, \sigma)$, the truth of a formula in ATL at a state of $\mathcal{S}$ is defined inductively on the structure of the formula as follows:

- $\mathcal{S}, q \vDash p$ iff $s \in V(p)$.
- $\mathcal{S}, q \vDash \neg\varphi$ iff $M, s \nvDash \varphi$.
- $\mathcal{S}, q \vDash \varphi \vee \psi$ iff $M, s \vDash \varphi$ or $M, s \vDash \psi$.
- $\mathcal{S}, q \vDash \langle\langle A\rangle\rangle\bigcirc\varphi$ iff there exists a strategy F_A for the coalition A such that for every $\lambda \in \text{out}(q, F_A)$, we have that $\mathcal{S}, \lambda[1] \vDash \varphi$.
- $\mathcal{S}, q \vDash \langle\langle A\rangle\rangle\Box\varphi$ iff there exists a strategy F_A for the coalition A such that for every $\lambda \in \text{out}(q, F_A)$ and for every $i \geq 0$, we have that $\mathcal{S}, \lambda[i] \vDash \varphi$.
- $\mathcal{S}, q \vDash \langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ iff there exists a strategy F_A for the coalition A such that for every $\lambda \in \text{out}(q, F_A)$, there is a number $i \geq 0$ where $\mathcal{S}, \lambda[i] \vDash \psi$ and for any $i > j \geq 0$, we have that $\mathcal{S}, \lambda[j] \vDash \varphi$.

For the formula $\langle\langle A\rangle\rangle\bigcirc\varphi$, an equivalent way to define its semantics is to use joint actions. That is $\mathcal{S}, q \models \langle\langle A\rangle\rangle\bigcirc\varphi$ iff there is a joint action $a_A \in D(q, A)$ such that for all $q' \in \text{out}(q, a_A)$, we have that $\mathcal{S}, q' \models \varphi$. Therefore, one can see that the modality $\langle\langle A\rangle\rangle\bigcirc$ is equivalent to the modality $[A]$ in CL. In the extreme case when $A = \varnothing$, the formula $\langle\langle\varnothing\rangle\rangle\bigcirc\varphi$ expresses the property that it is unavoidable that φ is true after the system transits to another state. That is for any joint action available to the grand coalition N, φ is true in the next state.

The $\Box$ modality in ATL has the semantics which is similar to that of G operator in CTL. A formula $\langle\langle A\rangle\rangle\Box\varphi$ is true at a state of a CGS iff there exists a strategy for the coalition A such that φ is true along any possible computation of the strategy. In the case when A is the empty coalition, the formula $\langle\langle\varnothing\rangle\rangle\Box\varphi$ has the same meaning as the formula $AG\varphi$ in CTL, that is φ unavoidable for the whole system. The last form of formula $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi$ in ATL says that there exists a strategy for the coalition A to keep φ true until ψ is eventually true.

As usual, a formula φ in ATL is satisfiable iff there exists a CGS and a state where φ is true. φ is valid in a CGS iff it is true at any state of the CGS; and φ is valid iff it is valid in any CGS.

2.4.3 Axiomatisation system

An axiomatisation system to generate valid formulas of ATL has been introduced in [Goranko & van Drimmelen, 2006]. We have mentioned that CL is a fragment of ATL, therefore there is no surprise that axioms for CL also appears as axioms for ATL. In the following, we list all the axioms and inference rules of the axiomatisation system for ATL. Notice that $A, A_1, A_2 \subseteq N$ are arbitrary coalitions of agents.

Axioms:

(PL) Tautologies of propositional variables

($\bot$) $\neg\langle\langle A\rangle\rangle\bigcirc\bot$

($\top$) $\langle\langle A\rangle\rangle\bigcirc\top$

(N) $\neg\langle\langle\varnothing\rangle\rangle\bigcirc\varphi \to \langle\langle N\rangle\rangle\bigcirc\neg\varphi$

S $\langle\langle A_1\rangle\rangle\bigcirc\varphi_1 \wedge \langle\langle A_2\rangle\rangle\bigcirc\varphi_2 \to \langle\langle A_1 \cup A_2\rangle\rangle\bigcirc(\varphi_1 \wedge \varphi_2)$ where $A_1 \cap A_2 = \varnothing$

FP$_\Box$ $\langle\langle A\rangle\rangle\Box\varphi \leftrightarrow \varphi \wedge \langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\Box\varphi$

GFP$_\Box$ $\langle\langle\varnothing\rangle\rangle\Box(\theta \to (\varphi \wedge \langle\langle A\rangle\rangle\bigcirc\theta)) \to \langle\langle\varnothing\rangle\rangle\Box(\theta \to \langle\langle A\rangle\rangle\Box\varphi)$

FP$_\mathcal{U}$ $\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi \leftrightarrow \psi \vee (\varphi \wedge \langle\langle A\rangle\rangle\bigcirc\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi)$

LFP$_\mathcal{U}$ $\langle\langle\varnothing\rangle\rangle\Box((\psi \vee (\varphi \wedge \langle\langle A\rangle\rangle\bigcirc\theta)) \to \theta) \to \langle\langle\varnothing\rangle\rangle\Box(\langle\langle A\rangle\rangle\varphi\mathcal{U}\psi \to \theta)$

Inference rules:

Modus Ponens: Given $\vdash_{\text{ATL}} \varphi \to \psi$ and $\vdash_{\text{ATL}} \varphi$, implies $\vdash_{\text{ATL}} \psi$

$\langle\langle A\rangle\rangle\bigcirc$**-monotonicity:** Given $\vdash_{\text{ATL}} \varphi \to \psi$, implies $\vdash_{\text{ATL}} \langle\langle A\rangle\rangle\bigcirc\varphi \to \langle\langle A\rangle\rangle\bigcirc\psi$

$\langle\langle\varnothing\rangle\rangle\Box$**-necessitation** Given $\vdash_{\text{ATL}} \varphi$, implies $\vdash_{\text{ATL}} \langle\langle\varnothing\rangle\rangle\Box\varphi$

As usual, we have the notions of theorem and consistency in ATL. A formula φ is a theorem of ATL iff $\vdash_{\text{ATL}} \varphi$. φ is consistent iff its negation is not a theorem of ATL, i.e. $\nvdash_{\text{ATL}} \neg\varphi$. The soundness and completeness of the above axiomatisation system for ATL have been proved in [Goranko & van Drimmelen, 2006]. In order to prove the completeness of ATL, the idea in [Goranko & van Drimmelen, 2006] is to construct a tree-like model for a consistent formula where each agent has a fix number of available actions at every state of the model. We also have the result of the satisfiability problem for ATL in [Goranko & van Drimmelen, 2006] that the complexity of ATL over a fixed and finite set of agents is EXPTIME-complete.

CHAPTER 3

BOUNDED MEMORY-COMMUNICATION LOGIC

3.1 Introduction

This chapter presents a logic, namely Bounded Memory-Communcation Logic (BMCL), to model and reason about systems of multiple reasoning agents whose resources of memory and communication are bounded. We extend the logic BML introduced in [Alechina et al., 2006a] which is for reasoning about systems of a single memory-bounded reasoner to the case of systems of multiple reasoning agents. Sometimes we also refer to reasoning agents as reasoners.

BML is used for reasoning about systems of single agents, the logic does not support modelling communication as well as reasoning about the effects of communication bounds on the abilities of the single agent. Since BMCL is an extension of BML where we take into account the fact multiple reasoning agents in a system can communicate with each other to exchange information. Communication between agents can be vital to them when some information, which they cannot derive by themselves, is needed in the middle of their reasoning processes. In such systems, communication is considered as a resource where the limitation on the total amount of exchanging information can affect the ability of some agents, hence, the whole system.

By setting the limitation on both memory of reasoners in the systems, i.e. the amount of information that each reasoner can hold in its memory at a time, and communication, i.e. the total amount of information that each reasoner can send and receive from other agents in the system, the logic BMCL, which is based on CTL, allows us to reason about the ability of the system to derive certain results possibly within a restriction of time. This section is organised as follows. In the next section, we discuss in more detail the models of systems of multiple reasoners. After that are the syntax and the semantics of the logic BMCL. Then, we study the satisfiability problem of BMCL. Although the problem of satisfiability of BML was missing in [Alechina et al., 2006a], the approach that we introduced in this chapter serves well for the case of BML. Finally, we introduce an axiomatisation system for BMCL which is shown to be sound and complete. Since BMCL is an extension of BML, this soundness and completeness result of BMCL also works well for the counterpart of BML.

3.2 Systems of multiple reasoners

We begin this section by a motivating example. Assume that we are requested to deploy a safety-critical system in a new building measuring environmental indexes such as temperature, humidity, the amount of carbon dioxide gas and the amount of leaking household gas. The system consists of multiple detecting agents (detectors) which are distributed within the building. For convenience, each detector must be relatively small in size, and can perform some simple forms of reasoning such as to infer the level of danger based on measurements of the environment. In order to increase the effectiveness and sensitivity of the system, detectors can communicate with each other to exchange their reasoning results. Because of the size requirement, each detector has a limitation on the memory to hold information, on the battery which leads to the limitation on how many messages they can send and receive through communication. During the design phase of the system, it is important to verify the correctness and effectiveness of the system such as how many detectors are necessary? Can it operate correctly (raise alarm when and only when some measurement bypass a threshold of safety)? How long can a detector operate without changing the battery?

In this section, we abstract such systems as systems of multiple reasoners about which the logic BMCL reasons. Formally, a system of multiple reasoners consists of a finite number $n \in \mathbb{N}$ of reasoners. Each reasoner has an internal memory and a set of inference rules for the purpose of reasoning about new information. Moreover, each reasoner is associated with a knowledge base whose size is considerably larger, so that it cannot fit into the internal memory. In order to derive a goal, a reasoner fetches necessary premises from its knowledge base, then applies an inference rule to derive the goal or intermediate results. If necessary, the reasoner repeats fetching more premises from the knowledge base and performing other inference rules again and again until the goal is achieved.

We have mentioned the size of the internal memory which holds formulas including premises from the knowledge base, intermediate results and possibly goals of a reasoner. Intuitively, one may assume that each cell of an internal memory can store a single symbol used in a formula, then the size of the internal memory is defined as the number of cells in the internal memory. In this situation, if we have a memory of 5 cells, it can hold only two formulas, $p \to q$ which contains three symbols and p (which contains one symbol). One may also prefer having an empty cell to separate these two formulas in the internal memory, hence, it is required that the total number of cells used to store two formulas is exactly 5, i.e. the internal memory is full and cannot hold another formula of any size. The internal memory cannot also hold two formulas $p \to q$ and $q \to r$ because the number of symbols in both formulas is six which bypasses the size of the internal memory. However, we take into account a simpler way to to calculate the size of a internal memory where cells of the internal memory is defined in a different way. We assume that each cell of an internal memory can hold one formula of arbitrary length. This means an internal memory of size 5 can hold at most 5 formulas. By using this definition of cell, the representation of any result relating to the logic BMCL is also technically simpler than the case when a cell can hold only a single symbol. Therefore, we make an assumption in this chapter that each cell of the internal memory of a reasoning agent can hold a formula of arbitrary length and the size of the internal memory is the maximal number of formulas which it can hold at the same time, i.e. the number of cells of the internal memory. It is worth noticing that the results of the logic BMCL where each cell of an internal memory is defined to hold only a single symbol can be adapted from the results we present in this chapter, however, the representation will be unnecessarily more complicated.

In a multi-agent system, agents have the ability to share reasoning results by exchanging information with each other. Such a communication ability might help an agent to reduce time for deriving a certain goal. For example, an agent i can carry out the derivation of a result which is one of the intermediate requirements for another agent j to conclude its goal. Then, while the agent j attempts to generate intermediate results, the agent i helps j to have one of them. This reduces the amount of time j needs to conclude the goal in comparison with the situation when j has to derive all the intermediate results by its own. To model the communication between agents, the common approach is to define a protocol which specifies how agents establish the connection between themselves and how to exchange information. A simple and usual way is to use the *ask-tell* protocol as presented in [Alechina et al., 2006b]. In *ask-tell* protocol, an agent requests information from another by sending an *ask* message where it specified the information that it would like to have. Let us assume that this action cost one step of time. If the other agent are holding the requested information, it simply replies to the asking agent a *tell* message where it the requested information is confirmed. Again, the replying action is assumed to consume another step of time. Overall, a successful communication by the ask-tell protocol requires at least two steps of time to complete. We further simplify the ask-tell protocol by introducing the *copy* protocol where the actions of sending an ask message and replying a tell message, which span in two steps of time, are combined into a single one. In the copy protocol, if an agent asks another for information and the other has that information, it will reply immediately. The asking and replying are combined into one action, namely *copy*, which can be carried out by an "asking" agent if and only if the requested information is available in the internal memory of the agent to be asked. Although the copy protocol may reduce the amount of time for an agent to derive a goal, it still can simulate the ask-tell protocol where the ability of performing the copy action corresponds to a "reply" action replying the ask message which could be sent some steps previously. When an agent is unable to perform the copy action, this would mean the asked agent does not have an answer (because either it is deriving the answer or it is impossible to do so) for an ask message which is sent some steps previously. Notice that using the ask-tell protocol, it is usually required to have a buffer in the model of each agent to hold the information which are requested by the others. However, it is not required when using the copy protocol. Hence, using the copy protocol also simplify the model of agents and communication between agents. Communication between agents can be restricted by setting a maximal amount of information which can be exchanged. Similar to the case of memory, the maximal amount of information can be understood as either the maximal number of symbols used to formalise formulas exchanged between agents or, in a simpler approach, the maximal number of messages an agent can send to others. In the case when we use the copy protocol to model communication between agents, we set the limitation on communication by restricting the maximal number of times each agent can perform a copy action.

In such a system of multiple reasoners, each reasoning agent has an internal memory whose size is bounded by some number. We assume that each agent also has the following actions:

Read: An agent retrieves a premise from its knowledge base and places it in some cell of its internal memory by using a *read* action. If the cell is not empty, the premise will overwrite the formula stored in the cell.

Infer: An agent uses an *infer* action to apply some inference rule over some premises in its internal memory. Similar to the *read* action, the resulting formula obtained by applying the inference rule is placed into the internal memory which may overwrite some formula if the chosen cell is not empty.

Copy: An agent performs a *copy* action if it wants to copy a formula which is stored in the internal memory of another agent into its memory. Similar to two actions above, storing a formula may overwrite some formula in the internal memory of the agent which performs the copy action if the chosen cell is not empty.

Idle: This action is given to every agent for the case when they do not want to do anything. When an agent performs the idle action, its state, i.e. the content of the internal memory, remains unchanged.

Let us consider an example of a system of two reasoning agents, namely agent 1 and agent 2. Each agent has an internal memory of size 2. The restriction on communication is 2, i.e. each agent can perform the copy action at most twice. They have the ability to reason within propositional logic and share the same set of inference rules which contains only conjunction introduction and modus ponens. These two rules are defined as follows.

Conjunction introduction

$$\frac{\varphi_1 \quad \varphi_2}{\varphi_1 \wedge \varphi_2}$$

Modus ponens

$$\frac{\varphi_1 \rightarrow \varphi_2 \quad \varphi_1}{\varphi_2}$$

Informally, the conjunction introduction inference rule means that if an agent has two formulas φ_1 and φ_2 in its memory, it concludes a new formula $\varphi_1 \wedge \varphi_2$ and places $\varphi_1 \wedge \varphi_2$ somewhere in its internal memory. Similarly, an agent performs the modus ponens inference rule if it has two formulas of the forms $\varphi_1 \rightarrow \varphi_2$ and φ_1, then the agent concludes φ_2 and places it φ_2 somewhere in its internal memory.

Two agents also share the same knowledge base which contains the following formulas:

- A_1, A_2, A_3, A_4
- $A_1 \wedge A_2 \rightarrow B_1, A_3 \wedge A_4 \rightarrow B_2$
- $B_1 \wedge B_2 \rightarrow C$

The goal for the system of these two agents is to conclude that C is true. Intuitively, C should be the case by implying B_1 and B_2 from A_1, A_2, A_3 and A_4 with the help of both inference rules: conjunction introduction and modus ponens. However, with the restriction on the size of the internal memory for both agents, it is impossible for one of them to imply C only by itself. The problem is that, once an agent obtains either B_1 or B_2, it must reserve one cell in the internal memory to hold this intermediate result. In order to imply C, the agent needs to have also the other intermediate result, either B_2 or B_1, respectively. However, it is required to have two cells of memory to do so in order to load either A_3 and A_4 or A_1 and A_2, respectively, into the internal memory. Nevertheless, with the cooperation of both agents, the system can conclude C. Figure 3.1 illustrates one of possible traces for the system to conclude C where each line describes the state of

Step	Memory	Op.	Memory	Op.
0	$\{\}$	Read	$\{\}$	Read
1	$\{A_3\}$	Read	$\{A_1\}$	Read
2	$\{A_3, A_4\}$	Infer	$\{A_1, A_2\}$	Infer
3	$\{A_3, A_3 \wedge A_4\}$	Read	$\{A_1 \wedge A2, A_2\}$	Read
4	$\{A_3 \wedge A_4 \rightarrow B_2, A_3 \wedge A_4\}$	Infer	$\{A_1 \wedge A2, A_1 \wedge A2 \rightarrow B_1\}$	Infer
5	$\{A_3 \wedge A_4 \rightarrow B_2, B_2\}$	Copy	$\{A_1 \wedge A2, B_1\}$	Idle
6	$\{B_1, B_2\}$	Infer	$\{A_1 \wedge A2, B_1\}$	Idle
7	$\{B_1, B_1 \wedge B_2\}$	Read	$\{A_1 \wedge A2, B_1\}$	Idle
8	$\{B_1 \wedge B_2 \rightarrow C, B_1 \wedge B_2\}$	Infer	$\{A_1 \wedge A2, B_1\}$	Idle
9	$\{C, B_1 \wedge B_2\}$		$\{A_1 \wedge A2, B_1\}$	

FIGURE 3.1: Two agents cooperate to derive C.

the internal memory for each agent and actions which are chosen correspondingly to perform. The result of these actions is the internal memory of two agents in the next line. At the initial step, i.e. step 0, both agents have empty memories and decide to perform a read action where agent 1 and 2 read the formulas A_3 and A_1 from their knowledge bases, respectively. At step 1, they perform another read action so that in the next step, A_4 and A_2 are added into the internal memory of agent 1 and 2, respectively. At this point, agent 1 and 2 perform the conjunction introduction inference rule to produce $A_3 \wedge A_4$ and $A_1 \wedge A2$, respectively. These intermediate results overwrite the formula A_4 in the internal memory of agent 1 and A_1 in the internal memory of agent 2, notice that the selection of formulas to be overwritten is arbitrary. Then, two agents load the "rule" formulas $A_3 \wedge A_4 \rightarrow B_2$ and $A_1 \wedge A_2 \rightarrow B_1$ and apply the modus ponens inference rule to conclude B_2 and B_1, respectively. At step 5, agent 1 performs a copy action to retrieve B_1 from the internal memory of agent 2. Then, it applies another conjunction introduction inference rule to obtain $B_1 \wedge B_2$ at step 6. After that, it loads the rule formula $B_1 \wedge B_2 \rightarrow C$ and performs the modus ponens inference rule to conclude C at step 9. Notice that from step 5, agent 2 does nothing. It constantly performs the idle action and the reasoning task of concluding C after step 5 is done only by agent 1. The above trace of computation is one of possible traces for the system to conclude C, it is easy to draw another trace where both agents or only agent 2 concludes C at the end.

In this example, it is easy to see that 9 steps of time is the minimal time required for the system to conclude C. This minimal time can change if we alter the restriction on memory and communication. For example, if we forbid communication by setting the limitation of communication to 0, it is not possible for the system to conclude C as both agents do not have the ability to conclude C on their own. We can relax the restriction on memory by increasing the size of the internal memories to 3. In this configuration, the minimal time to conclude C also increases to 13 steps of time, as depicted in Figure 3.2.

Step	Memory	Op.	Memory	Op.
0	$\{\}$	Read	$\{\}$	Idle
1	$\{A_3\}$	Read	$\{\}$	Idle
2	$\{A_3, A_4\}$	Infer	$\{\}$	Idle
3	$\{A_3, A_3 \wedge A_4\}$	Read	$\{\}$	Idle
4	$\{A_3 \wedge A_4 \to B_2, A_3 \wedge A_4\}$	Infer	$\{\}$	Idle
5	$\{A_3 \wedge A_4 \to B_2, B_2\}$	Read	$\{\}$	Idle
6	$\{A_1, A_3 \wedge A_4 \to B_2, B_2\}$	Read	$\{\}$	Idle
7	$\{A_1, A_2, B_2\}$	Infer	$\{\}$	Idle
8	$\{A_1 \wedge A_2, A_2, B_2\}$	Read	$\{\}$	Idle
9	$\{A_1 \wedge A_2, A_1 \wedge A_2 \to B_1, B_2\}$	Infer	$\{\}$	Idle
10	$\{A_1 \wedge A_2, B_1, B_2\}$	Infer	$\{\}$	Idle
11	$\{B_1 \wedge B_2, B_1, B_2\}$	Read	$\{\}$	Idle
12	$\{B_1 \wedge B_2, B_1 \wedge B_2 \to C, B_2\}$	Infer	$\{\}$	Idle
13	$\{C, B_1 \wedge B_2 \to C, B_2\}$		$\{\}$	

FIGURE 3.2: Agent 1 concludes C by itself.

In general, the ability of systems of reasoning agents varies depending on different settings on the restriction of memory and communication. Hence, the trade-offs are possible between the size of the internal memory, the restriction of communication and the required time for a derivation. When the restriction of communication between agents is tighter, agents tend to operate by its own and require more space for the internal memory to complete reasoning tasks. Furthermore, this setting also increases the required time to complete derivations. When the size of the internal memory decreases, agents often need help from others to produce intermediate results, which leads to the increasing of messages exchanged between them. However, the cooperation between agents helps their system to solve most problems faster, which reduces the required time to complete derivations.

3.3 Syntax and semantics of BMCL

In this section, we present a family of logics, namely, BMCL, each of which allows reasoning about a system of multiple reasoners as described in the previous section. Let us assume a system of multiple reasoners consisting of n agents, the set of agents in this system is denoted by $N = \{1, \ldots, n\}$. For simplicity, we also assume that all agents in the system agree on a logical language L, which they use for reasoning, and a finite set *IR* of inference rules. For simplicity, we assume that L is a finite set of of formulas. This assumption is reasonable since, in the context of resource bounds, an agent should not be able to hold in its internal memory a formula of arbitrary length. If we fix a maximal length for formulas, the language L must be finite. Each inference rule $r \in IR$ is defined as a pair (pre_r, con_r) where:

- $pre_r \subseteq \wp(L)$ is a set of subsets of required formulas for the rule to be applicable.
- $con_r : pre_r \rightarrow L$ is a function which specifies the conclusion when the rule r is applied.

We denote the memory bounds, the communication bounds and the knowledge bases for agents in N by three mappings n_{mem}, n_{com} and K, respectively, where

- $n_{mem}, n_{com} : N \rightarrow \mathbb{N}$ are two mappings which specify the restrictions of memory and communication for each agent in N.
- $K : N \rightarrow \wp(L)$ is a mapping which defines the knowledge base for each agent in N.

3.3.1 Syntax

The primitive formulas of BMCL are defined as follows:

- Formulas of the form $B_i\alpha$ where $\alpha \in L$ and $i \in N$: the meaning of the formula $B_i\alpha$ is that α is one of the formulas which are held in the internal memory of the agent i. For convenience, we define for each $i \in N$ that $L_{mem}(i) = \{B_i\alpha \mid \alpha \in L\}$, and $L_{mem} = \bigcup_{i \in N} L_{mem}(i)$.
- Formulas of the form c_i^k where $k \in \mathbb{N}$ such that $0 \leq k \leq n_{com}(i)$ and $i \in N$: the meaning of the formula c_i^k is that the agent i has communicated (or performed the copy action) exactly k times. For convenience, we define for every $i \in N$, $L_{com}(i) = \{c_i^k \mid 0 \leq k \leq n_{com}(i)\}$ and, $L_{com} = \bigcup_{i \in N} L_{com}(i)$. Then, L_{com} is called the set of message counters.

Let us now present the syntax of BMCL which is based on CTL.

$$\varphi ::= \top \mid B_i\alpha \mid c_i^k \mid \neg\varphi \mid \varphi \vee \psi \mid \mathrm{EX}\varphi \mid \mathrm{E}(\varphi\mathcal{U}\psi) \mid \mathrm{A}(\varphi\mathcal{U}\psi)$$

As usual, the cases of other logical operators are defined in terms of equivalence: $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$, $\varphi \to \psi \equiv \neg\varphi \vee \psi$ and $\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$. Moreover, we also define other modalities as in the case of CTL as follows:

$$\begin{aligned} &\mathrm{AX}\varphi \equiv \neg\mathrm{EX}\neg\varphi && \\ &\mathrm{AF}\varphi \equiv \mathrm{A}(\top\mathcal{U}\varphi) && \mathrm{EF}\varphi \equiv \mathrm{E}(\top\mathcal{U}\varphi) \\ &\mathrm{AG}\varphi \equiv \neg\mathrm{EF}\neg\varphi && \mathrm{EG}\varphi \equiv \neg\mathrm{AF}\neg\varphi \end{aligned}$$

The formula $\mathrm{AX}\varphi$ is the dual of $\mathrm{EX}\varphi$. Its meaning is that for any move a system performs, φ is true in the next state of the system. In other words, this formula epxresses a property which is unavoidable for the system. Similar to this formula, $\mathrm{AF}\varphi$ also expresses an unavoidable property of systems of multiple reasoners. A system has this propety iff for any execution, φ is finally true. The formula $\mathrm{EF}\varphi$ is used to express the ability of the whole system. It is true when all reasoners in the system can cooperate so that φ is finally true. The last two formulas express properties which are globally true. The formula $\mathrm{AG}\varphi$ means φ is globally true in a system, for any execution of the system, meanwhile the formula $\mathrm{EG}\varphi$ is to express the property where all reasoners in the system can cooperate to maintain φ true forever. Let us give some examples of properties expressed in BMCL. We reconsider the example of the system of two agents which use conjunction introduction and modus ponens for reasoning in page 28 of the previous section. We express the property that the system can conclude C by the following formula:

$$\mathrm{EF}(B_1C \vee B_2C)$$

Let us define the modality EX^k where $k \in \mathbb{N}$ by the following equivalence:

$$\mathrm{EX}^k\varphi \equiv \bigvee_{0 \le i \le k} \underbrace{\mathrm{EX}\ldots\mathrm{EX}}_{i \text{ times}}\varphi$$

Then, we can express the property for the restriction of 10 steps of time for the system to conclude C by the following formula:

$$\mathrm{EX}^{10}(B_1C \vee B_2C)$$

From the example, we already knew that it is impossible for the system to conclude C in less than 9 steps of time. Therefore, the formula $\mathrm{EX}^6(B_1C \vee B_2C)$ is not a property of the system.

3.3.2 Semantics

Similar to CTL, semantics of BMCL is defined by Kripke structure.

Definition 3. *A BMCL transition system is a triple $M = (S, R, V)$ where:*

1. *S is a non-empty set of states.*
2. *R is a total binary relation on S*
3. *$V : S \times N \to \wp(L \cup L_{com})$.*

Semantics of BMCL is defined by means of BMCL transition systems. A transitions system contains three components: a non-empty set of states S, a set R of transition relations in S and a mapping which assigns to each state in S and an agent i in N a subset of formulas from the reasoning language L and message counters in $L_{com}(i)$. The subset of formulas is used to describe the memory of the agent i and the message counters are used to record the number of copy actions which the agent i has performed. For convenience, we denote $V_{mem}(s, i) = V(s, i) \cap L$ as the memory of an agent i and $V_{com}(s, i) = V(s, i) \cap L_{com}(i)$ as message counters for the agent i.

Given a BMCL transition system $M = (S, R, V)$, we define a path in M as an infinite sequence $(s_0, s_1, s_2, \ldots)$ where $(s_i, s_{i+1}) \in R$ for all $i \geq 0$. The truth of a formula at a state of the BMCL transition system M is defined by induction as follows:

- $M, s \vDash \top$.
- $M, s \vDash B_i\alpha$ iff $\alpha \in V(s, i)$ for any $\alpha \in L$ and $i \in N$.
- $M, s \vDash c_i^k$ iff $c_i^k \in V(s, i)$ for any $i \in N$.
- $M, s \vDash \neg\varphi$ iff $M, s \nvDash \varphi$.
- $M, s \vDash \varphi \vee \psi$ iff $M, s \vDash \varphi$ or $M, s \vDash \psi$.
- $M, s \vDash \text{EX}\varphi$ iff there exists $s' \in S$ such that $(s, s') \in R$ and $M, s' \vDash \varphi$.
- $M, s \vDash \text{E}(\varphi\mathcal{U}\psi)$ iff there exists a path $(s_0 = s, s_1, s_2, \ldots)$ in M and $k \geq 0$ such that $M, s_i \vDash \varphi$ for all $i = 0, \ldots, k-1$ and $M, s_k \vDash \psi$.
- $M, s \vDash \text{A}(\varphi\mathcal{U}\psi)$ iff for any path $(s_0 = s, s_1, s_2, \ldots)$ in M, there exists $k \geq 0$ such that $M, s_i \vDash \varphi$ for all $i = 0, \ldots, k-1$ and $M, s_k \vDash \psi$.

Given a model $M = (S, R, V)$, for convenience, we write $V(s) = \{\varphi \mid M, s \models \varphi\}$ to denote the set of all formulas which are true at s of M.

A formula φ is satisfiable iff there exists a model M such that φ is true at some state of M. φ is satisfiable in a class of models iff there exists a model M of this class where φ is true at some state of M. φ is valid in a model M iff it is true at every state of the model. Finanlly, φ is valid iff it is valid in any model.

3.3.3 Models for systems of multiple agents

To reason about systems of multiple reasoning agents, we are interested in formulas which are valid in any model of systems of multiple reasoning agents. In this section, we present the class of BMCL transition systems which correspond to systems of multiple reasoning agents as defined in Section 3.2. In order to do so, we define several notions in a BMCL transition system to describe bounds on memory and communication, actions and the applicability of an action.

Given a BMCL transition system $M = (S, R, V)$, the bounds of memory are defined by restricting the cardinality of $V_{mem}(s, i)$ for all agents $i \in N$ and states $s \in S$. In other words, for each agent $i \in N$, the bound of memory is expressed by the condition $|V_{mem}(s, i)| \leq n_{mem}(i)$. Moreover, at each state, there should be only one counter for each agent to record the number of times an agent has performed the copy action. This condition is obtained by setting the constraint $|V_{com}(s, i)| = 1$.

We define the set *Act* of actions for each agent i in N which contains the following actions where $\alpha, \beta \in L$ and $\Gamma \subseteq L$.

- $read_{i,\alpha,\beta}$: Agent i loads a formula α from its knowledge base into its internal memory. The formula β indicates which formula in the internal memory will be overwritten, especially in the case when the internal memory is full. If β is not in the internal memory of the agent i, α will be put into an empty cell of its memory.

- $infer^{r}_{i,\Gamma,\alpha,\beta}$: Agent i performs the inference rule $r \in IR$ over the set of formulas Γ to conclude α. Similar to the previous action, the formula β is overwritten by α if it is in the internal memory of the agent i, otherwise α is loaded into an empty cell of the internal memory.

- $copy_{i,j,\alpha,\beta}$: Agent i copies a formula α from the internal memory of another agent $j \in N$ $(i \neq j)$. Similar to actions *read* and *infer*, the formula β is overwritten by α if it is the memory of the agent j, otherwise α is loaded into an empty cell of the internal memory.

- $Idle_i$: Agent i performs an idle action when it decides to not do anything.

We define the set $Act_{s,i}$ of applicable actions for each agent $i \in N$ at a state $s \in S$, assume that $\alpha, \beta \in L$ and $\Gamma \subseteq L$.

- $read_{i,\alpha,\beta} \in Act_{s,i}$ iff $\alpha \notin V_{mem}(s,i)$, $\alpha \in K_i$ and if $|V_{mem}(s,i)| \geq n_{mem}(i)$ then $\beta \in V_{mem}(s,i)$.
- $infer^r_{i,\Gamma,\alpha,\beta} \in Act_{s,i}$ iff $\Gamma \in pre_r$, $\alpha = con_r$, $\alpha \notin V_{mem}(s,i)$, $\Gamma \subseteq V_{mem}(s,i)$ and if $|V_{mem}(s,i)| \geq n_{mem}(i)$ then $\beta \in V_{mem}(s,i)$.
- $copy^r_{i,j,\alpha,\beta} \in Act_{s,i}$ iff $\alpha \notin V_{mem}(s,i)$, $\alpha \in V_{mem}(s,j)$ and if $|V_{mem}(s,i)| \geq n_{mem}(i)$ then $\beta \in V_{mem}(s,i)$.
- $Idle_i$ is always in $Act_{s,i}$.

Except the case of the action $Idle_i$, which is available to all agents at any state of a system, other actions require certain requirements to be applicable. The action $read_{i,\alpha,\beta}$ is applicable when α is available from the knowledge base of agent i. Moreover, when the internal memory of the agent i is full, β must be one of the formulas in the internal memory of the agent so that it is replaced with α. In the case of $infer^r_{i,\Gamma,\alpha,\beta}$, it is applicable when agent i has formulas required to fire the rule $r \in IR$. It is also required β to be one of the formulas in the internal memory when it is full. Finally, the action $copy^r_{i,j,\alpha,\beta}$ is applicable when the agent j has α in the internal memory so that i can copy α into its internal memory.

When an action is available for an agent to perform at a state of the system, the system moves to another state where changes of the internal memory and the message counter of the agent express the effect of the action. In the following, we define the effect of an action a by introducing its corresponding binary relation $R_a \subseteq S \times S$.

- $(s,t) \in R_{read_{i,\alpha,\beta}}$ iff $read_{i,\alpha,\beta} \in Act_{s,i}$ and $V_{mem}(t,i) = V_{mem}(s,i) \setminus \{\beta\} \cup \{\alpha\}$.
- $(s,t) \in R_{infer^r_{i,\Gamma,\alpha,\beta}}$ iff $infer^r_{i,\Gamma,\alpha,\beta} \in Act_{s,i}$ and $V_{mem}(t,i) = V_{mem}(s,i) \setminus \{\beta\} \cup \{\alpha\}$.
- $(s,t) \in R_{copy^r_{i,j,\alpha,\beta}}$ iff $copy^r_{i,j,\alpha,\beta} \in Act_{s,i}$ and $V_{mem}(t,i) = V_{mem}(s,i) \setminus \{\beta\} \cup \{\alpha\}$ and $V_{com}(t,i) = \{c_i^{n+1}\}$ where $V_{com}(s,i) = \{c_i^n\}$
- $(s,t) \in R_{Idle_i}$ iff $V(t,i) = V(s,i)$.

The action $Idle_i$ keeps the internal memory and message counter unchanged. Other actions only allow the formula α to be added into the internal memory of the agent, if β also appears in the internal memory, it will be overwritten by α.

Then, we define the class of models describing systems of multiple reasoning agents as follows.

Definition 4. *The class BMCM*$(K_i, n_{mem}(i), n_{com}(i))_{i \in A}$ *of models describing systems of multiple reasoning agents consists of BMCL transition systems* $M = (V, R, S)$ *satisfying the following conditions:*

1. $|V_{mem}(s,i)| \leq n_{mem}(i)$ *and* $|V_{com}(s,i)| = 1$ *for all* $i \in N$ *and* $s \in S$.
2. $\forall (s,t) \in R$*, there exists an joint action* $(a_1, \dots, a_n)$ *where* $a_i \in Act_{s,i}$ *for all* $i \in N$ *such that* $(s,t) \in R_{a_i}$ *for all* $i \in N$.
3. *Conversely, for all* $s \in S$ *and an joint action* $(a_1, \dots, a_n)$ *where* $a_i \in Act_{s,i}$ *for all* $i \in N$*, there exists* $t \in S$ *such that* $(s,t) \in R$ *and* $(s,t) \in R_{a_i}$ *for all* $i \in A$.

The definition of the class BMCM(K, n_{mem}, n_{com}) requires that a BMCL transition system is a model for a system of multiple reasoning agents when it satisfies three conditions. The first condition establishes bounds on memory and communication for agents in the system. It requires that the set of formulas which depicts the internal memory of an agent does not exceed the maximal bound on the internal memory of the agent. Then, the model is required to have only one message counter for each agent. The last two conditions are applied to the binary relation R. A state $s \in S$ is related to another state $t \in S$ by R if there is a joint action by agents of the system such that t is the resulting state of the joint action at state s. Moreover, models in the class are also required in the last condition that for each joint action which is available for the agents, there must be another state where the system will move to by performing the joint action.

In the remainder of this chapter, we present the satisfiability problem for formulas of BMCL by which we show that BMCL is decidable and a sound and complete deductive system for reasoning in BMCL about systems of multiple reasoning agents.

3.4 The satisfiability problem of BMCL

In this section, we present a procedure to decide the satisfiability of a formula of BMCL in the class BMCM(K, n_{mem}, n_{com}). The procedure bases on a characteristic of the class BMCM(K, n_{mem}, n_{com}) where there is a unique model used as a representative to determine the satisfiability of formulas. Notice that it is still possible to make use of the procedure for the decidability of CTL [Emerson, 1990] with suitable extensions corresponding to requirements for models in BMCM(K, n_{mem}, n_{com}).

We omit this approach with the purpose to emphasise the characteristic which models in BMCM(K, n_{mem}, n_{com}) share.

3.4.1 The canonical model of BMCM(K, n_{mem}, n_{com})

Let us consider an arbitrary model $M = (S, R, V)$ in BMCM(K, n_{mem}, n_{com}). At a state s in M, the mapping V describes the state of the internal memory and the message counter for each agent in the system. Moreover, the definition of BMCM(K, n_{mem}, n_{com}) determines path starting from s. That is, in a different model $M' = (S', R', V')$ of BMCM(K, n_{mem}, n_{com}), if there is a state s' which replicates s, i.e. $V(s,i) = V(s',i)$ for all $i \in N$, we have the same set of paths starting from s' as those starting from s in M when we do not differentiate states with the same value of V and V', respectively. This property of models in BMCM(K, n_{mem}, n_{com}) suggests considering a canonical model in the class BMCM(K, n_{mem}, n_{com}) which "contains" all models in the class.

Definition 5. *The canonical model CM is a triple* (S^c, R^c, V^c) *where*

1. $S^c = \Pi_{i \in N}(L_i \times \{0, \ldots, n_{com}(i)\})$ *where* $L_i = \{\Gamma \subseteq L \mid |\Gamma| \leq n_{mem}(i)\}$
2. $V^c((\Gamma_j, n_j)_{j \in N}, i) = \Gamma_i \cup \{c_i^{n_i}\}$ *for all* $i \in N$.
3. *For all* $s, t \in S^c$, $(s,t) \in R^c$ *iff there are* $a_i \in Act_{s,i}$ *for all* $i \in N$ *such that* $(s,t) \in R^c_{a_i}$ *for all* $i \in N$.

We firstly show that *CM* is in BMCM(K, n_{mem}, n_{com}).

Lemma 2. *CM is a model of the class BMCM*(K, n_{mem}, n_{com}).

Proof. The last two conditions in the definition of the class BMCM(K, n_{mem}, n_{com}) follow directly from the definition of *CM*. To complete the proof, we show the following:

By defintion, for any $s \in S^c$ and $i \in N$, ${V^c}_{mem}(s,i) = \Gamma$ for some $\Gamma \in L_i$. Hence, $|\Gamma| \leq n_{mem}(i)$.

Similarly, for any $s \in S^c$ and $i \in N$, ${V^c}_{com}(s,i) = \{c_i^n\}$ for some $i \in \{0, \ldots, n_{com}(i)\}$. Obviously, we have $|\{c_i^n\}| = 1$.

□

In the following, we prove that the satisfiability of a formula in BMCM(K, n_{mem}, n_{com}) can be determined by only checking the satisfiability of the formula on the canonical model *CM*.

Lemma 3. *A formula is satisfiable in BMCM(K, n_{mem}, n_{com}) iff it is satisfied by the canonical model CM.*

Proof. Let φ be a formula, $M = (S, R, V)$ a modal in BMCM(K, n_{mem}, n_{com}). For any state s of M, assume that $V_{mem}(s,i) = \Gamma_i^s$ and $V_{com}(s,i) = \{c_i^{n_i^s}\}$ for all $i \in N$, we define $s^c = (\Gamma_i^s, n_i^s)_{i \in N} \in S^c$ and say that s^c corresponds to s. To prove the lemma, we show by induction on the structure of φ that for any s of M, we have $M, s \models \varphi$ iff $CM, s^c \models \varphi$.

- If $\varphi = B_i\alpha$, we have

$$
\begin{aligned}
M, s \models B_i\alpha &\Leftrightarrow \alpha \in V_{mem}(s,i) = \Gamma_i^s = V_{mem}^c(s^c, i)\\
&\Leftrightarrow \alpha \in V_{mem}^c(s^c, i)\\
&\Leftrightarrow CM, s^c \models B_i\alpha
\end{aligned}
$$

- If $\varphi = c_i^k$, we have

$$
\begin{aligned}
M, s \models c_i^k &\Leftrightarrow c_i^k \in V_{com}(s,i) = \{c_i^k\} = V_{com}^c(s^c, i)\\
&\Leftrightarrow c_i^k \in V_{com}^c(s^c, i)\\
&\Leftrightarrow CM, s^c \models c_i^k
\end{aligned}
$$

- If $\varphi = \neg\psi$, we have

$$
\begin{aligned}
M, s \models \neg\psi &\Leftrightarrow M, s \not\models \psi\\
&\Leftrightarrow CM, s^c \not\models \psi \text{ by the induction hypothesis}\\
&\Leftrightarrow CM, s^c \models \neg\psi
\end{aligned}
$$

- If $\varphi = \varphi_1 \vee \varphi_2$, we have

$$
\begin{aligned}
M, s \models \varphi_1 \vee \varphi_2 &\Leftrightarrow M, s \models \varphi_1 \text{ or } M, s \models \varphi_2\\
&\Leftrightarrow CM, s^c \models \varphi_1 \text{ or } CM, s^c \models \varphi_2 \text{ by the induction hypothesis}\\
&\Leftrightarrow CM, s^c \models \varphi_1 \vee \varphi_2
\end{aligned}
$$

- If $\varphi = \mathrm{EX}\psi$, we have

$$
\begin{aligned}
M, s \models \mathrm{EX}\psi &\Leftrightarrow \exists t \in S : (s,t) \in R \text{ and } M, t \models \psi\\
&\Leftrightarrow t^c \in CM \text{ and } (s^c, t^c) \in R^c \text{ and } CM, t^c \models \psi\\
&\qquad\qquad \text{by the definition of } CM \text{ and the induction hypothesis}\\
&\Rightarrow CM, s^c \models \mathrm{EX}\psi
\end{aligned}
$$

For the reverse direction, let us assume that $\mathit{CM}, s^c \vDash \mathrm{EX}\psi$, this means $\exists t^c \in S^c : (s^c, t^c) \in R^c$ and $\mathit{CM}, t^c \vDash \psi$. By the definition of *CM*, there are $a_i \in \mathit{Act}_{s^c,i}$ such that $(s^c, t^c) \in R^c_{a_i}$ for all $i \in N$. However, this also implies that $a_i \in \mathit{Act}_{s,i}$ for all $i \in N$ and there exists $t \in S$ such that $(s,t) \in R_{a_i}$ for all $i \in N$ (hence, $(s,t) \in R$) and t^c corresponds to t. Since $\mathit{CM}, t^c \vDash \psi$, by the induction hypothesis, we have that $M, t \vDash \psi$. Therefore, $M, s \vDash \mathrm{EX}\psi$.

- If $\varphi = \mathrm{E}(\varphi_1 \mathcal{U} \varphi_2)$, we have

$$
\begin{aligned}
M, s \vDash \mathrm{E}(\varphi_1\mathcal{U}\varphi_2) &\Leftrightarrow \exists \text{ a path } (s_0, \ldots, s_k, \ldots) \text{ of } M \text{ where} \\
&\qquad s_0 = s, M, s_k \vDash \varphi_2 \text{ and } M, s_j \vDash \varphi_1 \texttt{ for all } 0 \le j < k \\
&\Leftrightarrow (s^c_0, \ldots, s^c_k, \ldots) \text{ is a path of } \mathit{CM} \text{ with} \\
&\qquad s^c_0 = s^c, \mathit{CM}, s^c_k \vDash \varphi_2 \text{ and } \mathit{CM}, s^c_j \vDash \varphi_1 \texttt{ for all } 0 \le j < k \\
&\qquad \text{by the definition of } \mathit{CM} \text{ and the induction hypothesis} \\
&\Rightarrow \mathit{CM}, s^c \vDash \mathrm{E}(\varphi_1\mathcal{U}\varphi_2)
\end{aligned}
$$

For the reverse direction, let us assume that $\mathit{CM}, s^c \vDash \mathrm{E}(\varphi_1\mathcal{U}\varphi_2)$, this means there exists a path $(s^c_0, \ldots, s^c_k)$ of *CM* where $s^c_0 = s^c$, $(s^c_{j-1}, s^c_j) \in R^c$ for all $0 < j \le k$, $\mathit{CM}, s^c_j \vDash \varphi_1$ for all $j < k$ and $\mathit{CM}, s^c_k \vDash \varphi_2$. By the definition of *CM*, for each s^c_j (where $j < k$), there are $a_{j,i} \in \mathit{Act}_{s^c_j,i}$ such that $(s^c_j, s^c_{j+1}) \in R^c_{a_{j,i}}$ for all $i \in N$. However, this also implies that $a_{0,i} \in \mathit{Act}_{s_0,i}$ (where $s_0 = s$) for all $i \in N$ and there exists $s_1 \in S$ such that $(s_0, s_1) \in R_{a_{0,i}}$ for all $i \in N$ (hence, $(s_0, s_1) \in R$) and s^c_1 corresponds to s_1. Similarly, we prove that there are also $s_2, \ldots, s_k$ where $(s_{j-1}, s_j) \in R_{a_{j-1,i}}$ for all $i \in N$ (hence, $(s_{j-1}, s_j) \in R$) and s^c_j corresponds to s_j, for all $2 \le j \le k$. Since $\mathit{CM}, s^c_j \vDash \varphi_1$ for all $j < k$ and $\mathit{CM}, s^c_k \vDash \varphi_2$, by the induction hypothesis, we have that $\mathit{CM}, s_j \vDash \varphi_1$ for all $j < k$ and $\mathit{CM}, s_k \vDash \varphi_2$. Therefore, $M, s \vDash \mathrm{E}(\varphi_1\mathcal{U}\varphi_2)$.

- If $\varphi = \mathrm{A}(\varphi_1\mathcal{U}\varphi_2)$, $M, s \vDash \mathrm{A}(\varphi_1\mathcal{U}\varphi_2)$ iff for any path $(s_0, s_1, \ldots)$ starting from s (i.e. $s_0 = s$), there exists $k \ge 0$ such that $M, s_k \vDash \varphi_2$ and $M, s_j \vDash \varphi_1$ for all $0 \le j < k$.

  We consider an arbitrary path $(s^c_0, s^c_1, \ldots)$ in *CM* starting from s^c (i.e. $s^c_0 = s^c$). Since $(s^c_0, s^c_1) \in R^c$, there must be a state s_1 of M such that $(s_0, s_1) \in R$ and s^c_1 corresponds to s_1 according to the definition of $\mathrm{BMCM}(K, n_{mem}, n_{com})$. Repeating the same argument, we have that for all $j > 0$, there exists $s_j \in S$ such that $(s_{j-1}, s_j) \in R$ and s^c_j corresponds to s_j. Figure 3.3 illustrates that there exists a path $(s_0, s_1, \ldots)$ in M corresponding to the path $(s^c_0, s^c_1, \ldots)$ in *CM*. Then, $(s_0, s_1, \ldots)$ where $s_0 = s$ is a path in M, therefore we have hat

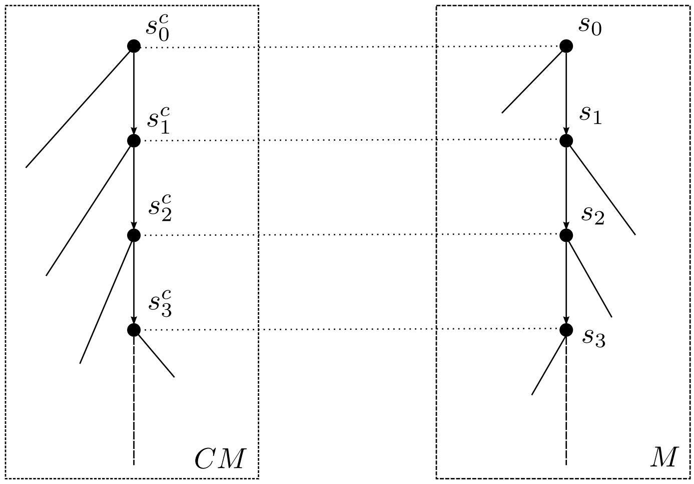

FIGURE 3.3: Each path is *CM* has a corresponding path in M.

there exists $k \geq 0$ such that $M, s_k \vDash \varphi_2$ and $M, s_j \vDash \varphi_1$ for all $0 \leq j < k$. By the induction hypothesis, we also have *CM*, $s_k^c \vDash \varphi_2$ and *CM*, $s_j^c \vDash \varphi_1$ for all $0 \leq j < k$. As the path $(s_0^c, s_1^c, \ldots)$ is arbitrary, we conclude that *CM*, $s_0^c \vDash \mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$.

For the reverse direction, let us assume that *CM*, $s^c \vDash \mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$, this means for any path $(s_0^c, s_1^c, \ldots)$ of *CM* where $s_0^c = s^c$, $(s_{j-1}^c, s_j^c) \in R^c$ for all $0 < j \leq k$, there exists $k \geq 0$ such that *CM*, $s_j^c \vDash \varphi_1$ for all $j < k$ and *CM*, $s_k^c \vDash \varphi_2$.

Consider an arbitrary path $(s_0, s_1, \ldots)$ in M starting from $s_0 = s$. Since $(s_j, s_{j+1}) \in R$, we much have s_j^c corresponds to s_j and $(s_j^c, s_{j+1}^c) \in R^c$ for all $j \geq 0$. Then, $(s_0^c, s_1^c, \ldots)$ is a path of *CM*. Since we have that *CM*, $s_j^c \vDash \varphi_1$ for all $j < k$ and *CM*, $s_k^c \vDash \varphi_2$ for some $k \geq 0$, by the induction hypothesis, we have that *CM*, $s_j \vDash \varphi_1$ for all $j < k$ and *CM*, $s_k \vDash \varphi_2$. Therefore, $M, s \vDash \mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$.

Since we have shown that $M, s \vDash \varphi$ iff *CM*, $s^c \vDash \varphi$, the proof of the lemma is straightforward.

□

The proof of the above lemma shows that there is a bisimulation [Blackburn et al., 2002, ch2.] between the canonical model *CM* and a model in BMCM(K, n_{mem}, n_{com}). Lemma 3 suggests us to use a model-checking algorithm for CTL to determine the satisfiability in the class BMCM(K, n_{mem}, n_{com}) where the canonical model and the candidate formula are inputs of the algorithm.

3.4.2 Checking satisfiability on *CM*

[Katoen, 1998] presented a model-checking procedure for CTL which decides whether a formula φ_0 is satisfied by a certain model. In principle, the procedure calculates the set of states satisfying the formula and replies *yes* if and only if that set is non-empty. For the satisfiability problem of BMCL, we utilise the model-checking algorithm procedure for CTL to determine the satisfiability of a formula on the canonical model *CM*.

In order to make this section to be self-contained, we recall the decision procedure for the model checking problem. For the correctness of this procedure, please refer to [Katoen, 1998]. Because we use the algorithm to check the satisfiability of φ only on the canonical model *CM*, *CM* is placed directly into the procedure rather than be used as the input of the procedure. In the following, we divide the algorithm into three sub-procedures. The *Sat* procedure is responsible for calculating the set of states in *CM* which satisfy the input formula. When the input formula is primitive, the procedure simply looks for states in *CM* where their valuation contains the input formula. When the input formula is composed by logical operators such as negation ($\neg$) and disjunction ($\vee$), the procedure calculates the set of state satisfying the sub-formulas of the input formula and uses the corresponding set operators to produce the result. The last two cases for formulas of the forms $\mathrm{E}(\varphi_1\mathcal{U}\varphi_2)$ and $\mathrm{A}(\varphi_1\mathcal{U}\varphi_2)$ are dealt with by using two sub-procedures Sat_{EU} and Sat_{AU}, respectively.

The procedure Sat which calculates the set of states in S^c satisfying an input formula φ is defined in Figure 3.4.

