A Framework for Coalitional Normative Systems

Jun Wu[1,2], Chongjun Wang[1]* and Junyuan Xie[1]
[1]National Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210093, China
[2]College of Computer & Information Engineering, Hohai University, Nanjing 210098, China
wujun@iip.nju.edu.cn,{chjwang, jyxie}@nju.edu.cn

ABSTRACT

We propose coalitional normative system (CNS), which can selectively restrict the joint behavior of a coalition, in this paper. We extend the semantics of ATL and propose *Coordinated* ATL (CO-ATL) to support the formalizing of CNS. We soundly and completely characterize the limitation of the normative power of a coalition by identifying two fragments of CO-ATL language corresponding to two types of system properties that are unchangeable by restricting the joint behavior of such a coalition. Then, we prove that the effectiveness checking, feasibility and synthesis problems of CNS are PTIME-complete, NP-complete and FNP-complete, respectively. Moreover, we define two concepts of optimality for CNS, that is, minimality and compactness, and prove that both minimality checking and compactness checking are co-NP-complete while the problem of checking whether a coalition is a minimal controllable coalition is DP-complete. The relation between NS and CNS is discussed, and it turns out that NSs intrinsically consists of a proper subset of CNSs and some basic problems related to CNS are no more complex than that of NS.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory

Keywords

normative systems, logic, model checking, complexity

1. INTRODUCTION

Normative system (NS) (or *social law*) was firstly proposed by [12, 13] as an off-line approach for coordinating multiagent systems, and then extended by, e.g., [17, 14, 1], based on introducing the formalisms of modal and temporal logics, especially *Alternating-time Temporal Logic* (ATL) [4, 5] and its variations, which can be used for the specification and verification of mechanisms such as social choice procedures. So work on this aspect is also considered as a part of the *logics for automated mechanism design* research [10, 16].

Although the various approaches to NSs proposed in the literature differ on technical details, they all share the same basic intuition that an NS is a set of constraints on the behavior of agents; by imposing these constraints, it is hoped that some desirable objectives will emerge [3], corresponding to a logic formula that is originally false to become true, or the reverse. The idea is that the imposing of an NS will lead to certain updating in the semantic model, and thus cause changes in the interpretations of some formulas.

But we find that NSs update the semantic model in a somewhat too coarse way, that is, when an action is forbidden in a state, all related transitions from this state are deleted [1]. This means the task of deleting a certain set of prescribed transitions, which corresponds to the necessary condition for fulfilling a certain objective, may exceed the abilities of all NSs. To overcome this shortcoming, we propose coalitional normative system (CNS), which is a set of behavioral constraints for a coalition (*i.e.*, agent set) that restrict its *joint actions*. By adopting a CNS, we can restrict the set of transitions to an arbitrary subset of it, thus we can achieve all possible updating in the semantic model.

Intuitively, the coalition represents a system we can control (it is a distributed open system formed by several agents); and the CNS specifies in every state for the coalition which sets of actions (that can be chosen by it) cannot be executed simultaneously, thus should be forbidden. We assume that the agents in the coalition will negotiate with each other before making any decisions on action selection in order to avoid adopting any joint actions that are forbidden by the CNS. So, compared with conventional NS, CNS can more effectively capture the overall effects of joint actions and prevent the destructive interactions from taking place. In this paper, we aim to present a framework for CNS based on ATL, and study its related reasoning and computational problems.

The remainder of this paper is structured as follows. We begin by introducing the basics of ATL and NS. Next, as the effects of CNSs cannot be captured by ATL directly, we propose CO-ATL to support the formalizing of CNS. Then for each coalition C, by identifying the $\mathcal{L}_C^+$ and $\mathcal{L}_C^-$ fragments of the CO-ATL language, we *"soundly and completely"* characterize its limitation of normative power. Afterward, we

*Chongjun Wang is the corresponding author.

Cite as: A Framework for Coalitional Normative Systems, Jun Wu, Chongjun Wang and Junyuan Xie, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 259-266.

[1]When concurrent action models are adopted, an action of an agent in a state may be related to several transitions.

establish the computational complexity of some key problems related to CNSs. Finally, we present some conclusions.