In the procedure *Sat*, the last two cases corresponding to formulas of the forms $\mathrm{E}(\varphi_1\mathcal{U}\varphi_2)$ and $\mathrm{A}(\varphi_1\mathcal{U}\varphi_2)$, respectively, are dealt with the help of two sub-procedures Sat_{EU} and Sat_{AU}. The procedure Sat_{EU} computes the set of states where φ_2 is satisfied and other states each of which is connected with one of the states satisfying φ_2 by a path along which φ_1 is satisfied before reaching the state satisfying φ_2. Figure 3.5 illustrates the procedure Sat_{EU}.

The procedure Sat_{AU} operates by collecting all states satisfying φ_2 and others where φ_1 is satisfied along any path starting from them satisfy before reaching a state satisfying φ_2. Sat_{AU} is presented in Figure 3.6.

As the procedure *Sat* calculates the set of states satisfying the input formula φ, we can determine the satisfiability of φ by checking whether the output of the procedure *Sat* is not empty.

input : A formula φ

output: The set of states in S^c satisfying φ

begin

  switch φ **do**

    case $\top$ return S^c;

    case $B_i\alpha$ return $\{s^c \in S^c \mid \alpha \in V^c_{mem}(s^c, i)\}$;

    case c^n_i return $\{s^c \in S^c \mid c^n_i \in V^c_{com}(s^c, i)\}$;

    case $\neg\psi$ return $S^c \setminus \mathit{Sat}(\psi)$;

    case $\varphi_1 \vee \varphi_2$ return $\mathit{Sat}(\varphi_1) \cup \mathit{Sat}(\varphi_2)$;

    case $\mathit{EX}\psi$ return $\{s^c \in S^c \mid \exists t^c \in \mathit{Sat}(\psi) : (s^c, t^c) \in S^c\}$;

    case $E(\varphi_1 \mathcal{U} \varphi_2)$ return $\mathit{Sat}_{EU}(\varphi_1, \varphi_2)$;

    case $A(\varphi_1 \mathcal{U} \varphi_2)$ return $\mathit{Sat}_{AU}(\varphi_1, \varphi_2)$;

  endsw

end

FIGURE 3.4: The procedure *Sat*.

If it is not empty, φ is satisfied in the canonical model *CM*; by Lemma 3, φ is satisfiable in the class $\mathrm{BMCM}(K, n_{mem}, n_{com})$. Before ending this section, let us discuss how difficult it is to solve the satisfiability problem of BMCL. Using the more efficient procedure presented in [Clarke et al., 1986], it is well-known that the time complexity of the model-checking problem for CTL is $O(|\varphi|.(|S|+|R|))$ where φ is the input formula and S and R are the set of states and the relation of the input model, respectively. Therefore, we determine roughly the size of the canonical model in order to provide an upper-bound for the time complexity of the satisfiability problem of BMCL. For simplicity, let us assume that there are a fixed number n of agents, the size of the logical language is a fixed number l, all agents have the same knowledge base of fixed size r. Moreover, we also assume that agents share the same bounds m for memory and c for communication. Then, the cardinality of the set of states in the canonical model is

$$(\sum_{k \leq m} \binom{k}{l} \times c)^n$$

Roughly, this number is bounded by $c^n \times (l^n)^m$. To estimate an upper-bound for the size of the relation of the canonical models, we determine the maximal number of actions available for each agent at every state. For reading a formula from the knowledge base, there are maximally r possible

input : Two formulas φ_1 and φ_2
output: The set of states in S^c satisfying $\mathrm{E}(\varphi_1 \mathcal{U} \varphi_2)$

begin
  $Q := Sat(\varphi_2)$;
  $Q' := \varnothing$; **while** $Q \neq Q'$ **do**
    $Q' := Q$;
    $Q := Q \cup (\{s^c \in S^c \mid \exists t^c \in Q : (s^c, t^c) \in R^c\} \cap Sat(\varphi_1))$;
  end
  return Q;
end

FIGURE 3.5: The procedure Sat_{EU}.

formulas to read. For applying inference rules, each subset of the formulas in the internal memory of the agent may trigger an inference rule (assume that for any inference rule applied on each subset, there is only one possible conclusion), hence there are maximally 2^m possible ways to apply inference rules. Finally, for the action Copy, there are at most $(n-1)m$ different formulas which can be copied from the internal memories of other agents, where n is the number of agents in the system, which is assumed to be a fixed number. In total, there are maximally $(r + 2^m + (n-1) \times m)$ different actions. However, we also need to take into account the fact each action also needs to choose which formula in the memory to be deleted to reserve the space for the new coming formula; therefore, the actual number of actions available for an agent is limited by $(r + 2^m + (n-1) \times m) \times m$. Then, from a state, there are maximally $((r + 2^m + (n-1) \times m) \times m)^n$ out-going transitions. Roughly, we may estimate the upper-bound for the number of transitions in the canonical model as $c^n \times (l^n)^m \times ((r+2^m+(n-1) \times m) \times m)^n$. As we assume that l, r and n are fixed numbers, this upper-bound can be written as $O((c \times m)^n \times ((2l)^n)^m)$. Overall, we set up an approximate upper-bound for the time complexity of the satisfiability problem of BMCL as $O(|\varphi| \times (c \times m)^n \times ((2l)^n)^m)$.

3.5 Axiomatisation for BMCL

In this section, we present a deductive system which allows us to reason about systems of multiple reasoning agents. In a given system of n agents, the knowledge base, bounds on memory and communication are characterised by the functions K, n_{mem} and n_{com}, respectively. We define a logic

input : Two formulas φ_1 and φ_2

output: The set of states in S^c satisfying $\mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$

begin

  $Q := Sat(\varphi_2)$;

  $Q' := \varnothing$;

  while $Q \neq Q'$ **do**

    $Q' := Q$;

    $Q := Q \cup (\{s^c \in S^c \mid \forall t^c \in S^c : (s^c, t^c) \in R^c \text{ implies } t^c \in Q\} \cap Sat(\varphi_1))$;

  end

  return Q;

end

FIGURE 3.6: The procedure Sat_{AU}.

BMCL(K, n_{mem}, n_{com}) which allows reasoning about the above system. Because BMCL is based on CTL, the axiomatisation for BMCL(K, n_{mem}, n_{com}) contains all axioms and inference rules of CTL. Moreover, it also contains axioms which correspond to properties of BMCM(K, n_{mem}, n_{com})

3.5.1 Axioms

Let us introduce some notations. For each state $s^c = (\Gamma_i, n_i)_{i \in N} \in S^c$ in the canonical model *CM* of the class BMCM(K, n_{mem}, n_{com}), we denote

$$\bigwedge s^c \equiv \bigwedge_{i \in N} \Big(\bigwedge_{\alpha \in \Gamma_i} B_i \alpha \wedge \bigwedge_{\alpha \in L \setminus \Gamma_i} \neg B_i \alpha \wedge c_i^{n_i} \Big)$$

This means the formula $\bigwedge s^c$ is the conjunction of all primitive formulas in $V^c(s^c)$. We write $R^c(s^c) = \{t^c \in S^c \mid (s^c, t^c) \in R^c\}$ to denote the set of all states related to s^c by the relation R^c. The axiomatisation for BMCL(K, n_{mem}, n_{com}) contains all axioms and inference rules of CTL (see Section 2.2.3), together with the following additional axioms:

A1 $\bigwedge_{\alpha \in \Gamma} B_i \alpha \to \neg B_i \alpha'$ for all $i \in N$, $\Gamma \subseteq L$ and $|\Gamma| > n_{mem}(i)$, and $\alpha' \in L$.

A2a $\bigvee_{n \in \{0, \ldots, n_{com}(i)\}} c_i^n$ for all $i \in N$.

A2b $c_i^k \to \neg c_i^{k'}$ for all $i \in N$, $k, k' \in \{0, \ldots, n_C(i)\}$ and $k \neq k'$.

A3a $\bigwedge s^c \rightarrow \text{EX}(\bigwedge t^c)$ for every $s^c, t^c \in S^c$ and $(s^c, t^c) \in R^c$.

A3b $\bigwedge s^c \rightarrow \text{AX}(\bigvee_{t^c \in R^c(s^c)}(\bigwedge t^c))$ for every $s^c \in S^c$.

It is worth noticing that the additional axioms correspond to exactly the requirements of models in the class BMCM(K, n_{mem}, n_{com}) and they will facilitate showing the soundness and completeness of the axiomatisation. In particular, the axiom **A1** makes sure that any agent in the system can hold in its internal memory maximally $n_{mem}(i)$ formulas. The axiom **A2a** states that there is a message counter for each agent while the axiom **A2b** says that if there is a message counter for an agent, the agent must not have another message counter. Hence, the axioms **A2a** and **A2b** together make sure that there is exactly one message counter for each agent. The axiom **A3a** corresponds to the requirement of models in the class BMCM(K, n_{mem}, n_{com}) that for every joint action for the agents in the system, there is move by the system to another state by performing this action. Similarly, the axiom **A3b** is about the requirement of models in the class BMCM(K, n_{mem}, n_{com}) that for from a state, the system can move to another state only by performing a joint action which is available for the agents.

As usual, we define that a formula φ is a theorem iff it can be proved in the axiomatisation, written as $\vdash_{\text{BMCL}} \varphi$. The logic BMCL$(K, n_{mem}, n_{com})$ is defined to be the set of all theorems in BMCL(K, n_{mem}, n_{com}). Moreover, φ is consistent iff its negation is not proved, i.e. $\nvdash_{\text{BMCL}} \neg\varphi$. We have the following result.

Theorem 1. *The logic BMCL(K, n_{mem}, n_{com}) is sound and complete with respect to the class BMCM(K, n_{mem}, n_{com}).*

In the next of this section, we present the proof of Theorem 1.

3.5.2 The Soundness and Completeness of BMCL

We show the soundness of BMCL(K, n_{mem}, n_{com}) by proving that all the additional axioms are valid in the class BMCM(K, n_{mem}, n_{com}). According to Lemma 3, a formula is satisfiable in BMCM(K, n_{mem}, n_{com}) iff it is satisfied in the canonical model *CM*. This is similar to say that a formula is valid in BMCM(K, n_{mem}, n_{com}) iff it is valid in the canonical model *CM*. Therefore, we show that all additional axioms are valid by using the canonical model. In the following, we show the validity of each additional axiom.

Let us consider the formula $\varphi = \bigwedge_{\alpha \in \Gamma} B_i\alpha \rightarrow \neg B_i\alpha'$ where $i \in N$ and $\alpha' \in L$, for some $\Gamma \subseteq L$ such that $|\Gamma| > n_{mem}(i)$. Let s^c be an arbitrary state in S^c, as $|V^c_{mem}(s^c, i)| \leq n_{mem}(i)$, there exists $\alpha \in \Gamma$ such that $\alpha \notin V^c_{mem}(s^c, i)$. Hence, we have that $CM, s^c \nvDash B_i\alpha$. This implies $CM, s^c \nvDash \bigwedge_{\alpha\in\Gamma} B_i\alpha$, hence $CM, s^c \vDash \varphi$. This means φ is true in any state of *CM*, i.e. $\bigwedge_{\alpha\in\Gamma} B_i\alpha \to \neg B_i\alpha'$ is valid in *CM*.

Let us consider the formula $\varphi = \bigvee_{k\in\{0,\dots,n_{com}(i)\}} c_i^k$ where $i \in N$. Let s^c be an arbitrary state in S^c, as $V^c_{com}(s^c, i) = \{c_i^k\}$ for some $k \in \{0, \dots, n_{com}(i)\}$, $CM, s^c \vDash c_i^k$, hence $CM, s^c \vDash \bigvee_{k\in\{0,\dots,n_{com}(i)\}} c_i^k$. This means φ is true in any state of *CM*, i.e. $\bigvee_{k\in\{0,\dots,n_{com}(i)\}} c_i^k$ is valid in *CM*.

Let us consider the formula $\varphi = c_i^k \to \neg c_i^{k'}$ where $i \in N$, $k, k' \in \{0, \dots, n_C(i)\}$ and $k \neq k'$. Let s^c be an arbitrary state in S^c, if $CM, s^c \vDash c_i^k$, we have that $V^c_{com}(s^c, i) = \{c_i^k\}$. Hence, $c_i^{k'} \notin V^c_{com}(s^c, i)$, which implies $CM, s^c \vDash \neg c_i^{k'}$. Therefore, $CM, s^c \vDash c_i^k \to \neg c_i^{k'}$. This means φ is true in any state of *CM*, i.e. $c_i^k \to \neg c_i^{k'}$ is valid in *CM*.

Let us consider the formula $\varphi = \bigwedge s^c \to \mathrm{EX}(\bigwedge t^c)$ where s^c, $t^c \in S^c$ and $(s^c, t^c) \in R^c$. Since only at state s^c, $CM, s^c \vDash \bigwedge s^c$, $CM, u^c \vDash \bigwedge s^c \to \mathrm{EX}(\bigwedge t^c)$ for all $u^c \neq s^c$. In the case of s^c, as $(s^c, t^c) \in R^c$ and we have that $CM, t^c \vDash \bigwedge t^c$, this implies $CM, s^c \vDash \bigwedge \mathrm{EX} t^c$. Therefore, $CM, s^c \vDash \bigwedge s^c \to \mathrm{EX}(\bigwedge t^c)$. This means φ is true in any state of *CM*, i.e. $\bigwedge s^c \to \mathrm{EX}(\bigwedge t^c)$ is valid in *CM*.

Let us consider the formula $\varphi = \bigwedge s^c \to \mathrm{AX}(\bigvee_{t^c\in R^c(s^c)}(\bigwedge t^c))$ where $s^c \in S^c$. Since only at state s^c, $CM, s^c \vDash \bigwedge s^c$, $CM, u^c \vDash \bigwedge s^c \to \mathrm{AX}(\bigvee_{t^c\in R^c(s^c)}(\bigwedge t^c))$ for all $u^c \neq s^c$. In the case of s^c, for each $u^c \in R^c(s^c)$, we have that $(s^c, u^c) \in R^c$ and $CM, u^c \vDash \bigwedge u^c$, this implies $CM, u^c \vDash \bigwedge \bigvee_{t^c\in R^c(s^c)}(\bigwedge t^c)$, hence $CM, s^c \vDash \bigwedge \mathrm{AX}(\bigvee_{t^c\in R^c(s^c)}(\bigwedge t^c))$. Therefore, $CM, s^c \vDash \bigwedge s^c \to \mathrm{AX}(\bigvee_{t^c\in R^c(s^c)}(\bigwedge t^c))$. This means φ is true in any state of *CM*, i.e. $\bigwedge s^c \to \mathrm{AX}(\bigvee_{t^c\in R^c(s^c)}(\bigwedge t^c))$ is valid in *CM*.

We have shown the validity of all additional axioms for BMCL(K, n_{mem}, n_{com}).

In the following, we prove the completeness of the axiomatisation. The usual approach to address the completeness (such as CTL [Emerson, 1990]) is to prove that any consistent formula is satisfiable. However, as shown by Lemma 3, a formula is satisfiable if and only if it is satisfiable in the canonical model *CM*. This suggests to us to show that any consistent formula is satisfiable in the canonical model *CM* rather than build a model in the class BMCL(K, n_{mem}, n_{com}) to satisfy it.

In order to show that a consistent formula of BMCL(K, n_{mem}, n_{com}) is satisfiable in the canonical model, we make use of the following result.

Lemma 4. *For any formula φ and a state $s^c \in S^c$, either $\bigwedge s^c \to \varphi$ or $\bigwedge s^c \to \neg\varphi$ is valid in the class BMCM(K, n_{mem}, n_{com}).*

Proof. Let us prove this lemma by using the canonical model *CM*. Notice that, to show a formula φ

is valid in the class BMCM(K, n_{mem}, n_{com}), we only need to prove that φ is valid in the canonical model *CM*.

Obviously, $CM, t^c \nvDash s^c$ for any $t^c \neq s^c$. Therefore, either $\bigwedge s^c \to \varphi$ or $\bigwedge s^c \to \neg\varphi$ is valid if and only if one of them is true at s^c.

Let us assume to the contrary that both formulas are not true at s^c. That is $CM, s^c \nvDash \bigwedge s^c \to \varphi$ and $CM, s^c \nvDash \bigwedge s^c \to \neg\varphi$. This implies that $CM, s^c \nvDash \neg\varphi$ and $CM, s^c \nvDash \varphi$, respectively, which is a contradiction. Therefore, either $\bigwedge s^c \to \varphi$ or $\bigwedge s^c \to \neg\varphi$ is true at s^c. Hence, either $\bigwedge s^c \to \varphi$ or $\bigwedge s^c \to \neg\varphi$ is valid in the canonical model *CM*. □

The above lemma implies that the conjunction of primitive formulas at a state of the canonical model contains enough information to conclude the truth of any formula, even those which contain temporal modalities such as AX, EX and other where the truth also depends on the truth of formulas at other states of the canonical model. This is not true in modal logic. For example, if we consider *Prop* = $\{p, q\}$ and a formula $\Diamond p$, then both formulas $p \wedge q \to \Diamond p$ and $p \wedge q \to \neg \Diamond p$ are not valid by models M_1 and M_2, respectively, as depicted in Figure 3.7. In particular, we have that $M_1, s \nvDash p \wedge q \to \Diamond p$ and $M_2, s \nvDash p \wedge q \to \neg \Diamond p$.

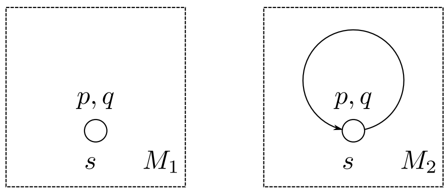

FIGURE 3.7: Models falsify $p \wedge q \to \Diamond p$ and $p \wedge q \to \neg \Diamond p$.

Furthermore, the validity of either $\bigwedge s^c \to \varphi$ or $\bigwedge s^c \to \neg\varphi$ suggests that either of them is a theorem of BMCL(K, n_{mem}, n_{com}) for any formula φ. If this can be proved, we have a consequence that any maximally consistent set of BMCL(K, n_{mem}, n_{com}) is determined by the primitive formulas in the set. This means if there are two maximally consistent sets which share the same set of primitive formulas, they must be the same. We show this by assuming to the contrary that this two sets Γ_1 and Γ_2 are not the same. That is there is a formula φ such that $\varphi \in \Gamma_1$ and $\neg\varphi \in \Gamma_2$. Because Γ_1 and Γ_2 share the same set of primitive formulas Γ, we have that either $\bigwedge_{\psi\in\Gamma} \psi \to \varphi$ or $\bigwedge_{\psi\in\Gamma} \psi \to \neg\varphi$ is a theorem. Without loss of generality, we assume that $\bigwedge_{\psi\in\Gamma} \psi \to \varphi$ is the case, which implies that $\varphi \in \Gamma_2$ which contradicts to the fact that Γ_2 is consistent. Therefore, Γ_1 and Γ_2 must be the same. Notice that the argument we have so far is based on the assumption that we can prove either $\bigwedge s^c \to \varphi$ or $\bigwedge s^c \to \neg\varphi$ is a theorem. If it is the case, we can select suitable states in the canonical model to satisfy a consistent formula which completes the proof of completeness for the logic BMCL(K, n_{mem}, n_{com}). Before showing that, let us prove our desired result.

Lemma 5. *For any formula φ and a state $s^c \in S^c$, either $\bigwedge s^c \to \varphi$ or $\bigwedge s^c \to \neg\varphi$ is a theorem of the logic BMCL(K, n_{mem}, n_{com}).*

Proof. The proof is done by the induction on the structure of the formula φ.

- Assume that φ is a primitive formula of the form $B_i\alpha$ where $i \in N$ and $\alpha \in L$. If $\alpha \in V^c_{mem}(s^c, i)$, then $B_i\alpha$ is one of the conjuncts in the formula $\bigwedge s^c$. Hence, by propositional tautologies, we have that $\vdash_{\text{BMCL}} \bigwedge s^c \to B_i\alpha$.

  Otherwise, $\alpha \notin V^c_{mem}(s^c, i)$, then $\neg B_i\alpha$ is one of the conjuncts in the formula $\bigwedge s^c$. Hence, by propositional tautologies, we have that $\vdash_{\text{BMCL}} \bigwedge s^c \to \neg B_i\alpha$.

- Assume that φ is a primitive formula of the form c_i^k where $i \in N$ and $k \in \{0, \ldots, n_{com}(i)\}$. If $c_i^k \in V^c_{com}(s^c, i)$, then c_i^k is one of the conjuncts in the formula $\bigwedge s^c$. Hence, by propositional tautologies, we have that $\vdash_{\text{BMCL}} \bigwedge s^c \to c_i^k$.

  Otherwise, $c_i^k \notin V^c_{com}(s^c, i)$, then $c_i^{k'}$ for some $k' \neq k$ is one of the conjuncts in the formula $\bigwedge s^c$. Hence, by axiom **A2b** and propositional tautologies, we have that $\vdash_{\text{BMCL}} \bigwedge s^c \to \neg c_i^k$.

- Assume that φ is of the form $\neg\psi$. By the induction hypothesis, we have that either $\bigwedge s^c \to \psi$ or $\bigwedge s^c \to \neg\psi$ is a theorem of the logic BMCL(K, n_{mem}, n_{com}). If $\bigwedge s^c \to \neg\psi$ is a theorem, the proof is immediate. If $\bigwedge s^c \to \psi$ is a theorem, then by propositional tautologies, we have that $\bigwedge s^c \to \neg\neg\psi$ is also a theorem.

- Assume that φ is of the form $\varphi_1 \vee \varphi_2$. By the induction hypothesis, we have that either $\bigwedge s^c \to \varphi_1$ or $\bigwedge s^c \to \neg\varphi_1$ and either $\bigwedge s^c \to \varphi_2$ or $\bigwedge s^c \to \neg\varphi_2$ are theorems of the logic BMCL(K, n_{mem}, n_{com}).

  If either $\bigwedge s^c \to \varphi_1$ or $\bigwedge s^c \to \varphi_2$ is a theorem, by propositional tautologies, we have that $\bigwedge s^c \to \varphi_1 \vee \varphi_2$ is a theorem.

  If both of $\bigwedge s^c \to \neg\varphi_1$ or $\bigwedge s^c \to \neg\varphi_2$ are theorems, by propositional tautologies, we have that $\bigwedge s^c \to \neg\varphi_1 \wedge \neg\varphi_2$ is a theorem. This implies that $\bigwedge s^c \to \neg(\varphi_1 \vee \varphi_2)$ is a theorem.

- Assume that φ is of the form EXψ. By the induction hypothesis, we have that either $\bigwedge t^c \to \psi$ or $\bigwedge t^c \to \neg\psi$ is a theorem where $t^c \in R^c(s^c)$.

If there exists $t^c \in R^c(s^c)$ such that $\bigwedge t^c \to \psi$ is a theorem, we have the following proof:

1. $\vdash_{\text{BMCL}} \bigwedge t^c \to \psi$
2. $\vdash_{\text{BMCL}} \text{EX} \bigwedge t^c \to \text{EX}\psi$ by 1 and CTL
3. $\vdash_{\text{BMCL}} \bigwedge s^c \to \text{EX} \bigwedge t^c$ by axiom **A3a**
4. $\vdash_{\text{BMCL}} \bigwedge s^c \to \text{EX}\psi$ by 2, 3 and propositional tautologies

If $\bigwedge t^c \to \neg\psi$ is a theorem for all $t^c \in R^c(s^c)$, we have the following proof:

1. $\vdash_{\text{BMCL}} \bigwedge t^c \to \neg\psi$ for all $t^c \in R^c(s^c)$
2. $\vdash_{\text{BMCL}} (\bigvee_{t^c \in R^c(s^c)} (\bigwedge t^c)) \to \neg\psi$ by 1 and propositional tautologies
3. $\vdash_{\text{BMCL}} \text{AX}(\bigvee_{t^c \in R^c(s^c)} (\bigwedge t^c)) \to \text{AX}\neg\psi$ by 2 and CTL
4. $\vdash_{\text{BMCL}} \bigwedge s^c \to \text{AX} \bigvee_{t^c \in R^c(s^c)} (\bigwedge t^c)$ by axiom **A3b**
5. $\vdash_{\text{BMCL}} \bigwedge s^c \to \text{AX}\neg\psi$ by 3, 4 and propositional tautologies
6. $\vdash_{\text{BMCL}} \bigwedge s^c \to \neg\text{EX}\psi$ by 5 and CTL

- Assume that φ is of the form $\text{E}(\varphi_1 \mathcal{U} \varphi_2)$. In order to prove that $\bigwedge s^c \to \text{E}(\varphi_1 \mathcal{U} \varphi_2)$ is a theorem, we attempt to construct a path which witnesses the fulfilment of $\text{E}(\varphi_1 \mathcal{U} \varphi_2)$ from s^c. We start the construction from s^c. By the induction hypothesis, we must know that either $\bigwedge s^c \to \varphi_1$ or $\bigwedge s^c \to \neg\varphi_1$ and either $\bigwedge s^c \to \varphi_2$ or $\bigwedge s^c \to \neg\varphi_2$ are theorems. If $\vdash_{\text{BMCL}} \bigwedge s^c \to \varphi_2$, s^c is enough to conclude that $\vdash_{\text{BMCL}} \bigwedge s^c \to \text{E}(\varphi_1 \mathcal{U} \varphi_2)$. Otherwise, for $\bigwedge s^c \to \text{E}(\varphi_1 \mathcal{U} \varphi_2)$ to be a theorem, we must at least have that $\bigwedge s^c \to \varphi_1$ is a theorem and pass the checking of the fulfilment for $\text{E}(\varphi_1 \mathcal{U} \varphi_2)$ on some successors of s^c (since we do not know which successor is the right choice, we try all of them). For each successor, we repeat the checking as we did with s^c. The construction terminates when either we reach a state t^c where $\bigwedge t^c \to \varphi_2$ is a theorem or there is no successor to consider.

  Let us consider an example of the construction as depicted in Figure 3.8. In this example, we draw a state s as a circle to assume that $\bigwedge s \to \varphi_1$ is a theorem and as a solid black circle to assume that $\bigwedge s \to \varphi_2$ is a theorem. States s where we assume both $\bigwedge s \to \neg\varphi_1$ and $\bigwedge s \to \neg\varphi_2$ are theorems are marked with the symbol "x". We start the construction at a state s_0. Since $\bigwedge s_0 \to \varphi_1$ is a theorem, we continue considering all successors of s_0 which are

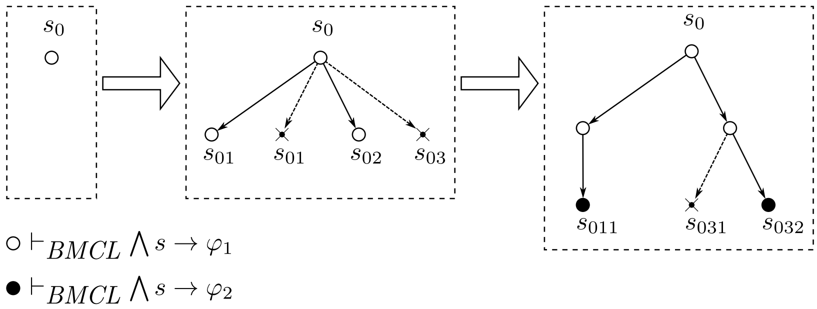

FIGURE 3.8: The construction of a path for the fulfilment of E$(\varphi_1 \mathcal{U} \varphi_2)$.

s_{01}, s_{02}, s_{03} and s_{04}. Because only $\bigwedge s_{01} \to \varphi_1$ and $\bigwedge s_{03} \to \varphi_1$ are theorems, we continue considering successors of s_{01} and s_{03} which are s_{011}, s_{031} and s_{032}. When we reach s_{032}, the construction terminates as $\bigwedge s_{032} \to \varphi_2$ is a theorem. Let us define the construction formally as follows. We say that a state $s^c \in S^c$ is potential for E$(\varphi_1 \mathcal{U} \varphi_2)$ if either $\vdash_{\text{BMCL}} \bigwedge s^c \to \varphi_1$ or $\vdash_{\text{BMCL}} \bigwedge s^c \to \varphi_2$. Given a sub-set of states $\Delta \subseteq S^c$, we denote

$$RE^c(\Delta) = \{t^c \in S^c \mid \exists u^c \in \Delta : \text{ and } t^c \in R^c(u^c)\}$$

Informally, $RE^c(\Delta)$ determines the set of successor states of some state in Δ according to R^c. Let us construct the set $RE^c_i(s^c)$ of states where $i \in \mathbb{N}$ incrementally as follows.

$$RE^c_0(s^c) = \begin{cases} \{s^c\} \text{ if } s^c \text{ is potential for E}(\varphi_1 \mathcal{U} \varphi_2) \\ \varnothing \text{ otherwise} \end{cases}$$

$$RE^c_{i+1}(s^c) = RE^c_i(s^c) \cup \{t^c \in S^c \setminus RE^c_i(s^c) \mid \begin{array}{c} t^c \in RE^c(RE^c_i(s^c)) \text{ and} \\ t^c \in S^c \text{ is potential for E}(\varphi_1 \mathcal{U} \varphi_2) \end{array} \}$$

At each step of the construction, we add into $RE^c_{i+1}(s^c)$ states which are potential to satisfy E$(\varphi_1 \mathcal{U} \varphi_2)$ and also related to some states in $RE^c_i(s^c)$. The construction terminates when either there exists a state $t^c \in RE^c_{i+1}(s^c)$ such that $\bigwedge t^c \to \varphi_2$ is a theorem or $RE^c_{i+1}(s^c) = RE^c_i(s^c)$ for some $i \geq 0$. Since S^c is finite, the construction must terminate at some $i = l \in \mathbb{N}$. We define $RE^c_*(s^c) = RE^c_l(s^c)$.

For convenience, we define a binary relation

$$E^c(RE^c_*(s^c)) = \{(t^c, u^c) \in R^c \mid \exists j < l : t^c \in RE^c_j(s^c) \text{ and } u^c \in RE^c_{j+1}(s^c) \setminus RE^c_j(s^c)\}$$

The binary relation $E^c(RE^c_*(s^c))$ is a subset of R^c over states in $RE^c_*(s^c)$ where loops are eliminated. This fact facilitates the following argument. According to the definition of

$RE^c_*(s^c)$, for any $t^c \in RE^c_*(s^c)$, there must be a path $s_0 s_1 \dots s_k$ in $E^c(RE^c_*(s^c))$ such that $s_0 = s^c$ and $s_k = t^c$ where $(s_j, s_{j+1}) \in E^c(RE^c_*(s^c))$ for all $j < k$.

If there exists $t^c \in RE^c_*(s^c)$ such that $\vdash_{\text{BMCL}} \bigwedge t^c \to \varphi_2$ is a theorem, there must be a path $s_0 s_1 \dots s_k$ in $E^c(RE^c_*(s^c))$ where $s_0 = s^c$ and $s_k = t^c$, and hence we have the following proof:

1. $\vdash_{\text{BMCL}} \bigwedge s_i \to \text{EX} \bigwedge s_{i+1}$ for all $i \leq k$, by axiom **A3a**
2. $\vdash_{\text{BMCL}} \bigwedge s_i \to \varphi_1$ for all $i < k$
3. $\vdash_{\text{BMCL}} \bigwedge s_k \to \varphi_2$
4. $\vdash_{\text{BMCL}} \bigwedge s_k \to \text{E}(\varphi_1 \mathcal{U} \varphi_2)$ by 3 and CTL
5. $\vdash_{\text{BMCL}} \text{EX} \bigwedge s_k \to \text{EXE}(\varphi_1 \mathcal{U} \varphi_2)$ by 4 and CTL
6. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \text{EX} \bigwedge s_k$ from 1
7. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \text{EXE}(\varphi_1 \mathcal{U} \varphi_2)$ from 5, 6 and propositional tautologies
8. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \varphi_1$ from 2
9. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \varphi_1 \wedge \text{EXE}(\varphi_1 \mathcal{U} \varphi_2)$ from 7, 8 and propositional tautologies
10. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \text{E}(\varphi_1 \mathcal{U} \varphi_2)$ from 9 and CTL

$\vdots$

11. $\vdash_{\text{BMCL}} \bigwedge s_0 \to \text{E}(\varphi_1 \mathcal{U} \varphi_2)$ by repeating 4,...,10 until s_0, notice that $s_0 = s^c$

Hence, $\bigwedge s^c \to \text{E}(\varphi_1 \mathcal{U} \varphi_2)$ is a theorem.

If, for all $t^c \in RE^c_*(s^c)$, $\vdash_{\text{BMCL}} \bigwedge t^c \to \neg\varphi_2$ is a theorem, according to the definition of $RE^c_*(s^c)$, we have that for every $t^c \in RE^c(RE^c_*(s^c)) \setminus RE^c_*(s^c)$, both $\vdash_{\text{BMCL}} \bigwedge t^c \to \neg\varphi_1$ and $\vdash_{\text{BMCL}} \bigwedge t^c \to \neg\varphi_2$ are theorems, otherwise t^c must be in $RE^c_*(s^c)$. We denote $\theta_1 \equiv \bigvee_{t^c \in RE^c_*(s^c)} \bigwedge t^c$ and $\theta_2 \equiv \bigvee_{t^c \in RE^c(RE^c_*(s^c)) \setminus RE^c_*(s^c)} \bigwedge t^c$. Let $\theta \equiv \theta_1 \vee \theta_2$, for any $t^c \in RE^c_*(s^c)$, we have that $R^c(t^c) \subseteq RE^c_*(s^c) \cup RE^c(RE^c_*(s^c))$, hence $\bigvee_{u^c \in R^c(t^c)} (\bigwedge u^c) \to \theta$ is a theorem. We have the following proof:

1. $\vdash_{\text{BMCL}} \bigwedge t^c \to \varphi_1$ for all $t^c \in RE^c_*(s^c)$
2. $\vdash_{\text{BMCL}} \bigwedge t^c \to \neg\varphi_2$ for all $t^c \in RE^c_*(s^c)$
3. $\vdash_{\text{BMCL}} \bigwedge t^c \to \neg\varphi_1$ for all $t^c \in RE^c(RE^c_*(s^c)) \setminus RE^c_*(s^c)$
4. $\vdash_{\text{BMCL}} \bigwedge t^c \to \neg\varphi_2$ for all $t^c \in RE^c(RE^c_*(s^c)) \setminus RE^c_*(s^c)$
5. $\vdash_{\text{BMCL}} \theta_1 \to \varphi_1$ by 1 and propositional tautologies

6. $\vdash_{\text{BMCL}} \theta_1 \to \neg\varphi_2$ by 2 and propositional tautologies
7. $\vdash_{\text{BMCL}} \theta_2 \to \neg\varphi_1$ by 3 and propositional tautologies
8. $\vdash_{\text{BMCL}} \theta_2 \to \neg\varphi_2$ by 4 and propositional tautologies
9. $\vdash_{\text{BMCL}} \theta \to \theta_1 \vee \theta_2$
10. $\vdash_{\text{BMCL}} (\theta \wedge \varphi_1) \to (\theta_1 \wedge \varphi_1) \vee (\theta_2 \wedge \varphi_1)$ by 9 and propositional tautologies
11. $\vdash_{\text{BMCL}} (\theta_2 \wedge \varphi_1) \to \bot$ by 7 and propositional tautologies
12. $\vdash_{\text{BMCL}} (\theta \wedge \varphi_1) \to (\theta_1 \wedge \varphi_1)$ by 10, 11 and propositional tautologies
13. $\vdash_{\text{BMCL}} (\theta \wedge \varphi_1) \to \theta_1$ by 12 and propositional tautologies
14. $\vdash_{\text{BMCL}} \bigwedge t^c \to \text{AX}(\bigvee_{u^c \in R^c(t^c)} (\bigwedge u^c))$ for all $t^c \in RE^c_*(s^c)$, by axiom **A3b**
15. $\vdash_{\text{BMCL}} \bigwedge t^c \to \text{AX}\theta$ for all $t^c \in RE^c_*(s^c)$, by 14, CTL and propositional tautologies
16. $\vdash_{\text{BMCL}} \theta_1 \to \text{AX}\theta$ by 15 and propositional tautologies
17. $\vdash_{\text{BMCL}} (\theta \wedge \varphi_1) \to \text{AX}\theta$ by 13, 16 and propositional tautologies
18. $\vdash_{\text{BMCL}} \theta \to (\varphi_1 \to \text{AX}\theta)$ by 17 and propositional tautologies
19. $\vdash_{\text{BMCL}} \theta \to \neg\varphi_2$ by 6, 8 and propositional tautologies
20. $\vdash_{\text{BMCL}} \theta \to (\neg\varphi_2 \wedge (\varphi_1 \to \text{AX}\theta))$ by 18,19 and propositional tautologies
21. $\vdash_{\text{BMCL}} \text{AG}(\theta \to (\neg\varphi_2 \wedge (\varphi_1 \to \text{AX}\theta)))$ by 20 and CTL
22. $\vdash_{\text{BMCL}} \theta \to \neg\text{E}(\varphi_1 \mathcal{U} \varphi_2)$ by 21 and CTL
23. $\vdash_{\text{BMCL}} \bigwedge s^c \to \theta$ by propositional tautologies
24. $\vdash_{\text{BMCL}} \bigwedge s^c \to \neg\text{E}(\varphi_1 \mathcal{U} \varphi_2)$ by 22, 23 propositional tautologies

Hence, $\bigwedge s^c \to \neg\text{E}(\varphi_1 \mathcal{U} \varphi_2)$ is a theorem.

- Assume that φ is of the form $\text{A}(\varphi_1 \mathcal{U} \varphi_2)$. In order to prove that $\bigwedge s^c \to \text{A}(\varphi_1 \mathcal{U} \varphi_2)$ is a theorem, we attempt to construct a sub-graph of the canonical model starting from s^c which contains the prefixes of any path starting from s^c in *CM*, and show that at all leaves s of the sub-graph, we have $\bigwedge s \to \varphi_2$ is a theorem and at other states t in the sub-graph, $\bigwedge t \to \varphi_1$ is a theorem. We start the construction from s^c, by the induction hypothesis, we must know that either $\bigwedge s^c \to \varphi_1$ or $\bigwedge s^c \to \neg\varphi_1$ and either $\bigwedge s^c \to \varphi_2$ or $\bigwedge s^c \to \neg\varphi_2$ are theorems. If $\bigwedge s^c \to \varphi_2$, it is enough to conclude that $\bigwedge s^c \to \text{A}(\varphi_1 \mathcal{U} \varphi_2)$. Otherwise, for $\bigwedge s^c \to \text{A}(\varphi_1 \mathcal{U} \varphi_2)$ to be a theorem, we must at least have that $\bigwedge s^c \to \varphi_1$ is a theorem and pass the checking of the fulfilment for $\mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ on all successors of s^c. For each successor, we repeat the checking as we did with s^c. The construction terminates when either we reach a state t^c where $\bigwedge t^c \rightarrow \neg\varphi_1$ and $\bigwedge t^c \rightarrow \neg\varphi_2$ are theorems or there is no successor to consider.

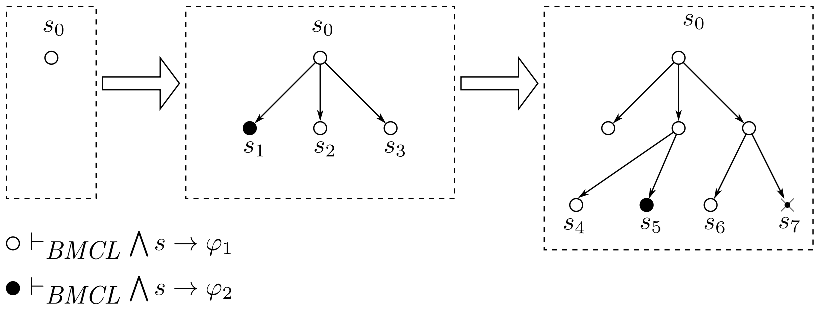

FIGURE 3.9: The construction of a sub-graph for the fulfilment of $\mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$.

Let us consider an example of the construction as depicted in Figure 3.9. Similar to the example in Figure 3.8, we also draw a state s as a circle to assume that $\bigwedge s \rightarrow \varphi_1$ is a theorem, and as a solid black circle to assume that $\bigwedge s \rightarrow \varphi_2$ is a theorem. States s where we assume both $\bigwedge s \rightarrow \neg\varphi_1$ and $\bigwedge s \rightarrow \neg\varphi_2$ are theorems are marked with the symbol "x". We start the construction at a state s_0. Since $\bigwedge s_0 \rightarrow \varphi_1$ is a theorem, we continue considering all successors of s_0 which are s_1, s_2 and s_3. Because only $\bigwedge s_1 \rightarrow \varphi_2$ is a theorem, we continue considering successors of s_2 and s_3 which are s_4, s_5, s_6 and s_7. When we reach s_7, the construction terminates because both $\bigwedge s_7 \rightarrow \neg\varphi_1$ and $\bigwedge s_7 \rightarrow \neg\varphi_2$ are theorems. In fact, s_7 can be used as a witness to prove that $\bigwedge s_0 \rightarrow \neg\mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ is a theorem.

We say that a state $s^c \in S^c$ is *potential* for $\mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ if either $\vdash_{\text{BMCL}} \bigwedge s^c \rightarrow \varphi_1$ or $\vdash_{\text{BMCL}} \bigwedge s^c \rightarrow \varphi_2$. Given a sub-set of states $\Delta \subseteq S^c$, we define

$$\Delta_{\varphi_1} = \{t^c \in \Delta \mid \vdash_{\text{BMCL}} \bigwedge t^c \rightarrow \varphi_1 \text{ and } \nvdash_{\text{BMCL}} \bigwedge t^c \rightarrow \varphi_2\}$$
$$\mathit{RA}^c(\Delta) = \{t^c \in S^c \mid \exists u^c \in \Delta_{\varphi_1} : t^c \in R^c(u^c)\}$$

The set $\mathit{RA}^c(\Delta)$ facilitates the construction by introducing candidate states at each step of the construction. We contruct the set $\mathit{RA}^c_i(s^c)$ of states where $i \in \mathbb{N}$ incrementally as follows.

$$\mathit{RA}^c_0(s^c) = \{s^c\}$$
$$\mathit{RA}^c_{i+1}(s^c) = \mathit{RA}^c_i(s^c) \cup \{t^c \in S^c \setminus \mathit{RA}^c_i(s^c) \mid t^c \in \mathit{RA}^c(\mathit{RA}^c_i(s^c))\}$$

The above construction terminates when either a state s which is not potential for for $\mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ is added into $RA_i^c(s^c)$ or $RA_{i+1}^c(s^c) = RA_i^c(s^c)$. Since S^c is finite, the construction must terminate at some step $i = l \in \mathbb{N}$. We define $RA_*^c(s^c) = RA_l^c(s^c)$.

We define a binary relation

$$A^c(RA_*^c(s^c)) = \{(t^c, u^c) \in R^c \mid t^c \in RA_*^c(s^c) \text{ and } u^c \in RA_*^c(s^c)\}$$

According to the definition of $RA_*^c(s^c)$, for any $t^c \in RA_*^c(s^c))$, there must be a path $s_0 s_1 \dots s_k$ in $RA^c(RA_*^c(s^c))$ such that $s_0 = s^c$ and $s_k = t^c$ where $(s_j, s_{j+1}) \in A^c(RA_*^c(s^c))$ for all $j < k$. The distance from s^c to t^c is the length of the longest path from s^c to t^c.

We say that a path $(s_0, \dots, s_k)$ in $RA_*^c(s^c)$ is a φ_1-cycle iff $s_i \in RA_*^c(s^c)$ for all $i \le k$, $(s_i, s_{i+1}) \in R^c$ for all $i < k$, $(s_k, s_0) \in R^c$, $\vdash_{\text{BMCL}} \bigwedge s_i \to \varphi_1$ for all $i \le k$ and $\vdash_{\text{BMCL}} \bigwedge s_i \to \neg\varphi_2$ for all $i \le k$.

If there is a φ_1-cycle $(t_0, \dots, t_k)$ in $RA_*^c(s^c)$, according to the construction of $RA_*^c(s^c)$, there must be a finite path $(s_0, \dots, s_l)$ for some $l \in \mathbb{N}$ where $s_0 = s^c$, $s_l = t_0$, $(s_i, s_{i+1}) \in R^c$ for all $i < l$, $\vdash_{\text{BMCL}} \bigwedge s_i \to \varphi_1$ for all $i \le l$ and $\vdash_{\text{BMCL}} \bigwedge s_i \to \neg\varphi_2$ for all $i \le l$. Let $\theta = \bigvee_{i \le l}(\bigwedge s_i) \vee \bigvee_{i \le k}(\bigwedge t_i)$. We have the following proof:

1. $\vdash_{\text{BMCL}} \bigwedge s_i \to \varphi_1$ for all $i \le l$
2. $\vdash_{\text{BMCL}} \bigwedge s_i \to \neg\varphi_2$ for all $i \le l$
3. $\vdash_{\text{BMCL}} \bigwedge t_i \to \varphi_1$ for all $i \le k$
4. $\vdash_{\text{BMCL}} \bigwedge t_i \to \neg\varphi_2$ for all $i \le k$
5. $\vdash_{\text{BMCL}} \theta \to \varphi_1$ by 1, 3 and propositional tautologies
6. $\vdash_{\text{BMCL}} \theta \to \neg\varphi_2$ by 2, 4 and propositional tautologies
7. $\vdash_{\text{BMCL}} \bigwedge s_i \to \mathrm{EX} \bigwedge s_{i+1}$ for all $i < l$, by axiom **A3a**
8. $\vdash_{\text{BMCL}} \bigwedge s_i \to \theta$ for all $i \le l$, by propositional tautologies
9. $\vdash_{\text{BMCL}} \mathrm{EX} \bigwedge s_i \to \mathrm{EX}\theta$ for all $i \le l$, by 8 and CTL
10. $\vdash_{\text{BMCL}} \bigwedge s_i \to \mathrm{EX}\theta$ for all $i < l$, by 7, 9 and propositional tautologies
11. $\vdash_{\text{BMCL}} \bigwedge t_i \to \mathrm{EX} \bigwedge t_{i+1}$ for all $i < k$, by axiom **A3a**
12. $\vdash_{\text{BMCL}} \bigwedge t_i \to \theta$ for all $i \le k$, by propositional tautologies
13. $\vdash_{\text{BMCL}} \mathrm{EX} \bigwedge t_i \to \mathrm{EX}\theta$ for all $i \le k$, by 12 and CTL
14. $\vdash_{\text{BMCL}} \bigwedge t_i \to \mathrm{EX}\theta$ for all $i < k$, by 13 and propositional tautologies

15. $\vdash_{\text{BMCL}} \bigwedge t_k \to \text{EX} \bigwedge t_0$ by axiom **A3a**
16. $\vdash_{\text{BMCL}} \bigwedge t_k \to \theta$ by propositional tautologies
17. $\vdash_{\text{BMCL}} \text{EX} \bigwedge t_k \to \text{EX}\theta$ by 16 and CTL
18. $\vdash_{\text{BMCL}} \bigwedge t_k \to \text{EX}\theta$ by 17 and propositional tautologies
19. $\vdash_{\text{BMCL}} \bigwedge \theta \to \text{EX}\theta$ by 10, 14, 18 and propositional tautologies
20. $\vdash_{\text{BMCL}} \bigwedge \theta \to (\varphi_1 \to \text{EX}\theta)$ by 19, 5 and propositional tautologies
21. $\vdash_{\text{BMCL}} \bigwedge \theta \to \neg\varphi_2 \wedge (\varphi_1 \to \text{EX}\theta)$ by 20, 6 and propositional tautologies
22. $\vdash_{\text{BMCL}} \bigwedge \theta \to \neg\text{A}(\varphi_1 \mathcal{U} \varphi_2)$ by 21 and CTL

If there is a state t^c in $RA^c_*(s^c)$ which is not potential for $\text{A}(\varphi_1 \mathcal{U} \varphi_2)$, there must be a path $s_0, \ldots, s_k$, where $s_0 = s^c$ and $s_k = t^c$, in $A^c(RA^c_*(s^c))$ such that for all $i < k$, $\bigwedge s_i \to \neg\varphi_2$ is a theorem (otherwise, t^c cannot be in $RA^c_*(s^c)$). We have the following proof:

1. $\vdash_{\text{BMCL}} \bigwedge s_k \to \neg\varphi_1$ since $t^c = s_k$ is not potential for $\text{A}(\varphi_1 \mathcal{U} \varphi_2)$
2. $\vdash_{\text{BMCL}} \bigwedge s_k \to \neg\varphi_2$ since $t^c = s_k$ is not potential for $\text{A}(\varphi_1 \mathcal{U} \varphi_2)$
3. $\vdash_{\text{BMCL}} \bigwedge s_k \to \neg\varphi_1 \vee \neg\text{AXA}(\varphi_1 \mathcal{U} \varphi_2)$ by 1 and propositional tautologies
4. $\vdash_{\text{BMCL}} \bigwedge s_k \to \neg\varphi_2 \wedge (\neg\varphi_1 \vee \neg\text{AXA}(\varphi_1 \mathcal{U} \varphi_2))$ by 2, 3 and propositional tautologies
5. $\vdash_{\text{BMCL}} \bigwedge s_k \to \neg\text{A}(\varphi_1 \mathcal{U} \varphi_2)$ by 4 and CTL
6. $\vdash_{\text{BMCL}} \bigwedge \text{EX}s_k \to \text{EX}\neg\text{A}(\varphi_1 \mathcal{U} \varphi_2)$ by 5 and CTL
7. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \text{EX} \bigwedge s_k$ by axiom **A3a**
8. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \text{EX}\neg\text{A}(\varphi_1 \mathcal{U} \varphi_2)$ by 6, 7 and propositional tautologies
9. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \neg\text{AXA}(\varphi_1 \mathcal{U} \varphi_2)$ by 8 and CTL
10. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \neg\varphi_1 \vee \neg\text{AXA}(\varphi_1 \mathcal{U} \varphi_2)$ by 10 and propositional tautologies
11. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \neg\varphi_2$
12. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \neg\varphi_2 \wedge (\neg\varphi_1 \vee \neg\text{AXA}(\varphi_1 \mathcal{U} \varphi_2))$ by 10, 11 and propositional tautologies
13. $\vdash_{\text{BMCL}} \bigwedge s_{k-1} \to \neg\text{A}(\varphi_1 \mathcal{U} \varphi_2)$ by 12 and CTL

$\vdots$

14. $\vdash_{\text{BMCL}} \bigwedge s_0 \to \neg\text{A}(\varphi_1 \mathcal{U} \varphi_2)$ by repeating 5,…,13 until s_0, notice that $s_0 = s^c$

Hence, $s^c \to \neg\text{A}(\varphi_1 \mathcal{U} \varphi_2)$ is a theorem.

Let us now assume that there is no φ_1-cycle and no state which is not potential for $\mathrm{A}(\varphi_1\mathcal{U}\varphi_2)$ in $RA^c_*(s^c)$. We first show that there must be a state $s \in RA^c_*(s^c)$ such that $\bigwedge s \to \varphi_2$ is a theorem. Assume to the contrary that there is no such state in $RA^c_*(s^c)$. This means for all states $s \in RA^c_*(s^c)$ we have that $\bigwedge s \to \varphi_1$ and $\bigwedge s \to \neg\varphi_2$ are theorems. Let us consider an arbitrary state $t^c \in RA^c_*(s^c)$, according to the construction of $RA^c_*(s^c)$, there must be a path $(s_0, \ldots, s_k)$ from s^c to t^c in $RA^c_*(s^c)$ where $s_0 = s^c$ and $s_k = t^c$. Since there is no φ_1-cycle in $RA^c_*(s^c)$, no state occurs twice in the path $(s_0, \ldots, s_k)$. Since s_k has at least one successor, according to the construction of $RA^c_*(s^c)$, the successor must be in $RA^c_*(s^c)$. However, this successor must not occur in the path $(s_0, \ldots, s_k)$ (otherwise, we have a φ_1-cycle). Let us call this successor as s_{k+1} and extend $(s_0, \ldots, s_k)$ to $(s_0, \ldots, s_{k+1})$. Then we repeat considering s_{k+1} as we did with s_k. Finally, we end up with a path $(s_0, \ldots, s_k, \ldots, s_l)$ which traverses every state in $RA^c_*(s^c)$ at most once. Again, we consider s_l which must have a successor. This successor must be in $RA^c_*(s^c)$, which implies that it is one of the states in the path $(s_0, \ldots, s_k, \ldots, s_l)$. Hence, we encounter a φ_1-cycle in $RA^c_*(s^c)$, which is a contradiction.