2. ATL AND NORMATIVE SYSTEMS

2.1 Alternating-time Temporal Logic

The syntax of *Alternating-time Temporal Logic* (ATL) [4, 5] is an extension of the syntax of CTL via replacing the path quantifiers $\forall$ and $\exists$ with the path quantifier $\langle\!\langle\rangle\!\rangle$, which can express the α-ability [15] in game theory. And the formulas of ATL are interpreted by *concurrent game structures* (CGSs).

A concurrent game structure is a tuple $S = \langle k, Q, \Pi, \pi, d, \delta\rangle$ with the following components:

- A natural number $k \geq 1$ of agents. We identify the agents with the numbers (or IDs) $1, ..., k$.
- A finite set Q of states.
- A finite set Π of propositions.
- For each state $q \in Q$, a set $\pi(q) \subseteq \Pi$ of propositions true at q. The function π is called labeling function.
- For each agent $a \in \{1, ..., k\}$ and each state $q \in Q$, a natural number $d_a(q) \geq 1$ of actions available to agent a at state q. We identify the actions of agent a at state q with the numbers $1, ..., d_a(q)$. A *joint action* of all the agents at state q is a tuple $\langle j_1, ..., j_k\rangle$ such that $1 \leq j_a \leq d_a(q)$ for each agent a. We write $D(q)$ for the set $\{1,...,d_1(q)\} \times ... \times \{1, ..., d_k(q)\}$ of joint actions. The function D is called move function.
- For each state $q \in Q$ and each joint action $\langle j_1, ..., j_k\rangle \in D(q)$, a state $\delta(q, j_1, ..., j_k) \in Q$ will result from state q if every agent $a \in \{1, ..., k\}$ chooses action j_a. The function δ is called transition function.

Note that, every agent set $A \subseteq \{1, ..., k\}$ can be seen as a coalition (with the agent set $\{1, ..., k\} \setminus A$ represents the environment). The *grand coalition* $\{1, ..., k\}$ is denoted as Ag. In the following of this paper, we will sometimes use $\vec{m}$ to refer to a joint action $\langle j_1, ..., j_k\rangle$ (of all the agents), use $\vec{m}_A$ to refer to a joint action of the coalition $A \subset Ag$[2] (called an *A-action*) and use $D_A(q)$ to refer to all the possible A-actions at the state q. Moreover, we introduce the notation $\vec{m}_A|A'$ to mean the joint action of the agent set $A \cap A'$ when the agent set A takes the joint action $\vec{m}_A$.

Some important concepts with respect to concurrent game structures are specified as follows: For two states q and q', q' is called a *successor* of q if there is a joint action $\vec{m} \in D(q)$ such that $q' = \delta(q, \vec{m})$. A *computation* of S is an infinite sequence $\lambda = q_0, q_1, q_2, ...$ of states such that for all positions $i \geq 0$, the state q_{i+1} is a successor of the state q_i. We refer to a computation starting from state q as a q-computation. For a computation λ and a position $i \geq 0$, we use $\lambda[i], \lambda[0, i]$,and $\lambda[i, \infty]$ to denote, respectively, the ith state of λ, the finite prefix $q_0, q_1, ..., q_i$ of λ, and the infinite suffix $q_i, q_{i+1}, ...$ of λ. A *strategy* for agent $a \in \Sigma$ is a function f_a that maps every nonempty finite state sequence $\lambda \in Q^+$ to an action such that if the last state of λ is q, then $f_a(\lambda) \in \{1, ..., d_a(q)\}$.

[2]For the joint actions of an arbitrary agent set $A \subseteq Ag$, we always consider them as action vectors arranged in order of increasing IDs of the corresponding agents in A, instead of action sets.

And, the *outcomes* of a set of strategies F_A, called an *A-strategy*, one for each agent in $A \subseteq Ag$, from a state $q \in Q$ is the set $out(q, F_A)$ of computations, such that a computation $\lambda = q_0, q_1, q_2, ...$ is in $out(q, F_A)$ if $q_0 = q$ and there is a joint action $\langle j_1, ..., j_k\rangle \in D(q_i)$ such that (1) $j_a = f_a(\lambda[0, i])$ for all agents $a \in A$, and (2) $\delta(q_i, j_1, ..., j_k) = q_{i+1}$.

The language of ATL $\mathcal{L}$ is generated by the following grammar:

$$\varphi ::= p|\neg\varphi|\varphi_1 \vee \varphi_2|\langle\!\langle A\rangle\!\rangle\bigcirc\varphi|\langle\!\langle A\rangle\!\rangle \square \varphi|\langle\!\langle A\rangle\!\rangle\varphi_1\mathcal{U}\varphi_2,$$

where $p \in \Pi$ is a proposition, and $A \subseteq Ag$ is a set of agents[3].

As an abbreviation, we write $\langle\!\langle A\rangle\!\rangle \diamond \varphi$ for $\langle\!\langle A\rangle\!\rangle\top\mathcal{U}\varphi$.

We write $S, q \vDash \varphi$ to indicate that the formula φ holds at state q of a CGS S. When S is clear from the context, we write $q \vDash \varphi$. The relation $\vDash$ is defined, for all states q of S, inductively as follows:

- For all $p \in \Pi$ we have $q \vDash p$ iff $p \in \pi(q)$.
- $q \vDash \neg\varphi$ iff $q \nvDash \varphi$.
- $q \vDash \varphi_1 \vee \varphi_2$ iff $q \vDash \varphi_1$ or $q \vDash \varphi_2$.
- $q \vDash \langle\!\langle A\rangle\!\rangle\bigcirc\varphi$ iff there exists a A-strategy, F_A, such that for all computations $\lambda \in out(q, F_A)$ we have $\lambda[1] \vDash \varphi$.
- $q \vDash \langle\!\langle A\rangle\!\rangle \square \varphi$ iff there exists a A-strategy, F_A, such that for all computations $\lambda \in out(q, F_A)$ and all positions $i \geq 0$, we have $\lambda[i] \vDash \varphi$.
- $q \vDash \langle\!\langle A\rangle\!\rangle\varphi_1\mathcal{U}\varphi_2$ iff there exists a A-strategy, F_A, such that for all computations $\lambda \in out(q, F_A)$ there exists a position $i \geq 0$ such that $\lambda[i] \vDash \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \vDash \varphi_1$.

2.2 Normative Systems

Given a concurrent game structure $S = \langle k, Q, \Pi, \pi, d, \delta\rangle$, a normative system (NS) is a function η such that

$$\eta(a, q) \subset \{1, ..., d_a(q)\}$$

for all agents $a \in Ag$ and states $q \in Q$.

Intuitively, $\eta(a, q)$ is the set of "*forbidden*" (or "*illegal*") actions for agent a in state q. The structure obtained from a CGS S by *implementing* an NS η, denoted as $S\dagger\eta$, is the structure obtained from S by deleting all the forbidden actions. Note that, NS is defined as a *proper subset* of all the available actions to guarantee every agent will has at least one available actions in every state after having implemented an NS. Apparently, $S\dagger\eta$ is still a concurrent game structure.

An existential and a universal sublanguage of ATL, denoted $\mathcal{L}^e$ and $\mathcal{L}^u$, respectively, were defined in [14] by the following grammars ϵ and υ respectively:

$$\epsilon ::= p|\epsilon \wedge \epsilon|\epsilon \vee \epsilon|\langle\!\langle Ag\rangle\!\rangle\bigcirc\epsilon|\langle\!\langle Ag\rangle\!\rangle \square \epsilon|\langle\!\langle Ag\rangle\!\rangle\epsilon\mathcal{U}\epsilon$$

$$\upsilon ::= p|\upsilon \wedge \upsilon|\upsilon \vee \upsilon|\langle\!\langle\rangle\!\rangle\bigcirc\upsilon|\langle\!\langle\rangle\!\rangle \square \upsilon|\langle\!\langle\rangle\!\rangle\upsilon\mathcal{U}\upsilon$$

where $p \in \Pi$.

Suppose we have a CGS S, an NS η, a state q in S, and formulas $\epsilon \in \mathcal{L}^e$, $\upsilon \in \mathcal{L}^u$. Then,

[3]We always assume that we are studying a fixed set Ag of agents and a fixed set Π of propositions. So, the language of ATL is a fixed set of formulas, and when we refer to "concurrent game structure" we actually mean a concurrent game structure with $|Ag|$ and Π as its components.