Therefore, if there is no φ_1-cycle and no state which is not potential for $\mathrm{A}(\varphi_1\mathcal{U}\varphi_2)$ in $RA^c_*(s^c)$, there must be a state $s \in RA^c_*(s^c)$ such that $\bigwedge s \to \varphi_2$ is a theorem. Given a state $s^c \in RA^c_*(s^c)$ where $\bigwedge s \to \varphi_2$ is a theorem, we define the distance from a state s^c to φ_2 is 0. Given a state $s^c \in RA^c_*(s^c)$ where $\bigwedge s \to \neg\varphi_2$ is a theorem, we define the distance from a state s^c to φ_2 is the length of the longest path from s^c to some first state $t^c \in RA^c_*(s^c)$ where $\bigwedge t^c \to \varphi_2$ is a theorem. Such a path must exist, otherwise, we can point out a φ_1-cycle in $RA^c_*(s^c)$. Furthermore, the length of the path is finite because there is no φ_1-cycle in $RA^c_*(s^c)$. In the following, we prove that, for every state $t^c \in RA^c_*(s^c)$, $\bigwedge t^c \to \mathrm{A}(\varphi_1\mathcal{U}\varphi_2)$ is a theorem by the induction on the distance from t^c to φ_2.

- In the base case, the distance from t^c to φ_2 is 0, we have the following proof:

  1. $\vdash_{\mathrm{BMCL}} \bigwedge t^c \to \varphi_2$
  2. $\vdash_{\mathrm{BMCL}} \bigwedge t^c \to \varphi_2 \vee (\varphi_1 \wedge \mathrm{AXA}(\varphi_1\mathcal{U}\varphi_2))$ by 1 and propositional tautologies
  3. $\vdash_{\mathrm{BMCL}} \bigwedge t^c \to \mathrm{A}(\varphi_1\mathcal{U}\varphi_2)$ by 2 and CTL

- In the induction step, the distance from t^c to φ_2 is greater than 0, we must have that

$$\vdash_{\mathrm{BMCL}} \bigwedge t^c \to \varphi_1 \tag{3.1}$$

$$\vdash_{\mathrm{BMCL}} \bigwedge t^c \to \neg\varphi_2 \tag{3.2}$$

According to the definition of $RA^c_*(s^c)$, for every $u^c \in R^c(t^c)$, we have $u^c \in RA^c_*(s^c)$. Of course, the distance from t^c to φ_2 is greater than that from any $u^c \in R^c(t^c)$. By the induction hypothesis, we have that for all $u^c \in R^c(t^c)$, $\bigwedge u^c \to \mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ is a theorem. We have the following proof:

1. $\vdash_{\text{BMCL}} \bigwedge u^c \to \mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ for all $u^c \in R^c(t^c)$
2. $\vdash_{\text{BMCL}} \bigvee_{u^c \in R^c(t^c)} \bigwedge u^c \to \mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ by 1 and propositional tautologies
3. $\vdash_{\text{BMCL}} \mathrm{AX}(\bigvee_{u^c \in R^c(t^c)} \bigwedge u^c) \to \mathrm{AXA}(\varphi_1 \mathcal{U} \varphi_2)$ by 2 and CTL
4. $\vdash_{\text{BMCL}} \bigwedge t^c \to \mathrm{AX}(\bigvee_{u^c \in R^c(t^c)} \bigwedge u^c)$ by axiom **A3b**
5. $\vdash_{\text{BMCL}} \bigwedge t^c \to \mathrm{AXA}(\varphi_1 \mathcal{U} \varphi_2)$ by 3, 4 and propositional tautologies
6. $\vdash_{\text{BMCL}} \bigwedge t^c \to \varphi_1$ from (3.1)
7. $\vdash_{\text{BMCL}} \bigwedge t^c \to (\varphi_1 \wedge \mathrm{AXA}(\varphi_1 \mathcal{U} \varphi_2))$ by 5, 6 propositional tautologies
8. $\vdash_{\text{BMCL}} \bigwedge t^c \to \varphi_2 \vee (\varphi_1 \wedge \mathrm{AXA}(\varphi_1 \mathcal{U} \varphi_2))$ by 7 and propositional tautologies
9. $\vdash_{\text{BMCL}} \bigwedge t^c \to \mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ by 8 and CTL

Since $s^c \in RA^c_*(s^c)$, we have that $\bigwedge s^c \to \mathrm{A}(\varphi_1 \mathcal{U} \varphi_2)$ is a theorem.

□

The above lemma makes the proof of completeness for BMCL$(K, n_{mem}, , n_{mem})$ straightforward. Assume that φ is a consistent formula. This means there must be a maximally consistent set Γ which contains φ. Since Γ is maximally consistent, for each $i \in N$, there must be $\Gamma_i \in L_i$ such that $B_i\alpha \in \Gamma$ for all $\alpha \in \Gamma_i$ and $\neg B_i\alpha \in \Gamma$ for all $\alpha \in L \setminus \Gamma_i$. Moreover, for each $i \in N$, there must be also $n_i \in \{0, \ldots, n_{com}(i)\}$ such that $c_i^{n_i} \in \Gamma$ and $\neg c_i^k \in \Gamma$ for all $k \in \{0, \ldots, n_{com}(i)\}$ and $k \neq n_i$. Let $s^c = (\Gamma_i, n_i)_{i \in N}$, we have that $\bigwedge s^c \in \Gamma$.

We show that $\bigwedge s^c \to \varphi$ is a theorem. Assume to the contrary that $\bigwedge s^c \to \varphi$ is not a theorem, this implies by Lemma 5 that $\bigwedge s^c \to \neg\varphi$ is a theorem. Since $\bigwedge s^c \in \Gamma$ and Γ is maximally consistent, we have that $\neg\varphi \in \Gamma$ which contradicts the fact that Γ is consistent. Hence, $\bigwedge s^c \to \varphi$ is a theorem. Since BMCL$(K, n_{mem}, , n_{mem})$ is sound, $\bigwedge s^c \to \varphi$ is valid. Therefore, we have that $CM, s^c \models \bigwedge s^c$ and $CM, s^c \models \bigwedge s^c \to \varphi$. This implies $CM, s^c \models \varphi$ which means the consistent formula φ is satisfied in CM at s^c.

3.6 Conclusion

In this chapter, we have introduced logics BMCL for each system of multiple reasoning agents. Agents in such systems perform reasoning under bounds on memory and communication. Each agent is equipped with a number of inference rules which allow them to derive new information from what they have in the memory. For each system, we define a corresponding logic based on CTL for reasoning about properties of the system under bounds on memory and communication. The logic is specific for the characteristics of the system such as the number of agents in the system, the knowledge bases of agents in the system and the bounds on memory and communication for each agent in the system. The semantics is also defined by Kripke structures as in the case of CTL, however, we are only interested in the class of models which describes systems of multiple reasoning agents. In particular, the valuation at each state of such a model must comply with the condition of bounds on memory and communication and a transition in a model must correspond to a correct behaviour of the system.

We have investigated the satisfiability problem of BMCL where it was proved that the logic is decidable. Rather than follow the approach for CTL, we utilise the characteristic of models in the class describing systems of multiple reasoning agents so that an ordinary model-checking algorithm for CTL can be used for solving the satisfiability problem. We also researched the soundness and completeness of the deductive system for the logic. Once again, the characteristic of models in the class describing systems of multiple reasoning agents is used to provide the proof of the soundness and completeness.

Comparing to the result in [Alechina et al., 2006a], the extension from BML to BMCL in this chapter not only introduces a logic for reasoning about the abilities of systems where memory and communication are bounded but also provides what has been missing in [Alechina et al., 2006a] such as the proof for the soundness and the completeness of BML and the study of its satisfiability problem. Nonetheless, BMCL has certain drawbacks. Although BMCL is for reasoning about systems of multiple reasoning agents, it is not clear how to use it to describe arbitrary multi-agent systems where resources available to agents in order to operate are bounded. Furthermore, BMCL can only express properties of systems as a whole. For example, it is possible to formulate a property where all agents in a system can cooperate to obtain a goal. However, it is impossible to express another property where some agent or a sub-group of agents in the system has the power to produce a certain result regardless of other agents. In a logic of BMCL, the bounds on memory and communication are fixed. Changing the setting of such bounds will produce a different logic.

Moreover, the logic gets much complicated if we want to reason about multi-agent systems where more types of resources except from memory, communication and time are involved. In order to overcome such drawbacks, in the next two chapters, we investigate two logical languages which extend CL and ATL, respectively.

CHAPTER 4

RESOURCE-BOUNDED COALITION LOGIC

4.1 Introduction

In previous chapter, we have introduced the logics BMCL which are for reasoning about the abilities of systems of multiple reasoning agents under a certain amount of resources. BMCL extends the branching-time temporal logic CTL where bounds of memory and communication are hard-coded into each of these logics. In other words, each of the logics BMCL allows us to reason about the ability of a system of multiple reasoning agents under immutable bounds of memory and communication. This implies that BMCL cannot express nested abilities under bounded resources such as agents in a system can cooperate to achieve a goal under some bound of resources, and then continue cooperating to achieve another goal under another bound of resources. Moreover, we also face the following drawbacks when working with BMCL:

- BMCL is designed for systems of multiple reasoning agents although it is possible to adapt the logical language for other multi-agent systems. Nevertheless, how to adapt is not straightforward and requires changes in the axiomatisation system of the logics.
- BMCL is based on CTL, which only allows specifying and reasoning about properties of systems as a whole. For instance, it is possible to formalise in BMCL the property that all agents in the system can cooperate to achieve some goal. However, it is not possible to use BMCL for formalising the properties about the abilities of individuals or a coalition of individuals in a multi-agent system.
- Because bounds of resources are hard-coded into a logic of BMCL, it is not possible to reason about the ability of a system of reasoning agents under different bounds of resource in the same logic.

Recently, there have been several studies on Coalition Logic (CL) such as [Pauly, 2001, Pauly, 2002, Wooldridge et al., 2007, Ågotnes et al., 2008a, Ågotnes et al., 2008b, Ågotnes et al., 2009a, Ågotnes et al., 2009b] which enables us to express many interesting properties about the abilities of coalitions. For instance, CL allows us to express a property where a coalition of agents in a system can cooperate to force a certain result regardless of the intervention from any agent outside the coalition. Because the semantics of CL is based on Game Frames, the logic CL allows expressing and reasoning about properties about coalitional abilities of multi-agent systems. Each game frame describes a multi-agent system including possible states that the system may have, actions available to agents in the system at each state as well as the outcome of a joint action by all the agents in the system at each state. Because there is no cost associated with actions available to agents, there is no natural way of expressing resource requirements in CL. For example, there is no easy way to verify properties of the form 'can a set of agents C cooperate to force a result without spending no more than a given resource bound b'. Essentially, this is the *successful coalition under resource bound* problem investigated by Wooldridge and Dunne in [Wooldridge & Dunne, 2006].

In this chapter, we extend CL with resource bounds in order to overcome problems with BMCL where the extended logic allows expressing and reasoning about individual and coalitional abilities under resource bounds. In particular, we expand the concept of Game Frames with sets of resources which are required by actions as well as costs to actions. Then, we also extend the syntax of CL with resource bounds for coalitions to express the ability of a coalition under a certain bound of resources. Furthermore, unlike Wooldridge and Dunne, the extended logic also accounts for multi-shot games where the agents need to perform a *sequence* of actions to achieve the goal. As a running example, let us reconsider the system of two reasoning agents described on page 28. There are two explicit resources required by agents in the system for operating which are memory and communication. Each action requires a different amount of resources to perform. The action Read, which loads some formula from the knowledge base into the internal memory, requires at least one cell of memory where the resulting formula could be loaded into. The action Infer, which performs the inference rule *modus ponens*, requires at least two cells of memory where the antecedents of the inference rule are stored. Notice that both actions, Read and Infer, do not require any mount of the resource communication. However, the action Copy does where in order to perform a copy action, an agent needs to have at least one memory cell to store the resulting formula and one unit of the resource communication to obtain the formula from the internal memory of other agents. Apart from the two explicit resources, actions in this system also require another type of resource, which is time. In other words, all those actions take one step of time to complete. Even though the action Idle, which means to do nothing, requires no memory and communication, one step of time is essential for Idle to perform. In this example, it is assumed that the bound of memory for each agent is two cells and the bound of communication is two messages, two reasoning agents in the system need at least seven steps of time to derive the goal formula C. Particularly, under the bounds of two cells of memory, two messages of communication and seven steps of time, the two reasoning agents of the system are able to derive C. Moreover, when we alter the bounds of resources where limitation of memory is increased to three cells, communication is forbidden and allowed time is extended to ten steps of times, agent 1 has the power to derive C on its own.

In this chapter, we present in detail the extension of CL, namely Resource-Bounded Coalition Logic (RBCL). In particular, we first discuss the notion of resources together with related ones which are costs of action and resource bounds. After that, we describe the extension of Game Frames with resources and costs of actions. Then, we give the definition of the syntax and semantics of RBCL. In the remainder of the chapter, we introduce a sound and complete axiomatisation of RBCL and study the satisfiability problem of RBCL.

4.2 Resources

In a multi-agent system, agents perform actions by spending certain amount of resources. For instance, actions available to our reasoning agents in the example on page 28 require two explicit resources which are memory and communication together with an implicit resource, namely time. Each action costs a different amount of resources such as the cost of the action Read is 1 cells of memory, 0 message of communication and 1 step of time. For convenience, we will write this cost as as a tuple $(1, 0, 1)$ where the first element of the tuple refers to the memory cost, the second to the communication cost and the third to the time cost. Likewise, the costs of the actions Infer by modus ponens and Copy are $(2, 0, 1)$ and $(1, 1, 1)$, respectively.

In the general case, we assume that each multi-agent system is associated with a set of resources which are the fuel for actions of agents in the system to be able to perform. For the sake of simplicity, we also make a further assumption where amounts of resources are expressed in terms of units. Let us first define the set of resource bounds.

Definition 6. *Given a finite set R of resources where $R = \{1, \dots, r\}$, the set of resource bounds is defined as $\mathbb{B} = \mathbb{N}^r$.*

There are two places where resource bounds are used. One the one hand, they can be used as the available amount of resources for a coalition of agents in properties about the ability of the coalition; in other words, they are the resource bounds of the coalition. On the other hand, we also use resource bounds to describe the cost of actions. We define the comparison over resource bounds as usual. Given a resource bound $b = (b_1, \ldots, b_r) \in \mathbb{B}$, we write b_i to denote the i$^{\text{th}}$ component of b. Then, given two resource bounds $b, d \in \mathbb{B}$, we say that $b \leq d$ iff $b_i \leq d_i$ for all $1 \leq i \leq r$.

In this thesis, we generalise the way in which the resource requirements of complex actions are calculated. We argue that not all resource costs should be combined using the addition operator. For example, if one of the resources is time and the agents execute their actions concurrently, then, if each individual action costs one unit of time, the parallel combination of those actions also costs one unit of time. If one of the resources is memory, one can argue that if action a_1 requires k units of memory and action a_2 requires m units of memory, then executing actions a_1 and a_2 sequentially requires $max(k, m)$ units of memory. For generality, we introduce two cost operators $\oplus_j$ and $\otimes_j$ for each resource $j \in R$ to express how resource requirements are combined in parallel and in sequence, respectively. These operators $\oplus_j$ and $\otimes_j$ are defined as mappings from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. For both of them, we only require that for any k and $m \in \mathbb{N}$, $k \leq k \oplus_j m$ and $k \leq k \otimes_j m$, for any $j \in R$. This requirement is natural since it makes sense to say that the combination of two amounts of resource must be greater than or at least equal to each of them. Given two resource bounds $b = (b_1, \ldots, b_r) \in \mathbb{B}$ and $d = (d_1, \ldots, d_r) \in \mathbb{B}$, we define that

$$b \oplus d = (b_1 \oplus_1 d_1, \ldots, b_r \oplus_r d_r)$$
$$b \otimes d = (b_1 \otimes_1 d_1, \ldots, b_r \otimes_r d_r)$$

Then, if two actions a_1 and a_2 cost $Res(a_1)$ and $Res(a_2)$, respectively, the cost of executing them in parallel is $Res(a_1) \oplus Res(a_2)$, and in sequence $Res(a_1) \otimes Res(a_2)$.

4.3 Formalising single step strategies

We assume a set of agents $N = \{1, \ldots, n\}$ and a set of resources $R = \{1, \ldots, r\}$. Agents can perform actions from a set $\Sigma = \cup_{i \in N} \Sigma_i$, where Σ_i is the set of actions that can be performed by the agent i. Each action $a \in \Sigma$ has an associated cost $Res(a)$, which is a resource bound. A joint action executed by a coalition $C \subseteq N$ is a tuple of actions $a_C = (a_1, \ldots, a_k)$ (we assume for simplicity unless otherwise stated that $C = \{1, \ldots, k\}$ for some $k \leq n$).

4.3.1 Syntax

The language of RBCL_1 is defined relative to the sets N and R and a set of propositional variables *Prop*. A formula is defined as follows:

$$p \mid \neg\varphi \mid \varphi \wedge \psi \mid [C^b]\varphi$$

where $p \in Prop$, $C \subseteq N$, and $b \in \mathbb{N}^r$. The intuitive meaning of $[C^b]\varphi$ for $C \neq \emptyset$ is that coalition C can force the outcome φ under the resource bound b, or, in other words, the agents in C have a strategy costing at most b which enables them to achieve a φ-state no matter what the agents in $\bar{C} = N \setminus C$ do. For the empty coalition, $[\emptyset^b]\varphi$ means that if the grand coalition N executes any joint action which together costs at most b, then the system will end up in a φ state; that is, φ is unavoidable if N acts within the resource bound b.

4.3.2 Semantics

We define models of RBCL_1 as transition systems, where in each state agents execute actions in parallel to determine the next state. These are essentially the same as the models for coalition logic with the addition of costs of actions. First we define *resource-bounded action frames* which underlie the models:

Definition 7. *A resource-bounded action (RBA) frame F is a tuple $(N, R, \Sigma = \cup_{i \in N}\Sigma_i, S, T, o, Res)$ where:*

- *N is a non-empty set of agents,*
- *R is a non-empty set of resources,*
- *Σ is the set of actions agents can perform,*
- *S is a non-empty set of states,*
- *$T : S \times N \to \wp(\Sigma_i)$ assigns to each state the set of actions available to the agent i in this state; there must be an action which requires the smallest cost $(0, \dots, 0)$.*
- *o is the outcome function which takes a state s and a joint action a_N and returns the state resulting from the execution of a_N by the agents in s.*
- *$Res : \Sigma \to \mathbb{N}^r$ is the resource requirement function.*

In the case of joint actions, we generalise the function T as follows: a joint action $a_C \in T(s, C)$ iff $a_i \in T(s, i)$ for all $i \in C$. By $Res(a_C)$ we denote the combined cost of $Res(a_i)$ for every $i \in C$, that is $Res(a_C) = \oplus_{i \in C} Res(a_i)$.

Definition 8. *A single-step resource-bounded action (RBA) model M is a pair (F, V) where F is an RBA frame and $V : S \to \wp(Prop)$ is an assignment function.*

The truth definition for single-step RBA models is as follows:

- $M, s \models p$ *iff* $p \in V(s)$
- $M, s \models \neg\varphi$ *iff* $M, s \not\models \varphi$
- $M, s \models \varphi \land \psi$ *iff* $M, s \models \varphi$ *and* $M, s \models \psi$
- $M, s \models [C^b]\varphi$ *for* $C \neq \varnothing$ *iff there is* $a_C \in T(s, C)$ *with* $Res(a_C) \leq b$ *such that for every joint action* $a_{\bar{C}} \in T(s, \bar{C})$ *by the agents not in* C*, the outcome of the resulting tuple of actions executed in* s *satisfies* φ*:* $M, o(s, (a_C, a_{\bar{C}})) \models \varphi$
- $M, s \models [\varnothing^b]\varphi$ *iff the outcome of any joint action* $a_N \in T(s, N)$ *with* $Res(a_N) \leq b$ *executed in* s *satisfies* φ*:* $M, o(s, a_N) \models \varphi$*.*

The notions of satisfiability and validity are standard. Let us call the set of all formulas valid in single-step RBA models RBCL$_1$ (where 1 refers to considering only one-step strategies, as in Coalition Logic).

Theorem 2. *RBCL$_1$ is completely axiomatised by the following set of axiom schemas and inference rules:*

A0 *All propositional tautologies*

A1 $[C^b]\top$

A2 $\neg[C^b]\bot$

A3 $\neg[\varnothing^b]\varphi \leftrightarrow [N^b]\neg\varphi$

A4 $[C^b](\varphi \land \psi) \to [C^b]\varphi$

A5 $[C^b]\varphi \to [C^d]\varphi$ *where* $d \geq b$ *if* $C \neq \varnothing$ *or* $d \leq b$ *if* $C = \varnothing$

A6a $[C^b]\varphi \land [D^d]\psi \to [(C \cup D)^{b \oplus d}](\varphi \land \psi)$ *where* C *and* D *are both disjoint and non-empty*

A6b $[\varnothing^b]\varphi \wedge [C^b]\psi \rightarrow [C^b](\varphi \wedge \psi)$ *where* C *is either* $\varnothing$ *or* N

MP $\vdash \varphi,\ \vdash \varphi \rightarrow \psi \Rightarrow \vdash \psi$

Equivalence $\vdash \varphi \leftrightarrow \psi \Rightarrow \vdash [C^b]\varphi \leftrightarrow [C^b]\psi$

The notions of derivability and consistency are standard. Note that if we erase the resource superscript in the axiomatisation above, we get the complete axiomatisation of Coalition Logic as given in [Pauly, 2002], and a trivial formula resulting from **A5**. The rule of monotonicity (**RM**) is derivable as in Coalition Logic, that is, if $\vdash \varphi \rightarrow \psi$, then $\vdash [C^b]\varphi \rightarrow [C^b]\psi$.

We omit the completeness proof here as it is a special case of completeness proof of RBCL given in the next sections.

4.3.3 Example

As an illustration, we show how to express some properties of coalitional resource games from [Wooldridge & Dunne, 2006] in RBCL_1.

A coalitional resource game (CRG) Γ is defined as a tuple $(N, G, R, G_1, \ldots, G_n, en, req)$ where

- $N = \{1, \ldots, n\}$ is a set of agents,
- $G = \{g_1, \ldots, g_m\}$ is a set of goals,
- $R = \{r_1, \ldots, r_t\}$ is a set of resources,
- $G_i \subseteq G$ is the set of goals for the agent i,
- $en : N \times R \rightarrow \mathbb{N}$ is the resource endowment function (how many units of a given resource is allocated to an agent),
- $req : G \times R \rightarrow \mathbb{N}$ is the resource requirement function (how many units of a particular resource is required to achieve a goal). It is assumed that each goal requires a non-zero amount for at least one resource.

In CRGs, the endowment of a coalition is equal to the sum of the endowments of its members: $en(C, r) = \Sigma_{i \in C} en(i, r)$. Furthermore, the cost of performing actions in parallel is defined by means of the sum operator.

As an example, we give a simple CRG from [Wooldridge & Dunne, 2006], where $N = \{1, 2, 3\}$; $G = \{g_1, g_2\}$; $R = \{r_1, r_2\}$; $G_1 = \{g_1\}$, $G_2 = \{g_2\}$, $G_3 = \{g_1, g_2\}$; $en(1, r_1) = 2$, $en(1, r_2) = 0$, $en(2, r_1) = 0$, $en(2, r_2) = 1$, $en(3, r_1) = 1$, $en(3, r_2) = 2$; $req(g_1, r_1) = 3$, $req(g_1, r_2) = 2$, $req(g_2, r_1) = 2$, and $req(g_2, r_2) = 1$. In RBCL$_1$, we can state properties such as the coalition of agents 1 and 3 can achieve g_1 under the resource bound corresponding to the sum of their endowments: $[1, 3^{\langle 3,2 \rangle}]g_1$. More generally, a decision problem which is called *coalition C is successful under resource bound b* in [Wooldridge & Dunne, 2006] can be expressed as

$$[C^b] \bigwedge_{i \in C} \bigvee_{g \in G_i} g.$$

.

4.4 Formalising multi-step strategies and arbitrary resource combinators

In this section, we generalise the logic described in the previous section. In particular, we consider multi-step strategies, as in Extended Coalition Logic with the $[C^*]$ operator [Pauly, 2001], or as in ATL. The reason for this is that we are interested in the resource requirements of strategies which involve multiple steps. For example, suppose a coalition C can enforce φ in three steps: $[C^{b_1}][C^{b_2}][C^{b_3}]\varphi$. We can deduce from this that the agents have a strategy to achieve φ which costs at most $b_1 \otimes b_2 \otimes b_3$. However expressing the fact in this way is rather clumsy. Even worse, to say that 'C has some strategy which achieves φ in three steps which costs at most b' in RBCL$_1$, we have to use a disjunction over all possible vectors of natural numbers b_1, b_2, b_3 which sum up to b: $\vee_{b_1 \otimes b_2 \otimes b_3 = b}[C^{b_1}][C^{b_2}][C^{b_3}]\varphi$. Hence we extend the set of actions, or strategies, with sequential compositions of actions.

In the rest of the chapter, we assume the following:

- The last resource r in the set of resources R is always time.
- $\oplus_r$ is the $\max$ function.
- $\otimes_r$ is the + operator.
- Every action costs exactly one unit of time.

As every action requires at least one step of time to perform, the smallest cost is redefined as $(0, \ldots, 0, 1)$. We denote by $t(b)$ the time component of cost vector b. In particular, $t(Res(a)) = 1$ for any $a \in \Sigma$. In the language, only operators $[C^b]$ with $t(b) \geq 1$ are allowed.

4.4.1 Strategies and multi-step RBA models

Given an RBA frame $F = (N, R, \Sigma, S, T, o, Res)$, a *strategy* for an agent $i \in N$ is a function $f_i : S^+ \to \Sigma_i$ from finite non-empty sequences of states to actions, such that $f_i(\lambda s) = a \in T(s, i)$, where λs is a sequence of states ending in state s. Intuitively, f_i says what action the agent i should perform in state s given the previous history of the system. A strategy for a coalition C is a set $F_C = \{f_1, \dots, f_k\}$ of strategies for each agent.

For a sequence $\lambda = s_0 s_1 \dots \in S^\omega$, we denote $\lambda[i] = s_i$ and $\lambda[i, j] = s_i \dots s_j$. The set of possible computations generated by a strategy F_C from a state s_0, $out(s_0, F_C)$, is

$$\{\lambda \mid \lambda[0] = s_0 \ \wedge \ \forall j \geq 0 : \lambda[j+1] \in o^*(\lambda[j], (f_i(\lambda[0, j]))_{i \in C})\}$$

where $o^*(s, a_C) = \{o(s, (a_C, a_{\bar{C}})) \mid a_{\bar{C}} \in T(s, \bar{C})\}$. Now we define the cost of a multi-step strategy. Let $\lambda \in out(s_0, F_C)$. The cost of F_C over a prefix $\lambda[0, m]$ where $m > 0$ is defined inductively as follows:

$cost(\lambda[0,1], F_C) = \oplus_{i \in C} Res(f_i(\lambda[0]))$, where $Res(f_i(\lambda[0]))$ is the cost of action of the agent i in $\lambda[0]$, and $\oplus_{i \in C}$ is the operator for combining the costs of actions executed in parallel by the agents in C;

$cost(\lambda[0, m], F_C) = cost(\lambda[0, m-1], F_C) \otimes (\oplus_{i \in C} Res\ (f_i(\lambda[0, m-1])))$ for $m > 1$; this is the cost of the previous $m-1$ steps in the strategy combined sequentially with the cost of the mth step.

In the following, we define the semantics of RBCL for the case of multi-step strategies. Notice that we only provide the definition of formulas of the form $[C^b]\varphi$ since the other cases are still the same as before.

Definition 9. *A multi-step resource-bounded action model M is a pair (F, V) where F is an RBA frame, and $V : S \to \wp(Prop)$ is an assignment function, and the truth definition for the $[C^b]$ modality is*

- $M, s \models [C^b]\varphi$ *for $C \neq \varnothing$ iff there is a strategy F_C such that for all $\lambda \in out(s, F_C)$, there exists $m > 0$ such that $cost(\lambda[0, m], F_C) \leq b$ and $M, \lambda[m] \models \varphi$,*

- $M, s \models [\varnothing^b]\varphi$ *iff for all strategies F_N, computations $\lambda \in out(s, F_N)$, and $m > 0$ such that $cost(\lambda[0, m], F_N) \leq b$, $M, \lambda[m] \models \varphi$.*

Note that under this definition, the meaning of $[C^b]\varphi$ (for non-empty C) becomes as follows: C has a multi-step strategy to bring about φ, and the cost of this strategy is less than b. The meaning of $[\varnothing^b]\varphi$ is that the outcome of *any* strategy of the grand coalition N which costs less than b, satisfies φ.

The set of all formulas valid in multi-step RBA models will be denoted by RBCL.

4.4.2 Example

As an illustration, we show how properties of coalitions of resource-bounded reasoners can be expressed by, once again, considering the example presented in page 28. As depicted in Figure 3.1, the system in this example has the ability to derive c under the resource bound 4 for memory, 1 for communication and 7 for time. In the logic BMCL, the resource bound (except time) is hard-coded into the logic and we did not have a way of expressing coalitional abilities of agents. We can however express in RBCL that, for example, reasoners 1 and 2 can derive c under such resource bound by the formula $[\{1,2\}^{(4,1,7)}]\ B_1 c$.

4.4.3 Effectivity structures

For proving completeness of RBCL, it is easier to work with an alternative semantics, given not in terms of multi-step RBA models, but in terms of effectivity structures. These are closely related to RBA models, and we will show that effectivity structures satisfying some natural properties give rise to an alternative semantics for RBCL.

Let $\wp(N)^{\mathbb{B}} = \{C^b \mid C \subseteq N,\ b \in \mathbb{N}^r,\ t(b) \geq 1\}$. Intuitively, this is the set of all possible coalitions with all possible resource allocations. An *effectivity structure* is a function $E : S \to (\wp(N)^{\mathbb{B}} \to \wp(\wp(S)))$ which describes, for each state in S, which subsets of S a coalition C can force under resource bound b.

Given an RBA frame F, *the effectivity structure corresponding to* F is defined as follows:

- for $C \neq \varnothing$, $X \in E(s)(C^b)$ iff there exists a strategy F_C such that for all $\lambda \in out(s, F_C)$, there exists $m > 0$ such that $cost(\lambda[0,m], F_C) \leq b$ and $\lambda[m] \in X$;
- $X \in E(s)(\varnothing^b)$ iff for all strategies F_N, sequences of states $\lambda \in out(s, F_N)$, and $m > 0$ such that $cost(\lambda[0,m], F_N) \leq b$, we have $\lambda[m] \in X$.

In other words, $X \in E(s)(C^b)$, where C is not the empty coalition, means that the coalition C has a strategy to bring about X within the bound b. $X \in E(s)(\varnothing^b)$ means that all strategies for the grand coalition which cost less b always result in a state in X, i.e., X is inevitable.

4.4.4 Characterising effectivity in RBA frames

Every RBA frame gives rise to an effectivity structure, but the reverse does not hold. In this section, we characterise properties which an effectivity structure should satisfy to be an effectivity structure corresponding to an RBA frame. Following Pauly in [Pauly, 2002], we call such effectivity structures playable (RB-playable, where RB stands for resource-bounded).

Below we state some useful properties of RB-playable effectivity structures. These are very similar (apart from the resource bound) to the properties of playable effectivity structures listed in [Pauly, 2002] and are given the same names:

- An effectivity structure E is *outcome monotonic* iff

$$X \in E(s)(C^b) \Rightarrow X' \in E(s)(C^b) \text{ for all } X' \supseteq X$$

- An effectivity structure E is *coalition monotonic* iff

$$X \in E(s)(C^b) \Rightarrow X \in E(s)(D^b)$$

  where $C \neq \varnothing$ and $D \supseteq C$; and

$$X \in E(s)(\varnothing^b) \Rightarrow X \in E(s)(N^b)$$

- An effectivity structure E is *N-maximal* iff

$$X \notin E(s)(\varnothing^b) \Rightarrow \overline{X} \in E(s)(N^b)$$

- An effectivity structure E is *N-minimal* iff

$$X \in E(s)(N^b) \wedge Y \notin E(s)(N^b) \Rightarrow X \setminus Y \in E(s)(N^b)$$

  Note that *N-minimality* is not listed in [Pauly, 2002], but its analogue is derivable.

- An effectivity structure E is N-*determinant*[1] iff

$$X \in E(s)(N^b) \Rightarrow \exists t \in X \text{ such that } \{t\} \in E(s)(N^b)$$

[1]Notice N-*determinacy* is also not listed in [Pauly, 2002], however, we need this property to prove Theorem 3 below and also its analogue in [Pauly, 2002]. The problem with Pauly's proof was pointed out to the author by Wojtek Jamroja, Valentin Goranko and Paolo Turrini, but the fix was developed independently.

We can also write this property in another way, that is E is N*-determinant* iff for any $X \subseteq S$, if $\forall t \in X$, $\{t\} \notin E(s)(N^b)$ then we have $X \notin E(s)(N^b)$.

- An effectivity structure E is *regular* iff for all coalitions C which are neither empty nor equal to N
$$X \in E(s)(C^b) \Rightarrow \overline{X} \notin E(s)(\overline{C}^{b'}) \text{ for all } t(b) = t(b') = 1$$
In the case where the time component is greater than one, we also have a similar property to regularity but for only the whole system (or the empty coalition). An effectivity structure E is *N-regular* iff $X \in E(s)(N^b) \Rightarrow \overline{X} \notin E(s)(\varnothing^b)$.

- An effectivity structure E is *super-additive* iff the following holds, for all b and d with $t(b) = t(d) = 1$, and $C \cap D = \varnothing$:
  - If $C \neq \varnothing$ and $D \neq \varnothing$, $X_1 \in E(s)(C^b)$ and $X_2 \in E(s)(D^d) \Rightarrow$
  $$X_1 \cap X_2 \in E(s)((C \cup D)^{b \oplus d})$$
  - If $C = \varnothing$ and $D = \varnothing$ or N, $X_1 \in E(s)(\varnothing^d)$ and $X_2 \in E(s)(D^d)$ then $X_1 \cap X_2 \in E(s)(D^d)$

  We have two different cases in the definition of super-additivity because in the notation $\varnothing^b$, b is not the resource bound for the coalition it annotates but for its complement. Therefore, it is not possible to combine the bounds as in the case when both coalition C and D are non empty. Notice that super-additivity requires the time component of both resource bounds to be equal to 1. When one of them is greater than one, such a property might not be true. We also have a more general property that if one of the coalitions in the property is empty as follows.

- An effectivity structure E is *general super-additive* iff it is super-additive and $X_1 \in E(s)(\varnothing^b)$ and $X_2 \in E(s)(C^b) \Rightarrow X_1 \cap X_2 \in E(s)(C^b)$ where C is either empty or the grand coalition.

We also have properties corresponding to sequential composition of strategies:

- An effectivity structure E is *super-transitive* iff the following holds for all $C \neq \varnothing$: $\{s' \in S \mid X \in E(s')(C^{b_2})\} \in E(s)(C^{b_1}) \Rightarrow X \in E(s)(C^{b_1 \otimes b_2})$ (if a set of states where X is obtainable under b_2 can be enforced under b_1, then X can be enforced by the combined strategy under $b_1 \otimes b_2$).

- An effectivity structure E is *transitive* iff for any b with $t(b) > 1$ and $C \neq \emptyset$: $X \in E(s)(C^b) \Rightarrow \exists b' < b : X \in E(s)(C^{b'})$ (X can be achieved under a tighter bound b') or $\exists b_1 \otimes b_2 = b : \{s' \in S \mid X \in E(s')(C^{b_2})\} \in E(s)(C^{b_1})$ (X can be achieved by combining two strategies costing b_1 and b_2 such that $b_1 \otimes b_2 = b$).

Finally, the following property is specific to resource bounds:

- An effectivity structure E is *bound-monotonic* iff
$X \in E(s)(C^b) \Rightarrow X \in E(s)(C^d)$ for all $d \geq b$ if $C \neq \emptyset$ or $d \leq b$ if $C = \emptyset$.

Bound-monotonicity is a very natural property: if a non-empty coalition can achieve something under the bound b, then it can achieve it with a more generous resource allowance. For $C = \emptyset$, this property means that if an outcome cannot be avoided when the grand coalition is restricted to strategies which cost at most b, then it cannot be avoided if N uses fewer resources (hence has fewer strategies available).

It is easy to prove that the properties above are true for any effectivity structure obtained from a RBA frame. Conversely, RB-playable effectivity structures defined below are effectivity structures of an RBA frame.

Definition 10. *An effectivity structure $E : S \to (\wp(N)^{\mathbb{B}} \to \wp(\wp(S)))$ is RB-playable iff, for every $s \in S$, E has the following properties:*

1. *For all $C^b \in \wp(N)^{\mathbb{B}}$, $S \in E(s)(C^b)$*
2. *For all $C^b \in \wp(N)^{\mathbb{B}}$, $\emptyset \notin E(s)(C^b)$*
3. *Outcome-monotonicity*
4. *N-maximality*
5. *N-determinacy*
6. *N-regularity*
7. *Super-additivity*
8. *Super-transitivity*
9. *Transitivity*

10. *Bound-monotonicity*

It can be shown that RB-playability implies the other properties listed above.

Lemma 6. *Let E be a RB-playable effectivity structure, then E has the following properties:*

1. *Coalition monotonicity*
2. *N-minimality*
3. *Regularity*
4. *General super-additivity*

In the following, we provide the proof of the above lemma. First general super-additivity is proved by induction on resource bounds using super-additivity. The proofs of the other properties are based on general super-additivity.

Proof. By super-transitivity, we have that, for any b and $b_1 \otimes b_2 = b$

$$\{s' \mid X \in E(s')(N^{b_2})\} \in E(s)(N^{b_1}) \Rightarrow X \in E(s)(N^{b_1 \otimes b_2})$$

Hence,

$$X \notin E(s)(N^{b_1 \otimes b_2}) \Rightarrow \{s' \mid X \in E(s')(N^{b_2})\} \notin E(s)(N^{b_1})$$

By N-regularity and N-maximality, we have $\overline{X} \in E(s)(\varnothing^{b_1 \otimes b_2}) \Rightarrow X \notin E(s)(N^{b_1 \otimes b_2})$ and $\{s' \mid X \in E(s')(N^{b_2})\} \notin E(s)(N^{b_1}) \Rightarrow \{s' \mid \overline{X} \in E(s')(\varnothing^{b_2})\} \in E(s)(\varnothing^{b_1})$, respectively. Therefore,

$$X \in E(s)(\varnothing^{b_1 \otimes b_2}) \Rightarrow \{s' \mid X \in E(s')(\varnothing^{b_2})\} \in E(s)(\varnothing^{b_1}) \tag{4.1}$$

We now prove general super-additivity by induction on the time component of b. The base case follows directly from super-additivity. Let $X \in E(s)(\varnothing^b)$ where the time component of b is greater than 1. Assume that $Y \in E(s)(C^b)$ where C is either $\varnothing$ or N. If $Y \in E(s)(C^{b'})$ for some $b' < b$, then bound-monotonicity for the empty coalition and induction hypothesis show that $X \cap Y \in E(s)(C^{b'})$. Hence, bound-monotonicity implies $X \cap Y \in E(s)(C^b)$. If $Y \notin E(s)(C^{b'})$ for all such b', we have there exists $b_1 \otimes b_2 = b$ such that

$$\{s' \mid Y \in E(s')(C^{b_2})\} \in E(s)(C^{b_1})$$

which follows from transitivity when $C = N$ or from (4.1) with arbitrary $b_1 \oplus b_2 = b$ when $C = \varnothing$. Note that we also have $\{s' \mid X \in E(s')(\varnothing^{b_2})\} \in E(s)(\varnothing^{b_1})$. Applying the induction hypothesis twice together with outcome-monotonicity, we have the following result:

$$\{s' \mid X \cap Y \in E(s')(C^{b_2})\} \in E(s)(C^{b_1})$$

Therefore, super-transitivity implies that $X \cap Y \in E(s)(C^b)$.

1. Assume that $X \in E(s)(\varnothing^b)$. By RB-playability, we have $S \in E(s)(N^b)$. Apply general super-additivity, we obtain $X \in E(s)(N^b)$.

   Let $\varnothing \neq C \subset N$, we prove by induction on the time component of b that $X \in E(s)(C^b) \Rightarrow X \in E(s)(D^b)$ for any $D \supset C$.

   In the base case, when time component of b is equal to 1, let $C' = D \setminus C$. We have $S \in E(s)(C'^{(0,\ldots,0,1)})$, thus super-additivity implies that $X = X \cap S \in E(s)(D^b)$.

   Let us assume that time component of b is greater than 1. If $X \in E(s)(C^{b'})$ for some $b' < b$, then it is obvious by the induction hypothesis that $X \in E(s)(D^{b'})$. Hence, bound-monotonicity shows that $X \in E(s)(D^b)$. If $X \notin E(s)(C^{b'})$ for any such b', then we have by transitivity that there exists $b_1 \otimes b_2 = b$ such that

   $$\{s' \mid X \in E(s')(C^{b_2})\} \in E(s)(C^{b_1})$$

   By the induction hypothesis, we have

   $$\{s' \mid X \in E(s')(C^{b_2})\} \in E(s)(D^{b_1})$$

   and

   $$\{s' \mid X \in E(s')(C^{b_2})\} \subseteq \{s' \mid X \in E(s')(D^{b_2})\}$$

   Thus, outcome-monotonicity implies that

   $$\{s' \mid X \in E(s')(D^{b_2})\} \in E(s)(D^{b_1})$$

   Therefore, we have by super-transitivity that $X \in E(s)(D^b)$.

2. Assume that $X \in E(s)(N^b)$ and $Y \notin E(s)(N^b)$. By N-maximality, we have $\overline{Y} \in E(s)(\varnothing^b)$. Therefore, general super-additivity implies that $X \cap \overline{Y} \in E(s)(N^b)$.

3. Assume that $\varnothing \neq C \subset N$ and $X \in E(s)(C^b)$ where the time component of b is equal to 1. Furthermore, assume to the contrary that $\overline{X} \in E(s)(\overline{C}^{b'})$ where time component of b' is also equal to 1. Applying super-additivity, we have $X \cap \overline{X} \in E(s)(N^{b \oplus b'})$ which contradicts the fact that E is RB-playable. Therefore, in general, E is regular.

□

Furthermore, notice that N-determinacy is derivable from other properties of a RB-playable effectivity structure E over a finite set S of states. From the proof of Lemma 6, we know that any RB-playable effectivity structure has N-minimality. Given any finite subset X of states where $X \in E(s)(N^b)$, we remove any state $s \in X$ such that $\{s\} \notin E(s)(N^b)$ and N-minimality shows that $X \setminus \{s\} \in E(s)(N^b)$. We repeat the removal for $X \setminus \{s\}$ until no states can be removed. Obviously, it must not happen that all $s \in X$ were removed as we would end up with $\varnothing \in E(s)(N^b)$ which violates the second requirement for a RB-playable effectivity structure. As X is finite, the removal must terminate and for any state s remained, we have that $\{s\} \in E(s)(N^b)$; hence N-determinacy is proved. In other words, this means that when proving an effectivity structure over a finite set of states to be RB-playable, we shall omit proving N-determinacy.

Theorem 3. *An effectivity structure is RB-playable iff it is the effectivity structure of some RBA frame.*

Proof. It is easy to check that effectivity structures obtained from RBA frames satisfy all properties of RB-playability. As a running example, let us prove that the corresponding effectivity structure E_F of a given RBA frame F (over a set S of states) satisfies N-determinacy. Assume that $X \in E_F(s)(N^b)$ where $X \subseteq S$, $s \in S$ and $b \in \mathbb{B}$. This means there is a strategy F_N for the coalition N such that for all $\lambda \in out(s, F_N)$, there exists $m > 0$ such that $cost(\lambda[0,m], F_N) \leq b$ and $\lambda[m] \in X$. Obviously, we also implies that $\{\lambda[m]\} \in E_F(s)(N^b)$ according to the definition of E_F. Hence, E_F satisfies N-determinacy.

In order to prove the other direction for a given RB-playable effectivity structure E, we need to construct a RBA frame such that its effectivity structure is identical to E.

Let E be an RB-playable effectivity structure. The construction of the RBA frame is similar to that in Coalition Logic extended with costs for actions. First, we define the set of possible actions for each agent at each state $s \in S$ with their associated costs Res. Then the construction is completed by defining the outcome function o.

In order to make the following proof easy to follow, let us provide an informal sketch of the argument. The main task of defining the RBA frame is to define actions available for each agent at a particular state. We define these actions so that it facilitates the definitions of costs of actions and the outcome function. Each action for an agent is a triple (g, t, h) where:

- g is a function which defines the preferred set of outcomes for each coalition where the agent participates and is willing to contribute a certain amount of resources (then, the cost of this action is this amount of resources). Given the actions of all agents, the component g of those actions will define the coalitions where the agents participate, hence also the preferred set of outcomes for each agent.
- t is a natural number which is used to determine which agent has the power to decide outcome.
- When we know which agent has the power to decide the outcome and its preferred set of outcomes, h is a function which determines the only outcome among those in the preferred set.

In the following, we present in detail how actions and outcomes of actions are defined.

For every $i \in N$, let b be a bound such that $t(b) = 1$, we define $\mathcal{C}_i^b = \{C^d \mid i \in C \wedge t(d) = 1 \wedge d \geq b\}$ which is the set of all coalitions where i may participate and contribute b amount of resources. Note that for all actions $t(b)$ is always 1.

For every $s \in S$, we define

$$\Gamma(s,i) = \{g^b_{(s,i)} : \mathcal{C}_i^b \to \wp(S) \mid g^b_{(s,i)}(C^d) \in E(s)(C^d)\}$$

$\Gamma(s,i)$ is the set of option functions for an agent i at state s. Each option function in $\Gamma(s,i)$ is a mapping $g^b_{(s,i)}$ where b is a resource bound such that $t(b) = 1$; $g^b_{(s,i)}$ determines the outcome when the agent i agrees to participate in a coalition. How an agent agrees to participate in a coalition will be specified later when we define the outcome function.

Let $H = \{h : \wp(S) \to S \mid h(X) \in X\}$ be the set of choice functions, that is, if an agent has the power to decide the outcome, it will use some h function to do so. We then define the set of available actions for an agent i at a state s as follows:

$$T(s,i) = \Gamma(s,i) \times \mathbb{N} \times H$$

Each action is a triple $(g^b_{(s,i)}, t, h)$ consisting of an option function $g^b_{(s,i)}$, an index t (a natural number) and a choice function h. Informally, option functions determine how the agents group together to form coalitions and then which outcome options they will choose. The index determines which agent has the power to decide the outcome based on its associated h function. We assign that $Res((g^b_{(s,i)}, t, h)) = b$. Note that for any action, we have $t(Res((g^b_{(s,i)}, t, h))) = 1$.

Let $\Sigma_i = \bigcup_{s \in S} T(s, i)$. We now define the outcome of a joint action $\sigma \in \Sigma_N$ at a state s. Assume that $\sigma = \{(g^{b_i}_{(s,i)}, t_i, h_i) \mid i = 1, \dots, n\}$ where $t(b_i) = 1$ for all $i \in N$. For any coalition $C \subseteq \Sigma$, let $b_C = \oplus_{i \in C} b_i$ and $g = (g^{b_i}_{(s,i)})_{i \in N}$. We denote $P(g, C)$ the coarsest partition $\langle C_1, \dots, C_m \rangle$ of C such that:

$$\forall l \leq m \ \forall i, j \in C_l : g^{b_i}_{(s,i)}(C^{b_C}) = g^{b_j}_{(s,j)}(C^{b_C})$$

We define how coalitions are formed based on g as follows:

$$\begin{array}{rcl}
P_0(g) & = & \langle N \rangle \\
P_1(g) & = & \langle P(g, N) \rangle = \langle C_{1,1}, \dots, C_{1,k_1} \rangle \\
P_2(g) & = & \langle P(g, C_{1,1}), \dots, P(g, C_{1,k_1}) \rangle \\
 & & = \langle C_{2,1}, \dots, C_{2,k_2} \rangle \\
 & \vdots & \\
P_\eta(g) & = & \langle C_{\eta,1}, \dots, C_{\eta,k_\eta} \rangle
\end{array}$$

As N is finite, the above computation reaches some η such that $P_\eta(g) = P_{\eta+1}(g)$. Let $P(g) = P_\eta(g)$ which shows how agents are grouped into coalitions.

Now, we define the core of the set $E(s)(N^b)$ containing all states which are the possible outcomes from s where all agents in the system spends less than b amount of resources. The core of $E(s)(N^b)$ is denoted as $E^o(s)(N^b)$ and we define that a state $t \in S$ is in $E^o(s)(N^b)$ iff $\{t\} \in E(s)(N^b)$. Obviously, $E^o(s)(N^b) \neq \varnothing$ as otherwise, $S \notin E(s)(N^b)$ according to the fact that E is N-determinant. Moreover, as $\forall t \in S \setminus E^o(s)(N^b)$, $\{t\} \notin E(s)(N^b)$, we have that $S \setminus E^o(s)(N^b) \notin E(s)(N^b)$. Thus, by N-maximality, we have that $E^o(s)(N^b) \in E(s)(\varnothing^b)$.

Assume that $P(g) = \langle C_1, \dots, C_m \rangle$. For convenience, let $g(C_l) = g^{b_i}_{(s,i)}({C_l}^{b_{C_l}})$ for some $i \in C_l$ where $l \leq m$.

We define $G(g) = \bigcap_{l \leq m} g(C_l) \cap (E^o(s)(N^{b_N}))$. Let us show that $G(g) \neq \varnothing$. By super-additivity, we have that $\bigcap_{l \leq m} g(C_l) \in E(s)(N^{b_N})$. Moreover, we already have that $E^o(s)(N^{b_N}) \in E(s)(\varnothing^{b_N})$. Apply super-additivity again, we obtain $G(g) = \bigcap_{l \leq m} g(C_l) \cap (E^o(s)(N^{b_N})) \in E(s)(N^{b_N})$. As E is RB-playable, it is straightforward that $G(g) \neq \varnothing$.

Let $t_0 = (\sum_{i \in N} t_i \mod n) + 1$. The outcome function is defined as follows: $o(s, \sigma) = h_{t_0}(G(g))$.

Before continuing the proof, let us consider an example in order to illustrate how outcome is determined for a given joint action as described above. For the sake of simplicity, we describe a RB-playable effectivity structure by considering the following resource bounded game frame at a state s_0. Assume that we have three agents 1, 2 and 3 in a system which is associated with two resources *utility* and *time* At s_0, each agent can either perform a cooperate or a defect action. We use C and D to denote these actions, respectively. While the cost of D for all the agents is (0,1), that is 0 for utility and 1 for time, the cost of C varies depending on who performs C. In particular, the cost of C for agent 1 is (1,1), for agent 2 is (2,1) and for agent 3 is (3,1). For convenience, we shall write $C^{(1,1)}$ to denote that action C costs (1,1). Moreover, as all actions cost 1 unit of time, we temporarily ignore the time component in the cost, hence $C^{(1,1)}$ is simply rewritten as C^1 which means that the action C costs 1 unit of utility. There are eight output states $s_1, \dots, s_8$ each is associated with a number which is the total of the utility contributed by every agent as they pay for the actions in order to get the corresponding outcome. In Figure 4.1, we illustrate the

3	C^3		D^0	
1 \ 2	C^2	D^0	C^2	D^0
C^1	s_1 6	s_2 4	s_5 3	s_6 1
D^0	s_3 5	s_4 3	s_7 2	s_7 0

FIGURE 4.1: Outcomes from s_0 and their associated numbers.

outcomes of each joint action. The first column contains actions performed by agent 1 together with corresponding costs. Actions performed by agent 2 are depicted in the second row while the first row is for describing actions of agent 3. In other cells, we define the outcomes of each joint action. For example, the cell on the third row, second column which contains a state s_1 and a number 6 says that the outcome of the joint action (C^1, C^2, C^3) is s_1 and this state is associated with the number 6. For convenience, for each number $k \in \{0, \dots, 6\}$, we define a proposition $(\geq k)$ which is true in a state s_i where $i \in \{1, \dots, 8\}$ iff n is smaller or equal to the number with which s_i is associated. For instance, (≥ 4) is true at s_2 and s_3 but not in s_5 and s_6. Furthermore, by abusing the notation, we also denote $(\geq k)$ is the subset of $\{s_1, \dots, s_8\}$ which contains only states where $(\geq k)$ is true at. Let us consider the effectivity structure E admitted by this example, at least at the state s_0. We consider the joint action $a = ((g_1^2, 5, h_1), (g_2^2, 6, h_2), (g^1, 8, h_1))$ in $T(s_0, 1) \times T(s_0, 2) \times T(s_0, 3)$

where three functions g_1^3, g_2^2 and g_3^1 are partially defined as follows:

$$\begin{array}{rlrlrl} g_1^2(\{1\}^2) = & (\geq 1) & g_2^2(\{2\}^2) = & (\geq 2) & g_3^1(\{3\}^1) = & S \\ g_1^2(\{1,2\}^4) = & (\geq 3) & g_2^2(\{1,2\}^4) = & (\geq 3) & g_3^1(\{1,3\}^3) = & S \\ g_1^2(\{1,3\}^3) = & (\geq 3) & g_2^2(\{2,3\}^3) = & (\geq 2) & g_3^1(\{2,3\}^3) = & S \\ g_1^2(\{1,2,3\}^5) = & (\geq 5) & g_2^2(\{1,2,3\}^5) = & (\geq 5) & g_3^1(\{1,2,3\}^5) = & S \end{array}$$

Moreover, the function h_1 (h_2) is defined so that it returns a state s_i from a subset $X \subseteq \{s_1, \dots, s_8\}$ where i the smallest (greatest) index in X. For example, $h_1(\{s_2, s_3, s_5\}) = s_2$ while $h_2(\{s_2, s_3, s_5\}) = s_5$. To determine the outcome of a at s_0, we first compute $P(g)$ as follows:

$$\begin{aligned} P_0(g) &= \langle N \rangle \text{ where } N = \{1,2,3\} \\ P_1(g) &= \langle P(g, N) \rangle = \langle \{1,2\}, \{3\} \rangle \\ P_2(g) &= \langle P(g, \{1,2\}), P(g, \{3\}) \rangle = \langle \{1,2\}, \{3\} \rangle \end{aligned}$$

Hence, we obtain $P(g) = \langle \{1,2\}, \{3\} \rangle$ where $g(\{1,2\}) = (\geq 3)$ and $g(\{3\}) = S$. Notice that we have $E^o(s_0)(N^5) = \{s_2, \dots, s_8\}$. Then, we have that $G(g) = (\geq 3) \cap S \cap E^o(s_0)(N^5) = (\geq 3) \setminus \{s_1\} = \{s_2, s_3, s_4, s_5\}$. Furthermore, we have that $t_0 = (5+6+8) \mod 3 + 1 = 2$, which means the function h_2 is used to decide the outcome. Thus, the outcome of a is $h_2(G(g)) = s_5$.