1. $S\dagger\eta, q \models \epsilon \Rightarrow S, q \models \epsilon$;

2. $S, q \models \upsilon \Rightarrow S\dagger\eta, q \models \upsilon$.

The first result tells us that the satisfaction of a $\mathcal{L}^e$ formula cannot be *established* by implementing a NS[4]. The second result, in contrast, tells us that the satisfaction of a $\mathcal{L}^u$ formula cannot be *avoided* by implementing a NS.

3. COALITIONAL NORMATIVE SYSTEMS

3.1 The Formal Framework

A coalitional normative system (CNS) for a concurrent game structure $S = \langle k, Q, \Pi, \pi, d, \delta\rangle$ is a tuple $\Gamma = \langle C, \vartheta\rangle$ with the following components:

- A coalition $C \subseteq \{1, ..., k\}$.
- For each state $q \in Q$, a set $\vartheta(q) \subset D_C(q)$ of C-actions the agents in coalition C cannot collaboratively choose. The function ϑ is called coordination function.

Sometimes we call a CNS $\Gamma = \langle C, \vartheta\rangle$ as a *C-norm*. When CNSs are taken into consideration, certain joint action choices and computations will be ruled out. As in [17], we adopt the prefix "Γ-*conformant*" to mean "*permitted by* Γ":

- A joint action $\vec{m} \in D(q)$ is called a Γ-*conformant joint action* iff $\nexists \vec{m}_C \in \vartheta(q)$ such that $\vec{m}|C = \vec{m}_C$.
- A state q' is called a Γ-*conformant successor* of state q if there is a Γ-conformant joint action $\vec{m} \in D(q)$ such that $q' = \delta(q, \vec{m})$.
- A Γ-*conformant computation* of S is an infinite sequence $\lambda = q_0, q_1, q_2, ...$ of states such that for all positions $i \geq 0$, the state q_{i+1} is a Γ-conformant successor of the state q_i.
- In each state $q \in Q$, an A-action $\vec{m}_A$ for the agent set $A \subseteq Ag$, is called a Γ-*conformant A-action* in q iff $\exists \vec{m}_C \notin \vartheta(q)$ such that $\vec{m}_A|A \cap C = \vec{m}_C|A \cap C$.
- A set $F_A = \{f_a | a \in A\}$ of strategies, one for each agent in A, is called a Γ-*conformant A-strategy* iff for all nonempty finite state sequences $\lambda \in Q^+$, the A-action $\vec{m}_A$, given by F_A, is a Γ-conformant A-action.
- Finally, the Γ-*conformant outcomes* of a Γ-conformant A-strategy from a state $q \in Q$ is the set $out_\Gamma(q, F_A)$, such that, a Γ-conformant computation $\lambda = q_0, q_1, q_2, ...$ is in $out_\Gamma(q, F_A)$ if $q_0 = q$ and there is a Γ-conformant joint action $\vec{m} \in D(q_i)$ such that (1) $j_a = f_a(\lambda[0, i])$ for all players $a \in A$, and (2) $\delta(q_i, \vec{m}) = q_{i+1}$.

Similarly, we can define the structure obtained from S by implementing a CNS $\Gamma = \langle C, \vartheta\rangle$, denoted as $S\dagger\Gamma$, as the structure obtained from S by deleting all the joint actions forbidden by Γ. Notice that, in most cases $S\dagger\Gamma$ is not an ordinary concurrent game structure any longer – agents in C will "*discuss*" in advance on which C-action should be selected, so a kind of *"coalitional coordination"* is explicitly represented in the structure.

[4]As result 1 is equivalent to $S, q \models \neg\epsilon \Rightarrow S\dagger\eta, q \models \neg\epsilon$.

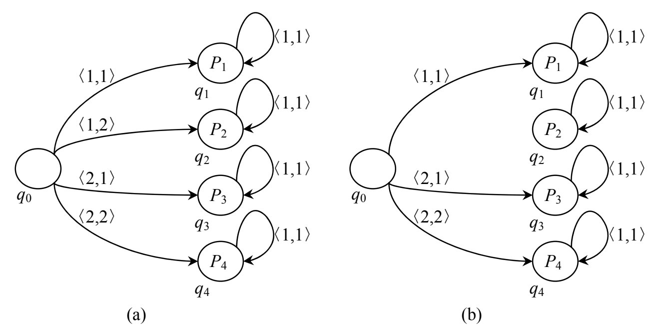

Figure 1: Implementing a CNS

CNS vs. NS: Coalitional normative system extends the concept of normative system by enabling selective restricting a coalition's joint behavior. It is not difficult to discover that every NS is intrinsically a special CNS. Suppose that an action i of agent a is ruled out by an NS η, it actually means all the joint actions of the grand coalition that adopting i as a member are ruled out. So, for an arbitrary NS η we can always find an *equivalent* CNS $\Gamma_\eta = \langle Ag, \vartheta_\eta\rangle$ for the grand coalition, just let $\forall q \in Q : \vartheta_\eta(q) = \eta(1, q) \times ... \times \eta(k, q)$. That is, the following result hold.

PROPOSITION 1. *Given a* CGS S. *For every* NS η, *there exists a* CNS Γ *such that* $S\dagger\eta = S\dagger\Gamma$.

But apparently there are some CNSs without equivalent NS. This means NS can be seen as a proper subset of CNS. Notice that although we can modify the original CGS in more ways by using CNSs, the resulting structure remains a CGS if and only if the CNS has an equivalent NS.

EXAMPLE 1. *Consider a* CGS S *depicted as Figure 1(a), that is,* $S = \langle k, Q, \Pi, \pi, d, \delta\rangle$ *where* $k = 2$; $Q = \{q_0, ..., q_4\}$; $\Pi = \{P_1, ..., P_4\}$; $\pi(q_i) = P_i$ *for all* $1 \leq i \leq 4$; $d_1(q_0) = d_2(q_0) = 2$; $d_1(q_i) = d_2(q_i) = 1$ *for all* $1 \leq i \leq 4$; *and* $\delta(q_0, 1, 1) = q_1$; $\delta(q_0, 1, 2) = q_2$; $\delta(q_0, 2, 1) = q_3$; $\delta(q_0, 2, 2) = q_4$; $\delta(q_i, 1, 1) = q_i$ *for all* $1 \leq i \leq 4$. *The following statements hold:*

1. *An* NS η *such that* $\eta(1, q_0) = \{1\}$; $\eta(2, q_0) = \{2\}$; $\eta(i, q_j) = \varnothing$ *for all* $i \in \{1, 2\}$ *and* $j \in \{1, ..., 4\}$ *has the same effect with the* CNS $\Gamma = \{C, \vartheta\}$ *where* $C = \{1, 2\}$; *and* $\vartheta(q_0) = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 2\rangle\}$; $\vartheta(q_i) = \varnothing$ *for all* $i \in \{1, ..., 4\}$.

2. *A* CNS $\Gamma' = \langle C', \vartheta'\rangle$, *where* $C' = \{1, 2\}$; *and* $\vartheta'(q_0) = \{\langle 1, 2\rangle\}$, $\vartheta'(q_i) = \varnothing$ *for all* $i \in \{1, ..., 4\}$, *can transform* S *to the structure depicted as Figure 1(b), but there doesn't exist any* NS *which can achieve this transformation.*

3. *The structure depicted as Figure 1(b)cannot be modeled by any* CGS.

3.2 Coordinated ATL

Consequently, in the presence of a CNS, we need to refine the interpretation for the ATL formulas. We call this new logic *Coordinated* ATL (CO-ATL), which directly inherits the ATL syntax but assumes slightly different semantics.