Let us now turn back to the proof. Assume E_F be the effectivity structure of the frame constructed above. We claim that $E = E_F$.

Firstly, we show the left-to-right inclusion by induction on bounds. In the base case, assume $X \in E(s)(C^b)$ where $t(b) = 1$. Choose the actions for agents in $C = \{1, \dots, k\}$ as follows,

$$\begin{array}{rcl} a_1 & = & (g_1^b, t_1, h_1) \\ a_2 & = & (g_2^{\overline{0}}, t_2, h_2) \\ & \vdots & \\ a_k & = & (g_k^{\overline{0}}, t_k, h_k) \end{array}$$

where $g_1^b(D^d) = g_i^{\overline{0}}(D^d) = X$ for all $i = 2, \dots, k$, $D \supseteq C$, $d \geq b$. Notice that the choices of g_1^b, $g_2^{\overline{0}}, \dots$, $g_k^{\overline{0}}$ must exist because of bound-monotonicity and coalition-monotonicity. Moreover, the choices of t_i and h_i, where $i = 1, \dots, k$, are arbitrary. Let $\sigma_C = \{(g_1^b, t_1, h_1), (g_2^{\overline{0}}, t_2, h_2), \dots, (g_k^{\overline{0}}, t_k, h_k)\}$.

Let $\sigma_{\overline{C}}$ be an arbitrary joint action for $\overline{C}$. Let $\sigma = (\sigma_C, \sigma_{\overline{C}})$ and let g be the set of the option functions from σ. By the choice of σ_C, C must be a subset of a partition C_l in $P(g)$. Then, we have

$$o(s, \sigma) = h_{t_0}(G(g)) \in G(g) \subseteq g(C_l) = X$$

Hence, $X \in E_F(s)(C^b)$.

For the induction step, let $X \in E(s)(C^b)$ where $t(b) > 1$. If $X \in E(s)(C^{b'})$ for some $b' < b$, by the induction hypothesis, we have $X \in E_F(s)(C^{b'})$. Therefore, bound-monotonicity implies that $X \in E_F(s)(C^b)$.

If $X \notin E(s)(C^{b'})$ for any $b' < b$, by transitivity there are $b_1 \otimes b_2 = b$ such that

$$\{s' \mid X \in E(s')(C^{b_2})\} \in E(s)(C^{b_1})$$

By the induction hypothesis, we have

$$\{s' \mid X \in E(s')(C^{b_2})\} \in E_F(s)(C^{b_1})$$

and

$$\begin{aligned}&\{s' \mid X \in E(s')(C^{b_2})\}\\ &\subseteq \{s' \mid X \in E_F(s')(C^{b_2})\}\end{aligned}$$

By outcome-monotonicity, we have

$$\{s' \mid X \in E_F(s')(C^{b_2})\} \in E_F(s)(C^{b_1})$$

Hence, by super-transitivity $X \in E_F(s)(C^b)$.

For the other direction, we consider two cases where $C = N$ and $C \subset N$. Assume that $X \notin E(s)(N^b)$. By N-maximality, we obtain $\overline{X} \in E(s)(\varnothing^b)$. However, the previous proof implies that $\overline{X} \in E_F(s)(\varnothing^b)$. As E_F is RB-playable, by regularity we have $X \notin E_F(s)(N^b)$.

For the case of $C \subset N$, the proof is done by induction on bounds. Assume that $X \notin E(s)(C^b)$ where $t(b) = 1$ and $C \subset N$, i.e. there is $i_0 \in N \setminus C$. Let $\sigma_C = \{(g^{b_i}_{(s,i)}, t_i, h_i) \mid i \in C\}$ be an joint action for C such that $Res(\sigma_C) \leq b$. We choose a strategy $\sigma_{\overline{C}} = \{(g^{b_i}_{(s,i)}, t_i, h_i) \mid i \in \overline{C}\}$ for $\overline{C}$ such that:

- $b_i = \overline{0}$ for all $i > k$
- $g^{b_i}_{(s,i)}(D^d) = S$ for all $i \in \overline{C}$, $D \supseteq \overline{C}$, $d \geq b_i$
- $(\sum_{i \in N} t_i \mod n) + 1 = i_0$
- h_i for $i \neq i_0$ is arbitrary, we will select h_{i_0} shortly

As before, let $\sigma = (\sigma_C, \sigma_{\overline{C}})$ and g the collection of option functions in σ. We use notation $b_D = \oplus_{i \in D} b_i$ for any $D \subseteq N$.

By the choice of option functions in $\sigma_{\overline{C}}$, it follows that $\overline{C}$ is the subset of some partition C_l of $P(g)$. For other partitions, super-additivity shows that $G(g) \in E(s)(\overline{C_l}^{b_{\overline{C_l}}})$. By coalition-monotonicity and bound-monotonicity, we have that $G(g) \in E(s)(C^b)$. As $X \notin E(s)(C^b)$, it follows that $G(g) \nsubseteq X$ by outcome-monotonicity, i.e. there is some $s_0 \in G(g) \setminus X$. Select h_{i_0} such that $h_{i_0}(G(g)) = s_0$, then

$$o(s,\sigma) = h_{i_0}(G(g)) = s_0 \notin X$$

Hence, $X \notin E_F(s)(C^b)$.

In the induction step, assume that $X \notin E(s)(C^b)$ where $t(b) > 1$. Bound-monotonicity shows that for all $b' \leq b$, $X \notin E(s)(C^{b'})$ and super-transitivity implies that for all $b_1 \otimes b_2 = b$,

$$\{s' \mid X \in E(s')(C^{b_2})\} \notin E(s)(C^{b_1})$$

By the induction hypothesis, we have that for all $b' < b$, $X \notin E_F(s)(C^{b'})$ and for all $b_1 \otimes b_2 = b$,

$$\{s' \mid X \in E(s')(C^{b_2})\} \notin E_F(s)(C^{b_1})\}$$

and $\{s' \mid X \in E(s')(C^{b_2})\} = \{s' \mid X \in E_F(s')(C^{b_2})$. Then, $\{s' \mid X \in E_F(s')(C^{b_2})\} \notin E_F(s)(C^{b_1})$. Therefore, transitivity implies that $X \notin E_F(s)(C^b)$.

□

4.5 Axiomatisation of RBCL

In this section we define models based on playable effectivity structures, and give a complete axiomatisation for the set of validities in those models.

Definition 11. *A resource-bounded effectivity model $M = (S, E, V)$ is a triple consisting of a non-empty set of states, a RB-playable effectivity structure and a valuation function $V : Prop \to \wp(S)$. The truth definition for $[C^b]$ modalities is as follows:*

- $M, s \vDash [C^b]\varphi$ *iff* $\varphi^M \in E(s)(C^b)$ *where* $\varphi^M = \{s' \mid M, s' \vDash \varphi\}$

Notice that in the above definition, we do not define the truth for $[C^b]$ modalities in two separate cases, one for non-empty coalitions C and one for empty coalitions. This is because the two cases have been covered by the RB-playable effectivity structure E, one may refer to the correspondence of effectivity structures to RBA frames in Section 4.4.3 for more details.

For convenience, we also extend the definition of the function V for a given model $M = (S, E, V)$ as follows, $V(\varphi) = \{s \in S \mid M, s \vDash \varphi\}$.

Theorem 4. *The sets of formulas valid in multi-step RBA models and in resource-bounded effectivity models are equal.*

This follows from the correspondence between RBA frames and RB-playable effectivity structures, and the correspondence between the two truth definitions. Therefore the next result also provides an axiomatisation for RBCL.

Theorem 5. *The following set of axiom schemas and inference rules provides a sound and complete axiomatisation of the set of validities over all resource-bounded effectivity models:*

A0-A5, MP *and* **Equivalence** *given above*

A6a $[C^b]\varphi \wedge [D^d]\psi \to [(C \cup D)^{b \oplus d}](\varphi \wedge \psi)$ *where* C *and* D *are both non-empty and disjoint, and* $t(b) = t(d) = 1$

A6b $[\varnothing^b]\varphi \wedge [C^b]\psi \to [C^b](\varphi \wedge \psi)$ *where* C *is either* $\varnothing$ *or* N

A7 $[C^{b_1}][C^{b_2}]\varphi \to [C^{b_1 \otimes b_2}]\varphi$ *for* $C \neq \varnothing$

A8 $[C^b]\varphi \to \bigvee_{b' < b}[C^{b'}]\varphi \vee \bigvee_{b_1 \otimes b_2 = b}[C^{b_1}][C^{b_2}]\varphi$ *for all* $C \neq \varnothing$ *and* $t(b) > 1$

Proof. The proof of soundness is straightforward. We prove completeness by first constructing a canonical model. Let us denote by $\vdash_\Lambda$ derivability in the axiom system above. Let S^Λ be the set of all Λ-maximally consistent sets. For any formula φ, we denote $\tilde{\varphi} = \{s \in S^\Lambda \mid \varphi \in s\}$. Then, we define the canonical valuation function $V^\Lambda(p) = \tilde{p}$.

We define the canonical effectivity structure E^Λ by induction on b as follows:

- For all b such that $t(b) = 1$ and $C \neq N$, $X \in E^\Lambda(s)(C^b)$ iff $\exists \tilde{\varphi} \subseteq X : [C^b]\varphi \in s$.
- For all b such that $t(b) = 1$, $X \in E^\Lambda(s)(N^b)$ iff $\overline{X} \in E^\Lambda(s)(\varnothing^b)$.
- For all b such that $t(b) > 1$ and $C \neq \varnothing$, $X \in E^\Lambda(s)(C^b)$ iff $X \in E^\Lambda(s)(C^{b'})$ for some $b' < b$ or there are $b_1 \otimes b_2 = b$ such that $\{s' \mid X \in E^\Lambda(s')(C^{b_2})\} \in E^\Lambda(s)(C^{b_1})$.
- For all b such that $t(b) > 1$, $X \in E^\Lambda(s)(\varnothing^b)$ iff $\overline{X} \notin E^\Lambda(s)(N^b)$.

The following property $(*)$ is crucial for the proof:

$$(*) \quad \tilde{\varphi} \in E^\Lambda(s)(C^b) \text{ iff } [C^b]\varphi \in s$$

We prove it by induction on the bounds. In the base case, assume that $\tilde{\varphi} \in E^{\Lambda}(s)(C^b)$ for some $t(b) = 1$. For $C \neq N$, $\tilde{\varphi} \in E^{\Lambda}(s)(C^b)$ iff $\exists \tilde{\psi} \subseteq \tilde{\varphi} : [C^b]\psi \in s$. Then we have $\psi \to \varphi \in s$; together with **RM**, it is implied that $[C^b]\varphi \in s$. In the inverse direction, $[C^b]\varphi \in s$ implies directly that $\tilde{\varphi} \in E^{\Lambda}(s)(C^b)$ by definition of E^{Λ}.

If $C = N$, we have $\tilde{\varphi} \in E^{\Lambda}(s)(N^b)$ iff $\widetilde{\neg\varphi} \notin E^{\Lambda}(s)(\varnothing^b)$ iff $\neg[\varnothing^b]\neg\varphi \in s$ (as just proved) iff $[N^b]\varphi \in s$ (by axiom $A3$).

For the induction step, assume that $\tilde{\varphi} \in E^{\Lambda}(s)(C^b)$ where $t(b) > 1$. For $C \neq \varnothing$, there are two cases to consider. (1) $\tilde{\varphi} \in E^{\Lambda}(s)(C^{b'})$ for some $b' < b$. By the induction hypothesis, we have $[C^{b'}]\varphi \in s$. Then, axiom **A5** implies that $[C^b]\varphi \in s$. (2) There are $b_1 \otimes b_2 = b$ such that

$$\{s' \mid \tilde{\varphi} \in E^{\Lambda}(s')(C^{b_2})\} \in E^{\Lambda}(s)(C^{b_1}).$$

Let $\psi = [C^{b_2}]\varphi$, by the induction hypothesis, we have $\tilde{\psi} = \{s' \mid \tilde{\varphi} \in E^{\Lambda}(s')(C^{b_2})\}$, thus, $\tilde{\psi} \in E^{\Lambda}(s)(C^{b_1})$. Again, induction hypothesis gives us $[C^{b_1}][C^{b_2}]\varphi \in s$. Therefore, by axiom **A7**, we have $[C^b]\varphi \in s$.

For the inverse direction, assume that $[C^b]\varphi \in s$ for some $t(b) > 1$. By axiom **A8**, there are two cases to consider. If $[C^{b'}]\varphi \in s$ for some $b' < b$, then the induction hypothesis implies that $\tilde{\varphi} \in E^{\Lambda}(s)(C^{b'})$. Hence, by the definition of E^{Λ}, we have $\tilde{\varphi} \in E^{\Lambda}(s)(C^b)$. In the second case, there are $b_1 \otimes b_2 = b$ such that $[C^{b_1}][C^{b_2}]\varphi \in s$. Similar to the proof above, let $\psi = [C^{b_2}]\varphi$, the induction hypothesis implies that $\tilde{\psi} \in E^{\Lambda}(s)(C^{b_1})$. As we have that $\tilde{\psi} = \{s' \mid \tilde{\varphi} \in E^{\Lambda}(s')(C^{b_2})\}$, this shows

$$\{s' \mid \tilde{\varphi} \in E^{\Lambda}(s')(C^{b_2})\} \in E^{\Lambda}(s)(C^{b_1}).$$

By the definition of E^{Λ}, we obtain $\tilde{\varphi} \in E^{\Lambda}(s)(C^b)$.

If $C = \varnothing$, we have $\tilde{\varphi} \in E^{\Lambda}(s)(\varnothing^b)$ iff $\widetilde{\neg\varphi} \notin E^{\Lambda}(s)(N^b)$ iff $\neg[N^b]\neg\varphi \in s$ (as just proved) iff $[\varnothing^b]\varphi \in s$ (by axiom **A3**).

Let us prove that E^{Λ} satisfies properties of RB-playability except N-determinacy by exploiting the property $(*)$, the definition of E^{Λ} and the axioms of Λ.

1. As $[C^b]\top \in s$ for all $s \in S^{\Lambda}$, it means by (*) that $S^{\Lambda} = \tilde{\top} \in E^{\Lambda}(s)(C^b)$.

2. Similarly, $[C^b]\bot \notin s$ for all $s \in S^{\Lambda}$, it implies by (*) that that $\varnothing = \tilde{\bot} \notin E^{\Lambda}(s)(C^b)$.

3. We prove outcome-monotonicity by induction on bounds. Assume that $X \in E^{\Lambda}(s)(C^b)$.

- If $t(b) = 1$ and $C \neq N$, $X \in E^{\Lambda}(s)(C^b)$ iff there exists φ such that $\tilde{\varphi} \subseteq X$ and $[C^b]\varphi \in s$. Hence, for all $X' \supseteq X$, we have that $\tilde{\varphi} \subseteq X'$. This implies by the definition of E^{Λ} that $X' \in E^{\Lambda}(s)(C^b)$
- If $t(b) = 1$, $X \in E^{\Lambda}(s)(N^b)$ iff $\overline{X} \notin E^{\Lambda}(s)(\varnothing^b)$. Let $X' \supseteq X$, it implies that $\overline{X'} \subseteq \overline{X}$. Assume to the contrary that $X' \notin E^{\Lambda}(s)(N^b)$. Then, $\overline{X'} \in E^{\Lambda}(s)(\varnothing^b)$. As $\overline{X'} \subseteq \overline{X}$, this implies that $\overline{X} \in E^{\Lambda}(s)(\varnothing^b)$ which is a contradiction.
- If $t(b) > 1$ and $C \neq \varnothing$. If $X \in E^{\Lambda}(s)(C^{b'})$ for some $b' < b$, the induction hypothesis shows that $X' \in E^{\Lambda}(s)(C^{b'})$ for all $X' \supseteq X$. Then, by the definition of E^{Λ} we have $X' \in E^{\Lambda}(C^b)(s)$. If $X \notin E^{\Lambda}(s)(C^{b'})$ for all $b' < b$. By the definition of E^{Λ}, there are $b_1 \otimes b_2 = b$ and
$$\{s' \mid X \in E^{\Lambda}(s')(C^{b_2})\} \in E^{\Lambda}(s)(C^{b_1})$$
Let $X' \supseteq X$, by the induction hypothesis we have
$$\begin{aligned}&\{s' \mid X \in E^{\Lambda}(s')(C^{b_2})\} \subseteq \{s' \mid X' \in E^{\Lambda}(s')(C^{b_2})\}\\ &\Rightarrow \{s' \mid X' \in E^{\Lambda}(s')(C^{b_2})\} \in E^{\Lambda}(s)(C^{b_1})\end{aligned}$$
By the definition of E^{Λ}, we have $X' \in E^{\Lambda}(s)(C^b)$.
- If $t(b) > 1$, $X \in E^{\Lambda}(s)(\varnothing^b)$ iff $\overline{X} \notin E^{\Lambda}(s)(N^b)$. Let $X' \supseteq X$, assume to the contrary that $X' \notin E^{\Lambda}(s)(\varnothing^b)$. This implies that $\overline{X'} \in E^{\Lambda}(s)(N^b)$. By the previous proof, we have $\overline{X} \in E^{\Lambda}(s)(N^b)$ as $\overline{X'} \subseteq \overline{X}$, which is a contradiction.

4. N-maximality follows directly from the definition of E^{Λ} for N when $t(b) = 1$ and $\varnothing$ when $t(b) > 1$.

5. Similarly, N-regularity also follows directly from the definition of E^{Λ} for N when $t(b) = 1$ and $\varnothing$ when $t(b) > 1$.

6. In order to show super-additivity, we consider the following three different cases. Let $t(b) = t(d) = 1$, $C \cap D = \varnothing$ with $X \in E^{\Lambda}(s)(C^b)$ and $Y \in E^{\Lambda}(s)(D^d)$.

   - If both C and D are not empty by the definition of E^{Λ}, we have that there are φ and ψ such that $\tilde{\varphi} \subseteq X$, $\tilde{\psi} \subseteq Y$, $[C^b]\varphi$ and $[D^d]\psi \in s$. According to axiom **A6a**, we have $[(C \cup D)^{b \oplus d}](\varphi \wedge \psi) \in s$. Obviously, $\tilde{\varphi} \cap \tilde{\psi} \subseteq X \cap Y$, hence $X \cap Y \in E^{\Lambda}(s)((C \cup D)^{b \oplus d})$.
   - If $C = \varnothing$, $b = d$ and $D = \varnothing$, the proof is similar to the one above except that axiom **A6b** gives us $[D^d](\varphi \wedge \psi) \in s$. Hence, $X \cap Y \in E^{\Lambda}(s)(D^d)$.

- If $C = \varnothing$, $b = d$ and $D = N$, we need to show that $X \cap Y \in E^{\Lambda}(N^b)(s)$. Assume to the contrary that $X \cap Y \notin E^{\Lambda}(N^b)(s)$, then N-maximality, which has been proved above, implies that $\overline{X \cap Y} \in E^{\Lambda}(\varnothing^b)(s)$. Then, by the previous case of super-additivity, we have $X \cap \overline{Y} \in E^{\Lambda}(\varnothing^b)(s)$. As we already showed outcome-monotonicity, $\overline{Y} \in E^{\Lambda}(\varnothing^b)(s)$. However, by N-regularity, we have $Y \notin E^{\Lambda}(N^b)(s)$ which is a contradiction.

7. Super-transitivity follows directly from the definition of E^{Λ} when $t(b) > 1$.

8. Similarly, transitivity follows directly from the definition of E^{Λ} when $t(b) > 1$.

9. Finally, we show that E^{Λ} is indeed bound-monotonic. Let us assume that $X \in E^{\Lambda}(s)(C^b)$.

   - If $t(b) = 1$ and $C \neq N$, $X \in E^{\Lambda}(s)(C^b)$ iff there exists φ such that $\tilde{\varphi} \subseteq X$ and $[C^b]\varphi \in s$. By axiom **A5**, we have for any $d \geq b$ or $d \leq b$ if $C \neq \varnothing$ or otherwise, respectively, $[C^d]\varphi \in s$. Then, by the definition of E^{Λ}, $X \in E^{\Lambda}(s)(C^d)$.
   - If $t(b) = 1$ and $C = N$, $X \in E^{\Lambda}(s)(N^b)$ iff $\overline{X} \notin E^{\Lambda}(s)(\varnothing^b)$. Then, axiom **A5** implies that $\overline{X} \notin E^{\Lambda}(s)(\varnothing^d)$ for any $d \geq b$. Once again, by the definition of E^{Λ}, we have $X \in E^{\Lambda}(s)(N^d)$.
   - If $t(b) > 1$ and $C \neq \varnothing$, it is straightforward from the definition of E^{Λ} that $X \in E^{\Lambda}(s)(C^d)$ for any $d \geq b$.
   - If $t(b) > 1$ and $C = \varnothing$, $X \in E^{\Lambda}(s)(\varnothing^b)$ iff $\overline{X} \notin E^{\Lambda}(s)(N^b)$. By the proof of the previous case, we have $\overline{X} \notin E^{\Lambda}(s)(N^d)$ for any $d \leq b$. Hence, $X \in E^{\Lambda}(s)(\varnothing^d)$.

Since we have already shown (*), the following truth lemma is straightforward:

$$(**) \quad M^{\Lambda}, s \models \varphi \text{ iff } \varphi \in s$$

As usual, we show (**) by induction on the structure of φ. The cases for proposition variables and usual Boolean connectives are trivial, so we omit them here.

- If $\varphi = [C^b]$, then,

$$\begin{aligned} M^{\Lambda}, s \models [C^b]\psi &\Leftrightarrow \psi^{M^{\Lambda}} \in E^{\Lambda}(s)(C^b) \\ &\Leftrightarrow \tilde{\psi} \in E^{\Lambda}(s)(C^b) \quad \text{by the induction hypothesis} \\ &\Leftrightarrow [C^b]\psi \in s \quad \text{by (*)} \end{aligned}$$

By showing (**), it is obvious that for any consistent formula φ, there is a state $s_0 \in S^\Lambda$ such that $\varphi \in s$, hence $M^\Lambda, s_0 \vDash \varphi$. In other words, φ is satisfiable in the canonical model M^Λ; however, notice that E^Λ is not proved to be RB-playable. Therefore, in order to provide a model which has an RB-playable effectivity structure and satisfies φ, we present the notion of filtration as for the case of CL [Hansen & Pauly, 2002].

Let $M = (S, E, V)$ be a resource-bounded effectivity model and Γ be a set of formulas, we define the equivalent relation $\equiv_\Gamma$ over Γ on S as follows: for any $s, t \in S$, $s \equiv_\Gamma t$ iff $\forall \varphi \in \Gamma$, $M, s \vDash \varphi$ iff $M, t \vDash \varphi$.

For convenience, we shall denote $|s| = \{t \in S \mid s \equiv_\Gamma t\}$, then $|X| = \{|s| \mid s \in X\}$ where $X \subseteq S$.

We define the notion of filtration for resource-bounded effectivity models, where the effectivity structure has all properties of RB-playability except N-determinacy, as follows.

Definition 12. *Given a resource-bounded effectivity model $M = (S, E, V)$ and a sub-formula closed set Γ of formulas, a model $M^f_\Gamma = (S^f, E^f, V^f)$ is a filtration of M over Γ iff the following conditions hold:*

1. $S^f = |S|$,
2. *For any $C \subseteq N$ and $b \in \mathbb{B}$ where $t(b) = 1$, $\forall \varphi \in \Gamma$: $\varphi^M \in E(s)(C^b)$ implies $|\varphi^M| \in E^f(|s|)(C^b)$,*
3. *For any $C \subseteq N$ and $b \in \mathbb{B}$ where $t(b) = 1$, if $X \in E^f(|s|)(C^b)$ then $\forall \varphi \in \Gamma$: $\{t \mid |t| \in X\} \subseteq \varphi^M$ implies $\varphi^M \in E(s)(C^b)$,*
4. *For any $C \neq \varnothing$ and $b \in \mathbb{B}$ where $t(b) > 1$, $X \in E^f(|s|)(C^b)$ iff $X \in E^f(|s|)(C^{b'})$ for some $b' < b$ or there are $b_1 \otimes b_2 = b$ such that $\{|s'| \mid X \in E^f(|s'|)(C^{b_2})\} \in E^f(|s|)(C^{b_1})$,*
5. *For any $b \in \mathbb{B}$ where $t(b) > 1$, $X \in E^f(|s|)(\varnothing^b)$ iff $\bar{X} \notin E^f(|s|)(\varnothing^b)$,*
6. $V^f(p) = |V(p)|$ *for all $p \in Prop$.*

We have the following result of truth preservation through filtration.

Lemma 7. *Given $M^f_\Gamma = (S^f, E^f, V^f)$ to be a filtration of $M = (S, E, V)$, for all $\varphi \in \Gamma$ and $s \in S$, we have that:*

$$M, s \vDash \varphi \textit{ iff } M^f_\Gamma, |s| \vDash \varphi$$

Proof. The proof is done by induction on the structure of φ:

- If $\varphi = p$ where $p \in Prop$, $M, s \models p$ iff $s \in V(p)$ iff $|s| \in |V(p)|$ iff $M^f_\Gamma, |s| \models p$.
- If $\varphi = \neg\psi$, $M, s \models \neg\psi$ iff $M, s \not\models \psi$ iff $M^f_\Gamma, |s| \not\models \psi$ (by the induction hypothesis) iff $M^f_\Gamma, |s| \models \neg\psi$.
- If $\varphi = \psi_1 \vee \psi_2$, $M, s \models \psi_1 \vee \psi_2$ iff $M, s \models \psi_1$ or $M, s \models \psi_2$. Without loss of generality, we assume that $M, s \models \psi_1$ iff $M^f_\Gamma, |s| \models \psi_1$ (by the induction hypothesis) iff $M^f_\Gamma, |s| \models \psi_1 \vee \psi_2$.
- If $\varphi = [C^b]\psi$, where $t(b) = 1$, $M, s \models [C^b]\psi$ iff $\psi^M \in E(s)(C^b)$ iff $|\psi^M| \in E^f(|s|)(C^b)$ (by the definition of E^f) iff $M^f_\Gamma, |s| \models [C^b]\psi$.
- The last case when $\varphi = [C^b]\psi$, where $t(b) > 1$, can be done similarly by using the definition of E^f and the induction hypothesis.

$\square$

Given a formula φ, we define $sub(\varphi)$ to be set of all sub-formulas of φ including itself, and $esub(\varphi)$ to be the boolean closure (closed under negations and disjunctions) of $sub(\varphi)$ upto tautology equivalence. Firstly, $esub(\varphi)$ is finite since $|sub(\varphi)| \leq |\varphi|$; hence the cardinality of $esub(\varphi)$ is no more than $2^{2^{|\varphi|}}$. We define the filtration of a resource-bounded effectivity model for a formula φ as follows:

Definition 13. *Given a resource-bounded effectivity model $M = (S, E, V)$ and a formula φ, the filtration model $M^\varphi = (S^\varphi, E^\varphi, V^\varphi)$ of M for φ is defined as follows:*

1. *$\Gamma = esub(\varphi)$,*
2. *$S^\varphi = |S|$ (then, S^φ has at most $2^{2^{2^{|\varphi|}}}$ states),*
3. *For any $C \neq N$ and b such that $t(b) = 1$, $X \in E^\varphi(|s|)(C^b)$ iff $\exists \varphi \in \Gamma$ such that $\varphi^M \subseteq \{t \mid |t| \in X\}$ and $\varphi^M \in E(s)(C^b)$,*
4. *For any b such that $t(b) = 1$, $X \in E^\varphi(|s|)(N^b)$ iff $\bar{X} \notin E^\varphi(|s|)(\varnothing^b)$,*
5. *For any $C \neq \varnothing$ and $b \in \mathbb{B}$ where $t(b) > 1$, $X \in E^\varphi(|s|)(C^b)$ iff $X \in E^\varphi(|s|)(C^{b'})$ for some $b' < b$ or there are $b_1 \otimes b_2 = b$ such that $\{|s'| \mid X \in E^\varphi(|s'|)(C^{b_2})\} \in E^\varphi(|s|)(C^{b_1})$,*
6. *For any $b \in \mathbb{B}$ where $t(b) > 1$, $X \in E^\varphi(|s|)(\varnothing^b)$ iff $\bar{X} \notin E^\varphi(|s|)(\varnothing^b)$,*

7. $V^\varphi(p) = |V(p)|$ *for all* $p \in$ *Prop.*

Let us show the following result.

Lemma 8. $M^\varphi = (S^\varphi, E^\varphi, V^\varphi)$ *is a filtration.*

Proof. The proof is straightforward. According to the definition of a filtration and M^φ, we only need to show that M^φ satisfies the second and third requirements in Definition 12.

- Let $C \neq N$, assume that $\psi \in esub(\varphi)$ and $\psi^M \in E(s)(C^b)$, it follows directly from the definition of M^φ that $|\psi^M| \in E(|s|)(C^b)$.

- Assume that $\psi \in esub(\varphi)$ and $\psi^M \in E(s)(N^b)$, we need to prove that $|\psi^M| \in E^\varphi(|s|)(N^b)$. Assume to the contrary that $|\psi^M| \notin E^\varphi(|s|)(N^b)$, this implies $|\neg\psi^M| \in E^\varphi(|s|)(\varnothing^b)$; thus, $\exists \psi' \in esub(\varphi)$ such that $\psi'^M \subseteq \{t \mid |t| \in |\neg\psi^M|\}$ and $\psi'^M \in E(s)(\varnothing^b)$. Then, $\psi'^M \subseteq \neg\psi^M$, hence, $\neg\psi^M \in E(s)(\varnothing^b)$ as E is outcome-monotonicity. By N-regularity, we obtain $\psi^M \notin E(s)(N^b)$ which is a contradiction.

- Let $C \neq N$, assume that $X \in E^\varphi(|s|)(C^b)$ and $\psi \in esub(\varphi)$ such that $\{t \mid |t| \in X\} \subseteq \psi^M$. We have by the definition of M^φ that $\exists \psi' \in esub(\varphi)$: $\psi'^M \subseteq \{t \mid |t| \in X\}$ and $\psi'^M \in E(s)(C^b)$. Thus, $\psi'^M \subseteq \psi^M$; and by outcome monotonic, we obtain $\psi^M \in E(s)(C^b)$.

- Assume that $X \in E^\varphi(|s|)(N^b)$ and $\psi \in esub(\varphi)$ such that $\{t \mid |t| \in X\} \subseteq \psi^M$. We have by the definition of M^φ that $\bar{X} \notin E^\varphi(|s|)(N^b)$ iff for all $\psi' \in esub(\varphi)$, if $\psi'^M \subseteq \{t \mid |t| \in \bar{X}\}$ then $\psi'^M \notin E(s)(\varnothing^b)$. As $\{t \mid |t| \in X\} \subseteq \psi^M$, we have that $\{t \mid |t| \in \bar{X}\} \supseteq \neg\psi^M$; hence $\neg\psi^M \notin E(s)(\varnothing^b)$. By N-regularity, we obtain $\psi^M \in E(s)(N^b)$.

□

Finally, we have the following result.

Lemma 9. *The effectivity structure* E^φ *of the filtration* M^φ *is RB-playability.*

The proof of the above lemma proceeds by first showing that E^φ has all properties of RB-playability except N-determinacy. We omit the proof as it is similar to that of showing E^Λ has all properties of RB-playability except N-determinacy. Since S^φ is finite, N-determinacy follows from the others properties of RB-playability. Therefore, E^φ is RB-playable. This result also implies the finite model property of RBCL where we have that any satisfiable formula in RBCL is also satisfied in a finite model.

We complete the proof of completeness for RBCL as follows. Let $M^{\Lambda,\varphi}$ be the filtration model of M^{Λ}, we have by Lemma 7 that $M^{\Lambda,\varphi}, |s_0| \vDash \varphi$. Let $\nvdash_{\Lambda} \varphi$, i.e. $\neg\varphi$ is consistent. Hence, $\neg\varphi$ is satisfiable. Therefore, φ is not valid.

□

4.6 Satisfiability problem

The last result about the finite model property of RBCL in the previous section suggests an exhausted way to search for a finite model to satisfy a given formula. In this section we give an alternative proof of decidability for RBCL, which is more efficient than the exhausted search, by providing an algorithm which determines the satisfiability of a given formula φ. Similar to Coalition Logic, our algorithm is developed by adopting the approach presented in [Vardi et al., 1989]. In principle, the algorithm will try to guess a suitable valuation for the set of more-or-less sub-formulas generated by φ which satisfies a number of conditions. Such existing valuation will help construct a model for φ, or in other words, assure the satisfiability of φ.

Given a formula φ, we define a set $sub(\varphi)$ inductively as follows.

- $sub(p) = \{p\}$ for any propositional variable p
- $sub(\neg\psi) = \{\neg\psi\} \cup sub(\psi)$
- $sub(\psi_1 \vee \psi_2) = \{\psi_1 \vee \psi_2\} \cup sub(\psi_1) \cup sub(\psi_2)$
- $sub([C^b]\psi) = \{[C^b]\psi\} \cup sub(\psi)$ for $t(b) = 1$ and $C \neq N$
- $sub([N^b]\psi) = \{[N^b]\psi\} \cup sub(\neg[\varnothing^b]\neg\psi)$ for $t(b) = 1$
- $sub([C^b]\psi) = \{[C^b]\psi\} \cup \bigcup_{b'<b} sub([C^{b'}]\psi) \cup \bigcup_{b_1 \otimes b_2 = b} sub([C^{b_1}][C^{b_2}]\psi)$ for $t(b) > 1$ and $C \neq \varnothing$
- $sub([\varnothing^b]\psi) = \{[\varnothing^b]\psi\} \cup sub(\neg[N^b]\neg\psi)$ for $t(b) > 1$

It is easy to show that $sub(\varphi)$ is finite. Then, we define the closure $cl(\varphi)$ of a given formula φ as follows.

$$\begin{aligned} cl(\varphi) \;=\; & \{\psi, \neg\psi \mid \psi \in sub(\varphi)\} \cup \\ & \{[\varnothing^b]\neg\psi, \neg[\varnothing^b]\neg\psi \mid [N^b]\psi \in sub(\varphi)\} \cup \\ & \{[N^b]\neg\psi, \neg[N^b]\neg\psi \mid [\varnothing^b]\psi \in sub(\varphi)\} \end{aligned}$$

Notice that we identify $\neg\neg\psi$ as ψ. We have the following definition of valuations. Moreover, we denote $\bar{0}$ as the smallest bound of which all components are 0 except for the time component which is 1.

Let us consider an example. Assume that $N = \{1,2\}$ (i.e. there are two agents in the system) and $|R| = 1$ and one of the resources in R is time. We consider the following formula $\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p$. Then, the set $cl(\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p)$ contains the following positive sub-formulas (apart from $\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p$):

$[\{1,2\}^{(2,1)}]p$	$[\varnothing^{(2,1)}]\neg p$	$[\{2\}^{0,1}][\{2\}^{0,1}]p$
$[\{2\}^{(0,1)}]p$	$[\{2\}^{1,1}]p$	$[\{2\}^{1,1}][\{2\}^{0,1}]p$
$[\{2\}^{0,2}]p$	$[\{2\}^{1,2}]p$	$[\{2\}^{0,1}][\{2\}^{1,1}]p$
p		

Then, $cl(\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p)$ contains the above formulas together with their negations.

Definition 14. *A valuation for a given formula φ is a mapping $v : cl(\varphi) \to \{0,1\}$ which satisfies the following conditions:*

1. $v(\varphi) = 1$
2. $v(\top) = 1$
3. $v(\neg\psi) = 1 - v(\psi)$
4. $v(\psi_1 \vee \psi_2) = \max(v(\psi_1), v(\psi_2))$
5. $v([\varnothing^b]\psi) = v(\neg[N^b]\neg\psi)$
6. $v([C^b]\psi) \leq v([C^d]\psi)$ *where* $b \leq d$ *if* $C \neq \varnothing$ *or* $b \geq d$ *otherwise*
7. $v([C^b]\psi) = \max\{\bigcup_{b'<b}\{v([C^{b'}]\psi)\} \cup \bigcup_{b_1\otimes b_2=b}\{v([C^{b_1}][C^{b_2}]\psi)\}\}$ *where* $t(b) > 1$ *and* $C \neq \varnothing$

For instance, Figure 4.2 depicts a valuation for the formulas in the set $cl(\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p)$. Notice that we only provide the valuation for the positive formulas in $cl(\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p)$ since the negation ones can be determined by the definition $v(\neg\psi) = 1 - v(\psi)$.

In the following lemma, we determine when such a valuation is qualified as a starting point to help building up a model for φ.

φ	$v(\varphi)$	φ	$v(\varphi)$	φ	$v(\varphi)$
$[\{1,2\}^{(2,1)}]p$	0	$[\varnothing^{(2,1)}]\neg$ p	1	$[\{2\}^{0,1}][\{2\}^{0,1}]p$	0
$[\{2\}^{(0,1)}]p$	0	$[\{2\}^{1,1}]p$	0	$[\{2\}^{1,1}][\{2\}^{0,1}]p$	0
$[\{2\}^{0,2}]p$	0	$[\{2\}^{1,2}]p$	1	$[\{2\}^{0,1}][\{2\}^{1,1}]p$	1
p	0				

FIGURE 4.2: A valuation for $cl(\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p)$.

Lemma 10. *A formula φ is satisfiable if and only if there exists a valuation v for φ such that*

1. *If there are $[C_1^{b_1}]\psi_1, \ldots, [C_k^{b_k}]\psi_k \in cl(\varphi)$ for some $k > 0$ such that:*
   - $t(b_j) = 1$ *for all* $j \leq k$
   - $C_1, \ldots, C_k$ *are pairwise disjoint*
   - *for any* $[C_j^{b_j}]\psi_j$ *such that* $C_j = \varnothing$, $b_j \geq \oplus_{C_{j'} \neq \varnothing} b_{j'}$
   - $v([C_j^{b_j}]\psi_j) = 1$ *for all* $j \leq k$

   then $\wedge_{j \leq k}\psi_j$ is satisfiable.

2. *If there are $[C_1^{b_1}]\psi_1, \ldots, [C_k^{b_k}]\psi_k \in cl(\varphi)$ for some $k > 0$ such that:*
   - $t(b_j) = 1$ *for all* $j \leq k$
   - $C_1, \ldots, C_{k-1}$ *are pairwise disjoint and all non-empty*
   - $\bigcup_{j<k} C_j \subseteq C_k$
   - $\oplus_{j<k} b_j = b_k$
   - $v([C_j^{b_j}]\psi_j) = 1$ *for all* $j < k$
   - $v([C_k^{b_k}]\psi_k) = 0$

   then $\wedge_{j<k}\psi_j \wedge \neg\psi_k$ is satisfiable.

Proof. Firstly, we prove the left-to-right direction by defining a valuation based on the model satisfying the formula φ. In particular, let us assume that φ is satisfiable by a model $M = (S, E, V)$ at some state $s \in S$. We define a valuation v for $cl(\varphi)$ as follows:

$$v(\psi) = \begin{cases} 1 & \text{if } M, s \models \psi \\ 0 & \text{otherwise} \end{cases}$$

Based on the definition of the semantics for RBCL, it is straightforward to show that the defined valuation v satisfies all conditions listed in Definition 14. What remains is to prove that it also has the two properties listed in the lemma.

1. Assume that there are $[C_1^{b_1}]\psi_1, \ldots, [C_k^{b_k}]\psi_k \in cl(\varphi)$ for some $k > 0$ such that:

   - $t(b_j) = 1$ for all $j \leq k$
   - $C_1, \ldots, C_k$ are pairwise disjoint
   - for any $[C_j^{b_j}]\psi_j$ such that $C_j = \varnothing$, $b_j \geq \oplus_{C_{j'} \neq \varnothing} b_{j'}$
   - $v([C_j^{b_j}]\psi_j) = 1$ for all $j \leq k$

   That is $M, s \vDash [C_j^{b_j}]\psi_j$ for all $j \leq k$.

   If there is some non-empty C_j then, by super-additivity, we have that $M, s \vDash [C^b](\wedge_{j \leq k, C_j \neq \varnothing}\psi_j)$ where $C = \bigcup_{j \leq k} C_j$ and $b = \oplus_{j \leq k, C_j \neq \varnothing} b_j$. By coalition monotonicity, we have that $M, s \vDash [N^b](\wedge_{j \leq k, C_j \neq \varnothing}\psi_j)$. Furthermore, super-additivity implies that, for all $C_j = \varnothing$, $M, s \vDash [N^b](\wedge_{j \leq k}\psi_j)$. Because of playability, $\varnothing \notin E(s)(N^b)$, thus $V(\wedge_{j \leq k}\psi_j) \neq \varnothing$. Therefore, there exists $s' \in V(\wedge_{j \leq k}\psi_j)$ and it is straightforward that $M, s' \vDash \wedge_{j \leq k}\psi_j$.

   If there is no non-empty C_j then super-additivity gives us directly that $M, s \vDash [\varnothing^b](\wedge_{j \leq k}\psi_j)$ where $b = \min\{b_j \mid j \leq k\}$. Apply the same argument for playability, we have that there exists $s' \in V(\wedge_{j \leq k}\psi_j)$ and it is straightforward that $M, s' \vDash \wedge_{j \leq k}\psi_j$.

2. Assume that there are $[C_1^{b_1}]\psi_1, \ldots, [C_k^{b_k}]\psi_k \in cl(\varphi)$ for some $k > 1$ such that:

   - $t(b_j) = 1$ for all $j \leq k$
   - $C_1, \ldots, C_{k-1}$ are pairwise disjoint and all non-empty
   - $\bigcup_{j<k} C_j \subseteq C_k$
   - $\oplus_{j<k} b_j \leq b_k$
   - $v([C_j^{b_j}]\psi_j) = 1$ for all $j < k$
   - $v([C_k^{b_k}]\psi_k) = 0$

   That is $M, s \vDash [C_j^{b_j}]\psi_j$ for all $j < k$ and $M, s \nvDash [C_k^{b_k}]\psi_k$. By super-additivity, we have that $M, s \vDash [C^b](\wedge_{j<k}\psi_j)$ where $C = \bigcup_{j<k} C_j$ and $b = \oplus_{j<k} b_j$. That is $V(\wedge_{j<k}\psi_j) \in E(s)(C^b)$.

By coalition monotonicity and bound monotonicity, we have $V(\wedge_{j<k}\psi_j) \in E(s)(C_k^{b_k})$. Moreover, we already have $M, s \nvDash [C_k^{b_k}]\psi_k$, thus, $V(\psi_k) \notin E(s)(C_k^{b_k})$. Then, outcome monotonicity implies that $V(\psi_k) \not\supseteq V(\wedge_{j<k}\psi_j)$. Since $V(\wedge_{j<k}\psi_j) \neq \varnothing$, there must exist $s' \in V(\wedge_{j<k}\psi_j) \setminus V(\psi_k)$ and it is straightforward that $M, s' \vDash \wedge_{j<k}\psi_j \wedge \neg\psi_k$.

In the case when $k = 1$, the proof is slightly different from above as we do not have the set $V(\wedge_{j<k}\psi_j)$. However, we make the use of the first requirement of playability which states that $S \in E(s)(C_k^{b_k})$, therefore, $V(\psi_k) \neq S$. Hence, there also exists $s' \in S \setminus V(\psi_k)$ and it is obvious that $M, s' \vDash \neg\psi_k$.

Let us now prove the right-to-left direction of the lemma. The idea is that we construct a model satisfying the formula φ by collecting models which witness the satisfaction of the formulas in the two conditions of the lemma. That is for any tuple $([C_j^{b_j}]\psi_j)_{j\leq k}$ of $cl(\varphi)$ which corresponds to one of the two conditions of the lemma, as $\wedge_{i\leq k}\psi_j$ (or $\wedge_{i<k}\psi_j \wedge \neg\psi_k$) is satisfiable, there is a model M' which satisfies $\wedge_{i\leq k}\psi_j$ (or $\wedge_{i<k}\psi_j \wedge \neg\psi_k$) at some state s' of M'. The model M we construct to satisfy φ will be the union of all such witnessing models M' together with a new state s_0 at which φ will be satisfied. We define the assignment function and the effectivity structure at a state of M by using the valuation function if the state is s_0 or the assignment function and the effectivity structures of the witness models if otherwise. After constructing the model M, we also have to show that the effectivity structure of M is RB-playable so that M then is a qualified model for φ. In the following, we detail the construction of M.

For each tuple of formulas $([C_j^{b_j}]\psi_j)_{j\leq k}$ of $cl(\varphi)$ which corresponds to one of two cases in the lemma, there is a finite model which satisfies its corresponding formula in form of either $\wedge_{i\leq k}\psi_j$ or $\wedge_{i<k}\psi_j \wedge \neg\psi_k$. Let $M_1, \ldots, M_n$ be the enumeration of the above witnessing models where $M_i = (S_i, E_i, V_i)$ such that, without loss of generality, all S_i's are assumed to be pairwise disjoint.

We construct a finite model $M = (S, E, V)$ as follows. The set of states S is the set $\bigcup_{i\leq n} S_i \cup \{s_0\}$ where s_0 is a new state; hence S is finite. In order to define V, we firstly introduce a mapping $V_0 : cl(\varphi) \to \wp(\{s_0\})$ where

$$V_0(\psi) = \begin{cases} \{s_0\} & \text{if } v(\psi) = 1 \\ \varnothing & \text{otherwise} \end{cases}$$

Then, we define an assignment $U : cl(\varphi) \to \wp(S)$ by $U(\psi) = \bigcup_{i=0,\ldots,n} V_i(\psi)$. Note that by the construction, we have $U(\neg\psi) = S \setminus U(\psi)$, $U(\psi_1 \vee \psi_2) = U(\psi_1) \cup U(\psi_2)$. Now, we define the mapping V for M by the projection of U on the set of propositional variables p, that is $V(p) = U(p)$ (without loss of generality, we can assume that all propositional variables are contained in $cl(\varphi)$).

Finally, we define the effectivity structure E in a way which is similar to that of the completeness proof.

For $C \neq N$ and b such that $t(b) = 1$, we put $X \subseteq S$ in $E(s)(C^b)$ if and only if $X = S$ or there are $[C_1^{b_1}]\psi_1, \ldots, [C_k^{b_k}]\psi_k \in cl(\varphi)$ for some $k > 0$ such that:

- $t(b_j) = 1$ for all $j \leq k$
- $C_1, \ldots, C_k$ are pairwise disjoint, and all non-empty if C is not empty
- $\bigcup_{j \leq k} C_j \subseteq C$
- $\oplus_{j \leq k} b_j \leq b$ if $C \neq \varnothing$ or $b \leq b_j$ for all $j \leq k$ otherwise
- $\bigcap_{j \leq k} U(\psi_j) \subseteq X$ for all $j \leq k$
- $v([C_j^{b_j}]\psi_j) = 1$ for all $j \leq k$ if $s = s_0$
- $M_i, s \vDash [C_j^{b_j}]\psi_j$ for all $j \leq k$ if $s \in S_i$ for some $i \leq n$

For $t(b) = 1$, $X \in E(s)(N^b)$ if and only if $\overline{X} \notin E(s)(\varnothing^b)$. For the case when $t(b) > 1$ and $C \neq \varnothing$, we define $E(s)(C^b)$ inductively as follows: $X \in E(s)(C^b)$ iff one of the following conditions holds,

1. There is $b' < b$ such that $X \in E(s)(C^b)$
2. There are $b_1 \oplus b_2 = b$ such that $\{s' \mid X \in E(s')(C^{b_2})\} \in E(s)(C^{b_1})$

Then, we define for $t(b) > 1$, $X \in E(s)(\varnothing^b)$ iff $\overline{X} \notin E(s)(N^b)$.

Before proving that the model M which we just construct is indeed a model for φ, it is required to show that E is an RB-playable effectivity structure.

Claim 1. *The effectivity structure E is RB-playable.*

Proof.

- We show the first two properties of RB-playability by induction on bounds.

  Let $t(b) = 1$ and $C \neq N$. The definition of E implies directly that $S \in E(s)(C^b)$.

  Moreover, $S \in E(s)(\varnothing^b)$ implies that $\varnothing \notin E(s)(N^b)$ also by the definition of E.

  Let $t(b) = 1$ and $C \neq N$. Assume to the contrary that $\varnothing \in E(s_0)(C^b)$. Hence, there are $[C_1^{b_1}]\psi_1, \ldots, [C_k^{b_k}]\psi_k \in cl(\varphi)$ for some $k > 0$ such that:

- $t(b_j) = 1$ for all $j \leq k$
- $C_1, \ldots, C_k$ are pairwise disjoint, and all non-empty if C is not empty
- $\bigcup_{j \leq k} C_j \subseteq C$
- $\oplus_{j \leq k} b_j \leq b$ if $C \neq \varnothing$ or $b \leq b_j$ for all $j \leq k$ otherwise
- $\bigcap_{j \leq k} U(\psi_j) \subseteq X$ for all $j \leq k$
- $v([C_j^{b_j}]\psi_j) = 1$ for all $j \leq k$ if $s = s_0$
- $M_i, s \vDash [C_j^{b_j}]\psi_j$ for all $j \leq k$ if $s \in S_i$ for some $i \leq n$

Then, $\wedge_{j \leq k} \psi_j \equiv \bot$ which contradicts the first condition of the lemma where $\bot$ is required to be satisfiable.

Similarly to the case when $s \neq s_0$, we can show that $\varnothing \notin E(s)(C^b)$ for $C \neq N$.

Then, $\varnothing \notin E(s)(\varnothing^b)$ implies that $S \in E(s)(N^b)$, by the definition of E, again.

In the induction step, let $t(b) > 1$ and $C \neq \varnothing$, we directly have that $S \in E(s)(C^b)$ as $S \in E(s)(C^{b'})$ for any $b' < b$ and $t(b') = 1$. $S \in E(s)(N^b)$ also implies that $\varnothing \notin E(s)(\varnothing^b)$. Moreover, by the induction hypothesis, we have that $\varnothing \notin E(s)(C^{b'})$ for any $b' < b$. Furthermore, for any $b_1 \otimes b_2 = b$, we have that $\{s' \mid \varnothing \in E(s')(C^{b_2})\} = \varnothing$ and $\varnothing \notin E(s)(C^{b_1})$ also because of the induction hypothesis. By the definition of E, it follows that $\varnothing \notin E(s)(C^b)$. Once agent, $\varnothing \notin E(s)(N^b)$ implies that $S \in E(s)(\varnothing^b)$.

- Let us now show outcome monotonicity.

  Let $t(b) = 1$ and $C \neq N$. Assume that $X \in E(s)(C^b)$ where $X \subset S$. By the definition of E, there are $[C_1^{b_1}]\psi_1, \ldots, [C_k^{b_k}]\psi_k \in cl(\varphi)$ for some $k > 0$ such that:

  - $t(b_j) = 1$ for all $j \leq k$
  - $C_1, \ldots, C_k$ are pairwise disjoint, and all non-empty if C is not empty
  - $\bigcup_{j \leq k} C_j \subseteq C$
  - $\oplus_{j \leq k} b_j \leq b$ if $C \neq \varnothing$ or $b \leq b_j$ for all $j \leq k$ otherwise
  - $\bigcap_{j \leq k} U(\psi_j) \subseteq X$ for all $j \leq k$
  - $v([C_j^{b_j}]\psi_j) = 1$ for all $j \leq k$ if $s = s_0$
  - $M_i, s \vDash [C_j^{b_j}]\psi_j$ for all $j \leq k$ if $s \in S_i$ for some $i \leq n$

It is straightforward that for any $X' \supseteq X$, we have $\bigcap_{j \leq k} U(\psi_j) \subseteq X \subseteq X'$. Hence, $X' \in E(s)(C^b)$.

In the case of the grand coalition, assume that $X \in E(s)(N^b)$. By the definition of E, we have $\bar{X} \notin E(s)(\varnothing^b)$. Assume to the contrary that $X' \notin E(s)(N^b)$ for some $X' \supseteq X$. It follows that $\bar{X'} \subseteq \bar{X}$. $X' \notin E(s)(N^b)$ implies that $\bar{X'} \in E(s)(\varnothing^b)$, hence, $\bar{X} \in E(s)(\varnothing^b)$ which is a contradiction.

Now, we provide a proof of outcome monotonicity for the case when $t(b) > 1$. It is easy to notice that it is similar to the proof of completeness of RBCL.

Let $t(b) > 1$ and $C \neq \varnothing$. . If $X \in E(s)(C^{b'})$ for some $b' < b$, induction hypothesis shows that $X' \in E(s)(C^{b'})$ for all $X' \supseteq X$. Then, by the definition of E, we have $X' \in E(C^b)(s)$. If $X \notin E^(s)(C^{b'})$ for all $b' < b$. By the definition of E, there are $b_1 \otimes b_2 = b$ and

$$\{s' \mid X \in E(s')(C^{b_2})\} \in E(s)(C^{b_1})$$

Let $X' \supseteq X$, by the induction hypothesis we have

$$\begin{aligned}&\{s' \mid X \in E(s')(C^{b_2})\} \subseteq \{s' \mid X' \in E(s')(C^{b_2})\}\\ \Rightarrow &\{s' \mid X' \in E(s')(C^{b_2})\} \in E(s)(C^{b_1})\end{aligned}$$

By the definition of E, we have $X' \in E(s)(C^b)$.
If $t(b) > 1$, $X \in E(s)(\varnothing^b)$ iff $\overline{X} \notin E(s)(N^b)$. Let $X' \supseteq X$, assume to the contrary that $X' \notin E(s)(\varnothing^b)$. This implies that $\overline{X'} \in E(s)(N^b)$. By the previous proof, we have $\overline{X} \in E(s)(N^b)$ as $\overline{X'} \subseteq \overline{X}$, which is a contradiction.

- N-maximality and regularity follow directly from the definition of E for $\varnothing$ when $t(b) = 1$ and also $t(b) > 1$. Therefore, we omit the proof here.

- Since S is finite, we ignore proving N-determinacy as it is derivable by other properties of RB-playability.

- Let us now prove super-additivity. Let $t(b) = t(d) = 1$, $C \cap D = \varnothing$ with $X \in E(s)(C^b)$ and $Y \in E(s)(D^d)$.

  - If both C and D are not empty. Assume that both X and Y are not equal to S. By definition of E, we have there are $[C_1^{b_1}]\psi_1, \ldots, [C_{k_C}^{b_{k_C}}]\psi_{k_C} \in cl(\varphi)$ and $[D_1^{d_1}]\psi'_1, \ldots, [D_{k_D}^{d_{k_D}}]\psi'_{k_D} \in cl(\varphi)$ for some $k_C > 0$ and $k_D > 0$ such that:

* $t(b_j) = 1$ for all $j \leq k_C$
* $t(d_j) = 1$ for all $j \leq k_D$
* $C_1, \ldots, C_{k_C}$ are pairwise disjoint, and all non-empty
* $D_1, \ldots, D_{k_D}$ are pairwise disjoint, and all non-empty
* $\bigcup_{j \leq k_C} C_j \subseteq C$
* $\bigcup_{j \leq k_D} D_j \subseteq D$
* $\oplus_{j \leq k_C} b_j \leq b$
* $\oplus_{j \leq k_D} d_j \leq d$
* $\bigcap_{j \leq k_C} U(\psi_j) \subseteq X$ for all $j \leq k_C$
* $\bigcap_{j \leq k_D} U(\psi'_j) \subseteq Y$ for all $j \leq k_D$
* $v([C_j^{b_j}]\psi_j) = 1$ for all $j \leq k_C$ if $s = s_0$
* $v([D_j^{d_j}]\psi'_j) = 1$ for all $j \leq k_D$ if $s = s_0$
* $M_i, s \vDash [C_j^{b_j}]\psi_j$ for all $j \leq k_C$ if $s \in S_i$ for some $i \leq n$
* $M_i, s \vDash [D_j^{d_j}]\psi'_j$ for all $j \leq k_D$ if $s \in S_i$ for some $i \leq n$

Then, it is straightforward that $X \cap Y \supseteq \bigcap_{j \leq k_C} U(\psi_j) \cap \bigcap_{j \leq k_D} U(\psi'_j)$. It follows that $X \cap Y \in E((C \cup D)^{b \oplus d})$.

In the case when Y is S, the proof is similar to above with the notice that $C \subseteq C \cup D$ and $b \leq b \oplus d$.

- If $C = D = \varnothing$ and $b = d$, we apply the same argument as the case above.
- If $C = \varnothing$, $b = d$ and $D = N$, we need to show that $X \cap Y \in E(N^b)(s)$. Assume to the contrary that $X \cap Y \notin E(N^b)(s)$, then N-maximality, which has been proved above, implies that $\overline{X \cap Y} \in E(\varnothing^b)(s)$. Then, by the previous case of super-additivity, we have $X \cap \overline{Y} \in E(\varnothing^b)(s)$. As we already showed outcome-monotonicity, $\overline{Y} \in E(\varnothing^b)(s)$. However, by N-regularity, we have $Y \notin E(N^b)(s)$ which is a contradiction.

- Super-transitivity follows directly from the definition of E when $t(b) > 1$.
- Similarly, transitivity follows directly from the definition of E when $t(b) > 1$.

□

Therefore, E is RB-playable. In order to show that M satisfies φ, we prove the following two claims.

Claim 2. *For any* $[C^b]\psi \in cl(\varphi)$, $U(\psi) \in E(s)(C^b)$ *iff* $v([C^b]\psi) = 1$ *if* $s = s_0$ *or* $M_i, s \vDash [C^b]\psi$ *if* $s \in S_i$ *for some* $i \leq n$.

Proof. The direction from right to left is straightforward according to the definition of E. Hence, we provide here only a proof for the other direction.

A trivial case is when $U(\psi) = S$, therefore we ignore it here. We prove the claim also by induction on the resource bounds.

Let $t(b) = 1$ and $C \neq N$. As $U(\psi) \in E(s)(C^b)$, there are $[C_1^{b_1}]\psi_1, \ldots, [C_k^{b_k}]\psi_k \in cl(\varphi)$ for some $k > 0$ such that:

- $t(b_j) = 1$ for all $j \leq k$
- $C_1, \ldots, C_k$ are pairwise disjoint, and all non-empty if C is not empty
- $\bigcup_{j \leq k} C_j \subseteq C$
- $\oplus_{j \leq k} b_j \leq b$ if $C \neq \varnothing$ or $b \leq b_j$ for all $j \leq k$ otherwise
- $\bigcap_{j \leq k} U(\psi_j) \subseteq U(\psi)$ for all $j \leq k$
- $v([C_j^{b_j}]\psi_j) = 1$ for all $j \leq k$ if $s = s_0$
- $M_i, s \vDash [C_j^{b_j}]\psi_j$ for all $j \leq k$ if $s \in S_i$ for some $i \leq n$

Suppose $s \in S_i$ for some $i \leq n$. As $M_i, s \vDash [C_j^{b_j}]\psi_j$ for all $j \leq k$, super-additivity implies that $M_i, s \vDash [\bigcup_{j \leq k} C_j^{\oplus_{j \leq k} b_j}](\wedge_{j \leq k}\psi_j)$ if $C \neq \varnothing$ or directly, $M_i, s \vDash [C^b](\wedge_{j \leq k}\psi_j)$ otherwise. In the former case, coalition monotonicity gives us $M_i, s \vDash [C^b](\wedge_{j \leq k}\psi_j)$. Then, in both cases, we can conclude by outcome-monotonicity that $M_i, s \vDash [C^b](\psi)$.

When $s = s_0$, assume by contradiction that $v([C^b]\psi) = 0$. Then there is a witnessing model M_i and $s' \in S_i$ such that $M_i, s' \vDash \wedge_{j \leq k}\psi_i \wedge \neg\psi$ which contradicts the fact that $\bigcap_{j \leq k} U(\psi_j) \subseteq U(\psi)$.

Let $t(b) = 1$ and $C = N$. By the definition of E, $U(\psi) \in E(s)(N^b)$ iff $U(\neg\psi) \notin E(s)(\varnothing^b)$. By the proof above, $U(\neg\psi) \notin E(s)(\varnothing^b)$ iff $v([\varnothing^b]\neg\psi) = 0$ if $s = s_0$ or $M_i, s \nvDash [\varnothing^b]\neg\psi$ if $s \in S_i$ for some $i \leq n$. By the definition of v, we have that $v([\varnothing^b]\neg\psi) = 0$ implies $v([N^b]\psi) = 1$. Moreover, by N-maximality, we also have $M_i, s \nvDash [\varnothing^b]\neg\psi$ implies that $M_i, s \vDash [N^b]\psi$.

Let $t(b) > 1$ and $C \neq \varnothing$. We have that $U(\psi) \in E(s)(C^b)$ iff $U(\psi) \in E(s)(C^{b'})$ for some $b' < b$ or there are $b_1 \otimes b_2 = b$ such that $\{s' \mid U(\psi) \in E(s)(C^{b_2})\} \in E(s)(C^{b_1})$. If $U(\psi) \in E(s)(C^{b'})$, by the induction hypothesis, $v([C^{b'}]\psi) = 1$ if $s = s_0$ or $M_i, s \vDash [C^{b'}]\psi$ if $s \in S_i$ for some $i \leq n$. By the definition of v, $v([C^{b'}]\psi) = 1$ implies that $v([C^b]\psi) = 1$. By RB-playability, $M_i, s \vDash [C^{b'}]\psi$ implies $M_i, s \vDash [C^b]\psi$.

If there are $b_1 \otimes b_2 = b$ such that $U([C^{b_2}]\psi) = \{s' \mid U(\psi) \in E(s)(C^{b_2})\} \in E(s)(C^{b_1})$, by the induction hypothesis, $v([C^{b_1}][C^{b_2}]\psi) = 1$ if $s = s_0$ or $M_i, s \vDash [C^{b_1}][C^{b_2}]\psi$ if $s \in S_i$ for some $i \leq n$. By the definition of v, $v([C^{b_1}][C^{b_2}]\psi) = 1$ implies that $v([C^b]\psi) = 1$. By RB-playability, $M_i, s \vDash [C^{b_1}][C^{b_2}]\psi$ implies $M_i, s \vDash [C^b]\psi$.

Let $t(b) > 1$ and $C = \varnothing$. By the definition of E, $U(\psi) \in E(s)(\varnothing^b)$ iff $U(\neg\psi) \notin E(s)(N^b)$. By the proof above, $U(\neg\psi) \notin E(s)(N^b)$ iff $v([N^b]\neg\psi) = 0$ if $s = s_0$ or $M_i, s \nvDash [N^b]\neg\psi$ if $s \in S_i$ for some $i \leq n$. By the definition of v, we have that $v([N^b]\neg\psi) = 0$ implies $v([\varnothing^b]\psi) = 1$. Moreover, by N-maximality, we also have $M_i, s \nvDash [N^b]\neg\psi$ implies that $M_i, s \vDash [\varnothing^b]\psi$. □

Claim 3. *V and U agree on $cl(\varphi)$.*

Proof. In the base case, the proof is trivial as according to the definition of V, they already agree on the set of propositions in $cl(\varphi)$. The proof for propositional connectives is also straightforward as we know that $U(\neg\psi) = S \setminus U(\psi)$ and $U(\psi_1 \vee \psi_2) = U(\psi_1) \cup U(\psi_2)$, and similarly for V. For the case of $[C^b]\psi$, the proof is done by induction on the resource bounds.

Assume that $s \in U([C^b]\psi)$, then by the definition of U, $v([C^b]\psi) = 1$ if $s = s_0$ or $M_i, s \vDash [C^b]\psi$ if $s \in S_i$ for some $i \leq n$.

If $t(b) = 1$ and $C \neq N$, then in both above cases, by the definition of E, we have that $U(\psi) \in E(s)(C^b)$. By the induction hypothesis, $U(\psi) = V(\psi)$, hence $V(\psi) \in E(s)(C^b)$, therefore $M, s \vDash [C^b]\psi$.

If $t(b) = 1$ and $C = N$, then we have $v([\varnothing^b]\neg\psi) = 0$ if $s = s_0$ or $M_i, s \nvDash [\varnothing^b]\neg\psi$ if $s \in S_i$ for some $i \leq n$. In both cases, by the definition of E, we have that $U(\neg\psi) \notin E(s)(\varnothing^b)$, otherwise, $U(\neg\psi) \in E(s)(\varnothing^b)$ will contradict Claim 2. Hence, $U(\psi) = V(\psi) \in E(s)(N^b)$ because E is RB-playable. Then, $M, s \vDash [N^b]\psi$. Therefore $s \in V([N^b]\psi)$.

Assume $t(b) > 1$ and $C \neq \varnothing$. If $s = s_0$ and $v([C^b]\psi) = 1$, then either $v([C^{b'}]\psi) = 1$ for some $b' < b$ or there are $b_1 \otimes b_2 = b$ such that $v([C^{b_1}][C^{b_2}]\psi) = 1$. In both cases, by the induction hypothesis together with the definition of E, we imply that $s \in V([C^b]\psi)$. Similarly, if $s \in S_i$ and $M_i, s \vDash [C^b]\psi$, either $M_i, s \vDash [C^{b'}]\psi$ or $M_i, s \vDash [C^{b_1}][C^{b_2}]\psi$. Again, in both cases, by the induction hypothesis together with the definition of E, we imply that $s \in V([C^b]\psi)$.

If $t(b) > 1$ and $C = \varnothing$, then we have $v([N^b]\neg\psi) = 0$ if $s = s_0$ or $M_i, s \nvDash [N^b]\neg\psi$ if $s \in S_i$ for some $i \leq n$. In both cases, by the definition of E, we have that $U(\neg\psi) \notin E(s)(N^b)$, otherwise, $U(\neg\psi) \in E(s)(N^b)$ will contradict Claim 2. Hence $U(\psi) = V(\psi) \in E(s)(\varnothing^b)$ because E is RB-playable. Then, $M, s \vDash [\varnothing^b]\psi$. Therefore $s \in V([\varnothing^b]\psi)$.

Assume that $s \in V([C^b]\psi)$, that is $M, s \vDash [C^b]\psi$. Therefore, $V(\psi) = U(\psi) \in E(s)(C^b)$. By the above claim, we have $v([C^b]\psi) = 1$ if $s = s_0$ or $M_i, s \vDash [C^b]\psi$ if $s \in S_i$ for some $i \leq n$. In both cases, the definition of U gives us $s \in U([C^b]\psi)$. □

Finally, we complete the proof for Lemma 10. Since $v(\varphi) = 1$, $s_0 \in U(\varphi)$. Therefore, by Claim 3, we have $s_0 \in V(\varphi)$, hence, $M, s_0 \vDash \varphi$. In other words, φ is satisfiable. □

We now return to the example. We have already computed the set $cl(\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p)$ as well as a valuation v as depicted in Figure 4.2. Let us consider any subset of formulas from $cl(\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p)$ which satisfy either the first or the second case of Lemma 10. For example, the set $\Gamma = \{[\{2\}^{(0,1)}][\{2\}^{(1,1)}]p, [\varnothing^{(2,1)}]\neg p, [\{1,2\}^{(2,1)}]p,\}$ which fits into the second case of Lemma 10, then is the formula $[\{2\}^{(1,1)}]p \wedge \neg p$ satisfiable? In order to answer this question, one possible way is to routinely compute the set $cl([\{2\}^{(1,1)}]p \wedge \neg p)$ and guess a valuation for it. However, the formula is simple enough for us to guess a model, which is illustrated in Figure 4.3. In Figure 4.3, only actions for agent 2 are drawn. For agent 1, there is only one

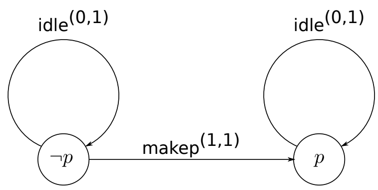

FIGURE 4.3: A model satisfies $[\{2\}^{(1,1)}]p \wedge \neg p$.

action at both states which is $idle^{(0,1)}$ which costs only one unit of time. Each transition drawn in Figure 4.3 defines the outcome of each joint action for both agents. A joint action in our example is a pair of $idle^{(0,1)}$ (performed by agent 1) and an action attached to the transition (performed by agent 2).

Beside the set Γ which we already considered above, there are also others sub-sets of formulas from $cl(\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p)$ which can be used in the first or the second case of Lemma 10. Let us list them as follows:

- $\Gamma = \{[\{2\}^{(0,1)}][\{2\}^{(1,1)}]p, [\{2\}^{(1,1)}]p,\}$
- $\Gamma = \{[\{2\}^{(0,1)}][\{2\}^{(1,1)}]p, [\{2\}^{(0,1)}]p,\}$

- $\Gamma = \{[\{2\}^{(0,1)}][\{2\}^{(1,1)}]p, [\{2\}^{(1,1)}][\{2\}^{(0,1)}]p, \}$
- $\Gamma = \{[\{2\}^{(0,1)}][\{2\}^{(1,1)}]p, [\{2\}^{(0,1)}][\{2\}^{(0,1)}]p, \}$

We can also routinely consider each case by computing the closure of their conjunction formulas and guessing the corresponding valuation. However, it is not difficult to see that they are also satisfied by the model depicted in Figure 4.3. In other words, Lemma 10 implies that $\neg[\{1,2\}^{(2,1)}]p \wedge [\{2\}^{1,2}]p$ is satisfiable.

As a consequence of the lemma, we finish this section by providing an algorithm for deciding the satisfiability problem of RBCL. Let us introduce some notation. As defined above, given a closure $cl(\varphi)$, let $CON(\varphi)$ be the set of all finite nonempty subsets $\{[C_1^{b_1}]\psi_1, \dots, [C_k^{b_k}]\psi_k\} \subseteq cl(\varphi)$ which appear in either the first or the second condition of Lemma 10. Moreover, each set $\Gamma = \{[C_1^{b_1}]\psi_1, \dots, [C_k^{b_k}]\psi_k\} \in CON(\varphi)$ is associated with a formula, denoted φ_Γ, which is in the form of either $\wedge_{i \leq k}\psi_i$ or $\wedge_{i<k}\psi_i \wedge \neg\psi_k$, depending on whether Γ is for the first or the second condition of Lemma 10, respectively. Then, the algorithm for the satisfiability problem, given a formula φ, is as follows.

1. Non-deterministically select a valuation v for $cl(\varphi)$.
2. For every set $\Gamma \in CON(\varphi)$, recursively check that φ_Γ is satisfiable.

Note that if we measure the size of the input to the algorithm (the formula φ) assuming that the resource bounds are written in *unary*, then the algorithm is PSPACE (since in this case the size of $cl(\varphi)$ is polynomial in $|\varphi|$). However, if the resource bounds are written in binary, then $|cl(\varphi)|$ is exponential in $|\varphi|$ and hence the algorithm requires exponential space to record the valuation.

4.7 Conclusion

In this chapter, we have introduced the logic RBCL (and its simplified version, RBCL_1) which allows us to reason about the ability of coalitions of agents under resource bounds in systems of multiple agents where every action is associated with a certain cost of resources. RBCL_1 is for reasoning about single step strategies of resource-bounded multi-agent system, which helps us to determine whether a coalition of agents could cooperate in one step of time in order to obtain a certain goal under a resource bound. We generalise the logic to RBCL for reasoning about multi-step strategies of coalitions of agnets under resource bounds. Rather than look at single step cooperation, we consider the case when agents in a coalition cooperate in a sequence of consecutive steps in order to force a desired result without spending more than a certain amount of resources.

Similar to the case of Coalition Logic, we study the soundness and completeness of RBCL by considering the notions of effectivity structures of resource-bounded concurrent game frames and resource-bounded playability. Apart from that, the chapter also discusses the satisfiability problem of RBCL where we have shown that RBCL is decidable.

Finally, let us remark on the relationship between RBCL and CL. Comparing to CL, RBCL is an extension where modalities of CL are attached with resource bounds in order to express abilities of a coalition of agents under such resource bounds. Although RBCL allows reasoning about multi-step strategies of coalitions, the approaches used in this chapter for results of RBCL are not proved to be applicable for the case of ECL [Pauly, 2002]. Moreover, the fact that every resource bound in a formula of RBCL contains only concrete numbers shows that RBCL is intuitively not expressive enough to define the modalities $[C^*]$ and $[C^\times]$ of ECL where the formula $[C^*]\varphi$ in ECL sets no time limitation on the strategy of the coalition C to force φ and the formula $[C^\times]\varphi$ in ECL says that the coalition C can maintain φ indefinitely.

CHAPTER 5

RESOURCE-BOUNDED ALTERNATING-TIME TEMPORAL LOGIC

5.1 Introduction

When we use RBCL to reason about coalitional ability under resource bounds in a multi-agent system, the formula of the form $[C^b]\varphi$ expresses the case that the coalition C has the ability to bring about φ under the resource bound b. In other words, RBCL is a logic which allows us to reason about coalitional ability to obtain certain goals under an explicit resource bound. Naturally, the logic is not expressive enough to specify other coalitional abilities of resource-bounded multi-agent systems. For instance, it is not possible the express the following properties in RBCL:

- A coalition C has the ability to maintain a condition under a resource bound b.
- A coalition C has the ability to maintain a condition until it obtains another goal under a resource bound b.

Without speaking about resource bounds, we also face a similar problem with Coalition Logic where the more expressive logic ATL is taken into account in order to express such properties with the help of temporal operators such as global ($\Box$) and until $\mathcal{U}$. Therefore, we present in this chapter a logic, namely resource-bounded alternating-time temporal logic (RB-ATL) which is an extension of ATL with resource bounds. The purpose of having RB-ATL is to have a more expressive logic for specifying and reasoning about properties of resource-bounded multi-agent systems. Furthermore, for a more flexible logic, we extend the notion of resource bounds in order to express properties where bounds do not need to cover for all resources used in a multi-agent system. Therefore, we could flexibly express properties such as a coalition C has the ability to maintain a condition φ

without spending more than an explicit amount of a resource r in the set of resources used by the multi-agent system. It is worth noticing that in this property, the usage of the resource r by the coalition C is explicitly limited while there are no bounds for all other resources of the multi-agent system. Therefore, the bound on the coalition C only covers the resource r and the coalition can spend an unlimited amount of other resources in order to maintain the condition φ.

The chapter is organised as follows. First, we introduce the extensions of notions of resource bounds and concurrent game structures. After that, the syntax and the semantics of the logic are presented. Then, we study the soundness and completeness of the logic. Finally, we investigate the satisfiability problem of the logic.

5.2 Extended resource bounds

In the previous chapter, we have defined the set of resource bounds $\mathbb{B}$ over a given finite set R of resources as $\mathbb{B} = \mathbb{N}^r$ where r is the cardinality of R. This definition means that a resource bound $b = (b_1, \ldots, b_r)$ determines the bounds on every resource in R.

Therefore, we could only express properties in RBCL such as $[C^b]\varphi$ where the bound b covers all resources used by a multi-agent system. For instance, let us return to the example on page 28. In this example, the system of two reasoning agents uses explicitly two resources: memory and communication and an implicit resource time. As we already knew, it is possible to express in RBCL the property that *Agents* 1 *and* 2 *can enforce* C *to become true without using more than 4 units of memory and 2 units of network bandwidth in maximally 7 steps of time* by the formula $[\{1,2\}^{(4,2,7)}]C$. However, if we would like to express the same property but we do not want to set a bound on how many messages can be exchanged between two agents (i.e. there is no limit on communication between them), it is not possible to do so in RBCL unless the language allows infinite disjunction. This is because the property can only be written as $\bigvee_{n\geq 0}[\{1,2\}^{(4,n,7)}]C$. In order to allow expressing such properties, we extend the notion of resource bounds by allowing the inclusion of an extra symbol ∞. The idea is that whenever no limit is required over a resource, we set the bound for this resource as ∞. Then, the previous property can be expressed by the formula $[\{1,2\}^{(4,\infty,7)}]c$ where ∞ means that there is no bound on communication while we still set limitations on the resources memory and time.

Let us formally define the set of extended resource bounds in the following.

Definition 15. *Given a finite set* R *of resources where* $R = \{1, \ldots, r\}$*, the set of extended resource bounds is defined as* $\mathbb{B}^\infty = (\mathbb{N} \cup \{\infty\})^r$.

In the previous chapter, we have mentioned that there are two places where resource bounds can be used. First, resource bounds b are used as the bounds in the formulas of the form $[C^b]\varphi$. Then, resource bounds are used to specify the cost of actions. However, it is not the case for extended resource bounds where the cost of actions must be concrete numbers rather than a symbol ∞. Therefore, we only use extended resource bounds as the limitation of resources in formalising properties of resource-bounded multi-agent systems. In other words, the set of resource bounds are still used to specify the cost of actions in models that are used to define the semantics of the logic RB-ATL.

The comparison operator between extended resource bounds are defined as usual. Given two extended resource bounds b and d, we say that $b \leq d$ iff $b_i \leq d_i$ for all $1 \leq i \leq r$ where the comparison involving ∞ is defined as follows.

$$n \leq \infty \text{ for all } n \in \mathbb{N}$$

$$\infty \leq \infty$$

This means ∞ is greater than any natural number in $\mathbb{N}$. For the sake of simplicity, we also assume that both resource bound sequence and parallel operators $\otimes$ and $\oplus$, which are presented in the previous chapter, are defined by means of the addition operator. The addition operator where ∞ is involved is defined as follows:

$$n + \infty = \infty + n = \infty \text{ for all } n \in \mathbb{N}$$

$$\infty + \infty = \infty$$

Then, when we add ∞ with any natural number, we also obtain ∞.

In this chapter, we denote the smallest (extended) resource bound by the symbol $\bar{0}$ where $\bar{0} = (0, \dots, 0)$, i.e. all components of $\bar{0}$ are 0.

5.3 Resource-bounded concurrent game structure

In this section, we present the extension of concurrent game structures [Alur et al., 2002] which will be used to define the semantics of the logic RB-ATL. Similar to the case of resource-bounded action frame defined in Section 4.3.2, we extend the concept of concurrent game structures by accompanying each of them a finite set of resources. Moreover, actions in concurrent game structures are also associated with costs each of which is a resource bound from $\mathbb{B}$.

In particular, we define the notion of resource-bounded concurrent game structures (RB-CGS) as follows.

Definition 16. *A Resource-bounded Concurrent Game Structure (RB-CGS) is a tuple*

$$S = (n, R, Q, Prop, \pi, d, c, \delta)$$

where:

- $n \geq 1$ *is the number of players (agents) in a resource-bounded multi-agent system, we denote the set of players* $\{1, \ldots, n\}$ *by* N
- R *is a finite set of resources where* $|R| = r$.
- Q *is a non-empty set of states*
- *Prop is a set of propositional variables*
- $\pi : Q \to \wp(Prop)$ *is a mapping which assigns each state in* Q *a subset of propositional variables*
- $d : Q \times N \to \mathbb{N}$ *is a mapping to indicate the number of available moves (actions) for each player* $a \in N$ *at a state* $q \in Q$ *such that* $d(q, a) \geq 1$. *At each state* $q \in Q$, *we denote the set of joint moves available for all players in* N *by* $D(q)$ *as follows:*

$$D(q) = \{1, \ldots, d(q, 1)\} \times \ldots \times \{1, \ldots, d(q, n)\}$$

- $c : Q \times N \times \mathbb{N} \to \mathbb{B}$ *is a mapping to indicate the cost of resources required by each move available to each agent at a state. Furthermore, we require that among actions* $\{1, \ldots, d(q, a)\}$ *available to an agent* $a \in N$ *at a state* $q \in Q$, *there is at least one action* $j \in \{1, \ldots, d(q, a)\}$ *where the cost of* j *is the minimal value* $\bar{0}$.
- $\delta : Q \times \mathbb{N}^n \to Q$ *is a mapping where* $\delta(q, m)$ *is the next state from* q *if all players opt the corresponding move in* $m \in D(q)$.

In the above definition, it is worth noticing that the extra requirement for the cost function c is to make the structure S to be total. That is there are always possible actions for agents so that the system can move forward without spending any amount of resources.

In the following, let us consider an example of an RB-CGS as depicted in Figure 5.1. We describe in this example possible ways for an agent to go from Nott (which stands for Nottingham) to Ams (Amsterdam) by train and by air. On the ways, the agent may have to get to intermediate destinations which are Lond (London) and Paris. There are two agents in this example, one is the agent that wants to go from Nott to Ams, and the other is an agent which tries to disrupt the trip of the first agent. If a trip is disrupted by the second agent, it causes the first agent to arrive the destination of the trip late as long as the first agent travels by air. We could think of the second agent as external factors which can delay trips by air such as bad weather or the strong activity of volcano. We assume that it is not likely to be disrupted when travelling by train, and hence, train is always on time. In this example, we assume that there are two resources which are the numbers

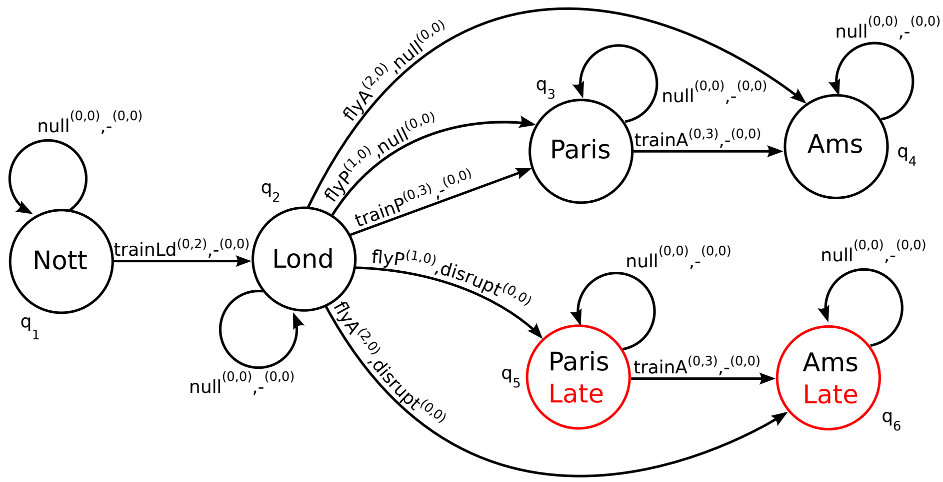

FIGURE 5.1: Possible ways to go from Nott to Ams.

of hours the first agent spends on a trip from one place to another by air and by train, respectively. From every state, each agent can perform a "null" action which costs no hour by air and by train, i.e. the cost of null is $\bar{0}$. Also from any state, the second agent can perform the disrupt action of which cost is also $\bar{0}$. From q_1, the first agent can perform the action trainLd which stands for going to London by train, hence, costs no hour by air but two hours by train. Similarly, from the state q_2, the first agent can go to Paris either by air or by train and only to Ams by air. The action flyA which means to go to Ams by air costs two hours on the flight but no hour by train. The action flyP is for going to Paris by air which costs one hour on the flight and no hour by train. The action trainP is for going to Paris by train; therefore, it costs no hour on the flight but three hours by train. Then, from Paris, the first agent can only go to Ams by train by performing the action trainA which costs no hour by air and three hours by train. In Figure 5.1, each transition from one state to another is the result of a joint action of both agents which is drawn next to the transition. An action in the figure is written in the form of $act^{(x,y)}$ to indicate that the action act costs x hours by air and y hours by train. We write $-^{(0,0)}$ to denote either the actions $null^{(0,0)}$ or $disrupt^{(0,0)}$. Notice that when the first agent performs an action flyX to go to the destination X by air and the second agent performs the action disrupt, we arrive at a state where the proposition Late is true to indicate that the action disrupt has caused delay to the trip of the first agent.

Then, we list in the following several properties of the system depicted in Figure 5.1:

- At q_1, the first agent cannot get to Ams on time (i.e. not late) without spending more than three hours by train and five hours by air.
- At q_1, the first agent can get to Ams on time without spending more than ten hours only by train.
- At q_1, the first agent cannot get to Ams on time without spending more than ten hours only by air.
- At q_1, the first agent can get to Ams on time, i.e. allowing spending any number of hours by air or by train.
- At q_1, the first agent can get to Ams on time only by train, i.e. there is no limitation on the number of hours spent on the train.
- At q_1, the first agent cannot get to Ams on time only by air, i.e. there is no limitation on the number of hours spent on the flight.
- At q_1, the first agent cannot get to Ams on time by spending no more than three hours by train and any number of hours by air.

In order to formalise and reason about the above properties, we present the logical language RB-ATL in the next section.

5.4 The language RB-ATL

In this section, we define the syntax and the semantics of RB-ATL. Moreover, we also present the normal form of RB-ATL which will be useful for us to investigate RB-ATL including the soundness and completeness problem of RB-ATL, and then the decidability of RB-ATL.

5.4.1 Syntax of RB-ATL

Assume we have a finite set N of agents where $N = \{1, \dots, n\}$, a finite set R of resources where $|R| = r$, the set $\mathbb{B}^\infty$ of extended resource bounds, and a set of propositional variables *Prop*. We shall write A to denote a non-empty coalition of agents, that is $A \subseteq N$ and $A \neq \varnothing$. Then, we define the syntax of RB-ATL as follows.

$$\varphi ::= p \mid \neg\varphi \mid \varphi \lor \psi \mid \langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi \mid \langle\!\langle A^b \rangle\!\rangle \Box \varphi \mid \langle\!\langle A^b \rangle\!\rangle \varphi \mathcal{U} \psi$$

where $p \in$ *Prop*, A is a coalition and $b \in \mathbb{B}^\infty$.

Let us discuss the informal meaning of temporal formulas of RB-ATL which are quite similar to those of ATL. The formula $\langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi$ means that agents in the coalition A have a strategy to force φ to become true in the next step without spending more than b amount of resources. Then, the formula $\langle\!\langle A^b \rangle\!\rangle \Box \varphi$ means that the coalition A has a strategy to make φ true forever without spending more than b amount of resources. Finally, the formula $\langle\!\langle A^b \rangle\!\rangle \varphi \mathcal{U} \psi$ is to say that the coalition A has a strategy to keep φ true until ψ is eventually true without spending more than b amount of resources. For the case of the empty coalition, we shall write $\langle\!\langle \varnothing^b \rangle\!\rangle \bigcirc \varphi$ to say that for any strategy of the grand coalition N which spends at most b amount of resources, φ is true in the next step. In other words, φ is unavoidable for the system if it is not allowed to spend more than b amount of resources. Similarly, the formula $\langle\!\langle \varnothing^b \rangle\!\rangle \Box \varphi$ describes that if the system can only spend no more than b amount of resources, φ is true forever. Then, $\langle\!\langle \varnothing^b \rangle\!\rangle \varphi \mathcal{U} \psi$ means that if the system can only spend no more than b amount of resources, φ keeps being true until ψ is true. We define these temporal operators for the case of the empty coalition in terms of equivalences as follows:

$$\begin{aligned}
\langle\!\langle \varnothing^b \rangle\!\rangle \bigcirc \varphi &\equiv \neg \langle\!\langle N^b \rangle\!\rangle \bigcirc (\neg\varphi) \\
\langle\!\langle \varnothing^b \rangle\!\rangle \Box \varphi &\equiv \varphi \land \neg \langle\!\langle N^b \rangle\!\rangle \top \mathcal{U} \neg\varphi \\
\langle\!\langle \varnothing^b \rangle\!\rangle \varphi \mathcal{U} \psi &\equiv \neg (\langle\!\langle N^b \rangle\!\rangle \neg\psi \mathcal{U} \neg(\varphi \lor \psi) \lor \langle\!\langle N^b \rangle\!\rangle \Box \neg\psi)
\end{aligned}$$

Before defining the semantics of RB-ATL, let us give some examples of formulas of RB-ATL. We turn back to the example in the previous section where the listed properties of the system depicted in Figure 5.1 can be written by the following formulas: $\neg\langle\!\langle \{1\}^{(5,3)} \rangle\!\rangle \top \mathcal{U}(Ams \land \neg Late)$, $\langle\!\langle \{1\}^{(0,10)} \rangle\!\rangle \top \mathcal{U}(Ams \land \neg Late)$, $\neg\langle\!\langle \{1\}^{(10,0)} \rangle\!\rangle \top \mathcal{U}(Ams \land \neg Late)$, $\langle\!\langle \{1\}^{(\infty,\infty)} \rangle\!\rangle \top \mathcal{U}(Ams \land \neg Late)$, $\langle\!\langle \{1\}^{(0,\infty)} \rangle\!\rangle \top \mathcal{U}(Ams \land \neg Late)$, $\neg\langle\!\langle \{1\}^{(\infty,0)} \rangle\!\rangle \top \mathcal{U}(Ams \land \neg Late)$, and $\neg\langle\!\langle \{1\}^{(\infty,3)} \rangle\!\rangle \top \mathcal{U}(Ams \land \neg Late)$, respectively.

5.4.2 Semantics of RB-ATL

As we have already mentioned, the semantics of RB-ATL is defined by means of resource-bounded concurrent game structures. Given a RB-CGS $S = (n, R, Q, Prop, \pi, d, c, \delta)$, let us define the notion of moves, strategies and b-strategies where $b \in \mathbb{B}^{\infty}$.

Firstly, given a RB-CGS S, we denote the set of infinite sequences of states by Q^{ω} as usual. Let $\lambda = q_0 q_1 \ldots \in Q^{\omega}$, we denote $\lambda[i] = q_i$ and $\lambda[i, j] = q_i \ldots q_j$ for any $i, j \in \mathbb{N}$ such that $0 \le i \le j$.

Then, we define the notion of moves as follows.

Definition 17. *Given a RB-CGS S and a state $q \in Q$, a move for a coalition $A \subseteq N$ is a tuple $\sigma_A = (\sigma_a)_{a \in A}$ such that $1 \le \sigma_a \le d(q, a)$.*

For convenience, we denote $D_A(q)$ to be the set of all moves for A at q. Furthermore, given $m \in D(q)$, we denote $m_A = (m_a)_{a \in A}$. Then, we define the set of all possible outcomes by a move $\sigma_A \in D_A(q)$ at a state q as follows

$$out(q, \sigma_A) = \{q' \in Q \mid \exists m \in D(q) : m_A = \sigma_A \wedge q' = \delta(q, m)\}$$

The cost of a move $\sigma_A \in D_A(q)$ then is defined as $cost(q, \sigma_A) = \Sigma_{a \in A} c(q, a, \sigma_a)$.

Let us come back to the example in Figure 5.1. At state q_2, flyP is a move of the coalition $\{1\}$ which contains only the first agent. The cost of this move is simply the cost of the action $flyP$, i.e. $(1, 0)$. The set of possible outcomes of this move is $out(q_2, (flyP)) = \{q_3, q_5\}$. Let us consider another move (flyA,null) of the coalition $\{1, 2\}$ which consists of both agents where fltyA is the action of the first agent and null of the second one. The cost of this action is $(2, 0) + (0, 0) = (2, 0)$. Then, the set of possible outcomes of this move is $out(q_2, (flyA, null)) = \{q_4\}$.

From the notion of moves, let us now define strategies.

Definition 18. *Given a RB-CGS S, a strategy for a coalition $A \subseteq N$ is a mapping F_A which associates each sequence $\lambda q \in Q^{+}$ to a move in $D_A(q)$.*

A computation $\lambda \in Q^{\omega}$ is consistent with F_A iff for all $i \ge 0$ we have that

$$\lambda[i + 1] \in out(\lambda[i], F_A(\lambda[0, i]))$$

We denote $out(q, F_A)$ the set of all sequences λ consistent with F_A which start from q, i.e. $q = \lambda[0]$. Given a bound $b \in \mathbb{B}^{\infty}$, a computation $\lambda \in out(q, F_A)$ is b-consistent with F_A iff, for every

$i \geq 0$, $\sum_{j=0}^{i} cost(\lambda[i], F_A(\lambda[0,i])) \leq b$. We denote $out(q_0, F_A, b)$ the set of all such b-consistent sequences with F_A. A strategy F_A is a b-strategy iff $out(q, F_A) = out(q, F_A, b)$ for any $q \in Q$.

For example, let us consider some strategies of the system in Figure 5.1. Below is a strategy $F_{\{1\}}$ for the first agent to get to Ams on time:

- $F_{\{1\}}(q_1)$ = trainLd
- $F_{\{1\}}(q_1 q_2)$ = trainP
- $F_{\{1\}}(q_1 a_2 q_3)$ = trainA
- $F_{\{1\}}(q_1 a_2 q_3 q_4 q_4^*)$ = null

Then, it is easy to see that this strategy has only a single consistent computation, we have that $out(q_1, F_{\{1\}}) = \{q_1 q_2 q_3 q_4 q_4 \dots\}$. Moreover, we also have that $F_{\{1\}}$ is a $(0, 8)$-strategy.

Let us consider another strategy $F'_{\{1\}}(\lambda)$ also for the first agent to get to Ams as follows:

- $F'_{\{1\}}(q_1)$ = trainLd
- $F'_{\{1\}}(q_1 q_2)$ = flyA
- $F'_{\{1\}}(q_1 a_2 q_4 q_4^*)$ = null
- $F'_{\{1\}}(q_1 a_2 q_6 q_6^*)$ = null

Then, we have that $out(q_1, F'_{\{1\}}) = \{q_1 q_2 q_3 q_4 q_4 \dots, q_1 q_2 q_3 q_6 q_6 \dots\}$ and $F'_{\{1\}}$ is a (2,2)-strategy.

Using the notions of moves and strategies, let us define the semantics of RB-ATL. Given a RB-CGS $S = (n, R, Q, Prop, \pi, d, c, \delta)$, the truth of a RB-ATL formula is defined inductively as follows:

- $S, q \models p$ iff $p \in \pi(q)$
- $S, q \models \neg\varphi$ iff $S, q \not\models \varphi$
- $S, q \models \varphi \vee \psi$ iff $S, q \models \varphi$ or $S, q \models \psi$
- $S, q \models \langle\langle A^b \rangle\rangle \bigcirc \varphi$ iff there exists a b-strategy F_A such that for all $\lambda \in out(q, F_A)$, $S, \lambda[1] \models \varphi$ iff there is a move $\sigma_A \in D_A(q)$ such that for all $q' \in out(q, \sigma_A)$, $S, q' \models \varphi$
- $S, q \models \langle\langle A^b \rangle\rangle \Box \varphi$ iff there exists a b-strategy F_A such that for all $\lambda \in out(q, F_A)$, $S, \lambda[i] \models \varphi$ for all $i \geq 0$

- $S, q \vDash \langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi$ iff there exists a b-strategy F_A such that for all $\lambda \in out(q, F_A)$, there is a position $i \geq 0$ such that $S, \lambda[i] \vDash \psi$ and $S, \lambda[j] \vDash \psi$ for all $j \in \{0, \ldots, i-1\}$

Before the end of this section, let us consider the truth of some properties we have listed for the example in Figure 5.1. For convenience, let S be the name of the transition system depicted in Figure 5.1. We have that $S, q_1 \vDash \langle\langle \{1\}^{(0,\infty)} \rangle\rangle \top \mathcal{U}(Ams \wedge \neg Late)$ since the first agent can apply the strategy $F_{\{1\}}$, then $out(q_1, F_{\{1\}}) = \{q_1 q_2 q_3 q_4 q_4 \ldots\}$ and we have that $S, q_4 \vDash Ams \wedge \neg Late$. As we know that $F_{\{1\}}$ is a (0,8)-strategy and $(0,8) \leq (0,\infty)$, which means that $F_{\{1\}}$ is also a $(0,\infty)$-strategy.

5.5 Normal form RB-ATL

For the sake of simplicity when dealing with the soundness and completeness of RB-ATL, as well as the satisfiability problem of RB-ATL, we work with normal form formulas rather than arbitrary ones of RB-ATL. In this section, we present the syntax and semantics of the normal form RB-ATL.

5.5.1 Syntax of normal form RB-ATL

A formula of RB-ATL is said to be in normal form iff the negation symbol can only appear in front of a propositional variable or a temporal operator. Given a finite set N of agents where $N = \{1, \ldots, n\}$, a finite set R of resources where $|R| = r$, the set $\mathbb{B}^\infty$ of extended resource bounds, and a set of propositional variables *Prop*, we formally define the syntax of normal form RB-ATL as follows.

$$\varphi ::= p \mid \neg p \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \langle\langle A^b \rangle\rangle \bigcirc \varphi \mid \neg \langle\langle A^b \rangle\rangle \bigcirc \varphi \mid \langle\langle A^b \rangle\rangle \Box \varphi \mid \neg \langle\langle A^b \rangle\rangle \Box \varphi \mid \langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi \mid \neg \langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi$$

where $p \in$ *Prop*, A is a non-empty coalition and $b \in \mathbb{B}^\infty$.

It is clear that any formula of RB-ATL can be equivalently converted into normal form by applying the De Morgan's law where $\neg(\varphi \wedge \varphi) \equiv \neg\varphi \vee \neg\varphi$ and $\neg(\varphi \vee \varphi) \equiv \neg\varphi \wedge \neg\varphi$. For a normal form formula φ, we denote $\sim\varphi$ as the equivalent normal form formula of $\neg\varphi$.

Before defining the semantics of normal form RB-ATL, we need more definitions about the counterpart of moves and strategies from RB-ATL in order to define the semantics for formulas of the forms $\neg\langle\langle A^b \rangle\rangle \bigcirc \varphi$, $\neg\langle\langle A^b \rangle\rangle \Box \varphi$ and $\neg\langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi$ where the negation symbol appears in front of a temporal operator. In the following, we define the notion of co-moves.

Definition 19. *Given a RB-CGS S and a state $q \in Q$, a co-move for a coalition $A \subseteq N$ is a mapping $\sigma_A^c : D_A(q) \to Q$ such that $\sigma_A^c(\sigma_A) \in out(q, \sigma_A)$ for any $\sigma_A \in D_A(q)$.*

Let $D^c_A(q)$ denote the set of all co-moves for A at a state $q \in Q$. A state q' is consistent with a co-move σ^c iff there is some move σ_A such that $\sigma^c(\sigma_A) = q'$. We define the set of consistent outcomes for a co-move σ^c by

$$out(q, \sigma^c) = \{q' \in Q \mid q' \text{ is consistent with } \sigma^c\}$$

Given a bound $b \in \mathbb{B}^\infty$, a state q' is b-consistent with a co-move σ^c at q iff there is some move $\sigma_A \in D_A(q)$ with $cost(q, \sigma_A) \leq b$ such that $\sigma^c(\sigma_A) = q'$. We denote the set of b-consistent outcomes for a co-move σ^c by

$$out(q, \sigma^c, b) = \{q' \in Q \mid q' \text{ is } b\text{-consistent with } \sigma^c \text{ at } q\}$$

Then, we define the notion of co-strategy as follows.

Definition 20. *Given a RB-CGS S, a co-strategy for a subset of players $A \subseteq N$ is a mapping F^c_A which assigns each sequence $\lambda q \in Q^+$ to a co-move in $D^c_A(q)$.*

We say a computation $\lambda \in Q^\omega$ is consistent with F^c_A iff, for all $i \geq 0$, $\lambda[i+1] \in out(\lambda[i], F^c_A(\lambda[0,i]))$. Let us define $out(q, F^c_A)$ to be the set of all such consistent computations where $\lambda[0] = q$.

Given a bound $b \in \mathbb{B}^\infty$, we say a computation $\lambda \in out(q, F^c_A)$ is b-consistent with F^c_A iff there is an infinite sequence of moves $(\sigma^i_A)_{i\geq 0}$ for the coalition A where $\sigma^i_A \in D_A(\lambda[i])$ and $\lambda_{i+1} = F^c_A(\lambda[0,i])(\sigma^i_A)$ for all $i \geq 0$, such that $\sum_j cost(\lambda[j], \sigma^j_A) \leq b$ for all $j \geq 0$. Let us denote $out(q, F^c_A, b)$ be the set of all such b-consistent computations where $\lambda[0] = q$.

5.5.2 Fixed-point characterisations of temporal operators

In this section, we study fixed-point characterisations of temporal operators in RB-ATL, which will reveal how useful the notions of co-moves and co-strategies are for defining the semantics of normal form RB-ATL. Furthermore, fixed-point characterisations also suggest key axioms for the axiomatisation of RB-ATL.

Given a RB-CGS S, we define $\|\varphi\| = \{q \in Q \mid S, q \models \varphi\}$, i.e. the set of states where φ is true. Let us first have the following definition.

Definition 21. *The function $[\langle\langle A^b\rangle\rangle\bigcirc] : \wp(Q) \to \wp(Q)$ is defined as*

$$[\langle\langle A^b\rangle\rangle\bigcirc](X) = \{q \mid \exists\sigma \in D_A(q) : cost(\sigma) \leq b \wedge out(q, \sigma) \subseteq X\}$$

Informally, $[\langle\langle A^b\rangle\rangle\bigcirc](X)$ where $X \subseteq Q$ is the set of all states q where agents in the coalition A have a move costing at most b so that the outcome is within X; in this situation, we shall say that the coalition A is b-effective for X at q. Obviously, we have that:

$$[\langle\langle A^b\rangle\rangle\bigcirc](\|\varphi\|) = \|\langle\langle A^b\rangle\rangle\bigcirc\varphi\|$$

Similarly, we also define the following function.

Definition 22. *The function* $[\neg\langle\langle A^b\rangle\rangle\bigcirc] : \wp(Q) \to \wp(Q)$ *is defined as*

$$[\neg\langle\langle A^b\rangle\rangle\bigcirc](X) = \{q \mid \exists\sigma^c \in D^c_A(q) : out(q, \sigma^c, b) \subseteq Q \setminus X\}$$

In contrast to $[\langle\langle A^b\rangle\rangle\bigcirc]$, the function $[\neg\langle\langle A^b\rangle\rangle\bigcirc](X)$ defines the set of states q where A is not b-effective for X. It is straightforward that:

$$[\neg\langle\langle A^b\rangle\rangle\bigcirc](\|\varphi\|) = Q \setminus [\langle\langle A^b\rangle\rangle\bigcirc](\|\varphi\|) = \|\neg\langle\langle A^b\rangle\rangle\bigcirc\varphi\|$$

Before discussing fixed-point characterisations in more details, let us define some notions. Given three bounds $b, d, e \in \mathbb{B}^\infty$, we say that $b +^\infty d = e$ iff for every $i = 1, \ldots, r$,

$$\begin{aligned} b_i + d_i = e_i &\text{ if } e_i \neq \infty \\ b_i = d_i = \infty &\text{ if } e_i = \infty \end{aligned}$$

Then, $+^\infty$ only allows performing addition over natural numbers rather than the infinity symbol ∞. Then, given a bound $e \in \mathbb{B}^\infty$, the set $\{(b, d) \in \mathbb{B}^\infty \times \mathbb{B}^\infty \mid b +^\infty d = e\}$ must be finite. For example, let us consider the bound $(1, 2, \infty)$, the set of $(b, d) \in \mathbb{B}^\infty \times \mathbb{B}^\infty$ where $b +^\infty d = (1, 2, \infty)$ contains the following pairs: $(1, 2, \infty)$ and $(0, 0, \infty)$, $(0, 2, \infty)$ and $(1, 0, \infty)$, $(1, 1, \infty)$ and $(0, 1, \infty)$, $(1, 0, \infty)$ and $(0, 2, \infty)$, and the last one $(0, 0, \infty)$ and $(1, 2, \infty)$.