CO-ATL Semantics: We write $S, \Gamma, q \models \varphi$ to indicate that the formula φ holds at state q of a concurrent game structure S under the CNS Γ. When S and Γ is clear from the context, we write $q \models \varphi$. The relation $\models$ is defined, for all states q of S, inductively as follows:

- For all $p \in \Pi$, we have $q \models p$ iff $p \in \pi(q)$.
- $q \models \neg\varphi$ iff $q \not\models \varphi$.
- $q \models \varphi_1 \vee \varphi_2$ iff $q \models \varphi_1$ or $q \models \varphi_2$.
- $q \models \langle\!\langle A \rangle\!\rangle \bigcirc \varphi$ iff there exists a Γ-conformant A-strategy, F_A, such that for all Γ-conformant computations $\lambda \in out_\Gamma(q, F_A)$ we have $\lambda[1] \models \varphi$.
- $q \models \langle\!\langle A \rangle\!\rangle \Box \varphi$ iff there exists a Γ-conformant A-strategy, F_A, such that for all Γ-conformant computations $\lambda \in out_\Gamma(q, F_A)$ and all positions $i \geq 0$, we have $\lambda[i] \models \varphi$.
- $q \models \langle\!\langle A \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2$ iff there exists a Γ-conformant A-strategy, F_A, such that for all Γ-conformant computations $\lambda \in out_\Gamma(q, F_A)$ there exists a position $i \geq 0$ such that $\lambda[i] \models \varphi_2$ and for all positions $0 \leq j < i$ we have $\lambda[j] \models \varphi_1$.

Model Checking CO-ATL: Because of its significance both in theory and practice, model checking is an important computational problem for any modal or temporal logic. The model checking problem for CO-ATL is defined as follows.

CO-ATL MODEL CHECKING:
Given: CGS $S = \langle k, Q, \Pi, \pi, d, \delta \rangle$, CNS $\Gamma = \langle C, \vartheta \rangle$,
state $q \in Q$, and CO-ATL formula φ.
Question: Is $S, \Gamma, q \models \varphi$?

The lower bound of CO-ATL MODEL CHECKING is trivial, for we can reduce the ATL model checking problem, which is PTIME-complete [5], to it in polynomial time.

To show the upper bound, we can find a polynomial time algorithm for this problem. The algorithm is obtained from the ATL model checking algorithm proposed in [5] by rewriting the *Pre* function which given a set $A \subseteq \Sigma$ of agents and a set $\rho \subseteq Q$ of states, returns the set of states q such that form q the agents in A can cooperate and enforce the next state to lie in ρ. Formally, in the algorithm for CO-ATL, $Pre(A, \rho)$ contains state $q \in Q$ if there exists a Γ-conformant A-action $\vec{m_A}$ in q such that for all Γ-conformant joint actions $\vec{m}$ in q satisfying $\vec{m}|A = \vec{m}_A$, we have $\delta(q, \vec{m}) \in \rho$. Finally, we can prove that this algorithm for CO-ATL is still a polynomial time algorithm.

THEOREM 2. CO-ATL MODEL CHECKING *is* PTIME-*complete, and can be solved in time $\mathcal{O}(m \cdot l)$ for a* CGS *with m transitions and a* CO-ATL *formula φ of length l. The problem is* PTIME-*hard even for a fixed formula.*

PROOF. We follow the steps of the proof for the ATL model checking complexity given by Alur *et al.* [5]. We reduce games played on CGSs with the constraint of a CNS to games played on turn-based synchronous game structures. The only difference is in building the corresponding 2-player turn-based synchronous game structure $S_A = \langle 2, Q_A, \Pi_A, \pi_A, \sigma_A, R_A \rangle$ with respect to a CGS S, an NS Γ, and a set of agents(players) $A \in \Sigma$. The components in S_A are defined as usual, but we have to redefine some related basic concepts based on the semantics of CO-ATL: For a state $q \in Q$, an *A-move* c at q is a Γ-conformant A-action defined in this paper; An state $q' \in Q$ is a *c-successor* of q if there is a Γ-conformant joint action $\vec{m}$ such that (1) $\vec{m}|A = c$, and (2) $q' = \delta(q, \vec{m})$. It is easy to see that the aforementioned changes add no additional complexity to the structure of S_A, that is, if the original game structure S has m transitions, the turn-based synchronous structure S_C has $\mathcal{O}(m)$ states and transitions. This means we can find an algorithm for CO-ATL MODEL CHECKING which requires time $\mathcal{O}(m \cdot l)$. □

So, with respect to the computational complexity of CO-ATL model checking we get the same result with that of ATL model checking.

CO-ATL vs. ATL: By adopting the empty CNS $\Gamma^C_\varnothing = \langle C, \vartheta_\varnothing \rangle$ where $\vartheta_\varnothing(q) = \varnothing$ for all $q \in Q$, we can identify the following relation between CO-ATL and ATL.