Furthermore, we define a projection d_b of bound $d \in \mathbb{B}^\infty$ with respect to a bound b in $\mathbb{B}^\infty$ as follows, for all $i = 1, \ldots, |r|$

$$(d_b)_i = \begin{cases} d_i \text{ if } b_i \neq \infty \\ \infty \text{ otherwise} \end{cases}$$

For example, we have that $\bar{0}_{(1,2,\infty)} = (0, 0, \infty)$, $(3, 3, 1)_{(1,2,\infty)} = (3, 3, \infty)$ and $(\infty, 3, 1)_{(1,2,\infty)} = (\infty, 3, \infty)$.

Then, we define the following macros for lengthy formulas:

$$\langle\!\langle A^b\rangle\!\rangle\bigcirc\square\varphi = \bigvee_{\substack{b_1+^\infty b_2=b\\ b_1\neq\bar{0}_b}}\langle\!\langle A^{b_1}\rangle\!\rangle\bigcirc\langle\!\langle A^{b_2}\rangle\!\rangle\square\varphi$$
$$\neg\langle\!\langle A^b\rangle\!\rangle\bigcirc\square\varphi = \bigwedge_{\substack{b_1+^\infty b_2=b\\ b_1\neq\bar{0}_b}}\neg\langle\!\langle A^{b_1}\rangle\!\rangle\bigcirc\langle\!\langle A^{b_2}\rangle\!\rangle\square\varphi$$
$$\langle\!\langle A^b\rangle\!\rangle\bigcirc\varphi\mathcal{U}\psi = \bigvee_{\substack{b_1+^\infty b_2=b\\ b_1\neq\bar{0}_b}}\langle\!\langle A^{b_1}\rangle\!\rangle\bigcirc\langle\!\langle A^{b_2}\rangle\!\rangle\varphi\mathcal{U}\psi$$
$$\neg\langle\!\langle A^b\rangle\!\rangle\bigcirc\varphi\mathcal{U}\psi = \bigwedge_{\substack{b_1+^\infty b_2=b\\ b_1\neq\bar{0}_b}}\neg\langle\!\langle A^{b_1}\rangle\!\rangle\bigcirc\langle\!\langle A^{b_2}\rangle\!\rangle\varphi\mathcal{U}\psi$$

We have the following result.

Lemma 11. *For all $q\in Q$, the following fixed-point characterisations hold:*

1. $q\in\|\langle\!\langle A^b\rangle\!\rangle\square\varphi\|$ *iff* $q\in\nu X.\|\varphi\|\cap(\|\langle\!\langle A^b\rangle\!\rangle\bigcirc\square\varphi\|\cup[\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](X))$ *iff there is a b-strategy F_A for A such that for all $\lambda\in out(q,F_A)$, $\lambda[i]\in\|\varphi\|$ for all $i\geq 0$*
2. $q\in\|\langle\!\langle A^b\rangle\!\rangle\varphi\mathcal{U}\psi\|$ *iff* $q\in\mu X.\|\psi\|\cup(\|\varphi\|\cap(\|\langle\!\langle A^b\rangle\!\rangle\bigcirc\varphi\mathcal{U}\psi\|\cup[\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](X)))$ *iff there is a b-strategy F_A for A such that for all $\lambda\in out(q,F_A)$, there exists $i\geq 0$ such that $\lambda[i]\in\|\psi\|$ and $\lambda[j]\in\|\varphi\|$ for all $j\leq i$*

Proof. We will only provide the proof for the first case as the second can be done in a similar way. For convenience, let us denote $f(X)=\|\varphi\|\cap(\|\langle\!\langle A^b\rangle\!\rangle\bigcirc\square\varphi\|\cup[\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](X))$. We firstly show that $f(X)$ is monotone. Let $X_1\subseteq X_2\subseteq Q$. Let $q\in f(X_1)$, then $q\in\|\varphi\|$ and either $q\in\|\langle\!\langle A^b\rangle\!\rangle\bigcirc\square\varphi\|$ or $q\in[\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](X_1)$. According to definition of $[\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc]()$, it is easy to see that $q\in[\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](X_2)$ if $q\in[\langle\!\langle A^{0}\rangle\!\rangle\bigcirc](X_1)$, hence $q\in f(X_2)$.

Therefore, $f(X)$ is monotone and there is the greatest fixed point $\nu X.f(X)$. We now show that $Y=\|\langle\!\langle A^b\rangle\!\rangle\square\varphi\|$ is a post-fixed point of $f(X)$, i.e. $f(Y)\subseteq Y$. Let $q\in Y$, by the semantics definition, we have that there is a b-strategy F_A such that for any $\lambda\in out(q,F_A)$, $\lambda[i]\in\|\varphi\|$ for all $i\geq 0$. Then, $q=\lambda[0]\in\|\varphi\|$. Assume that $b'=cost(q,F_A(q))$, let b'' be an extended resource bound such that $b'_b+^\infty b''=b$. For every $q'\in out(q,F_A(q))$, we define a b-strategy F'_A which is the remainder of F_A from q' as follows, $F'_A(\kappa)=F_A(q\kappa)$ for all $\kappa\in Q^+$ starting at q'. Then, for all $\kappa\in out(q',F'_A)$, we have that $q\kappa\in out(q,F_A)$. It is straightforward that any computation in $out(q',F'_A)$ costs at most b''. Then, for all $i\geq 0$, we have that $\kappa[i]\in\|\varphi\|$, hence $q'\in\|\langle\!\langle A^{b''}\rangle\!\rangle\square\varphi\|$. Thus, $q\in[\langle\!\langle A^{b'}\rangle\!\rangle\bigcirc](\|\langle\!\langle A^{b''}\rangle\!\rangle\square\varphi\|)$. If $b'\neq\bar{0}$, we have that $q\in\|\langle\!\langle A^b\rangle\!\rangle\bigcirc\square\varphi\|$, otherwise $q\in[\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](\|\langle\!\langle A^b\rangle\!\rangle\square\varphi\|)$. This means that $q\in f(\|\langle\!\langle A^b\rangle\!\rangle\square\varphi\|)$.

In order to show that $Y=\|\langle\!\langle A^b\rangle\!\rangle\square\varphi\|$ is, in fact, the greatest fixed point of $f(X)$, we show that for every post-fixed point Z, $Z\subseteq Y$. Considering that b is the projection of a bound $d\in\mathbb{B}$ and b, we show the inclusion by induction on d.

The base case:

In the base case, $d = \bar{0}$, we have $f(X) = \|\varphi\| \cap [\langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc](X)$. Assume $q \in Z$, then $q \in \|\varphi\| \cap [\langle\langle A^{0} \rangle\rangle \bigcirc](Z)$ as Z is a post-fixed point of $f(X)$. We now define a $\bar{0}_b$-strategy F_A which will maintain φ for any consistent computation. The definition proceeds by induction on the length of inputs for F_A. Moreover, we only define F_A for inputs which will be used later for the coalition to determine which joint action to perform in order to maintain φ. Let Λ^n denote the set of such inputs of length n. Initially, $\Lambda^1 = \{q\}$. We will define F_A and Λ^{i+1} inductively on i such that the last element of any member of Λ^{i+1} is always in Z.

- When $i = 1$, we have that $q \in \|\varphi\|$ and there is a move $\sigma_A \in D_A(q)$ with $cost(q, \sigma_A) = \bar{0}_b$ such that $out(q, \sigma_A) \subseteq Z$. Let $F_A(q) = \sigma_A$ and $\Lambda^2 = \{qq' \mid q' \in out(q, F_A(q))\}$. For all such states q', we have $q' \in Z \subseteq f(Z)$.

- When $i > 1$, for any $\lambda \in \Lambda^i$, we have that $\lambda[i-1] \in Z \subseteq f(Z)$ by the induction hypothesis. We have that $\lambda[i-1] \in \|\varphi\|$ and there is a move $\sigma_A \in D_A(\lambda[i-1])$ with $cost(\lambda[i-1], \sigma_A) = \bar{0}_b$ such that $out(\lambda[i-1], \sigma_A) \subseteq Z$. Let $F_A(\lambda) = \sigma_A$ and $\Lambda^{i+1}(\lambda) = \{\lambda q' \mid q' \in out(\lambda[i-1], F_A(\lambda))\}$.

  Finally, we define $\Lambda^{i+1} = \bigcup_{\lambda \in \Lambda^i} \Lambda^{i+1}(\lambda)$. By the definition of $\Lambda^{i+1}(\lambda)$, it is easy to see that for any $\lambda' \in \Lambda^{i+1}$, $\lambda'[i] \in Z \subseteq f(Z)$.

After defining F_A, we have that for any $\lambda \in out(q, F_A)$ and $i \geq 0$, $\lambda[0, i] \in \Lambda^{i+1}$, hence $\lambda[i] \in Z \subseteq f(Z)$. Therefore, $\lambda[i] \in \|\varphi\|$. This shows that $q \in Y$.

The induction step:

In the induction step, $d > \bar{0}$, we have $f(X) = \|\varphi\| \cap (\|\langle\langle A^{d_b} \rangle\rangle \bigcirc \Box \varphi\| \cup [\langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc](Z))$. Assume $q \in Z$, then $q \in \|\varphi\|$ and either $q \in \|\langle\langle A^{d_b} \rangle\rangle \bigcirc \Box \varphi\|$ or $q \in [\langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc](Z)$. Similar to the base case, we also define a d_b-strategy F_A which will maintain φ for any consistent computation. The definition will proceed by induction on the length of inputs for F_A. Moreover, we only define F_A for inputs which will be used later for the coalition to determine which joint action to perform in order to maintain φ. Let Λ^n denote the set of such inputs of length n. Initially, $\Lambda^1 = \{q\}$. We will define F_A and Λ^{i+1} inductively on i such that the last element of any member of Λ^{i+1} is always either in $\|\langle\langle A^{b_2} \rangle\rangle \Box \varphi\|$ if the accumulated cost along that member is no more than b_1 for some $b_1 +^{\infty} b_2 = d_b$ or in Z if the same cost is less than $\bar{0}_b$.

- When $i = 1$, we have that $q \in \|\varphi\|$ and either $q \in \|\langle\langle A^{d_b}\rangle\rangle \bigcirc \square \varphi\|$ or $q \in [\langle\langle A^{\bar{0}_b}\rangle\rangle \bigcirc](Z)$.

  If $q \in \|\langle\langle A^{d_b}\rangle\rangle \bigcirc \square \varphi\|$, there is $b_1 +^\infty b_2 = d_b$ such that $q \in [\langle\langle A^{b_1}\rangle\rangle \bigcirc](\|\langle\langle A^{b_2}\rangle\rangle \square\varphi\|)$. Then, there is a move $\sigma_A \in D_A(q)$ with $cost(q, \sigma_A) \leq b_1$ such that $out(q, \sigma_A) \subseteq \|\langle\langle A^{b_2}\rangle\rangle \square\varphi\|$. By the induction hypothesis, for any $q' \in out(q, \sigma_A)$, there is another b_2-strategy F'_A from q' to maintain φ, we define $F_A(qq'\lambda) = F'_A(q'\lambda)$ for all $\lambda \in Q^*$. Let $F_A(q) = \sigma_A$ and $\Lambda^2 = \{qq' \mid q' \in out(q, \sigma_A)\}$. It is obvious that all such states $q' \in \|\langle\langle A^{b_2}\rangle\rangle \square\varphi\|$ and the cost along qq' is at most b_1.

  If $q \in [\langle\langle A^{\bar{0}_b}\rangle\rangle \bigcirc](Z)$, there is a move $\sigma_A \in D_A(q)$ with $cost(q, \sigma_A) \leq \bar{0}_b$ such that $out(q, \sigma_A) \subseteq Z$. Let $F_A(q) = \sigma_A$ and $\Lambda^2 = \{qq' \mid q' \in out(q, \sigma_A)\}$. It is obvious that all such $q' \in Z$ and the cost along qq' is less than $\bar{0}_b$.

- When $i > 1$, for any $\lambda \in \Lambda^i$, we have that either $\lambda[i-1] \in \|\langle\langle A^{b_2}\rangle\rangle \square\varphi\|$ if

$$\sum_{j<i-1} cost(\lambda[j], F_A(\lambda[0, j])) \leq b_1$$

  for some $b_1 +^\infty b_2 = d_b$ or $\lambda[i-1] \in Z \subseteq f(Z)$ if

$$\sum_{j<i-1} cost(\lambda[j], F_A(\lambda[0, j])) = \bar{0}_b$$

  by the induction hypothesis.

  - If $\lambda[i-1] \in \|\langle\langle A^{b_2}\rangle\rangle \square\varphi\|$, then F_A has been defined. Let $\Lambda^{i+1}(\lambda) = \{\lambda q' \mid q' \in out(\lambda[i-1], F_A(\lambda[0, i-1]))\}$. Assume that $b' = cost(\lambda[i-1], F_A(\lambda[0, i-1]))$ and let $b'' \in \mathbb{B}^\infty$ such that $b'_d + b'' = b_2$. By the induction hypothesis, as $b_2 < b$, we have that all such $q' \in \|\langle\langle A^{b''}\rangle\rangle \square\varphi\|$ and $\sum_{j<i} cost(\lambda[j], F_A(\lambda[0, j])) \leq b_1 + b'_d$.
  - If $\lambda[i-1] \in Z \subseteq f(Z)$, then $\lambda[i-1] \in \|\varphi\|$ and either $\lambda[i-1] \in \|\langle\langle A^{d_b}\rangle\rangle \bigcirc \square \varphi\|$ or $\lambda[i-1] \in [\langle\langle A^{\bar{0}_b}\rangle\rangle \bigcirc](Z)$.
    * If $\lambda[i-1] \in \|\langle\langle A^{d_b}\rangle\rangle \bigcirc \square \varphi\|$, there exists $b_1 +^\infty b_2 = d_b$ such that

$$\lambda[i-1] \in [\langle\langle A^{b_1}\rangle\rangle \bigcirc](\|\langle\langle A^{b_2}\rangle\rangle \square\varphi\|)$$

      Then, there is a move $\sigma_A \in D_A(\lambda[i-1])$ with $cost(\lambda[i-1], \sigma_A) \leq b_1$ such that $out(\lambda[i-1], \sigma_A) \subseteq \|\langle\langle A^{b_2}\rangle\rangle \square\varphi\|$. By the induction hypothesis, for any $q' \in out(\lambda[i], \sigma_A)$, there is another b_2-strategy from q to maintain φ, we define

$$F_A(\lambda q'\kappa) = F'_A(q'\kappa)$$

for all $\kappa \in Q^*$. Let $F_A(\lambda) = \sigma_A$ and $\Lambda^{i+1}(\lambda) = \{\lambda q' \mid q' \in out(\lambda[i-1], \sigma_A)\}$. Then, for all such states q' we have $q' \in \|\langle\!\langle A^{b_2} \rangle\!\rangle \bigcirc \Box \varphi\|$ and $\sum_{j<i} cost(\lambda[j], F_A(\lambda[0,j])) \leq b_1$.

* If $\lambda[i-1] \in [\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc](Z))$, there is a move $\sigma_A \in D_A(\lambda[i-1])$ with $cost(\lambda[i-1], \sigma_A) \leq \bar{0}_b$ such that $out(\lambda[i-1], \sigma_A) \subseteq Z$. Let $F_A(\lambda) = \sigma_A$ and $\Lambda^{i+1}(\lambda) = \{\lambda q' \mid q' \in out(\lambda[i-1], \sigma_A)\}$. Then, for all such q', we have that $q' \in Z$ and $\sum_{j<i} cost(\lambda[j], F_A(\lambda[0,j])) \leq \bar{0}_b$.

Then, $\Lambda^{i+1} = \bigcup_{\lambda \in \Lambda^i} \Lambda^{i+1}(\lambda)$.

After defining F_A, we have that for any $\lambda \in out(q, F_A)$ and $i \geq 0$, $\lambda[0,i] \in \Lambda^{i+1}$, hence $\lambda[i] \in Z \subseteq f(Z)$. Therefore, $\lambda[i] \in \|\varphi\|$. This shows that $q \in Y$.

Therefore, Y is the greatest post-fixed point of $f(X)$, hence also the greatest fixed point of $f(X)$. $\Box$

Similarly, we also have the following result.

Lemma 12. *For all $q \in Q$, the following fixed-point characterisations hold:*

1. $q \in \|\neg\langle\!\langle A^b \rangle\!\rangle \Box \varphi\|$ *iff* $q \in \mu X. \| \sim\varphi\| \cup (\|\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \Box \varphi\| \cap [\neg\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc](Q \setminus X))$ *iff there is a co-strategy F_A^c such that for all $\lambda \in out(q, F_A, b)$, $\lambda[i] \in \|\neg\varphi\|$ for some $i \geq 0$*

2. $q \in \|\neg\langle\!\langle A^b \rangle\!\rangle \varphi \mathcal{U} \psi\|$ *iff* $q \in \nu X. \| \sim\psi\| \cap (\| \sim\varphi\| \cup (\|\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi \mathcal{U} \psi\| \cap [\neg\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc](Q \setminus X)))$ *iff there is a co-strategy F_A such that for all $\lambda \in out(q, F_A, b)$, if there is $i \geq 0$ such that $\lambda[i] \in \|\psi\|$, then there exists $j < i$ such that $\lambda[j] \in \|\neg\varphi\|$*

Proof. Similar to the previous lemma, we only show the first case, others can be also done in a similar way.

We have that

$$\begin{aligned} \|\neg\langle\!\langle A^b \rangle\!\rangle \Box \varphi\| &= Q \setminus \|\langle\!\langle A^b \rangle\!\rangle \Box \varphi\| \\ &= Q \setminus \nu X. \|\varphi\| \cap (\|\langle\!\langle A^b \rangle\!\rangle \bigcirc \Box \varphi\| \cup [\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc](X)) \\ &= \mu X. \| \sim\varphi\| \cup (\|\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \Box \varphi\| \cap [\neg\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc](Q \setminus X)) \end{aligned}$$

Let $f(X) = \| \sim\varphi\| \cup (\|\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \Box \varphi\| \cap [\neg\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc](Q \setminus X))$. Let $Z \subseteq Q$ be the set of states q where there is a co-strategy F_A^c such that for every $\lambda \in out(q, F_A^c, b)$, $\lambda[i] \in \| \sim\varphi\|$ for some $i \geq 0$. Let us now show that Z is a pre-fixed point of $f(X)$, i.e. $f(Z) \subseteq Z$.

Assume $q \in f(Z)$. If $q \in \| \sim\varphi\|$, then for any co-strategy F^c_A, every computation $\lambda \in out(q, F^c_A)$ from q has that $\lambda[0] \in \| \sim\varphi\|$. If $q \in \|\neg\langle\!\langle A^b\rangle\!\rangle\bigcirc \Box \varphi\| \cap [\neg\langle\!\langle A^0\rangle\!\rangle\bigcirc](Q \setminus Z)$, we continue the proof by induction on components of a bound b which are not ∞. In order to do so, the proof proceeds by induction on the projection d_b of a bound $b \in \mathbb{B}$ and an arbitrary bound $d \in \mathbb{B}^\infty$.

The base case:

Let $d = \bar{0}$, then $q \in [\neg\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](Q \setminus Z)$. This means there is a co-move $\sigma^c \in D^c_A(q)$ such that $out^c(q, \sigma^c, \bar{0}_b) \subseteq Q \setminus (Q \setminus Z) = Z$. Then, for each $q' \in out^c(q, \sigma^c, \bar{0}_b)$, we have a co-strategy $F^c_{A,q'}$ such that for each computation $\lambda \in out^c(q', F^c_{A,q'}, \bar{0}_b)$, $\lambda[i] \in \| \sim\varphi\|$ for some $i \geq 0$. We just need to define a strategy $F^c_A(q) = \sigma^c$ and $F^c_A(qq'\kappa) = F^c(q'\kappa)$ for all such states q' and $\kappa \in Q^*$. It is easy to see that for any $\lambda \in out^c(q', F^c_{A'}, \bar{0}_b)$, $\lambda[i] \in \| \sim\varphi\|$ for some $i \geq 1$.

The induction step:

Let $d > \bar{0}$, then we have both $q \in \|\neg\langle\!\langle A^{d_b}\rangle\!\rangle\bigcirc \Box \varphi\|$ and $q[\neg\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](Q \setminus Z))$.

- Since $q \in \|\neg\langle\!\langle A^b\rangle\!\rangle\bigcirc \Box \varphi\|$, then for every pair of extended resource bounds b_1 and b_2 such that $b_1 +^\infty b_2 = b$, we have $q \in [\neg\langle\!\langle A^{b_1}\rangle\!\rangle\bigcirc](\|\langle\!\langle A^{b_2}\rangle\!\rangle\Box\varphi\|))$. This means there is a co-move $\sigma^c_{b_1} \in D^c_A(q)$ such that $out^c(q, \sigma^c_{b_1}, b_1) \subseteq \|\neg\langle\!\langle A^{b_2}\rangle\!\rangle\Box\varphi\|$. Let us pick a co-move $\sigma^c \in D^c_A(q)$ as follows, for each $\sigma \in D_A(q)$ such that $cost(q, \sigma) = b_1 \leq b$, $\sigma^c(\sigma) = \sigma^c_{b_1}(\sigma)$. Then, we define $F^c_A(q) = \sigma^c$. For each $q' \in out^c(q, \sigma^c, b)$ where $q' = \sigma^c(\sigma)$ for some $cost(q, \sigma) = b_1 \leq b$, by the induction hypothesis, there is a co-strategy $F^c_{A,q'}$ such that for all $\lambda \in out(q', F^c_{A,q'}, b_2)$, $\lambda[i] \in \| \sim\varphi\|$ for some $i \geq 0$. Let $F^c_A(qq'\kappa) = F^c_{A,q'}(q'\kappa)$ for all such q' and $\kappa \in Q^*$. Then, it is easy to see that for every $\lambda \in out(q, F^c_A, b)$, $\lambda[i] \in \| \sim\varphi\|$ for some $i \geq 1$.
- Since $q \in [\neg\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](Q \setminus Z))$, the proof can proceed similarly to the base case.

In order to prove that Z is in fact the least fixed point of $f(X)$, we show that Z is included in all pre-fixed points of $f(X)$. Assume that Y is a pre-fixed point of $f(X)$, i.e. $Y \supseteq f(Y)$, we now prove for every $q \notin Y$ that $q \notin Z$.

Let us consider b as the projection of a bound $d \in \mathbb{B}$ and $b \in \mathbb{B}^\infty$. Then, the proof is done by induction on d. In the following, the base case is included in the second case of the argument. As $q \notin Y$, hence $q \notin f(Y)$, then $q \notin \| \sim\varphi\|$ and $q \notin \|\neg\langle\!\langle A^{d_b}\rangle\!\rangle\bigcirc \Box \varphi\| \cap [\neg\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](Q \setminus Y)$. Then, either $q \notin [\neg\langle\!\langle A^{b_1}\rangle\!\rangle\bigcirc](\|\langle\!\langle A^{b_2}\rangle\!\rangle\Box\varphi\|)$ for some $b_1 +^\infty b_2 = d_b$ or $q \notin [\neg\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc](Q \setminus Y)$. Let us consider an arbitrary co-strategy F^c_A. Assume that $F^c_A(q) = \sigma^c$ for some $\sigma^c \in D^c_A(q)$.

- If $q \notin [\neg\langle\langle A^{b_1}\rangle\rangle\bigcirc](\|\langle\langle A^{b_2}\rangle\rangle\Box\varphi\|)$ for some $b_1 +^\infty b_2 = d_b$, we have that for any co-move $\sigma^c \in D^c_A(q)$
$$out^c(q, \sigma^c, b_1) \nsubseteq \|\neg\langle\langle A^{b_2}\rangle\rangle\Box\varphi\|$$
Therefore, there exists a state $q_1 \in out^c(q, \sigma^c, b_1)$ such that $q_1 \in \|\langle\langle A^{b_2}\rangle\rangle\Box\varphi\|$. Then, by the previous lemma, there is a b_2-strategy F_A such that for every computation $\lambda \in out(q_1, F_A)$, $\lambda[i] \notin \| \sim\varphi\|$ for all $i \geq 0$. Consider a computation $\lambda \in Q^\omega$ defined such that $\lambda[0] = q_1$ and $\lambda[i+1] = F^c_A(\lambda[0,i])(F_A(\lambda[0,i]))$ for all $i \geq 0$. It is straightforward that $\lambda \in out(q_1, F_A)$ and $out^c(q_1, F^c_A, b_2)$. This implies $\lambda[i] \notin \| \sim\varphi\|$ for all $i \geq 0$. Hence, we have a computation $\lambda' = q\lambda \in out^c(q, \sigma^c, d_b)$ where $\lambda'[i] \notin \| \sim\varphi\|$ for all $i \geq 0$.

- If $q \notin [\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc](Q \setminus Y)$, by a similar argument, we have a state $q_1 \in out^c(q, \sigma^c, \bar{0}_b)$ such that $q_1 \notin Y$. Hence, $q_1 \notin \| \sim\varphi\|$. By proceeding in the same manner, we can find a successor $q_2 \notin \| \sim\varphi\|$ of q_1. In this way, we construct a computation $\lambda = qq_1q_2\dots$ which is in $out^c(q, F^c_A, d_b)$ and satisfies $\lambda[i] \notin \| \sim\varphi\|$ for all $i \geq 0$.

In summary, we have shown that for any co-strategy F^c_A, there exists a computation $\lambda \in out^c(q, F^c_A, d_b)$ such that $\lambda[i] \notin \| \sim\varphi\|$ for all $i \geq 0$. Thus, $q \notin Z$ as well. $\Box$

5.5.3 Semantics of normal form RB-ATL

Provided the fixed-point characterisations of temporal operators in RB-ATL which have been shown in the previous section, we present the semantics for normal form RB-ATL which is also equivalent to that of RB-ATL.

Given a RB-CGS $S = (n, R, Q, Prop, \pi, d, c, \delta)$, the truth of a normal form RB-ATL formula is defined inductively as follows:

- $S, q \models p$ iff $p \in \pi(q)$
- $S, q \models \neg p$ iff $p \notin \pi(q)$
- $S, q \models \varphi \vee \psi$ iff $S, q \models \varphi$ or $S, q \models \psi$
- $S, q \models \varphi \wedge \psi$ iff $S, q \models \varphi$ and $S, q \models \psi$
- $S, q \models \langle\langle A^b\rangle\rangle\bigcirc\varphi$ iff there exists a b-strategy F_A such that for all $\lambda \in out(q, F_A)$, $S, \lambda[1] \models \varphi$ iff there is a move $\sigma_A \in D_A(q)$ such that for all $q' \in out(\sigma_A)$, $S, q' \models \varphi$

- $S, q \models \neg\langle\langle A^b\rangle\rangle\bigcirc\varphi$ iff there exists a co-strategy F_A^c such that for all $\lambda \in out(q, F_A, b)$, $S, \lambda[1] \models\sim \varphi$ iff there is a co-move $\sigma^c \in D_A^c(q)$ such that for all $\sigma_A \in D_A(q)$ and $cost(\sigma_A) \leq b$, $S, \sigma^c(\sigma_A) \models\sim\varphi$

- $S, q \models \langle\langle A^b\rangle\rangle\Box\varphi$ iff there exists a b-strategy F_A for any $\lambda \in out(q, F_A)$, $S, \lambda[i] \models \varphi$ for all $i \geq 0$

- $S, q \models \neg\langle\langle A^b\rangle\rangle\Box\varphi$ iff there exists a co-strategy F_A^c for any $\lambda \in out(q, F_A^c, b)$, $S, \lambda[i] \models \varphi$ for all $i \geq 0$

- $S, q \models \langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ iff there exists a b-strategy F_A such that for all $\lambda \in out(q, F_A)$, there is a position $i \geq 0$ such that $S, \lambda[i] \models \psi$ and $S, \lambda[j] \models \psi$ for all $j \in \{0, \ldots, i-1\}$

- $S, q \models \neg\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ iff there exists a co-strategy F_A^c such that for all $\lambda \in out(q, F_A, b)$, either $S, \lambda[i] \models \psi$ for all $i \geq 0$ or if there is a position $i \geq 0$ such that $S, \lambda[i] \models \psi$ then there exists $0 \leq j < i$ such that $S, \lambda[j] \models\sim\varphi$

5.6 Axiomatisation of RB-ATL

In this section, we present an axiomatisation system for RB-ATL. Then, we prove that the logic generated by the axiomatisation system for RB-ATL is sound and complete.

5.6.1 Axiomatisation of RB-ATL

The axiomatisation system for RB-ATL consists of the following axioms and inference rules. Let A be a non-empty coalition ($A \subseteq N$), and b, d, b_1, b_2 extended resource bounds, i.e. in $\mathbb{B}^\infty$.

Axioms

(PL) Tautologies of Propositional Logic

($\bot$) $\neg\langle\langle A^b\rangle\rangle\bigcirc\bot$

($\top$) $\langle\langle A^b\rangle\rangle\bigcirc\top$

(B) $\langle\langle A^b\rangle\rangle\bigcirc\varphi \rightarrow \langle\langle A^d\rangle\rangle\bigcirc\varphi$
where $b \leq d$

(S) $\langle\!\langle A_1^{b_1}\rangle\!\rangle\bigcirc\varphi \wedge \langle\!\langle A_2^{b_2}\rangle\!\rangle\bigcirc\psi \to \langle\!\langle (A_1 \cup A_2)^{b_1+b_2}\rangle\!\rangle\bigcirc(\varphi\wedge\psi)$
where both $A_1 \subseteq N$ and $A_2 \subseteq N$ are non-empty and $A_1 \cap A_2 = \varnothing$

(S$_\varnothing$) $\langle\!\langle \varnothing^{b}\rangle\!\rangle\bigcirc\varphi \wedge \langle\!\langle \varnothing^{b}\rangle\!\rangle\bigcirc\psi \to \langle\!\langle \varnothing^{b}\rangle\!\rangle\bigcirc(\varphi\wedge\psi)$

(S$_N$) $\langle\!\langle N^{b}\rangle\!\rangle\bigcirc\varphi \wedge \langle\!\langle \varnothing^{b}\rangle\!\rangle\bigcirc\psi \to \langle\!\langle N^{b}\rangle\!\rangle\bigcirc(\varphi\wedge\psi)$

(FP$_\Box$) $\langle\!\langle A^{b}\rangle\!\rangle\Box\varphi \leftrightarrow \varphi \wedge (\langle\!\langle A^{b}\rangle\!\rangle\bigcirc\Box\varphi \vee \langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc(\langle\!\langle A^{b}\rangle\!\rangle\Box\varphi))$

(FP$_\mathcal{U}$) $\langle\!\langle A^{b}\rangle\!\rangle\varphi\mathcal{U}\psi \leftrightarrow \psi \vee (\varphi \wedge (\langle\!\langle A^{b}\rangle\!\rangle\bigcirc\varphi\mathcal{U}\psi \vee \langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc(\langle\!\langle A^{b}\rangle\!\rangle\varphi\mathcal{U}\psi)))$

(N$_\bigcirc$) $\langle\!\langle \varnothing^{b}\rangle\!\rangle\bigcirc\varphi \leftrightarrow \neg\langle\!\langle N^{b}\rangle\!\rangle\bigcirc(\neg\varphi)$

(N$_\Box$) $\langle\!\langle \varnothing^{b}\rangle\!\rangle\Box\varphi \leftrightarrow \varphi \wedge \neg\langle\!\langle N^{b}\rangle\!\rangle\top\mathcal{U}\neg\varphi$

(N$_\mathcal{U}$) $\langle\!\langle \varnothing^{b}\rangle\!\rangle\varphi\mathcal{U}\psi \leftrightarrow \neg(\langle\!\langle N^{b}\rangle\!\rangle\neg\psi\mathcal{U}\neg(\varphi\vee\psi) \vee \langle\!\langle N^{b}\rangle\!\rangle\Box\neg\psi)$

Inference rules

(MP) $\dfrac{\varphi, \varphi\to\psi}{\psi}$

($\langle\!\langle A^{b}\rangle\!\rangle\bigcirc$-Monotonicity) $\dfrac{\varphi\to\psi}{\langle\!\langle A^{b}\rangle\!\rangle\bigcirc\varphi \to \langle\!\langle A^{b}\rangle\!\rangle\bigcirc\psi}$

($\langle\!\langle \varnothing^{b}\rangle\!\rangle\Box$-Necessitation) $\dfrac{\varphi}{\langle\!\langle \varnothing^{b}\rangle\!\rangle\Box\varphi}$

($\langle\!\langle A^{b}\rangle\!\rangle\Box$-Induction) $\dfrac{\theta \to (\varphi \wedge (\langle\!\langle A^{b}\rangle\!\rangle\bigcirc\Box\varphi \vee \langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc\theta))}{\theta \to \langle\!\langle A^{b}\rangle\!\rangle\Box\varphi}$

($\langle\!\langle A^{b}\rangle\!\rangle\mathcal{U}$-Induction) $\dfrac{(\psi \vee (\varphi \wedge (\langle\!\langle A^{b}\rangle\!\rangle\bigcirc\varphi\mathcal{U}\psi) \vee \langle\!\langle A^{\bar{0}_b}\rangle\!\rangle\bigcirc\theta))) \to \theta}{\langle\!\langle A^{b}\rangle\!\rangle\varphi\mathcal{U}\psi \to \theta}$

As usual, we define that a formula φ is a theorem of RB-ATL iff it is derivable from the above axiomatisation system, denoted as $\vdash_{\text{RB-ATL}} \varphi$. Then, a formula φ is consistent if its negation $\neg\varphi$ is not a theorem, i.e. $\nvdash_{\text{RB-ATL}} \varphi$.

In the rest of this section, we prove the soundness and completeness of the above axiomatisation system. As usual the soundness is omitted as it is straightforward. In the following, we present the proof of completeness. We show that by constructing a RB-CGS for each consistent formula φ_0 such that φ_0 is satisfied in the RB-CGS. Each RB-CGS structure is in the form of tree models. We formally define them in the following. The approach of the proof is based on the idea from [Goranko & van Drimmelen, 2006], but extends it for resource bounds.

5.6.2 Labelled tree models

Given a finite alphabet Θ, we denote the sets of finite words and infinite words of Θ by Θ^* and Θ^ω, respectively.

Definition 23. *A tree T is a subset of $\mathbb{N}^*$ where for any $x \cdot c \in T$, where $x \in \mathbb{N}^*$ and $c \in \mathbb{N}$:*

- $x \in T$
- $x \cdot c' \in T$ *for all* $0 \leq c' \leq c$

Given a tree T, ϵ is the root of T. Nodes of T are elements of T. We define $succ : T \to 2^T$ as a function to return the successors of a node $x \in T$. Formally, $succ(x) = \{x \cdot c \in T \mid c \in \mathbb{N}\}$. The degree $d(x)$ of a node x is defined as the cardinality of $succ(x)$, i.e. $d(x) = |succ(x)|$. A node x is a leaf iff $d(x) = 0$. A node x is an interior node iff $d(x) > 0$.

Definition 24. *Given a set Θ, a Θ-labelled tree is a pair (T, V) where T is a tree and $V : T \to \Theta$ is a mapping which labels each node of T with an element of Θ.*

Given a finite set of agents $N = \{1, \ldots, n\}$, for the purpose of constructing models for consistent formulas of RB-ATL, we are interested in a special form of Θ-labelled trees (T, V) where Θ is the set 2^{Prop} of subsets of propositions and the degree of every node of T is fixed by some given number $k \in \mathbb{N}$, i.e. $deg(x) = k^n$ for all $x \in T$. Then, a 2^{Prop}-labelled tree (T, V) with a fixed degree k^n can be considered as the skeleton of a model for RB-ATL formulas. We call a tree with a fixed degree k^n as a k^n-tree. Informally, each node of T is considered as a state. From each state $x \in T$, there are k^n transitions to its successors, namely from $x \cdot 0$ to $x \cdot k^n - 1$. We can name each transition from x to $x \cdot c$ by a tuple $(a_1, \ldots, a_n)$ where

1. $1 \leq a_i \leq k$
2. $encode((a_1, \ldots, a_n)) = c$

Where $encode : \{1, \ldots, k\}^n \to \{0, \ldots, k^n - 1\}$ is a bijective function which is defined as

$$encode((x_1, \ldots, x_n)) = (x_1 - 1)k^{n-1} + (x_2 - 1)k^{n-2} + \ldots + (x_n - 1)$$

For convenience, we call the inverse function of $encode$ as $decode$. Then, each transition from x to $x \cdot c$ can be considered as the effect of the joint action of n agents in N where agent i performs the action a_i for all $i \in \{1, \ldots, n\}$ and $(a_1, \ldots, a_n) = decode(c)$. Moreover, to become a model for RB-ATL formulas, we need to supply for each 2^{Prop}-labelled k^n-tree (T, V) a costing function which defines the cost of each action of an agent at a node on the tree. We have the following definition.

Definition 25. *A 2^{Prop}-labelled k^n-costed-tree is a tuple (T, V, C) where (T, V) is a 2^{Prop}-labelled k^n-tree and $C : T \times N \times \{1, \ldots, k\} \to \mathbb{B}$ is a costing function.*

Given a 2^{Prop}-labelled k^n-costed-tree (T, V, C), we define the corresponding RB-CGS

$$S_{(T,V,C)} = (n, T, Prop, V, d, C, \delta)$$

where $d(x, i) = k$ for all $x \in T$ and $i \in N$ and $\delta(x, (a_1, \ldots, a_n)) = x \cdot encode((a_1, \ldots, a_n))$. It is straightforward that $S_{(T,V,C)}$ is well-defined. We shall write $(T, V, C), x \models \varphi$ for $S_{(T,V,C)}, x \models \varphi$ and $(T, V, C) \models \varphi$ for $(T, V, C), \epsilon \models \varphi$. Furthermore, we also have that in $S_{(T,V,C)}$, the available joint actions for any coalition A at any state are the same, i.e. $D_A(x) = D_A(x')$ for any $x, x' \in T$, hence we shall write Δ_A for $D_A(x)$. For convenience, the cost of a joint action $\sigma \in \Delta_A$ at a state $t \in T$ is defined as $C(t, \sigma) = \sum_{i \in A} \sigma_i$. Similarly, we also have that $out(x, \sigma) = out(x', \sigma)$ for all $\sigma \in \Delta_A$ and $x, x' \in T$ (i.e. the outcomes of the same action are the same at any state), we shall write $out(\sigma)$ instead of $out(x, \sigma)$ for simplicity.

Notice that when constructing the tree model for a consistent formula, we build k^n-costed-trees which are labelled by subsets of formulas rather than only a subset of propositional variables. However, we can consider them as models for RB-ATL formulas by restricting the labeling function V over the set of propositions, i.e. $V(t) \cap Prop$. Finally, we define a simple tree as a tree which consists of only a root and its children.

5.6.3 Completeness of RB-ATL

In this section, we present the proof for the completeness of the logic RB-ATL.

Firstly, we define the closure $cl(\varphi_0)$ of a given consistent formula φ_0 which provides the ingredients for labelling nodes of the tree model during the construction.

Definition 26. *The closure $cl(\varphi_0)$ is the smallest set of formulas that satisfies the following conditions:*

- *All sub-formulas of φ_0 including itself are in $cl(\varphi_0)$*
- *If $\langle\!\langle A^b \rangle\!\rangle \Box \varphi$ is in $cl(\varphi_0)$, then so are $\langle\!\langle A^{b_1} \rangle\!\rangle \bigcirc \langle\!\langle A^{b_2} \rangle\!\rangle \Box \varphi$ for all $b_1 +^\infty b_2 = b$ and also $\langle\!\langle A^{0_b} \rangle\!\rangle \bigcirc \langle\!\langle A^b \rangle\!\rangle \Box \varphi$*

- *If* $\langle\!\langle A^b \rangle\!\rangle \varphi \mathcal{U} \psi$ *is in* $cl(\varphi_0)$*, then so are* $\langle\!\langle A^{b_1} \rangle\!\rangle \bigcirc \langle\!\langle A^{b_2} \rangle\!\rangle \varphi \mathcal{U} \psi$ *for all* $b_1 +^\infty b_2 = b$ *and also* $\langle\!\langle A^{0_b} \rangle\!\rangle \bigcirc \langle\!\langle A^b \rangle\!\rangle \varphi \mathcal{U} \psi$

- *If* φ *is in* $cl(\varphi_0)$*, then so is* $\sim\varphi$

- $cl(\varphi_0)$ *is also closed under finite positive boolean operators* ($\vee$ *and* $\wedge$) *up to tautology equivalence.*

Obviously, $cl(\varphi_0)$ is finite as its cardinality is bounded by $2^{2^{m^r \times |\varphi|}}$ where m is the maximal bound of any resource appearing in φ_0 and r is the number of resources. We denote $cl(\varphi_0)_\bigcirc$ to be the set of all formulas of form $\langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi$ or $\neg \langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi$ in $cl(\varphi_0)$.

Then, the following three lemmas describe each step of the construction of the tree model. We only provide the proof of the last lemma.

Lemma 13. *Let* $\Phi = \{\langle\!\langle A_1^{b_1} \rangle\!\rangle \bigcirc \varphi_1, \ldots, \langle\!\langle A_k^{b_k} \rangle\!\rangle \bigcirc \varphi_k, \neg \langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi\}$ *be a consistent set of formulas where:*

- *All* A_i *are both non-empty and pair-wise disjoint*

- $\bigcup_i A_i \subseteq A$

- $\sum_i b_i \leq b$

We have $\Psi = \{\varphi_1, \ldots, \varphi_k, \sim\varphi\}$ *is also consistent.*

Proof. Let $A' = \bigcup_i A_i$, $b' = \sum_i b_i$ and $\varphi' = \bigwedge_i \varphi_i$.

By axiom (S), we have $\vdash \bigwedge_i \langle\!\langle A_i^{b_i} \rangle\!\rangle \bigcirc \varphi_i \to \langle\!\langle A'^{b'} \rangle\!\rangle \bigcirc \varphi'$. As $A' \subseteq A$ and $b' \leq b$, we have that

$$\vdash \bigwedge_i \langle\!\langle A_i^{b_i} \rangle\!\rangle \bigcirc \varphi_i \to \langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi' \tag{5.1}$$

Now, let us assume that Ψ is inconsistent, that is $\vdash \varphi' \to \varphi$. Then, applying $\langle\!\langle A^b \rangle\!\rangle \bigcirc$-monotonicity, we have that $\vdash \langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi' \to \langle\!\langle A^b \rangle\!\rangle \bigcirc (\varphi)$.

By (5.1), we have that

$$\vdash \bigwedge_i \langle\!\langle A_i^{b_i} \rangle\!\rangle \bigcirc \varphi_i \to \langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi$$

Hence $\Phi \cup \{\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi\}$ is consistent, which is a contradiction $\square$

Similarly, we can prove the following lemma.

Lemma 14. *Let* $\Phi = \{\langle\langle A_1^{b_1}\rangle\rangle\bigcirc\varphi_1, \dots, \langle\langle A_k^{b_k}\rangle\rangle\bigcirc\varphi_k, \langle\langle \varnothing^{e_1}\rangle\rangle\bigcirc\chi_1, \dots, \langle\langle \varnothing^{e_m}\rangle\rangle\bigcirc\chi_m\}$ *be a consistent set of formulas where:*

- *All* A_i *are both non-empty and pair-wise disjoint*
- $\sum_i b_i \leq e_j$ *for all* j

We have $\Psi = \{\varphi_1, \dots, \varphi_k, \chi_1, \dots, \chi_m\}$ *is also consistent.*

We now use the above lemma to construct a simple tree which is locally consistent for a consistent set of formulas.

Definition 27. *A tree* (T, V, C) *is locally consistent if and only if for any interior node* $t \in T$*:*

1. *If* $\langle\langle A^b\rangle\rangle\bigcirc\varphi$ *in* $V(t)$*, then there is a move* σ_A *such that* $C(t, \sigma_A) \leq b$ *and for any* $c \in out(t, \sigma_A)$ *we have* $\varphi \in V(c)$
2. *If* $\neg\langle\langle A^b\rangle\rangle\bigcirc\varphi$ *in* $V(t)$*, then for any move* σ_A *with* $C(t, \sigma_A) \leq b$*, there exists* $c \in out(t, \sigma_A)$ *where* $\sim\varphi \in V(c)$

Lemma 15. *Let* Φ *be a finite consistent set of formulas,* $\Phi_\bigcirc$ *the subset of* Φ *which contains all formulas of the forms* $\langle\langle A^b\rangle\rangle\bigcirc\varphi$ *or their negations from* Φ *and* k *some number where* $|\Phi_\bigcirc| < k$*, there is a simple* k^n*-costed-tree* (T, V, C) *which is locally consistent such that* $V(\epsilon) = \Phi$.

Proof. Firstly, we have $\neg\langle\langle N^b\rangle\rangle\bigcirc\varphi$ and $\neg\langle\langle \varnothing^b\rangle\rangle\bigcirc\varphi$ are equivalent to $\langle\langle \varnothing^b\rangle\rangle\bigcirc \sim\varphi$ and $\langle\langle N^b\rangle\rangle\bigcirc \sim\varphi$, respectively. Therefore, we only consider the case when $\Phi_\bigcirc$ does not contain formulas of the form $\neg\langle\langle N^b\rangle\rangle\bigcirc\varphi$ and $\neg\langle\langle \varnothing^b\rangle\rangle\bigcirc\varphi$.

Assume that

$$\begin{array}{rcl}\Phi_\bigcirc & = & \{\langle\langle A_1^{b_1}\rangle\rangle\bigcirc\varphi_1, \dots, \langle\langle A_m^{b_m}\rangle\rangle\bigcirc\varphi_m\}\cup \\ & & \{\neg\langle\langle B_1^{d_1}\rangle\rangle\bigcirc\psi_1, \dots, \neg\langle\langle B_l^{d_l}\rangle\rangle\bigcirc\psi_l\}\cup \\ & & \{\langle\langle \varnothing^{e_1}\rangle\rangle\bigcirc\chi_1, \dots, \langle\langle \varnothing^{e_h}\rangle\rangle\bigcirc\chi_h\}\end{array}$$

where all A_i's are non-empty, all B_i's are both non-empty and not equal to the grand coalition N. We define a bound $\max \in \mathbb{B}$ where each component of $\max$ is the maximal bound other than the infinity infinity symbol of the corresponding resource appearing in $\Phi_\bigcirc$. In the case that there is no maximal bound, then the component of $\max$ is set to 0. For example, assume that $|r| = 2$ and $\Phi_\bigcirc = \{\neg\langle\langle\{1,2\}^{(2,2)}\rangle\rangle\bigcirc p, \langle\langle\{1\}^{(3,\infty)}\rangle\rangle\bigcirc p\}$, then $\max = (3, 2)$; in another case, if $\Phi_\bigcirc = \{\langle\langle\{1\}^{(3,\infty)}\rangle\rangle\bigcirc p\}$ then $\max = (3, 0)$.

Then, we define a function *deinf* : $\mathbb{B}^{\infty} \to \mathbb{B}$ which removes infinity from a bound as follows: *deinf*$(b) = b'$ where for all $i = 1, \ldots, |r|$

$$b'_i = \begin{cases} b_i \text{ if } b_i \neq \infty \\ \max_i + 1 \text{ otherwise} \end{cases}$$

Let e be a bound of resources such that $e >$ *deinf*(e_i) for all $i \in \{1, \ldots, h\}$.

We construct a tree with a root labelled by Φ and k^n children, each is denoted by $c = encode(a_1, \ldots, a_n)$ where $a_i \in \{1, \ldots, k\}$. Intuitively, we allow each agent i to perform k different actions and the special action k for each agent will be considered as the costless idle-action. We shall denote $c(i) = a_i$ for the action performed by agent i with the corresponding outcome c. In the following, we define the labelling function $V(c)$ for each node c and the cost function $C(\epsilon, i, a)$ for each agent i and action $a \in \{1, \ldots, k\}$.

For each $\langle\langle A_p^{b_p} \rangle\rangle \bigcirc \varphi_p \in \Phi_{\bigcirc}$ where $A_p \neq \varnothing$, φ_p is added to $V(c)$ whenever $c(i) = p$ for all $i \in A_p$. Let $\min_{A_p}$ be the smallest number in A_p, we assign the cost of action p performed by $\min_{A_p}$ to be b_p, i.e. $C(\epsilon, \min_{A_p}, p) =$ *deinf*(b_p). For actions of other agents i in A_p, we assign $C(\epsilon, i, p) = \bar{0}$.

After considering all $\langle\langle A_p^{b_p} \rangle\rangle \bigcirc \varphi_p \in \Phi_{\bigcirc}$, for all other unassigned-cost actions, i.e. actions $a > m$ but $a < k$ for all agents, we simply set their costs to be e. The action k performed by all agents is defined to associate with the cost $\bar{0}$. We denote $C(c) = \sum_{i \in N} C(\epsilon, i, c(i))$. Then, for each $\langle\langle \varnothing_p^{e_p} \rangle\rangle \bigcirc \chi_p \in \Phi_{\bigcirc}$, χ_p is added to $V(c)$ whenever $C(c) \leq e_p$.

Finally, we will add at most one formula from the negation formulas of $\Phi_{\bigcirc}$ to $V(c)$. We denote $C(c, A) = \sum_{i \in A} C(\epsilon, i, c(i))$. For each c, let $\Phi_{\bigcirc}^{-}(c) = \{\neg\langle\langle B^d \rangle\rangle \bigcirc \psi \in \Phi_{\bigcirc} \mid C(c, B) \leq d\} = \{\neg\langle\langle B_{i_1}^{d_{i_1}} \rangle\rangle \bigcirc \psi_{i_1}, \ldots, \neg\langle\langle B_{i_{l_c}}^{d_{l_c}} \rangle\rangle \bigcirc \psi_{l_c}\}$ where $i_1 < i_2 < \ldots < i_{l_c}$. Let $I = \{i \mid m < c(i) \leq m + l_c\}$ and $j = \sum_{i \in I} (c(i) - 1 - m) \mod l_c + 1$. Consider $\neg\langle\langle B_{i_j}^{d_{i_j}} \rangle\rangle \bigcirc \psi_{i_j}$: if $N \setminus B_{i_j} \subseteq I$, then $\sim\psi_{i_j}$ is added into $V(c)$.

We now need to show that our simple tree is locally consistent. In the first step, we show that all labels are consistent. It is obvious that $V(\epsilon) = \Phi$ is consistent.

Let us firstly consider every child c of the root where $\sim\psi_q \in V(c)$ from some negation formula in $\Phi_{\bigcirc}$. This will imply that there will be no $\chi \in V(c)$ from the formulas of the form $\langle\langle \varnothing^b \rangle\rangle \bigcirc \chi$ in $\Phi_{\bigcirc}$. The reason is that because some $\sim\psi_q \in V(c)$, there must be some agent performing an action $a \in \{m+1, \ldots, m+l_c\}$ as otherwise $I = \varnothing$ and the condition $N \setminus B_{i_j} \subseteq I$ fails since $B_{i_j} \neq N$. We know that the cost of this action is e, then $C(c) \geq e$, therefore, no χ will be added into $V(c)$.

When there is no $\varphi \in V(c)$ from the formulas of the form $\langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi$ in $\Phi_{\bigcirc}$, the proof is trivial as there is only one $\sim\psi_q \in V(c)$. If there are some $\varphi_p \in V(c)$ where $\langle\!\langle A_p^{b_p} \rangle\!\rangle \bigcirc \varphi_p \in \Phi_{\bigcirc}$, then for each p, $c(i) = p < m$ for all $i \in A_p$. Hence, all A_p are pair-wise disjoint. Moreover, we have that $N \setminus B_q \subseteq I$ where $I = \{i \in N \mid m < c(i) \leq m + l_c\}$. Then, $B_q \supseteq N \setminus I \supseteq \{i \in N \mid c(i) \leq m\}$, which implies that $\bigcup_{\varphi_p \in V(c)} A_p \subseteq B_q$. This simply shows that the set of $\langle\!\langle A_p^{b_p} \rangle\!\rangle \bigcirc \varphi_p \in \Phi_{\bigcirc}$ where $\varphi_p \in V(c)$ and $\sim\langle\!\langle B_q^{b_q} \rangle\!\rangle \bigcirc \psi_q$ satisfies the conditions of Lemma 13. Therefore, $V(c)$ is consistent.

Now, we consider every child c of the root where there is no $\sim\psi \in V(c)$ from some negation formula in $\Phi_{\bigcirc}$.

When there is no $\varphi \in V(c)$ from the formulas of the form $\langle\!\langle A^b \rangle\!\rangle \bigcirc \varphi$ in $\Phi_{\bigcirc}$, the proof is trivial as there are only some $\chi_q \in V(c)$. If there are some $\varphi_p \in V(c)$ where $\langle\!\langle A_p^{b_p} \rangle\!\rangle \bigcirc \varphi_p \in \Phi_{\bigcirc}$ and $A_p \neq \emptyset$, then for each p, $c(i) = p < m$ for all $i \in A_p$. Hence, all A_p are pair-wise disjoint. For any $\chi_q \in V(c)$ by some $\langle\!\langle \emptyset^{e_q} \rangle\!\rangle \bigcirc \chi_q \in \Phi_{\bigcirc}$, we have that $e_q \geq C(c) \geq \sum_p b_p$. This simply shows that the set of $\langle\!\langle A_p^{b_p} \rangle\!\rangle \bigcirc \varphi_p \in \Phi_{\bigcirc}$ where $\varphi_p \in V(c)$ and $\langle\!\langle \emptyset_q^{e_q} \rangle\!\rangle \bigcirc \chi_q$ satisfies the conditions of Lemma 14. Therefore, $V(c)$ is consistent.

Let us now check the conditions of local consistency on the newly built tree.

For $\langle\!\langle A_p^{b_p} \rangle\!\rangle \bigcirc \varphi_p \in \Phi_{\bigcirc}$, it is straightforward that the move σ_{A_p} where all agents in A_p performs action $p \leq m$ which cost no more than b_p and for any $c \in out(\sigma_{A_p})$, $\varphi_p \in V(c)$.

For $\neg\langle\!\langle B_p^{d_p} \rangle\!\rangle \bigcirc \psi_p \in \Phi_{\bigcirc}$ and σ being an arbitrary move of agents in B_p of which cost is at most equal to d_p, we will point out an output $c \in out(\epsilon, \sigma)$ where $\sim\psi \in V(c)$ and the actions of agents out of B_p are within $m+1$ and $m+l$, which always cost e amount of resources. Even though we do not know the exact actions of agents out of B_p, the costs of those unspecified actions are known to be e. Hence, we can determine the set $\Phi_{\bigcirc}^{-}(c) = \{\neg\langle\!\langle B_{i_1}^{d_{i_1}} \rangle\!\rangle \bigcirc \psi_{i_1}, \ldots, \neg\langle\!\langle B_{i_{l_c}}^{d_{i_{l_c}}} \rangle\!\rangle \bigcirc \psi_{i_{l_c}}\}$ as well as l_c. It is obvious that $\neg\langle\!\langle B_p^{d_p} \rangle\!\rangle \bigcirc \psi_p \in \Phi_{\bigcirc}^{-}(c)$, then $p = i_r$ for some $1 \leq r \leq l_c$. Let σ_i be the action performed by agent i in B_p, we define $c(i) = \sigma_i$ for all $i \in B_p$. Let $I' = \{i \in B_q \mid m < c(i) \leq m + l_c\}$ and $j' = \sum_{i \in I'}(c(i) - 1 - m)) \mod l_c$. We select an arbitrary $i' \notin B_p$ and set $c(i') = m + (r - 1 - j') \mod l_c + 1$. For all other $i \notin B_p$, let $c(i) = m + 1$. Then, we have $I = \{i \mid m < c(i) \leq m + l_c\} = (N \setminus B_p) \cup I'$. Therefore, $\sum_{i \in I}(c(i) - 1 - m) \mod l_c + 1 = \sum_{i \in I' \cup \{i'\}}(c(i) - 1 - m) \mod l_c + 1 = (j' + c(i') - 1 - m) \mod l_c + 1 = (r - 1) \mod l_c + 1 = r$, and $N \setminus B_p \subseteq I$ because $I = (N \setminus B_p) \cup I'$. By choosing such outcome c, according to the construction of the simple tree model, we must have that $\sim\psi_p \in V(c)$. □

Let us consider an example of building such a locally consistent tree. Consider a system of 2 agents, i.e. $N = \{1, 2\}$, 1 resource, i.e. $|r| = 1$, and the following set $\Phi_{\bigcirc}$ of RB-ATL formulas.

$$\Phi_{\bigcirc} = \{\langle\langle 1^1 \rangle\rangle \bigcirc p, \langle\langle 2^{\infty} \rangle\rangle \bigcirc (p \to q), \neg\langle\langle 1^2 \rangle\rangle \bigcirc q, \neg\langle\langle 2^2 \rangle\rangle \bigcirc p, \langle\langle \varnothing^2 \rangle\rangle \bigcirc (\neg q)\}$$

It is easy to see that $\max = 2$ and we can pick $e = 3$. We now construct a simple tree which is locally consistent and the root is labelled by $\Phi_{\bigcirc}$. As $|\Phi_{\bigcirc}| = 5$, let us consider the number of actions for each agent $k = 6$. Then, the set of outcomes is $O = \{(i, j) \mid 1 \le i, j \le 6\}$.

Consider the formula $\langle\langle 1^1 \rangle\rangle \bigcirc p \in \Phi_{\bigcirc}$, we add to the label of every $V((1, j))$ the formula p, for any $1 \le j \le 6$. The cost of action 1 of agent 1 is 1.

Consider the formula $\langle\langle 2^{\infty} \rangle\rangle \bigcirc (p \to q) \in \Phi_{\bigcirc}$, we add to the label of every $V((i, 2))$ the formula $p \to q$, for any $1 \le i \le 6$. The cost of action 2 of agent 2 is $\max + 1 = 3$.

As we mean the action 6 for both agents to be the idle action, we simply assign the cost 0 for 6 of both agents. Then we assign the cost $e = 3$ for all cost-unassigned actions of both agents. After this step, we add $\neg q$ to every outcome (i, j) of which the total cost of i and j is no more than 2.

We have the assignment of labels $V((i, j))$ for every $1 \le i, j \le 6$ so far as in Figure 5.2 where each column (row) corresponds to an action of agent 1 (2) together with its cost.

AC	1^1	2^3	3^3	4^3	5^3	6^0
1^3	$\{p\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
2^3	$\{p, p \to q\}$	$\{p \to q\}$	$\{p \to q\}$	$\{p \to q\}$	$\{p \to q\}$	$\{p \to q\}$
3^3	$\{p\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
4^3	$\{p\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
5^3	$\{p\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
6^0	$\{p, \neg q\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\neg q\}$

FIGURE 5.2: The assignment of labels $V((i, j))$ for every $1 \le i, j \le 6$.

Let us consider the negation formulas in $\Phi_{\bigcirc}$. We take each outcome into account to decide whether one of the sub-formulas of the negation formulas in $\Phi_{\bigcirc}$ is included in the label of the outcome.

We consider the outcome $c = (1^1, 1^3)$, then $\Phi_{\bigcirc}(c) = \{\neg\langle\langle 1^2 \rangle\rangle \bigcirc q\}$. Then $l_c = 1$, $I = \{i \mid 2 < c(i) \le 2 + 1\} = \varnothing$. Therefore, as $N \setminus \{1\} \nsubseteq I$, $\neg q$ is not included in $V(c)$.

We consider the outcome $c = (1^1, 3^3)$, then $\Phi_{\bigcirc}(c) = \{\neg\langle\langle 1^2 \rangle\rangle \bigcirc q\}$. Then $l_c = 1$, $I = \{i \mid 2 < c(i) \le 2 + 1\} = \{2\}$. Therefore, as $N \setminus \{1\} \subseteq I$, $\neg q$ is included in $V(c)$.

We consider the outcome $c = (3^3, 6^0)$, then $\Phi_{\bigcirc}(c) = \{\neg\langle\langle 2^2\rangle\rangle\bigcirc p\}$. Then $l_c = 1$, $I = \{i \mid 2 < c(i) \leq 2+1\} = \{1\}$. Therefore, as $N \setminus \{2\} \subseteq I$, $\neg p$ is included in $V(c)$.

We can apply similar argument, and obtain the final assignment of labels as shown in Figure 5.3.

AC	1^1	2^3	3^3	4^3	5^3	6^0
1^3	$\{p\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
2^3	$\{p, p \to q\}$	$\{p \to q\}$	$\{p \to q\}$	$\{p \to q\}$	$\{p \to q\}$	$\{p \to q\}$
3^3	$\{p, \neg q\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\neg q\}$
4^3	$\{p\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
5^3	$\{p\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$	$\{\}$
6^0	$\{p, \neg q\}$	$\{\}$	$\{\neg p\}$	$\{\}$	$\{\}$	$\{\neg q\}$

FIGURE 5.3: The final assignment of labels $V((i,j))$ for every $1 \leq i, j \leq 6$.

In the following, Γ is the finite set of all maximal consistent sets of formulas from $cl(\varphi_0)$. As $cl(\varphi_0)$ is finite, Γ is also finite. We extend the construction for satisfying eventuality formulas which are in forms of $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ and $\langle\langle A^b\rangle\rangle\Box\varphi$ by the following lemma. We also omit the proof.

Firstly, we say that a formula $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ ($\neg\langle\langle A^b\rangle\rangle\Box\varphi$) is realised from a node t of a Γ-labelled tree (T, V, C) if there exists a strategy (co-strategy) F_A such that for all $\lambda \in out(t, F_A, b)$ ($\lambda \in out(t, F_A^c, b)$), there is some i such that $\psi \in V(\lambda[i])$ and $\varphi \in V(\lambda[j])$ for all $j \in \{0, i-1\}$ ($\sim\varphi \in V(\lambda[i])$).

Definition 28. *A formula $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ is realised from a node t of a Γ-labelled tree (T, V, C) if there exists a strategy F_A such that for all $\lambda \in out(t, F_A)$, there is some i such that $cost(\lambda[0,i], F_A) \leq b$, $\psi \in V(\lambda[i])$ and $\varphi \in V(\lambda[j])$ for all $j \in \{0, i-1\}$.*

Lemma 16. *For any subset $Y \subseteq \Gamma$, there is a formula $\chi_Y \in cl(\varphi_0)$, called the characterised formula of Y, such that for every $y \in \Gamma$, $\chi_Y \in y$ iff $y \in Y$.*

Proof. The proof is a repetition of that of a similar lemma in [Goranko & van Drimmelen, 2006].

For any $y \in Y$, we define $\chi_{\{y\}} = \wedge y = \bigwedge\{\varphi \mid \varphi \in y\}$.

Note that $\chi_{\{y\}} \in cl(\varphi_0)$ as $cl(\varphi_0)$ is closed under finite conjunctions. Then, $\chi_{\{y\}}$ is the characterised formula of $\{y\}$. For any other $y' \in Y$, as both y and y' are maximal consistent set of formulas from $cl(\varphi_0)$, there is a formula $\theta \in cl(\varphi_0)$ such that $\theta \in y$ but $\neg\theta \in y'$. Then, θ is a conjunction of $\chi_{\{y\}}$ and then $\chi_{\{y\}} \wedge \neg\theta$ is inconsistent. Hence, $\chi_{\{y\}} \notin y'$ as y' is consistent.

For any $Y \in \Gamma$, we define $\chi_Y = \bigvee\{\chi_{\{y\}} \mid y \in Y\}$. Then, for any $y \in Y$, we have $\vdash \chi_{\{y\}} \to \chi_Y$, that is $\chi_Y \in y$. Conversely, for any $y' \notin Y$, $\chi_{\{y\}} \notin y'$ for any $y \in Y$, hence $\chi_Y \notin y'$. □

Lemma 17. *For each formula $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ and $x \in \Gamma$, there is finite Γ-labelled k^n-costed-tree (T, V, C) where:*

- $k = |\Psi_{\bigcirc}| + 1$
- (T, V, C) *is locally consistent*
- $V(\epsilon) = x$
- *If $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \in x$ then (T, V, C) realises $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ from ϵ*

Proof. Most of the proof is based on that of the similar lemma in [Goranko & van Drimmelen, 2006].

For the sake of readability, we refer to finite Γ-labelled k^n-costed-trees by finite trees.

Consider a specific formula $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$. Let $Z \subseteq \Gamma$ be a set of maximal consistent sets of formulas where for every $x \in Z$, there is a finite tree obeying the conditions of the lemma. Hence, we prove the lemma by showing that $Z = \Gamma$. If $x \in \Gamma$ does not contain $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$, we just need to construct a tree (T, V, C) which has only a single root with label $V(\epsilon) = x$. Obviously, $x \in Z$.

Let us now consider the more interesting case, where $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \in x$. As we want to show that $x \in Z$, it suffixes to prove that $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \to \chi_Z$ is a theorem. This is because $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \in x$, hence $\chi_Z \in x$, i.e. $x \in Z$ by Lemma 16. However, to show $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \to \chi_Z$ is a theorem, we only need to show

$$(\psi \vee (\varphi \wedge (\langle\langle A^b\rangle\rangle\bigcirc\varphi\mathcal{U}\psi \vee (\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\chi_Z)))) \to \chi_Z \tag{5.2}$$

is also a theorem. If it is the case, then by $\langle\langle A^b\rangle\rangle\mathcal{U}$-induction, we have that $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \to \chi_Z$ is a theorem as well.

We prove (5.2) being an theorem by showing that it belongs to every maximal consistent set q (not only formulas from $cl(\varphi_0)$). Note that $\chi_Z \in q$ iff $q \cap cl(\varphi_0) \in Z$.

Let us consider the first two easy cases, when either $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \notin q$ or $(\psi \vee (\varphi \wedge (\langle\langle A^b\rangle\rangle\bigcirc\varphi\mathcal{U}\psi \vee (\langle\langle A^0\rangle\rangle\bigcirc\chi_Z)))) \notin q$.

If $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \notin q$, it is straightforward for us to construct a tree containing only a single root. Hence, $q \cap cl(\varphi_0) \in Z$. Then, $\chi_Z \in q$ and we must have $(5.2) \in q$. If $(\psi \vee (\varphi \wedge (\langle\langle A^b\rangle\rangle\bigcirc\varphi\mathcal{U}\psi \vee (\langle\langle A^0\rangle\rangle\bigcirc\chi_Z)))) \notin q$, it is even easier for us as we directly have $(5.2) \in q$.

In the last case, consider that b is the projection of a bound $d \in \mathbb{B}$ with b, the proof proceeds by induction on the bound $d \in \mathbb{B}$. Let us consider the base case where $d = \bar{0}$. Assume that both $\langle\langle A^{\bar{0}_b} \rangle\rangle \varphi \mathcal{U} \psi$ and $(\psi \vee (\varphi \wedge \langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc \chi_Z)) \in q$. We show $(5.2) \in q$ by proving that $q \cap cl(\varphi_0) \in Z$. As $(\psi \vee (\varphi \wedge \langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc \chi_Z)) \in q$, there are two cases to consider:

(a) $\psi \in q$, then we just need to construct a tree (T, V, C) with only a single root and $V(\epsilon) = q \cap cl(\varphi_0)$. It is straightforward that $\langle\langle A^{\bar{0}_b} \rangle\rangle \varphi \mathcal{U} \psi$ is realised at ϵ as $\psi \in V(\epsilon)$.

(b) $\varphi \wedge \langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc \chi_Z \in q$, we construct a tree with a root with the label $q \cap cl(\varphi_0)$ and k^n children defined as follows.

Let Φ' be the set containing all formulas of the form $\langle\langle A^b \rangle\rangle \bigcirc \phi$ or $\neg \langle\langle A^b \rangle\rangle \bigcirc \phi$ from $q \cap cl(\varphi_0)$ and also the formula $\langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc \chi_Z$. Then, $|\Phi'| \leq k + 1$, by Lemma 15, we have a locally consistent tree (T', V', C') of branching degree k^n with $V'(\epsilon) = \Phi'$.

For each child $c < k^n$, we assign $V(c)$ be an arbitrary set from Γ such that $V(c) \supseteq V'(c)$. This preserves the local consistency at ϵ according to Lemma 15.

For every child $c < k^n$ such that $\chi_Z \in V(c)$, we have that $V(c) \in Z$. This means there is a locally consistent tree (T_c, V_c, C_c) satisfying conditions of the lemma. Then, we replace c by (T_c, V_c, C_c). The result tree (T, V, C) is also locally consistent and of branching degree k^n.

We now show that (T, V, C) realises $\langle\langle A^{\bar{0}_b} \rangle\rangle \varphi \mathcal{U} \psi$ at ϵ. Let σ be the move generated because of $\langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc \chi_Z \in V'(\epsilon)$ according to Lemma 15. Then, for every $c \in out(\sigma)$, we have that $\chi_Z \in V(c)$, hence $V(c) \in Z$. This is also means that there is a strategy $F_{A,c}$ which realises $\langle\langle A^0 \rangle\rangle \varphi \mathcal{U} \psi$ from c. Let us consider a strategy F_A such that $F_A(\epsilon) = \sigma$ and $F_A(c\lambda) = F_{A,c}(\lambda)$. It is straightforward that F_A realises $\langle\langle A^0 \rangle\rangle \varphi \mathcal{U} \psi$ from the root ϵ.

In the induction step, where $b > \bar{0}$, the proof proceeds in the similar manner. Assume that both $\langle\langle A^{\bar{0}_b} \rangle\rangle \varphi \mathcal{U} \psi$ and $(\psi \vee (\varphi \wedge (\langle\langle A^{d_b} \rangle\rangle \bigcirc \varphi \mathcal{U} \psi \vee (\langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc \chi_Z)))) \in q$. Similar to the base case, there are also three cases to consider:

(a) $\psi \in q$, the proof is the repetition of that for the base case.

(b) φ and $\langle\langle A^{\bar{0}_b} \rangle\rangle \bigcirc \chi_Z \in q$, the proof is the repetition of that for the base case.

(c) φ and $\langle\langle A^{b_1} \rangle\rangle \bigcirc \langle\langle A^{b_2} \rangle\rangle \varphi \mathcal{U} \psi \in q$ for some $b_1 +^{\infty} b_2 = d_b$ where $b_2 < d_b$, we construct a tree with a root with the label is $q \cap cl(\varphi_0)$ and k^n children defined as follows.

Let Φ' be the set containing all formulas of the form $\langle\langle A^b \rangle\rangle \bigcirc \phi$ or $\neg\langle\langle A^b \rangle\rangle \bigcirc \phi$ from $q \cap cl(\varphi_0)$. Notice that one of them is $\langle\langle A^{b_1} \rangle\rangle \bigcirc \langle\langle A^{b_2} \rangle\rangle \varphi \mathcal{U} \psi$. It is obvious that $|\Phi'| < k + 1$, by Lemma 15, we have a locally consistent tree (T', V', C') of branching degree k^n with $V'(\epsilon) = \Phi'$.

For each child $c < k^n$, we assign $V(c)$ be an arbitrary set from Γ such that $V(c) \supseteq V'(c)$. This preserves the local consistency at ϵ according to Lemma 15.

For every child $c < k^n$ such that $\langle\langle A^{b_2} \rangle\rangle \varphi \mathcal{U} \psi \in V(c)$, as $b_2 < d_b$, by the induction hypothesis, there is a locally consistent tree (T_c, V_c, C_c) realising $\langle\langle A^{b_2} \rangle\rangle \varphi \mathcal{U} \psi$ that its root. Then, we replace c by (T_c, V_c, C_c). The result tree (T, V, C) is also locally consistent and of branching degree k^n.

We now show that (T, V, C) realises $\langle\langle A^{d_b} \rangle\rangle \varphi \mathcal{U} \psi$ at ϵ. Let σ be the move generated because of $\langle\langle A^{b_1} \rangle\rangle \bigcirc \langle\langle A^{b_2} \rangle\rangle \varphi \mathcal{U} \psi \in V'(\epsilon)$ according to Lemma 15, hence $cost(\sigma) \leq b_1$. Then, for every $c \in out(\sigma)$, we have that $\langle\langle A^{b_2} \rangle\rangle \varphi \mathcal{U} \psi \in V(c)$. By the induction hypothesis, $\langle\langle A^{b_2} \rangle\rangle \varphi \mathcal{U} \psi$ is realised at c. This is also means that there is a strategy $F_{A,c}$ which realises $\langle\langle A^{b_2} \rangle\rangle \varphi \mathcal{U} \psi$ from c which spends at most b_2 amount of resources. Let us consider a strategy F_A such that $F_A(\epsilon) = \sigma$ and $F_A(c\lambda) = F_{A,c}(\lambda)$. It is straightforward that F_A realises $\langle\langle A^0 \rangle\rangle \varphi \mathcal{U} \psi$ from the root ϵ which costs at most $b_1 + b_2 = d_b$.

$\square$

In the following, we extend the notions of realisation to other types of formulas.

Definition 29. *A formula $\neg\langle\langle A^b \rangle\rangle \Box \varphi$ is realised from a node t of a tree (T, V, C) over Γ if there exists a co-strategy F^c_A such that for all $\lambda \in out(t, F^c_A, b)$, there is some i such that $\neg\varphi \in V(\lambda[i])$.*

Definition 30. *A formula $\langle\langle A^b \rangle\rangle \Box \varphi$ is realised from a node t of a tree (T, V, C) over Γ if there exists a b-strategy F_A such that for all $\lambda \in out(t, F_A)$, $\varphi \in V(\lambda[i])$ for all i.*

Definition 31. *A formula $\neg\langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi$ is realised from a node t of a tree (T, V, C) over Γ if there exists a co-strategy F^c_A such that for all $\lambda \in out(t, F^c_A, b)$, if there is some i such that $\psi \in V(\lambda[i])$, then there is some $j < i$ such that $\neg\varphi \in V(\lambda[j])$.*

Lemma 18. *For each formula $\neg\langle\langle A^b \rangle\rangle \Box \varphi$ and $x \in \Gamma$, there is finite tree (T, V, C) over Γ such that:*

- *(T, V, C) is of fixed branching degree k^n where $k = |\Psi_\bigcirc| + 1$*
- *(T, V, C) is locally consistent*

- $V(\epsilon) = x$
- *If* $\neg\langle\langle A^b\rangle\rangle\Box\varphi \in x$ *then* (T, V, C) *realises* $\neg\langle\langle A^b\rangle\rangle\Box\varphi$ *from* ϵ

Proof. The proof is also done in a similar manner to that of the previous lemma.

Consider a specific formula $\neg\langle\langle A^b\rangle\rangle\Box\varphi$. Let $Z \subseteq \Gamma$ be a set of maximal consistent sets of formulas where for every $x \in Z$, there is a finite tree obeying the conditions of the lemma. Hence, we prove the lemma by showing that $Z = \Gamma$. If $x \in \Gamma$ does not contain $\neg\langle\langle A^b\rangle\rangle\Box\varphi$, we just need to construct a tree (T, V, C) which has only a single root with label $V(\epsilon) = x$. Obviously, $x \in Z$.

Let us now consider the more interesting case, where $\neg\langle\langle A^b\rangle\rangle\Box\varphi \in x$. As we want to show that $x \in Z$, it suffices to prove that $\neg\langle\langle A^b\rangle\rangle\Box\varphi \to \chi_Z$ is a theorem. This is because if $\neg\langle\langle A^b\rangle\rangle\Box\varphi \in x$, then $\chi_Z \in x$, i.e. $x \in Z$. However, to show $\neg\langle\langle A^b\rangle\rangle\Box\varphi \to \chi_Z$ is a theorem, we only need to show

$$(\neg\varphi \vee (\neg\langle\langle A^b\rangle\rangle\bigcirc\Box\varphi \wedge \neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z)) \to \chi_Z \tag{5.3}$$

is also a theorem. If it is the case, then by $\langle\langle A^b\rangle\rangle\Box$-induction, we have that $\neg\langle\langle A^b\rangle\rangle\Box\varphi \to \chi_Z$ is a theorem as well.

By considering b as the projection of a resource bound $d \in \mathbb{B}$ and b, we prove (5.3) being a theorem by showing inductively on d that (5.3) belongs to every maximal consistent set q (not only formulas from $cl(\varphi_0)$). Note that $\chi_Z \in q$ iff $q \cap cl(\varphi_0) \in Z$.

The base case:

Assume that $d = \bar{0}$, let us consider the first two easy cases, when either $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi \notin q$ or $\neg\varphi \vee (\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z) \notin q$. In the first case, we just need to consider a trivial tree containing only a root. In the later case, it is straightforward that $(5.3) \in q$

Let us now assume that both $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi$ and $\neg\varphi \vee (\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z) \in q$.

- If $\neg\varphi \in q$, we construct a tree containing only a root ϵ with the label $V(\epsilon) = q \cap cl(\varphi_0)$. As $\neg\varphi \in q$, it is straightforward that $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi$ is realised at the root ϵ.
- If $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z \in q$, let us construct a tree (T, V, C) as follows. Let Ψ' be the set of all formulas of form $\langle\langle A^b\rangle\rangle\bigcirc\varphi$ or $\neg\langle\langle A^b\rangle\rangle\bigcirc\varphi$ from $q \cup cl(\varphi_0)$ and also the formula $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z$. We have that $|\Psi'| \leq k + 1$ where $k = |\Psi_{\bigcirc}| + 1$. By Lemma 15, there is a locally consistent simple tree (T', V', C') of branching degree k^n such that $V'(\epsilon) = \Psi'$. Moreover, because $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z \in \Psi'$, for every $\sigma \in \Delta_A$ where $C(\epsilon, \sigma) \leq \bar{0}_b$, there is at least a child $c_\sigma \in out(\epsilon, \sigma)$ such that $\neg\neg\chi_Z = \chi_Z \in V(c_\sigma)$. Now, let us define k^n children for the root of

(T, V, C) of which each successor $c < k^n$ is labelled such that $V(c)$ is some set in Γ which contains $V'(c)$. This still maintains the local consistency of (T, V, C). For every child c, if $\chi_Z \in V(c)$, then $V(c) \in Z$. Hence, there is a tree (T_c, V_c, C_c) satisfying the conditions of the lemma. Replace the child c in (T, V, C) with the tree (T_c, V_c, C_c).

We also need to show that $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi$ is realised from the root of (T, V, C). We define a co-strategy F^c_A for A with, initially, $F^c_A(\epsilon) = \sigma^c$ where the co-move σ^c is defined so that for every $\sigma \in \Delta_A$ such that $c(c, \sigma) \leq \bar{0}_b$, $\sigma^c(\sigma) = c_\sigma$. Notice that according to the construction in Lemma 15, we have that $\chi_Z \in V(c_\sigma)$. Then, for every $c \in out^c(\epsilon, \sigma^c, \bar{0}_b)$, we have that $\chi_Z \in V(c)$. As (T_c, V_c, C_c) satisfies the conditions of the lemma, there is a co-strategy F'^c_A in (T_c, V_c, C_c) realising $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi$ at c. We define $F^c_A(c\lambda) = F'^c_A(\lambda)$ for all $\lambda \in Q^*$. It follows from this construction of F^c_A that $\neg\langle\langle A^{0}\rangle\rangle\Box\varphi$ is realised in (T, V, C) from the root ϵ.

The induction step

Assume that $d > \bar{0}$, let us consider the first two easy cases, we repeat the argument as for the base case, when either $\neg\langle\langle A^{b}\rangle\rangle\Box\varphi \notin q$ or $\neg\varphi \vee (\neg\langle\langle A^{d_b}\rangle\rangle\bigcirc\Box\varphi \wedge \neg\langle\langle A^{d_b}\rangle\rangle\bigcirc\neg\chi_Z) \notin q$. In the first case, we just need to consider a trivial tree containing only a root. In the later case, it is straightforward that $(5.3) \in q$

Let us now assume that both $\neg\langle\langle A^{d_b}\rangle\rangle\Box\varphi$ and $\neg\varphi \vee (\neg\langle\langle A^{d_b}\rangle\rangle\bigcirc\Box\varphi \wedge \neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z) \in q$.

- If $\neg\varphi \in q$, we construct a tree containing only a root ϵ with the label $V(\epsilon) = q \cap cl(\varphi_0)$. As $\neg\varphi \in q$, it is straightforward that $\neg\langle\langle A^{d_b}\rangle\rangle\Box\varphi$ is realised at the root ϵ.

- If $(\neg\langle\langle A^{d_b}\rangle\rangle\bigcirc\Box\varphi \wedge \neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z) \in q$, let us construct a tree (T, V, C) as follows. Let Ψ' be the set of all formulas of form $\langle\langle A^{b}\rangle\rangle\bigcirc\varphi$ or $\neg\langle\langle A^{b}\rangle\rangle\bigcirc\varphi$ from $q \cup cl(\varphi_0)$ and notice that it also contains the formula $\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\chi_Z$. We have that $|\Psi'| \leq k + 1$ where $k = |\Psi_\bigcirc| + 1$. By Lemma 15, there is a locally consistent simple tree (T', V', C') of branching degree k^n such that $V'(\epsilon) = \Psi'$. Moreover, because $\neg\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\neg\chi_Z \in \Psi'$, for every $\sigma \in \Delta_A$ where $C(\epsilon, \sigma) \leq \bar{0}_b$, there is at least a child $c_\sigma \in out(\epsilon, \sigma)$ such that $\neg\neg\chi_Z = \chi_Z \in V(c_\sigma)$. Similarly, $\neg\langle\langle A^{b_1}\rangle\rangle\bigcirc\langle\langle A^{b_2}\rangle\rangle\Box\varphi \in \Psi'$ where $b_1 +^\infty b_2 = d_b$ and $b_2 < d_b$, for every $\sigma \in \Delta_A$ where $C(\epsilon, \sigma) \leq \bar{b}_1$, there is at least a child $c_\sigma \in out(\epsilon, \sigma)$ such that $\neg\langle\langle A^{b_2}\rangle\rangle\Box\varphi \in V(c_\sigma)$. Now, let us define k^n children for the root of (T, V, C) of which each successor $c < k^n$ is labelled such that $V(c)$ is some set in Γ which contains $V'(c)$. This still maintains the local consistency of (T, V, C).

  - For every child c, if $\chi_Z \in V(c)$, then $V(c) \in Z$. Hence, there is a tree (T_c, V_c, C_c)

satisfying the conditions of the lemma. Replace the child c in (T, V, C) with the tree (T_c, V_c, C_c).

– For every child c, if $\neg\langle\langle A^{b_2}\rangle\rangle\Box\varphi \in V(c)$ for some $b_2 < d_b$. By the induction hypothesis, there is there is a tree (T_c, V_c, C_c) satisfying the conditions of the lemma. Replace the child c in (T, V, C) with the tree (T_c, V_c, C_c).

We also need to show that $\neg\langle\langle A^b\rangle\rangle\Box\varphi$ is realised from the root of (T, V, C). We define a co-strategy F^c_A for A with, initially, $F^c_A(\epsilon) = \sigma^c$ where the co-move σ^c is defined so that for every move σ where $c(\epsilon, \sigma) \leq d_b$:

– If $c(\epsilon, \sigma) \leq \bar{0}_b$, we assign $\sigma^c(\sigma) = c_\sigma$ where, according to the construction in Lemma 15, we have that $\chi_Z \in V(c_\sigma)$,

– If $c(\epsilon, \sigma) \leq \bar{b}_1$ for some $b_1 +^\infty b_2 = d_b$ with $b_2 < d_b$, we assign $\sigma^c(\sigma) = c_\sigma$ where, according to the construction in Lemma 15, we have that $\neg\langle\langle A^{b_2}\rangle\rangle\Box\varphi \in V(_\sigma)$.

Then, for every $c \in out^c(\epsilon, \sigma^c, d_b)$, we have that either χ_Z or $\neg\langle\langle A^b_2\rangle\rangle\Box\varphi \in V(c)$ for some $b_2 < d_b$. Therefore, as (T_c, V_c, C_c) satisfies the conditions of the lemma, there is a co-strategy F'^c_A in (T_c, V_c, C_c) realising $\neg\langle\langle A^d_b\rangle\rangle\Box\varphi$ or $\neg\langle\langle A^b_2\rangle\rangle\Box\varphi$ at c, respectively for each case. We define $F^c_A(c\lambda) = F'^c_A(\lambda)$ for all $\lambda \in Q^*$. It follows from this construction of F^c_A that $\neg\langle\langle A^b\rangle\rangle\Box\varphi$ is realised in (T, V, C) from the root ϵ.

□

The above lemmas give us the ingredients to finally construct the model for the considered consistent formula φ_0. In more detail, for each consistent set x in Γ and an eventual formula φ of $cl(\varphi_0)$, we have a finite tree $(T_{x,\varphi}, V_{x,\varphi}, C_{x,\varphi})$ which realises φ with the root having label x. Let the eventual formulas in $cl(\varphi_0)$ be listed as $\varphi^e_0, \ldots, \varphi^e_m$. In the following, we have the definition of the final tree.

Definition 32. *The final tree $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$ is constructed inductively as follows.*

- *Initially, select an arbitrary $x \in \Gamma$ such that $\varphi_0 \in x$. As the formula φ_0 is consistent, such a set exists. Let $(T_{x,\varphi^e_0}, V_{x,\varphi^e_0}, C_{x,\varphi^e_0})$ be the initial tree.*

- *Given the tree constructed so far and the last used eventual formula φ^e_i. Then, for every leaf labelled by $y \in \Gamma$ of the currently constructed tree, we replace it with the tree $(T_{y,\varphi^e_j}, V_{y,\varphi^e_j}, C_{y,\varphi^e_j})$ where $j = i + 1$ if $i < m$ or $j = 0$ if otherwise.*

Before proving the truth lemma for the final model, we show the following lemmas which confirm the realisation of eventual formulas.

Lemma 19. *If $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ or $\neg\langle\langle A^b\rangle\rangle\Box\varphi$ is in the label of some node t of $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$, it is realised from t.*

Proof. Let us consider the first case when $\varphi_i^e = \langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \in V(t)$ where t is a node of $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$.

- If t happens to be the root of the sub-tree $(T_{t,\varphi_i^e}, V_{t,\varphi_i^e}, C_{t,\varphi_i^e})$, then the proof is done as φ_i^e is realised within this sub-tree at t, hence also in the final tree.
- If it is not that case, we consider b as the projection of a bound $d \in \mathbb{B}$ and b and define inductively on d a d_b-strategy as follows.

  Base case

  Assume that $d = 0$, since $\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi\mathcal{U}\psi \in V(t)$, as $V(t)$ is a maximally consistent set, we have that $\psi \vee (\varphi \wedge \langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi\mathcal{U}\psi) \in V(t)$.

  - If $\psi \in V(t)$, the proof is done as $\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi\mathcal{U}\psi$ is immediately realised at t.
  - Otherwise, we have $\varphi \wedge \langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi\mathcal{U}\psi \in V(t)$. Then $\varphi \in V(t)$ and by Lemma 15, there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ such that, for all $c \in out(t, \sigma)$, we have $\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi\mathcal{U}\psi \in V(tc)$. Let $F_A(t) = \sigma$. Then, we can continue with the same argument to define the strategy F_A until a node t' in $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$ is reached. Such a node must exist because of the construction of the $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$, we add the sub-tree such that eventual formulas in $cl(\varphi_0)$ are used in a circle order.

  Induction Step

  Assume that $d > 0$, since $\langle\langle A^{d_b}\rangle\rangle\varphi\mathcal{U}\psi \in V(t)$, and $V(t)$ is a maximally consistent set, we have that $\psi \vee (\varphi \wedge (\langle\langle A^{d_b}\rangle\rangle\bigcirc\varphi\mathcal{U}\psi \vee \langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\langle\langle A^{d_b}\rangle\rangle\varphi\mathcal{U}\psi)) \in V(t)$.

  - If $\psi \in V(t)$, the proof is done as $\langle\langle A^{d_b}\rangle\rangle\varphi\mathcal{U}\psi$ is immediately realised at t.
  - If φ and $\langle\langle A^{b_1}\rangle\rangle\bigcirc\langle\langle A^{b_2}\rangle\rangle\varphi\mathcal{U}\psi \in V(t)$ for some $b_1 +^\infty b_2 = d_b$ with $b_2 < d_b$, we have that $\varphi \in V(t)$ and by Lemma 15 and there is a move $\sigma \in \Delta_A$ which costs no more than b_1 such that, for all $c \in out(t, \sigma)$, we have $\langle\langle A^{b_2}\rangle\rangle\varphi\mathcal{U}\psi \in V(tc)$. Let $F_A(t) = \sigma$. As $b_2 < d_b$,

by the induction hypothesis, there is a strategy $F_{A,c}$ which realises $\langle\langle A^{b_2}\rangle\rangle\varphi\mathcal{U}\psi$ from tc. Hence, we just need to define $F_A(tc\lambda) = F_{A,c}(c\lambda)$. This simply gives us a b-strategy which realises $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ from t.

- Otherwise, we have φ and $\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\langle\langle A^{d_b}\rangle\rangle\varphi\mathcal{U}\psi \in V(t)$. Let us repeat the argument in the base case where $\varphi \in V(t)$ and by Lemma 15 and we have that there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ such that, for all $c \in out(t,\sigma)$, we have $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi \in V(tc)$. Let $F_A(t) = \sigma$. Then, we can continue with the same argument to define the strategy F_A until a node t' in $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$ is reached. Such a node must exist because of the construction of the $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$, we add the sub-tree such that eventual formulas in $cl(\varphi_0)$ are used in a circle order.

The proof for the case of $\varphi_i^e = \neg\langle\langle A^b\rangle\rangle\Box\varphi$ is also done similarly as that of the previous case. However, we construct a co-strategy F_A^c instead. If t happens to be the root of the sub-tree $(T_{t,\varphi_i^e}, V_{t,\varphi_i^e}, C_{t,\varphi_i^e})$, then the co-strategy is the one which realised φ_i^e within $(T_{t,\varphi_i^e}, V_{t,\varphi_i^e}, C_{t,\varphi_i^e})$ at t. Otherwise, we proceed the construction along the tree by choosing co-moves, which confirms the condition of local consistency of $\neg\varphi \vee (\neg\langle\langle A^b\rangle\rangle\bigcirc\Box\varphi \wedge \neg\langle\langle A^0\rangle\rangle\bigcirc\langle\langle A^b\rangle\rangle\Box\varphi)$ as done in Lemma 15 until we reach the root of a sub-tree which belongs to the eventual formula $\neg\langle\langle A^b\rangle\rangle\Box\varphi$.

$\square$

Lemma 20. *If $\neg\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ or $\langle\langle A^b\rangle\rangle\Box\varphi$ is in the label of some node t of $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$, it is realised from t.*

Proof. Let us firstly consider the case when $\langle\langle A^b\rangle\rangle\Box\varphi \in V(t)$. By considering b as the projection of a bound $d \in \mathbb{B}$ and b, the proof below is done by induction on d where we construct a d_b-strategy F_A also by induction on the length of the input.

The base case:

Assume that $d = \bar{0}$. As $\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi \in V(t)$ and $V(t)$ is a maximally consistent set, we have that φ and $\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi \in V(t)$. By Lemma 15, there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ such that for all $c \in out(t,\sigma)$, $\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi \in V(tc)$. We define $F_A(t) = \sigma$.

Assume that we already construct F_A for all inputs $\lambda \in Q^*$ of length $i \geq 1$. We now need to define F_A for all inputs λc where $c \in out(\lambda[i-1], F_A(\lambda))$. The resource we spend so far no more than $\bar{0}_b$. As $\langle\langle A^{\bar{0}_b}\rangle\rangle\Box\varphi \in V(\lambda c)$, repeating the above argument, we have that φ and

$\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle \bigcirc \langle\!\langle A^{\bar{0}_b}\rangle\!\rangle \Box\varphi \in V(\lambda c)$. By Lemma 15, there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ such that for all $c' \in out(c,\sigma)$, $\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle \Box\varphi \in V(\lambda cc')$. We define $F_A(\lambda c) = \sigma$.

This construction also shows inductively on the length of input that for any $\lambda \in out(t, F_A)$, $\varphi \in V(\lambda[0,i])$ for all $i \geq 0$.

The induction step:

Assume that $d > \bar{0}$. As $\langle\!\langle A^{d_b}\rangle\!\rangle \Box\varphi \in V(t)$ and $V(t)$ is a maximally consistent set, φ and either $\langle\!\langle A^{b_1}\rangle\!\rangle \bigcirc \langle\!\langle A^{2}\rangle\!\rangle \Box\varphi$ for some $b_1 +^\infty b_2 = d_b$ with $b_2 < d_b$ or $\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle \bigcirc \langle\!\langle A^{d_b}\rangle\!\rangle \Box\varphi \in V(t)$.

- If $\langle\!\langle A^{b_1}\rangle\!\rangle \bigcirc \langle\!\langle A^{b_2}\rangle\!\rangle \Box\varphi \in V(t)$, by Lemma 15, there is a move $\sigma \in \Delta_A$ of cost no more than b_1 such that for all $c \in out(t,\sigma)$, $\langle\!\langle A^{b_2}\rangle\!\rangle \Box\varphi \in V(t)$. We define $F_A(t) = \sigma$. Moreover, as $b_2 < b$, by the induction hypothesis, we have a b_2-strategy $F_{A,c}$ which realises $\langle\!\langle A^{b_2}\rangle\!\rangle \Box\varphi$ at tc. Then, we simply define $F_A(tc\lambda) = F_{A,c}(c\lambda)$ for all $\lambda \in Q^*$. It is straightforward that for any $\lambda \in out(t, F_A)$, $\varphi \in V(\lambda[0,i])$ for all $i \geq 0$.

- If $\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle \bigcirc \langle\!\langle A^{d_b}\rangle\!\rangle \Box\varphi \in V(t)$, by Lemma 15, there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ such that for all $c \in out(t,\sigma)$, $\langle\!\langle A^{d_b}\rangle\!\rangle \Box\varphi \in V(tc)$. We define $F_A(t) = \sigma$.

  Assume that we already constructed F_A for all inputs $\lambda \in Q^*$ of length $i \geq 1$. We now need to define F_A for all inputs λc where $c \in out(\lambda[i-1], F_A(\lambda))$. The resource we spend so far no more than $\bar{0}_b$. As $\langle\!\langle A^{b}\rangle\!\rangle \Box\varphi \in V(\lambda c)$, repeating the above argument, we have that φ and either $\langle\!\langle A^{b_1}\rangle\!\rangle \bigcirc \langle\!\langle A^{b_2}\rangle\!\rangle \Box\varphi$ for some $b_1 +^\infty b_2 = d_b$ or $\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle \bigcirc \langle\!\langle A^{d_b}\rangle\!\rangle \Box\varphi \in V(\lambda c)$.

  - If $\langle\!\langle A^{b_1}\rangle\!\rangle \bigcirc \langle\!\langle A^{b_2}\rangle\!\rangle \Box\varphi \in V(\lambda c)$. By Lemma 15, there is a move $\sigma \in \Delta_A$ of cost no more than b_1 such that for all $c' \in out(\lambda c,\sigma)$, $\langle\!\langle A^{b_2}\rangle\!\rangle \Box\varphi \in V(\lambda cc')$. We define $F_A(\lambda c) = \sigma$. Moreover, as $b_2 < b$, by the induction hypothesis, we have a b_2-strategy $F_{A,c}$ which realises $\langle\!\langle A^{b_2}\rangle\!\rangle \Box\varphi$ at $\lambda cc'$. Then, we simply define $F_A(\lambda cc'\kappa) = F_{A,c}(c'\kappa)$ for all $\lambda \in Q^*$. Then, it is straightforward that for any $\lambda \in out(t, F_A)$, $\varphi \in V(\lambda[0,i])$ for all $i \geq 0$.
  - If $\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle \bigcirc \langle\!\langle A^{d_b}\rangle\!\rangle \Box\varphi \in V(\lambda c)$. By Lemma 15, there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ such that for all $c' \in out(\lambda c,\sigma)$, $\langle\!\langle A^{d_b}\rangle\!\rangle \Box\varphi \in V(\lambda cc')$. We define $F_A(\lambda c) = \sigma$. This construction also shows inductively on the length of input that for any $\lambda \in out(t, F_A)$, $\varphi \in V(\lambda[0,i])$ for all $i \geq 0$.

The proof for the case of $\neg\langle\!\langle A^{b}\rangle\!\rangle \varphi\mathcal{U}\psi$ is done in the similar way as that of the previous case. However, we construct a co-strategy F^c_A instead. As $\neg\langle\!\langle A^{b}\rangle\!\rangle \varphi\mathcal{U}\psi \in V(t)$, we also have that

$\neg\psi$, $\neg\langle\langle A^0\rangle\rangle\bigcirc\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ and $\neg\langle\langle A^{b_1}\rangle\rangle\bigcirc\langle\langle A^{b_2}\rangle\rangle\varphi\mathcal{U}\psi \in V(t)$ for all $b_1 +^\infty b_2 = b$. Then, the construction of F_A^c is done from the co-moves implied by $\neg\psi$, $\neg\langle\langle A^0\rangle\rangle\bigcirc\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ and $\neg\langle\langle A^{b_1}\rangle\rangle\bigcirc\langle\langle A^{b_2}\rangle\rangle\varphi\mathcal{U}\psi \in V(t)$ for all $b_1 +^\infty b_2 = b$. Moreover, the construction may end for a given computation if we reach a node t' where $\sim\varphi \in V(t')$.

□

Let S_{φ_0} be the model which is based on $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$.

Finally, we show the following truth lemma.

Lemma 21. *For every node t of $(T_{\varphi_0}, V_{\varphi_0}, C_{\varphi_0})$ and every formula $\varphi \in cl(\varphi_0)$, if $\varphi \in V_{\varphi_0}(t)$ then $S_{\varphi_0}, t \models \varphi$.*

Proof. The proof is done by induction on the structure of φ.

- For the cases of propositions, negations and disjunctions, the proofs are trivial.
- Assume $\varphi = \langle\langle A^b\rangle\rangle\bigcirc\psi$, Lemma 15 makes sure that there is a move $\sigma \in \Delta_A$ of cost at most b such that for all $c \in out(t,\sigma)$, we have $\psi \in V(tc)$. Then by the induction hypothesis, we have that $S_{\varphi_0}, tc \models \psi$. Then, $S_{\varphi_0}, t \models \langle\langle A^b\rangle\rangle\bigcirc\psi$
- Assume $\varphi = \neg\langle\langle A^b\rangle\rangle\bigcirc\psi$, Lemma 15 makes sure that there is a co-move $\sigma \in \Delta_A$ such that for all $c \in out(t,\sigma,b)$, we have $\sim\psi \in V(tc)$. Then by the induction hypothesis, we have that $S_{\varphi_0}, tc \models \sim\psi$. Then, $S_{\varphi_0}, t \models \neg\langle\langle A^b\rangle\rangle\bigcirc\psi$
- For the cases of $\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$, $\neg\langle\langle A^b\rangle\rangle\Box\varphi$, $\neg\langle\langle A^b\rangle\rangle\varphi\mathcal{U}\psi$ and $\langle\langle A^b\rangle\rangle\Box\varphi$, the proofs are trivial with the help of the two previous lemmas.

□

Finally, we have the following proposition.

Proposition 1. *The axiomatisation system for RB-ATL is sound and complete.*

We have the following corollary which is useful for the satisfiability problem of RB-ATL.

Corollary 1. *Every satisfiable RB-ATL formula is satisfied by a fixed-branching degree tree model where the cost of any action in the model is limited by some resource bound which only depends on the formula.*

Proof. It follows from the soundness of RB-ATL that every satisfiable RB-ATL formula φ is a consistent formula. Therefore, according to the completeness of RB-ATL, there is a tree model satisfying φ. This tree model has a fixed-branching degree which is k^n where $k = |cl(\varphi)_{\bigcirc}| + 1$. Let b_0 be defined as follow, the i-th component of b_0 is define to be the maximal i-th component of all bounds appearing in φ for all $1 \leq i \leq r$ plus one. According to the assignment of costs for actions presented in Lemma 15, it is possible to assign $e = b_0$. Thus, it is straightforward that no action has cost more than b_0. $\square$

5.7 Satisfiability of RB-ATL

Similar to the satisfiability problem of ATL, we also apply the automaton-based approach as presented in [Goranko & van Drimmelen, 2006] to determine the satisfiability of a RB-ATL formula φ_0. The proof of the correctness of the decision procedure introduced in this section is based on its counterpart for ATL in [Goranko & van Drimmelen, 2006] with the extension for dealing with resource bounds. Firstly, we recall the notion of Alternating Büchi Tree Automata. Then, we define for each RB-ATL formula an alternating Büchi tree automaton which only accepts a class of fixed-branching degree models satisfying the formula. Therefore, the algorithm for deciding the emptiness of alternating Büchi tree automata gives us a procedure for the satisfiability of RB-ATL, given that the number of resources and the number of agents are fixed.

5.7.1 Alternating Büchi tree automata

Firstly, we recall the notion of positive Boolean formulas which will be used later to define transitions of the automata.

Definition 33. *Given a set X, $B^+(X)$ is the set of positive formulas which are defined inductively from elements of X in the following way:*

- $\top$, $\bot$ *and any element of X are positive formulas,*
- *If θ_1 and θ_2 are positive formulas, so are $\theta_1 \wedge \theta_2$ and $\theta_1 \vee \theta_2$.*

A set $Y \subseteq X$ satisfies a formula $\theta \in B^+(X)$ iff assigning true to every element in Y and false to every element in $Y \setminus X$ makes θ true.

Note that if Y_1 satisfies θ_1 and Y_2 satisfies θ_2, then $Y_1 \cup Y_2$ satisfies $\theta_1 \wedge \theta_2$. In the following, we give the definition of alternating Büchi tree automata.

Definition 34. *A (finite) alternating Büchi automaton (ATA) is a tuple* $\mathcal{A} = (\Theta, k, S, s^*, \rho, F)$ *where:*

- Θ *is a finite alphabet,*
- k *is a finite branching degree,*
- S *is a finite set of states,*
- $s^* \in S$ *is an initial state,*
- $\rho : S \times \Theta \to B^+(\{0, \dots, k-1\} \times S)$ *is a partial transition function, and*
- $F \subseteq S$ *is a set of acceptance states.*

Inputs of ATA automata are Θ-labelled leafless k-trees (T, V, C). A run of an ATA automaton over a tree (T, V, C) is also a tree (T_r, r) where nodes are labelled by elements of $\mathbb{N}^* \times S$. The label of a node on (T_r, r) and its children have to satisfy the following conditions:

1. $r(\epsilon) = (\epsilon, s^*)$,
2. If $y \in T_r$, $r(y) = (x, s)$ and $\rho(s, V(x)) = \theta$, there is a set $Q = \{(c_0, s_0), \dots, (c_p, s_p)\} \subseteq \{0, \dots, k-1\} \times S$ such that:
   - Q satisfies θ and
   - For any $0 \le i \le p$, we have that $y \cdot i \in T_r$ and $r(y \cdot i) = (x \cdot c_i, s_i)$.

Given a path λ in a run (T_r, r), $\inf(\lambda)$ denotes the set of all states which appear infinitely often on λ. A run is accepting if every infinite path λ of the run satisfies $\inf(\lambda) \cap F \neq \emptyset$, i.e. there is at least a state in F appearing infinitely often on λ. An input tree (T, V, C) is accepted by an ATA automaton iff it has an accepting run. We denote the set of all trees which are accepted by an ATA automaton $\mathcal{A}$ by $\mathcal{T}_\omega(\mathcal{A})$.

5.7.2 ATA automata for RB-ATL formulas

Given a RB-ATL formula φ_0, we have the closure $cl(\varphi_0)$ and $k = |cl(\varphi_0)_{\bigcirc}| + 1$. Let b_0 be the limited bound as defined in the proof of Corollary 1 with respect to φ_0. For the definition of the transition function, we introduce a notation $ca(\Delta)$ which denotes the set of all possible cost assignments for actions in Δ. Recall that given k, $\Delta = \{\{1, \dots, k\}^n\}$ is the set of all joint actions for agents in N. Each assignment a in $ca(\Delta)$ defines the cost of each action $1 \le j \le k$ for every agent i to be some value between $\bar{0}$ and b_0. We denote this assignment as $a(j,i)$. We define an ATA automaton $\mathcal{A}_\varphi$ for φ_0 as follows.

Definition 35. *Let φ_0 be a RB-ATL formula over a set of propositions Prop and N be the set of agents with $|N| = n$, the corresponding ATA automaton $\mathcal{A}_{\varphi_0}$ of φ_0 is defined as $\mathcal{A}_{\varphi_0} = (\wp(Prop) \times ca(\Delta), k^n, cl(\varphi_0), \varphi_0, \rho, F)$ where:*

- *The transition function ρ is defined as follows:*
  - $\rho(p, (\pi, a)) = \top$ *if* $p \in \pi$
  - $\rho(p, (\pi, a)) = \bot$ *if* $p \notin \pi$
  - $\rho(\neg p, (\pi, a)) = \bot$ *if* $p \in \pi$
  - $\rho(\neg p, (\pi, a)) = \top$ *if* $p \notin \pi$
  - $\rho(\varphi_1 \wedge \varphi_2, (\pi, a)) = \rho(\varphi_1, (\pi, a)) \wedge \rho(\varphi_2, (\pi, a))$
  - $\rho(\varphi_1 \vee \varphi_2, (\pi, a)) = \rho(\varphi_1, (\pi, a)) \vee \rho(\varphi_2, (\pi, a))$
  - $\rho(\langle\langle A^b \rangle\rangle \bigcirc \varphi, (\pi, a)) =$
    $\bigvee_{\sigma \in \Delta_A : \sum_{i \in A} a(\sigma_i, i) \leq b} (\bigwedge_{c \in out(\sigma)} (c, \varphi))$
  - $\rho(\neg\langle\langle A^b \rangle\rangle \bigcirc \varphi, (\pi, a)) =$
    $\bigwedge_{\sigma \in \Delta_A : \sum_{i \in A} a(\sigma_i, i) \leq b} (\bigvee_{c \in out(\sigma)} (c, \sim\varphi))$
  - $\rho(\langle\langle A^b \rangle\rangle \Box \varphi, (\pi, a)) =$
    $\rho(\varphi, (\pi, a)) \wedge$
    $\bigvee_{b_1 +^\infty b_2 = b} (\rho(\langle\langle A^{b_1} \rangle\rangle \bigcirc \langle\langle A^{b_2} \rangle\rangle \Box \varphi, (\pi, a)))$
  - $\rho(\neg\langle\langle A^b \rangle\rangle \Box \varphi, (\pi, a)) =$
    $\rho(\sim\varphi, (\pi, a)) \vee$
    $\bigwedge_{b_1 +^\infty b_2 = b} (\rho(\neg\langle\langle A^{b_1} \rangle\rangle \bigcirc \langle\langle A^{b_2} \rangle\rangle \Box \varphi, (\pi, a)))$
  - $\rho(\langle\langle A^b \rangle\rangle \varphi_1 \mathcal{U} \varphi_2, (\pi, a)) =$
    $\rho(\varphi_2, (\pi, a)) \vee (\rho(\varphi_1, (\pi, a)) \wedge$
    $\bigvee_{b_1 +^\infty b_2 = b} (\rho(\langle\langle A^{b_1} \rangle\rangle \bigcirc \langle\langle A^{b_2} \rangle\rangle \varphi_1 \mathcal{U} \varphi_2, (\pi, a))))$
  - $\rho(\neg\langle\langle A^b \rangle\rangle \varphi_1 \mathcal{U} \varphi_2, (\pi, a)) =$
    $\rho(\sim\varphi_2, (\pi, a)) \wedge (\rho(\sim\varphi_1, (\pi, a)) \vee$
    $\bigwedge_{b_1 +^\infty b_2 = b} (\rho(\neg\langle\langle A^{b_1} \rangle\rangle \bigcirc \langle\langle A^{b_2} \rangle\rangle \varphi_1 \mathcal{U} \varphi_2, (\pi, a))))$

- *The set F of final state is defined as*

$$F = \{\langle\!\langle A^b \rangle\!\rangle \Box \varphi \in cl(\varphi_0)\} \cup \{\neg \langle\!\langle A^b \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2 \in cl(\varphi_0)\}$$

We have the following theorem.

Lemma 22. *Given a RB-ATL formula φ_0, then $\mathcal{T}_\omega(\mathcal{A}_{\varphi_0})$ is exactly the set of tree models of φ_0 where it has a fixed branching degree k^n and no action costs more than b_0 amount of resources.*

Proof. In the following, we show that the automaton $\mathcal{A}_{\varphi_0}$ accepts exactly the set $Tree(k^n, b_0)$ of tree models of φ_0 where each model has a fixed branching degree k^n and no action costs more than b_0 amount of resources. For convenience, we extend the definition of the function V in a model $(T, V, C) \in Tree(k^n, b_0)$ such that $V(x) = (\pi, C(x))$ where $\pi \subseteq Prop$ is the set of propositions labeling x (that is exactly $V(x)$ as before) and $C(x)$ is the cost assignment for actions in Δ at x, that is $C(x)(j, i) = C(x, i, j)$ for all $1 \le j \le k$ and $1 \le i \le n$. This extension allows the labels on those tree models of $Tree(k^n, b_0)$ are members of the alphabet of the automaton $\mathcal{A}_{\varphi_0}$ so that it makes sense to run the tree models on $\mathcal{A}_{\varphi_0}$.

In the first part of the proof, we show the direction where if a tree model $(T, V, C) \in Tree(k^n, b_0)$ has a successful run (T_r, r) on $\mathcal{A}_{\varphi_0}$, then it satisfies φ_0.

Firstly, we introduce the notion of sub-tree models and sub-runs.

Given a node $x \in T$, a sub-tree model of (T, V, C) at x, denoted as (T^x, V^x, C^x), is defined as follows:

- $T^x = T$
- $V^x(x') = V(x \cdot x')$
- $C^x(x') = C(x \cdot x')$

Intuitively, the sub-tree model (T^x, V^x, C^x) is the sub-tree of (T, V, C) which starts from the node x in T.

We define the notion of sub-runs of (T_r, r) as follows. For a node $y \in T_r$ and a formula $\varphi \in cl(\varphi_0)$ such that $r(y) = (x, \varphi)$, we define a sub-runs $(T_r^{y,\varphi}, r^{y,\varphi})$ where

- $z \in T_r^{y,\varphi}$ iff $y \cdot z \in T_r$ and
- $r^{y,\varphi}(z) = (x', s')$ iff $r(y \cdot z) = (x \cdot x', s')$.

We also define $\mathcal{A}^{\varphi}_{\varphi_0}$ as $\mathcal{A}_{\varphi_0}$ with the initial state replaced by φ.

We have the following claim.

Claim 4. *Given $y \in T_r$ and $r(y) = (x, \varphi)$, $(T^{y,\varphi}_r, r^{y,\varphi})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}^{\varphi}_{\varphi_0}$.*

Proof. The proof is straightforward. Firstly, we have that $r^{y,\varphi}(\epsilon) = (\epsilon, \varphi)$ as $r(y \cdot \epsilon) = (x \cdot \epsilon, \varphi)$. Moreover, let us consider any $z \in T^{y,\varphi}_r$, we have that $r^{y,\varphi}(z) = (x', \varphi')$ where $r(y \cdot z) = (x \cdot x', \varphi')$. As (T_r, r) is a successful run on $\mathcal{A}_{\varphi_0}$, we have that $\rho(\varphi', V(x \cdot x')) = \theta$ is satisfied by some subset $Q = \{(c_0, \varphi_0), \dots, (c_p, \varphi_p)\} \subset \{0, \dots, k^n - 1\} \times cl(\varphi_0)$. However, because of the definition of (T^x, V^x, C^x), we also have that $\rho(\varphi', V(x \cdot x')) = \rho(\varphi', V^x(x'))$, then Q also satisfies $\rho(\varphi', V^x(x'))$. Moreover, we also have that $y \cdot z \cdot i \in T_r$ and $r(y \cdot z \cdot i) = (c_i, \varphi_i)$ for all $0 \le i \le p$, thus $z \cdot i \in T^{y,\varphi}_r$ and $r^{y,\varphi}(z \cdot i) = (c_i, \varphi_i)$ for all $0 \le i \le p$. Therefore, $(T^{y,\varphi}_r, r^{y,\varphi})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}^{\varphi}_{\varphi_0}$. □

Notice that $(T^{\epsilon,\varphi_0}_r, r^{\epsilon,\varphi_0}) = (T_r, r)$ and $\mathcal{A}^{\varphi_0}_{\varphi_0} = \mathcal{A}_{\varphi_0}$.

We are going to prove the following claim.

Claim 5. *For any $y \in T_r$ with $r(y) = (x, \varphi)$, if $(T^{y,\varphi}_r, r^{y,\varphi})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}^{\varphi}_{\varphi_0}$, then $(T, V, C), x \models \varphi$.*

Proof. The proof is done inductively on the structure of φ.

- Assume that φ is a proposition p. As $(T^{y,p}_r, r^{y,p})$ is an accepting run on $\mathcal{A}^{p}_{\varphi_0}$, we must have $\rho(p, V(x)) = \top$. Then, assume that $V(x) = (\pi, a)$, we must have $p \in \pi$. Hence, $(T, V, C), x \models p$.

- Assume that φ is a proposition $\neg p$. As $(T^{y,\neg p}_r, r^{y,\neg p})$ is an accepting run on $\mathcal{A}^{\neg p}_{\varphi_0}$, we must have $\rho(\neg p, V(x)) = \top$. Then, assume that $V(x) = (\pi, a)$, we must have $p \notin \pi$. Hence, $(T, V, C), x \models \neg p$.

- Assume that φ is $\varphi_1 \wedge \varphi_2$. As $(T^{y,\varphi_1\wedge\varphi_2}_r, r^{y,\varphi_1\wedge\varphi_2})$ is an accepting run on $\mathcal{A}^{\varphi_1\wedge\varphi_2}_{\varphi_0}$, there must be a set $Q \subseteq \{0, k^n - 1\} \times cl(\varphi_0)$ satisfying $\rho(\varphi_1 \wedge \varphi_2, V(x)) = \rho(\varphi_1, V(x)) \wedge \rho(\varphi_2, V(x))$. Then, Q satisfies both $\rho(\varphi_1, V(x))$ and $\rho(\varphi_2, V(x))$. Using Q as a set which satisfies $\rho(\varphi_1, V(x))$ and repeat the proof of Claim 4, we imply that $(T^{y,\varphi_1}_r, r^{y,\varphi_1})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}^{\varphi_1}_{\varphi_0}$. By the induction hypothesis, we have that $(T, V, C), x \models \varphi_1$.

Apply the same argument as above, we also have that $(T, V, C), x \vDash \varphi_2$. Therefore, $(T, V, C), x \vDash \varphi_1 \wedge \varphi_2$.

- Assume that φ is $\varphi_1 \vee \varphi_2$. As $(T_r^{y,\varphi_1 \vee \varphi_2}, r^{y,\varphi_1 \vee \varphi_2})$ is an accepting run on $\mathcal{A}_{\varphi_0}^{\varphi_1 \vee \varphi_2}$, there must be a set $Q \subseteq \{0, k^n - 1\} \times cl(\varphi_0)$ satisfying $\rho(\varphi_1 \vee \varphi_2, V(x)) = \rho(\varphi_1, V(x)) \vee \rho(\varphi_2, V(x))$. Then, Q satisfies either $\rho(\varphi_1, V(x))$ or $\rho(\varphi_2, V(x))$. Without loss of generality, let us assume that Q satisfies $\rho(\varphi_1, V(x))$. Using Q as a set which satisfies $\rho(\varphi_1, V(x))$ and repeat the proof of Claim 4, we imply that $(T_r^{y,\varphi_1}, r^{y,\varphi_1})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}_{\varphi_0}^{\varphi_1}$. By the induction hypothesis, we have that $(T, V, C), x \vDash \varphi_1$. Therefore, $(T, V, C), x \vDash \varphi_1 \wedge \varphi_2$.

- Assume that φ is $\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi$. As $(T_r^{y,\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi}, r^{y,\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}_{\varphi_0}^{\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi}$, there must be a set $Q \subseteq \{0, k^n - 1\} \times cl(\varphi_0)$ satisfying

$$\rho(\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi, V(x)) = \bigvee_{\sigma \in \Delta_A : \sum_{i \in A} C(x,i,\sigma_i) \leq b} (\bigwedge_{c \in out(\sigma)} (c, \psi))$$

  Then, there exists a move $\sigma \in \Delta_A$ with the cost $\sum_{i \in A} C(x, i, \sigma_i) \leq b$ such that for all $c \in out(\sigma)$, $(c, \psi) \in Q$. Let us denote the index of each $(c, \psi) \in Q$ by i_c, then $y \cdot i_c \in T_r$ and $r(y \cdot i_c) = (x \cdot c, \psi)$. Then, by Claim 4, we have that $(T_r^{y \cdot i_c, \psi}, r^{y \cdot i_c, \psi})$ is an accepting run of $(T^{x \cdot c}, V^{x \cdot c}, C^{x \cdot c})$ on $\mathcal{A}_{\varphi_0}^{\psi}$. By the induction hypothesis, we have that $(T, V, C), x \cdot c \vDash \psi$ for all $c \in out(\sigma)$. As the cost $\sum_{i \in A} C(x, i, \sigma_i)$ of σ is no more than b, we have that $(T, V, C), x \vDash \langle\!\langle A^b \rangle\!\rangle \bigcirc \psi$.

- Assume that φ is $\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi$. As $(T_r^{y,\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi}, r^{y,\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}_{\varphi_0}^{\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi}$, there must be a set $Q \subseteq \{0, k^n - 1\} \times cl(\varphi_0)$ satisfying

$$\rho(\neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi, V(x)) = \bigwedge_{\sigma \in \Delta_A : \sum_{i \in A} C(x,i,\sigma_i) \leq b} (\bigvee_{c \in out(\sigma)} (c, \sim\psi))$$

  Then, for every move $\sigma \in \Delta_A$ with the cost $\sum_{i \in A} C(x, i, \sigma_i) \leq b$, there is an outcome $c_\sigma \in out(\sigma)$ such that $(c_\sigma, \sim\psi) \in Q$. Let us denote the index of each $(c_\sigma, \sim\psi) \in Q$ by i_{c_σ}, then $y \cdot i_{c_\sigma} \in T_r$ and $r(y \cdot i_{c_\sigma}) = (x \cdot c_\sigma, \sim\psi)$. Then, by Claim 4, we have that $(T_r^{y \cdot i_{c_\sigma}, \psi}, r^{y \cdot i_{c_\sigma}, \psi})$ is an accepting run of $(T^{x \cdot c_\sigma}, V^{x \cdot c_\sigma}, C^{x \cdot c_\sigma})$ on $\mathcal{A}_{\varphi_0}^{\sim\psi}$. By the induction hypothesis, we have that $(T, V, C), x \cdot c_\sigma \vDash \sim\psi$ for all $\sigma \in \Delta_A$ with the cost $\sum_{i \in A} C(x, i, \sigma_i) \leq b$. In other words, we have that $(T, V, C), x \vDash \neg\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi$.

- Assume that φ is $\langle\!\langle A^b \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$. As $(T_r^{y,\langle\!\langle A^b \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2}, r^{y,\langle\!\langle A^b \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2})$ is an accepting run of

(T^x, V^x, C^x) on $\mathcal{A}_{\varphi_0}^{\langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2}$, there must be a set $Q \subseteq \{0, k^n - 1\} \times cl(\varphi_0)$ satisfying

$$\rho(\langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2, V(x)) =$$
$$\rho(\varphi_2, V(x)) \vee (\rho(\varphi_1, V(x)) \wedge \bigvee_{b_1 +^{\infty} b_2 = b}(\rho(\langle\langle A^{b_1}\rangle\rangle\bigcirc\langle\langle A^{b_2}\rangle\rangle\varphi_1\mathcal{U}\varphi_2, V(x))))$$

Let us construct a b-strategy for A to satisfy $\langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2$. We consider b as in the projection of a bound $d \in \mathbb{B}$ and b and the construction is done by induction on d.

The base case:

When $d = \bar{0}$, we have that

$$\rho(\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi_1\mathcal{U}\varphi_2, V(x)) =$$
$$\rho(\varphi_2, V(x)) \vee (\rho(\varphi_1, , V(x)) \wedge \rho(\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi_1\mathcal{U}\varphi_2, V(x)))$$

We define a subtree G of (T, V, C) rooted at x on which the strategy is based. A node $z \in G$ is called internal if there is a node $z \cdot c \in G$ for some $c \in \{0, k^n - 1\}$, otherwise it is called external. Then, we define G inductively as follows.

- Initially, G contains only x,
- In the base case, we consider the external node $x \in G$
  - $\star$ If Q satisfies $\rho(\varphi_2, V(x))$, applying the proof of Claim 4, we have that $(T_r^{y,\varphi_2}, r^{y,\varphi_2})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}_{\varphi_0}^{\varphi_2}$. By the induction hypothesis, we have $(T, V, C), x \vDash \varphi_2$.
  - $\star$ Otherwise, Q must satisfy

    $$\rho(\varphi_1, V(x)) \wedge \rho(\langle\langle A^{\bar{0}_b}\rangle\rangle\bigcirc\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi_1\mathcal{U}\varphi_2, V(x))$$

    Then applying the same argument as the previous case, we have $(T, V, C), x \vDash \varphi_1$. Moreover, applying the same argument as in the case of $\langle\langle A^b\rangle\rangle\bigcirc\psi$, there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ at x such that for all $c \in out(\sigma)$, $(c, \langle\langle A^{\bar{0}_b}\rangle\rangle\varphi_1\mathcal{U}\varphi_2) \in Q$. We denote the index of $(c, \langle\langle A^{\bar{0}_b}\rangle\rangle\varphi_1\mathcal{U}\varphi_2)$ in Q by i_c, then $r(y \cdot i_c) = (x \cdot c, \langle\langle A^{\bar{0}_b}\rangle\rangle\varphi_1\mathcal{U}\varphi_2)$. We add every $x \cdot c$ where $c \in out(\sigma)$ into G. We also assign $\sigma_{x \cdot c} = \sigma$.
- In the induction step, we consider an external node $x \cdot c_1 \cdots c_m$ where $r(y \cdot i_{c_1} \cdots i_{c_m}) = (x \cdot c_1 \cdots c_m, \langle\langle A^{\bar{0}_b}\rangle\rangle\varphi_1\mathcal{U}\varphi_2)$. Then, there is a subset $Q' \subseteq \{0, \ldots, k^n - 1\} \times cl(\varphi_0)$ satisfying $\rho(\langle\langle A^{\bar{0}_b}\rangle\rangle\varphi_1\mathcal{U}\varphi_2, V(x \cdot c_1 \cdots c_m))$.

* If Q' satisfies $\rho(\varphi_2, V(x \cdot c_1 \dots c_m))$, we have that

$$(T_r^{y \cdot i_{c_1} \cdots i_{c_m}, \varphi_2}, r^{y \cdot i_{c_1} \cdots i_{c_m}, \varphi_2})$$

is an accepting run of $(T^{x \cdot c_1 \dots c_m}, V^{x \cdot c_1 \dots c_m}, C^{x \cdot c_1 \dots c_m})$ on $\mathcal{A}_{\varphi_0}^{\varphi_2}$. By the induction hypothesis, we have $(T, V, C), x \cdot c_1 \cdots c_m \vDash \varphi_2$.

* Otherwise, Q' must satisfy

$$\rho(\varphi_1, V(x \cdot c_1 \dots c_m)) \wedge \rho(\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc \langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2, V(x \cdot c_1 \dots c_m))$$

Then, we have $(T, V, C), x \cdot c_1 \cdots c_m \vDash \varphi_1$ and there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ at $x \cdot c_1 \cdots c_m$ such that for all $c \in out(\sigma)$, $(c, \langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2) \in Q'$. We denote the index of $(c, \langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$ in Q' by i_c, then $r(y \cdot i_{c_1} \cdots i_{c_m} \cdot i_c) = (x \cdot c_1 \cdots c_m \cdot c, \langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$. We add every $x \cdot c_1 \cdots c_m \cdot c$ into G. We also assign $\sigma_{x \cdot c_1 \cdots c_m \cdot c} = \sigma$.

The construction of G terminates when no new node is added. G must be finite as otherwise, there is an infinite path $\lambda = x, x \cdot c_1, x \cdot c_1 \cdot c_2, \dots$ in G such that $r(y \cdot i_{c_1} \cdots i_{c_m}) = (x \cdot c_1 \cdots c_m, \langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$ for all $m \geq 0$. Then, $\inf(\lambda) \cap F = \varnothing$ which contradicts the fact that (T_r, r) is an accepting run. Then we define a strategy F_A by assigning $F_A((x) \dots (x \cdot c_1 \cdots c_m)) = \sigma_{x \cdot c_1 \cdots c_{m+1}}$ for every path $(x) \dots (x \cdot c_1 \cdots c_m)$ where $(x \cdot c_1 \cdots c_m) \in G$ for all $m \geq 0$ and there is some $x \cdot c_1 \cdots c_{m+1} \in G$. For all other inputs of F_A, we simply define the output as idle actions for all agents in A which cost $\bar{0}$.

By defining such F_A, we have that for every $\lambda \in out(x, F_A)$ whose cost for agents in A is always $\bar{0}_b$, there is a prefix which is a finite path in G and ending with an external node in G such that φ_1 is satisfied on all internal nodes of G on this prefix and φ_2 is satisfied on the external node of G at the end of the prefix. In other words, $(T, V, C), x \vDash \langle\!\langle A^{\bar{0}} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$.

The induction case:

When $d > \bar{0}$, we have that

$$\begin{aligned}&\rho(\langle\!\langle A^{d_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2, V(x)) = \\ &\rho(\varphi_2, V(x)) \vee (\rho(\varphi_1, V(x)) \wedge \textstyle\bigvee_{b_1 +^{\infty} b_2 = d_b} (\rho(\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc \langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2, V(x))))\end{aligned}$$

We also define a subtree G of (T, V, C) rooted at x on which the strategy is based. G is defined inductively as follows.

- Initially, G contains only x,
- In the base case, we consider the external node $x \in G$
  * If Q satisfies $\rho(\varphi_2, V(x))$, applying Claim 4, we have that $(T_r^{y,\varphi_2}, r^{y,\varphi_2})$ is an accepting run of (T^x, V^x, C^x) on $\mathcal{A}_{\varphi_0}^{\varphi_2}$. By the induction hypothesis, we have $(T, V, C), x \models \varphi_2$.
  * Otherwise, if Q satisfies

$$\rho(\varphi_1, V(x)) \wedge \rho(\langle\!\langle A^{b_1}\rangle\!\rangle \bigcirc \langle\!\langle A^{b_2}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2, V(x))$$

  for some $b_1 +^{\infty} b_2 = d_b$ where $b_2 < d_b$, then we have $(T, V, C), x \models \varphi_1$. Moreover, there is a move $\sigma \in \Delta_A$ costing at most b_1 at x such that for all $c \in out(\sigma)$, $(c, \langle\!\langle A^{b_2}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2) \in Q$. We denote the index of $(c, \langle\!\langle A^{b_2}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$ in Q by i_c, then $r(y \cdot i_c) = (x \cdot c, \langle\!\langle A^{b_2}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$. We add every $x \cdot c$ into G. We also assign $\sigma_{x \cdot c} = \sigma$.
  * Otherwise, Q must satisfy

$$\rho(\varphi_1, V(x)) \wedge \rho(\langle\!\langle A^{\bar{0}_b}\rangle\!\rangle \bigcirc \langle\!\langle A^{d_b}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2, V(x))$$

  Then we have $(T, V, C), x \models \varphi_1$. Moreover, there is a move $\sigma \in \Delta_A$ which costs no more than $\bar{0}_b$ at x such that for all $c \in out(\sigma)$, $(c, \langle\!\langle A^{d_b}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2) \in Q$. We denote the index of $(c, \langle\!\langle A^{d_b}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$ in Q by i_c, then $r(y \cdot i_c) = (x \cdot c, \langle\!\langle A^{d_b}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$. We add every $x \cdot c$ into G. We also assign $\sigma_{x \cdot c} = \sigma$.
- In the induction step, we consider an external node $x \cdot c_1 \cdots c_m$ at which $r(y \cdot i_{c_1} \cdots i_{c_m}) = (x \cdot c_1 \cdots c_m, \langle\!\langle A^{d_b}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$. Then, there is a subset $Q' \subseteq \{0, \ldots, k^n - 1\} \times cl(\varphi_0)$ satisfying $\rho(\langle\!\langle A^{d_b}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2, V(x \cdot c_1 \cdots c_m))$.
  * If Q' satisfies $\rho(\varphi_2, V(x \cdot c_1 \ldots c_m))$, by Claim 4, we have that $(T_r^{y \cdot i_{c_1} \cdots i_{c_m}, \varphi_2}, r^{y \cdot i_{c_1} \cdots i_{c_m}, \varphi_2})$ is an accepting run of $(T^{x \cdot c_1 \ldots c_m}, V^{x \cdot c_1 \ldots c_m}, C^{x \cdot c_1 \ldots c_m})$ on $\mathcal{A}_{\varphi_0}^{\varphi_2}$. By the induction hypothesis, we have $(T, V, C), x \cdot c_1 \cdots c_m \models \varphi_2$.
  * Otherwise, if Q' satisfies

$$\rho(\varphi_1, V(x \cdot c_1 \ldots c_m)) \wedge \rho(\langle\!\langle A^{b_1}\rangle\!\rangle \bigcirc \langle\!\langle A^{b_2}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2, V(x \cdot c_1 \ldots c_m))$$

  for some $b_1 +^{\infty} b_2 = d_b$ where $b_2 < d_b$, then we have $(T, V, C), x \cdot c_1 \cdots c_m \models \varphi_1$ and there is a move $\sigma \in \Delta_A$ costing at most b_1 at $x \cdot c_1 \cdots c_m$ such that for all $c \in out(\sigma)$, $(c, \langle\!\langle A^{b_2}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2) \in Q'$. We denote the index of $(c, \langle\!\langle A^{b_2}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$ in Q by i_c, then $r(y \cdot i_{c_1} \cdots i_{c_m} \cdot i_c) = (x \cdot c_1 \cdots c_m \cdot c, \langle\!\langle A^{b_2}\rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$. We add every $x \cdot c_1 \cdots c_m \cdot c$ into G. We also assign $\sigma_{x \cdot c_1 \cdots c_m \cdot c} = \sigma$.

* Otherwise, Q' must satisfy

$$\rho(\varphi_1, V(x \cdot c_1 \dots c_m)) \wedge \rho(\langle\!\langle A^{\bar{0}_b} \rangle\!\rangle \bigcirc \langle\!\langle A^{d_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2, V(x \cdot c_1 \dots c_m))$$

Then we have $(T, V, C), x \cdot c_1 \cdots c_m \models \varphi_1$ and there is a $\bar{0}_b$-cost move $\sigma \in \Delta_A$ at $x \cdot c_1 \cdots c_m$ such that for all $c \in out(\sigma)$, $(c, \langle\!\langle A^{d_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2) \in Q$. We denote the index of $(c, \langle\!\langle A^{d_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$ in Q by i_c, then $r(y \cdot i_{c_1} \cdots i_{c_m} \cdot i_c) = (x \cdot c_1 \cdots c_m \cdot c, \langle\!\langle A^{d_b} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$. We add every $x \cdot c_1 \cdots c_m \cdot c$ into G. We also assign $\sigma_{x \cdot c_1 \cdots c_m \cdot c} = \sigma$.

The construction of G terminates when no new node is added. G must be finite as otherwise, there is an infinite path $\lambda = x, x \cdot c_1, x \cdot c_1 \cdot c_2, \dots$ in G such that $r(y \cdot i_{c_1} \cdots i_{c_m}) = (x \cdot c_1 \cdots c_m, \langle\!\langle A^b \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$ for all $m \geq 0$. Then, $\inf(\lambda) \cap F = \varnothing$ which contradicts the fact that (T_r, r) is an accepting run. Then we define a strategy F_A by assigning $F_A((x) \dots (x \cdot c_1 \cdots c_m)) = \sigma_{x \cdot c_1 \cdots c_{m+1}}$ for every path $(x) \dots (x \cdot c_1 \cdots c_m)$ where $(x \cdot c_1 \cdots c_m) \in G$ for all $m \geq 0$ and there is some $x \cdot c_1 \cdots c_{m+1} \in G$. For external nodes $x \cdot c_1 \cdots c_m$ of G where $r(y \cdot i_{c_1} \cdots i_{c_m}) = (x \cdot c_1 \cdots c_m, \langle\!\langle A^{b_2} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2)$ for some $b_2 < d_b$, by the induction hypothesis, there is a b_1-strategy F'_A at $x \cdot c_1 \cdots c_m$ for A which makes $\langle\!\langle A^{b_2} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$ true. We simply define $F_A((x) \dots (x \cdot c_1 \cdots c_m)\lambda) = F'_A(\lambda)$. It is also straightforward to prove that $\sum_{i \leq m} cost(\sigma_{x \cdot c_1 \cdots c_i}) \leq b_1$ where $b_1 +^{\infty} b_2 = d_b$. For all other input of F_A, we simply define the output as idle actions for all agents in A which cost $\bar{0}$.

By defining such F_A, we have that for every $\lambda \in out(x, F_A)$ whose cost for agents in A is always no more than d_b, there is a prefix which is a finite path in G and ends with an external node in G such that φ_1 is satisfied on all internal nodes of G on this prefix and φ_2 or $\langle\!\langle A^{b_2} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$ where $b_1 +^{\infty} b_2 = d_b$ and $b_2 < d_b$ is satisfied on the external node of G at the end of the prefix. In the later case, i.e. the external node satisfies $\langle\!\langle A^{b_2} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$, by the induction hypothesis, there is a b_1-strategy for A at the external node to make $\langle\!\langle A^{b_2} \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$ while the total cost of actions from x leading to this external node is no more than b_1. In other words, $(T, V, C), x \models \langle\!\langle A^b \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$.

- Proofs are similar for the case of $\neg\langle\!\langle A^b \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$, $\langle\!\langle A^b \rangle\!\rangle \Box \psi$ and $\neg\langle\!\langle A^b \rangle\!\rangle \Box \psi$.

□

Therefore, the result of Claim 5 shows us that $(T, V, C), \epsilon \models \varphi_0$.

Let us now prove the other direction of the lemma. We show that for every tree model in *Tree*(k^n, b_0) which satisfies the formula φ_0, there is an accepting run on the automaton $\mathcal{A}_{\varphi_0}$.

Assume that we have a tree model $(T,V,C) \in \mathit{Tree}(k^n, b_0)$ where $(T,V,C), \epsilon \models \varphi_0$. We now construct an acceptance run (T_r, r) on $\mathcal{A}_{\varphi_0}$. The construction is done inductively as follows.

Initially, we add ϵ to T_r and assign $r(\epsilon) = (\epsilon, \varphi_0)$. We continue adding nodes to the run and always make sure that if y is added into T_r with $r(y) = (x, \varphi)$, we must have $(T,V,C), x \models \varphi$. It is straightforward to see that this property holds for the root ϵ of T_r.

We now consider a leaf y in T_r with $r(y) = (x, \varphi)$ and $(T,V,C), x \models \varphi$. We will prove the following:

- There exists a set $Q \subseteq \{0, k^n - 1\} \times cl(\varphi_0)$ which satisfies the formula $\rho(\varphi, V(x))$. Then, we expand T_r as follows. For every $(c, \varphi_c) \in Q$, let i_c be the index of c in Q, we add i_c as a child of y and assign $r(y \cdot i_c) = (x \cdot c, (\varphi_c, d_c))$.

- $(T,V,C), x \cdot c \models \varphi_c$.

The proof proceeds by induction on the structure of φ.

- If φ is p, as $(T,V,C), x \models p$, let $V(x) = (\pi, C(x))$, we must have $p \in \pi$. Therefore, $\rho(p, V(x))$ is satisfied by an empty Q.

- If φ is $\neg p$, as $(T,V,C), x \models \neg p$, let $V(x) = (\pi, C(x))$, we must have $p \notin \pi$. Therefore, $\rho(p, V(x))$ is also satisfied by an empty Q.

- If $\varphi = \varphi_1 \wedge \varphi_2$, then $(T,V,C), x \models \varphi_1 \wedge \varphi_2$ implies that $(T,V,C), x \models \varphi_1$ and $(T,V,C), x \models \varphi_2$. By the induction hypothesis, there are two sets Q_1 and Q_2 which satisfy $\rho(\varphi_1, V(x))$ and $\rho(\varphi_2, V(x))$, respectively. Moreover, for every $(c, \varphi_c) \in Q_1$ or Q_2, we have that $(T,V,C), x \cdot c \models \varphi_c$. Because $\rho(\varphi_1 \wedge \varphi_2, V(x)) = \rho(\varphi_1, V(x)) \wedge \rho(\varphi_2, V(x))$ is a positive Boolean formula, we have that $Q = Q_1 \cup Q_2$ satisfies $\rho(\varphi_1 \wedge \varphi_2, V(x))$.

- If $\varphi = \varphi_1 \vee \varphi_2$, then $(T,V,C), x \models \varphi_1 \vee \varphi_2$ implies that $(T,V,C), x \models \varphi_1$ or $(T,V,C), x \models \varphi_2$. Without loss of generality, let us assume that $(T,V,C), x \models \varphi_1$. By the induction hypothesis, there is a set Q which satisfy $\rho(\varphi_1, V(x))$ and for every $(c, \varphi_c) \in Q$, we have that $(T,V,C), x \cdot c \models \varphi_c$. It is also straightforward that Q satisfies $\rho(\varphi_1 \vee \varphi_2, V(x))$.

- If $\varphi = \langle\!\langle A^b \rangle\!\rangle \bigcirc \psi$, $(T,V,C), x \models \langle\!\langle A^b \rangle\!\rangle \bigcirc \psi$ implies that there is a move $\sigma \in \Delta_A$ with the cost $\sum_{i \in A} C(x, i, \sigma_i) \leq b$ such that for every $c \in out(\sigma)$, $(T,V,C), x \cdot c \models \psi$. We define $Q = \{(c, \psi) \mid c \in out(\sigma)\}$. It is straightforward that Q satisfies $\rho(\langle\!\langle A^b \rangle\!\rangle \bigcirc \psi, V(x))$.

- If $\varphi = \neg\langle\langle A^b\rangle\rangle\bigcirc\psi$, $(T,V,C),x \models \neg\langle\langle A^b\rangle\rangle\bigcirc\psi$ implies that for any move $\sigma \in \Delta_A$ with the cost $\sum_{i\in A} C(x,i,\sigma_i) \leq b$, there exists an outcome $c_\sigma \in out(\sigma)$, $(T,V,C),x\cdot c_\sigma \models \sim\psi$. We define $Q = \{(c_\sigma, \sim\psi) \mid \sigma \in \Delta_A : \sum_{i\in A} C(x,i,\sigma_i) \leq b\}$. It is straightforward that Q satisfies $\rho(\neg\langle\langle A^b\rangle\rangle\bigcirc\psi, V(x))$.

- If $\varphi = \langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2$, then $(T,V,C),x \models \langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2$. This implies that either $(T,V,C),x \models \varphi_2$ or $(T,V,C),x \models \varphi_1 \wedge \langle\langle A^{b_1}\rangle\rangle\bigcirc\langle\langle A^{b_2}\rangle\rangle\varphi_1\mathcal{U}\varphi_2$ for some $b_1 +^\infty b_2 = b$.

  In the first case, by the induction hypothesis, there is a set Q which satisfies $\rho(\varphi_2, V(x))$.

  In the second case, firstly, by the induction hypothesis, there must be a set Q_1 which satisfies $\rho(\varphi_1, V(x))$. Secondly, during the construction T_r, we may attach to a node $y \in T_r$ a strategy. The reason of this attachment will be explained later. If y has not been attached with a b-strategy, because $(T,V,C),x \models \langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2$, there must be a b-strategy F_A for coalition A to make $\langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2$ true at x. Let b' be the cost of $\sigma = F_A(x)$ and $b_2 \in \mathbb{B}^\infty$ such that $b'_b +^\infty b_2 = b$. If y is already attached with b-strategy F_A, similarly, let b' be the cost of $\sigma = F_A(x)$ and $b_2 \in \mathbb{B}^\infty$ such that $b'_b +^\infty b_2 = b$.

  Then, for every $c \in out(\sigma)$, we have that $(T,V,C),x\cdot c \models \langle\langle A^{b_2}\rangle\rangle\varphi_1\mathcal{U}\varphi_2$. We define $Q_2 = \{(c, \langle\langle A^{b_2}\rangle\rangle\varphi_1\mathcal{U}\varphi_2) \mid c \in out(\sigma)\}$. We simple choose $Q = Q_1 \cup Q_2$, then Q satisfies $\rho(\langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2, V(x))$. Moreover, for every $c \in out(\sigma)$ where $(c, \langle\langle A^{b_2}\rangle\rangle\varphi_1\mathcal{U}\varphi_2)$ has the index i_c in Q, by the construction of T_r, i_c is added into T_r as a child of y with the assignment $r(y\cdot i_c) = (x\cdot c, \langle\langle A^{b_2}\rangle\rangle\varphi_1\mathcal{U}\varphi_2)$, we attach a b_2-strategy F'_A to $y\cdot i_c$ which is defined as $F'_A(\lambda) = F_A(x\cdot\lambda)$. The idea of attaching a strategy to a node in the run T_r is that when the construction continues at $y\cdot i_c$, we still follow the strategy F_A which has been chosen at y to satisfy the eventually formula $\langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2$. As this formula is satisfied in the model, following a fixed strategy helps us to not generate a formula in the form $\langle\langle A^{b''}\rangle\rangle\varphi_1\mathcal{U}\varphi_2$ infinitely often along some specific branch of the run T_r.

- The case when $\varphi = \neg\langle\langle A^b\rangle\rangle\Box\psi$ is treated in a similar way as for the above case. However, instead of using strategies, we can make use of the co-strategies. Because of the similarity, we omit the proof here.

- The cases when $\varphi = \langle\langle A^b\rangle\rangle\Box\psi$ and $\varphi = \neg\langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2$ is also treated in a similar ways as for the above cases. However, making use of strategies (or co-strategy, respectively) will produce on the run T_r infinite branches on which states of the form $(\langle\langle A^b\rangle\rangle\Box\psi, d)$ (or

($\neg\langle\langle A^b\rangle\rangle\varphi_1\mathcal{U}\varphi_2, d$), respectively) appear infinitely often. This fact satisfies the condition for accepting the run T_r on infinite branches.

□

We have the following decidability result.

Proposition 2. *RB-ATL is decidable.*

The proof is rather straightforward. Given a formula φ in RB-ATL, we construct the corresponding alternating-tree automaton as described above. Then, the decidability of the satisfiability of φ is determined by deciding whether the corresponding alternating-tree automaton is non-empty. As this problem is decidable, hence RB-ATL is decidable. We already know from the definition of $cl(\varphi_0)$ that the size of $cl(\varphi_0)$ is bounded by $2^{2^{m^r \times |\varphi_0|}}$, where m is the maximal bound of any resource appearing in φ_0 and r is the number of resources. Moreover, from [Goranko & van Drimmelen, 2006] we have that the emptiness problem for ATA automata is decidable in exponential time with respect to the size of the input automaton. Therefore, the algorithm for deciding the emptyness problem of ATA automata gives us an triple exponential time procedure for the satisfiability problem of RB-ATL.

5.8 Conclusion

In comparison with BMCL and RBCL, RB-ATL is the most fully-fledged logic so far for specifying and reasoning about resource-bounded multi-agent systems. The logic RB-ATL is an extension of ATL where resource bounds are attached to every coalition appearing in a formula of ATL. Similar to the case of RB-CL, resource bounds attached to a coalition restrict the abilities of the coalition where the greater resource bounds are attached, the more abilities a coalition may have. Moreover, we also introduce the symbol ∞ in resource bounds in order to remove the limitation of particular resources. Rather than always set a concrete bound on every resource like RBCL, the extension of the set of resource bounds in RB-ATL enables us to reason about more properties of resource-bounded multi-agent systems. For example, the formula $\langle\langle A^{(x,\infty,y,\infty)}\rangle\rangle\top\mathcal{U}\varphi$ where x, y are some natural numbers allows us to reason about the ability of the coalition A when there is no restriction on the usage of particular resources (those have the bound ∞, respectively). Similarly, RB-ATL also allows us to investigate the importance of resources when a coalition does not have an expected ability when it is not allowed to spend any of such resources while other resources have no restriction on usage. Then, we could use the formula $\langle\!\langle A^{(0,\infty,0,\infty)} \rangle\!\rangle \top \mathcal{U} \varphi$ where 0's are the bounds for resources which are considered to be investigated.

In this chapter, we have defined the syntax and semantics of RB-ATL. In order to simplify the proof of the soundness and completeness of the logic, as well as the satisfiability of RB-ATL, we also presented the normal form RB-ATL where negation can only appear in front of a propositional variables or temporal operators. Then, to define the semantics of the normal RB-ATL, apart from the notions of moves and strategies as in the semantics of RB-ATL, we presented the notions of co-moves and co-strategies which help the definition of semantics for formulas where negation is in front of temporal operators.

Before ending this chapter, let us briefly discuss the relationship between RBCL and RB-ATL. Firstly, RB-ATL allows modelling more types of coalitional strategies through the help of until ($\mathcal{U}$) operator. Therefore, the statements such as a coalition can maintain a certain condition until it achieves some goal under a resource bound is not expressible in RBCL. Furthermore, RBCL does not allow modelling coalitional abilities with unbounded conditions for some resources. However, RB-ATL is defined for reasoning about resource-bounded multi-agent systems where combined costs are defined only by means of addition operator. Therefore, the question of whether the results of RB-ATL that we have in this chapter are still hold for the case of general cost-combining operators, as defined for RBCL, is still open.

CHAPTER 6

CONCLUSION

In this thesis we have discussed logic-based formalisms for specifying and reasoning about resource-bounded multi-agent systems. In our models of resource-bounded multi-agent systems, in order to perform a particular action, each agent has to pay a certain amount of resources. The bounds on the amount of resources which can be used by agents in the system effectively limit the abilities of each agent or group of agents. The thesis has presented a series of logic-based formalisms which allow us to specify and reason about the abilities of agents or groups of agents in resource-bounded multi-agent systems under resource bounds.

6.1 Review of the chapters

The first logic for modelling resource-bounded multi-agent systems presented in Chapter 3 of this thesis is named Bounded Memory Communication Logic (BMCL). The logic, which is an extension of CTL, allows reasoning about the ability of systems of multiple reasoning agents. Agents in those systems are assumed to operate by using only two types of resources, namely memory and communication. In the semantics of BMCL, each model is associated with a fixed resource bound for each resource and the usage of every resource is recorded at each state so that actions available for each agent at a state are limited by the cost of the action, the fixed resource bound and the recorded usage of the resources at that state. In other words, BMCL allows reasoning about the ability of a system of multiple reasoning agents under a certain resource bound where the only resources are memory and communication. Moreover, as BMCL is based on CTL, we cannot reason about the properties of individual agents or groups of agents in the system.

In order to overcome the drawbacks of BMCL, we have introduced Resource-Bounded Coalition Logic (RBCL) and Resource-Bounded Alternating-time temporal logic (RB-ATL) in Chapter 4 and Chapter 5, respectively. As its name suggests, RBCL is an extension of Coalition Logic where each coalitional modality is extended with a resource bound determining the maximal amount of resource agents in a coalition can use. In the semantics of RBCL, a model associates each action with a finite set of resources that the agents in the system must use when performing this action. Rather than record the amount of resources which has been used by agents from state to state, we define for each action a certain cost. Then, the question whether an agent can perform an action from a state is answered by comparing the cost of the action with the resource bound of the agent. Furthermore, RBCL not only allows reasoning about single strategies under resource bounds but also enables us to reason about multi-step strategies for an individual agent or a group of agents under resource bounds for obtaining a certain goal. As usual, we study the soundness, the completeness and the satisfiability problems of RBCL in Chapter 4.

In Chapter 5, we have presented a more expressive logic for reasoning about resource-bounded multi-agent systems, namely RB-ATL. Because of basing on ATL, RB-ATL enables the reasoning about strategies for an individual agent or a coalition of agents under resource bounds where strategies are not only for obtaining a certain goal but also for maintaining a condition. Furthermore, we introduce the unlimited symbol (∞) in resource bounds so that bounds on certain resources can be ignored while reasoning about the ability of a coalition. Therefore, RB-ATL allows more properties to be formalised. We have also shown in Chapter 5 the soundness and the completeness of RB-ATL, and that RB-ATL is decidable.

6.2 Future work

The thesis has presented a theoretical framework for reasoning about resource-bounded multi-agent systems. For each logic introduced in the thesis, we concentrated on theoretical results which are sound and complete axiomatisation systems for these logics and their satisfiability problems. In the case of BMCL, practical results in the model-checking problem of BMCL have been presented in [Alechina et al., 2008c, Alechina et al., 2008b]. Although the model-checking algorithms for both RBCL and RB-ATL have been presented in [Alechina et al., 2009b, Alechina et al., 2010b], there is no implementation for them yet. Therefore, in the future, one direction is to devote more effort to more practical results on the work with resource-bounded multi-agent systems where a framework for verifying properties of resource-bounded multi-agent systems is developed by implementing model-checking algorithms for RBCL and RB-ATL.

For the future work, there is also other direction to extend the current theoretical results of the thesis. In this thesis, we have presented logic-based formalisms for modelling and reasoning about resource-bounded multi-agent systems. In these systems, agents only spend resources rather than produce them. In the consequence, logical languages presented in this thesis only allow us to work with multi-agent systems where agents do not have the ability to produce resources. Theoretical work on such systems have been initiated in [Bulling & Farwer, 2009] where the authors has extended CTL and CTL* for formalising properties of those system. However, they have not had axiomatisation and decidability results where the model-checking problem is only solved in for a limited sub-logic. In other words, the questions of modelling and reasoning about such systems where agents can both consume and produce resource are still open and a direction for the future work is to extend our current results (particularly, RBCL and RB-ATL) to cover those systems. Furthermore, we would also like to study the relationship between RB-ATL with other derivations of ATL such as ATLBM [Ågotnes & Walther, 2009] (Alternative-time Temporal Logic with Bounded Memory) where bounded memory means that the size of agents' strategies is limited and agents have bounded recall.

References

[Ågotnes, 2004] Ågotnes, T. (2004). *A logic of finite syntactic epistemic states*. PhD thesis, Department of Informatics, University of Bergen.

[Agotnes & Alechina, 2006] Agotnes, T. & Alechina, N. (2006). Knowing Minimum/Maximum n Formulae. *Frontiers In Artificial Intelligence And Applications*, 141, 317.

[Ågotnes et al., 2008a] Ågotnes, T., Van Der Hoek, W., & Wooldridge, M. (2008a). Quantified coalition logic. *Synthese*, 165(2), 269–294.

[Ågotnes et al., 2008b] Ågotnes, T., van der Hoek, W., & Wooldridge, M. (2008b). Quantifying over coalitions in epistemic logic. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2* (pp. 665–672).: International Foundation for Autonomous Agents and Multiagent Systems.

[Ågotnes et al., 2009a] Ågotnes, T., Van Der Hoek, W., & Wooldridge, M. (2009a). Logics for qualitative coalitional games. *Logic Journal of IGPL*, 17(3), 299.

[Ågotnes et al., 2009b] Ågotnes, T., van der Hoek, W., & Wooldridge, M. (2009b). Reasoning about coalitional games. *Artificial Intelligence*, 173(1), 45–79.

[Ågotnes & Walther, 2009] Ågotnes, T. & Walther, D. (2009). A logic of strategic ability under bounded memory. *Journal of Logic, Language and Information*, 18(1), 55–77.

[Alechina et al., 2006a] Alechina, N., Bertoli, P., Ghidini, C., Jago, M., Logan, B., & Serafini, L. (2006a). Verifying space and time requirements for resource-bounded agents. In P. Stone & G. Weiss (Eds.), *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2006)* (pp. 217–219). Hakodate, Japan: IEEE Press.

[Alechina et al., 2006b] Alechina, N., Jago, M., & Logan, B. (2006b). Modal Logics for Communicating Rule-Based Agents. *Frontiers In Artificial Intelligence And Applications*, 141, 322.

[Alechina et al., 2008a] Alechina, N., Logan, B., Nguyen, H. N., & Rakib, A. (2008a). Verifying resource requirements for distributed rule-based systems. In *RuleML* (pp. 31–38).

[Alechina et al., 2008b] Alechina, N., Logan, B., Nguyen, H. N., & Rakib, A. (2008b). Verifying time and communication costs of rule-based reasoners. In *MoChArt* (pp. 1–14).

[Alechina et al., 2008c] Alechina, N., Logan, B., Nguyen, H. N., & Rakib, A. (2008c). Verifying time, memory and communication bounds in systems of reasoning agents. In *AAMAS '08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems* (pp. 736–743). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

[Alechina et al., 2009a] Alechina, N., Logan, B., Nguyen, H. N., & Rakib, A. (2009a). Expressing properties of coalitional ability under resource bounds. In *LORI* (pp. 1–14).

[Alechina et al., 2009b] Alechina, N., Logan, B., Nguyen, H. N., & Rakib, A. (2009b). A logic for coalitions with bounded resources. In *IJCAI* (pp. 659–664).

[Alechina et al., 2009c] Alechina, N., Logan, B., Nguyen, H. N., & Rakib, A. (2009c). Reasoning about other agents' beliefs under bounded resources. In J.-J. Meyer & J. Broersen (Eds.), *Post-proceedings of KR2008-workshop on Knowledge Representation for Agents and Multi-Agent Systems (KRAMAS), Sydney, September 2008*, volume 5605 of *Lecture Notes in Artificial Intelligence* (pp. 1–15).: Springer.

[Alechina et al., 2010a] Alechina, N., Logan, B., Nguyen, H. N., & Rakib, A. (2010a). Resource-bounded alternating-time temporal logic. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1* (pp. 481–488).: International Foundation for Autonomous Agents and Multiagent Systems.

[Alechina et al., 2010b] Alechina, N., Logan, B., Nguyen, H. N., & Rakib, A. (2010b). Resource-bounded alternating-time temporal logic. In W. van der Hoek, G. Kaminka, Y. Lesperance, M. Luck, & S. Sen (Eds.), *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)* Toronto, Canada: IFAAMAS IFAAMAS. (to appear).

[Alur et al., 2002] Alur, R., Henzinger, T., & Kupferman, O. (2002). Alternating-time temporal logic. *Journal of the ACM (JACM)*, 49(5), 672–713.

[Blackburn et al., 2002] Blackburn, P., De Rijke, M., & Venema, Y. (2002). *Modal logic*. Cambridge Univ Pr.

[Bulling & Farwer, 2009] Bulling, N. & Farwer, B. (2009). RTL and RTL*: Expressing abilities of resource-bounded agents. In J. Dix, M. Fisher, & P. Novak (Eds.), *Proceedings 10th International Workshop Computational Logic in Multi-Agent Systems* (pp. 2–19).

[Clarke & Emerson, 1982] Clarke, E. & Emerson, E. (1982). Design and synthesis of synchronization skeletons using branching time temporal logic. *Logics of Programs*, (pp. 52–71).

[Clarke et al., 1986] Clarke, E. M., Emerson, E. A., & Sistla, A. P. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8, 244–263.

[Emerson, 1990] Emerson, E. (1990). Temporal and modal logic. *Handbook of theoretical computer science*, 8, 995–1072.

[Emerson & Halpern, 1982] Emerson, E. & Halpern, J. (1982). Decision procedures and expressiveness in the temporal logic of branching time. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing* (pp. 169–180).: ACM New York, NY, USA.

[Fagin et al., 1995] Fagin, R., Halpern, J. Y., Moses, Y., & Vardi, M. Y. (1995). *Reasoning about knowledge*. The MIT Press, Cambridge, Massachuse.

[Goranko, 2001] Goranko, V. (2001). Coalition games and alternating temporal logics. In *Proceedings of the 8th conference on Theoretical aspects of rationality and knowledge* (pp. 259–272).: Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

[Goranko & van Drimmelen, 2006] Goranko, V. & van Drimmelen, G. (2006). Complete axiomatization and decidability of Alternating-time temporal logic. *Theoretical Computer Science*, 353(1-3), 93–117.

[Hansen & Pauly, 2002] Hansen, H. & Pauly, M. (2002). Axiomatising Nash-consistent coalition logic. *Logics in Artificial Intelligence*, (pp. 394–406).

[Hintikka, 1962] Hintikka, J. (1962). *Knowledge and belief: an introduction to the logic of the two notions*. Cornell University Press.

[Hintikka, 1978] Hintikka, J. (1978). Impossible possible worlds vindicated. *Game-Theoretical Semantics*, (pp. 367–379).

[Jamroga & van der Hoek, 2004] Jamroga, W. & van der Hoek, W. (2004). Agents that know how to play. *Fundamenta Informaticae*, 63(2), 185–219.

[Katoen, 1998] Katoen, J.-P. (1998). *Concepts, Algorithms, and Tools for Model Checking*. Friedrich-Alexander Universität Erlangen-Nürnberg: Lecture Notes.

[Nguyen, 2010] Nguyen, H. N. (2010). An extension of RB-ATL. In *Proceedings of the Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW 2010)*, volume CEUR-WS 627 Lyon, France.

[Pauly, 2001] Pauly, M. (2001). *Logic for social software*. Institute for Logic, Language and Computation, Universiteit van Amsterdam.

[Pauly, 2002] Pauly, M. (2002). A Modal Logic for Coalitional Power in Games. *Journal of Logic and Computation*, 12(1), 149–166.

[Schobbens, 2004] Schobbens, P. (2004). Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2), 82–93.

[van der Hoek & Wooldridge, 2003] van der Hoek, W. & Wooldridge, M. (2003). Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75(1), 125–157.

[Vardi et al., 1989] Vardi, M., Res, I., & San Jose, C. (1989). On the complexity of epistemic reasoning. In *Logic in Computer Science, 1989. LICS'89, Proceedings., Fourth Annual Symposium on* (pp. 243–252).

[Wooldridge et al., 2007] Wooldridge, M., Agotnes, T., Dunne, P., & van der Hoek, W. (2007). Logic for automated mechanism design-a progress report. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 22: Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[Wooldridge & Dunne, 2006] Wooldridge, M. & Dunne, P. (2006). On the computational complexity of coalitional resource games. *Artificial Intelligence*, 170(10), 835–871.