PROPOSITION 3. *Given a* CGS $S = \langle k, Q, \Pi, \pi, d, \delta \rangle$, *then for all $q \in Q$, $C \subseteq Ag$ and $\varphi \in \mathcal{L}$, we have*

$$S, \Gamma^C_\varnothing, q \models \varphi \Leftrightarrow S, q \models_{ATL} \varphi.$$

Remark that, compared with ATL, CO-ATL can represent and reason about α-abilities in a wider range of structures, that is, the class of structures obtained from concurrent game structures by implementing CNSs[5]. To differentiate between the two semantics, we use "$\models_{ATL}$" to denote the satisfaction relation in ATL.

3.3 Objectives and Effectiveness

We define the concept of objective for expressing the aim of the designer in CNS synthesis. Formally, an *atomic objective* is a state-formula pair, e.g. $\langle q, \varphi \rangle$, indicating that in state q the CO-ATL formula φ should be satisfied; Then *objectives* are generated by the following grammar o:

$$o ::= \langle q, \varphi \rangle | \neg o | o_1 \wedge o_2 | o_1 \vee o_2$$

where q is a state and φ is a CO-ATL formula. That is, an objective is a Boolean combination of atomic objectives which can express complex requirements about the system.

We adopt the expression "$S \dagger \Gamma \rightsquigarrow o$" to mean the CNS Γ is *effective* for the objective o (in the CGS S), where the relation "$\rightsquigarrow$" is inductively defined as follows:

- $S \dagger \Gamma \rightsquigarrow \langle q, \varphi \rangle$ iff $S, \Gamma, q \models \varphi$;
- $S \dagger \Gamma \rightsquigarrow \neg o$ iff *not* $S \dagger \Gamma \rightsquigarrow o$;
- $S \dagger \Gamma \rightsquigarrow o_1 \wedge o_2$ iff $S \dagger \Gamma \rightsquigarrow o_1$ *and* $S \dagger \Gamma \rightsquigarrow o_2$;
- $S \dagger \Gamma \rightsquigarrow o_1 \vee o_2$ iff $S \dagger \Gamma \rightsquigarrow o_1$ *or* $S \dagger \Gamma \rightsquigarrow o_2$.

Intuitively, an effective CNS is a CNS that can fulfill the aim of the CNS designer, which is expressed as an objective.

3.4 Concepts of Optimality

Usually, there might be more than one effective CNSs for an objective. So it would be helpful if we can define some concepts of optimality for selecting among effective CNSs.

Minimality: The idea of minimality was firstly proposed in [6, 7], attempting to minimize the amount of constraints

[5]All concurrent game structures are in this class, because an arbitrary concurrent game structure S is also the structure obtained from S by implementing an empty CNS.

set on the agents and as such, capture the notion of maximal individual flexibility. We are going to transplant this idea to coalitional normative systems.

Given two CNSs $\Gamma_1 = \langle C, \vartheta_1 \rangle$ and $\Gamma_1 = \langle C, \vartheta_2 \rangle$. Γ_1 is said to be *less restrictive* than Γ_2, denoted as $\Gamma_1 \preccurlyeq \Gamma_2$, if and only if $\forall q \in Q\ \vartheta_1(q) \subseteq \vartheta_2(q)$. Γ_1 is said to be *strictly less restrictive* than Γ_2, denoted as $\Gamma_1 \prec \Gamma_2$, if and only if $\Gamma_1 \preccurlyeq \Gamma_2$ and $\exists q \in Q\ \vartheta_1(q) \subset \vartheta_2(q)$. And Γ is a *minimal* CNS if and only if $\nexists \Gamma'$ such that Γ' is effective and $\Gamma' \prec \Gamma$.

So a minimal CNS is one of the effective CNSs that put the least amount of constraints on the coalition.

Compactness: Notice that, in our framework the agents in the coalition have to negotiate with each other before selecting any joint actions. Basically, this process requires agents in the coalition sending messages to each other until an agreement on action selection has been reached. Obviously, more agents in the coalition means more complex the process is and more prone to cause error.

In this sense, with respect to an objective o if both $\Gamma = \langle C, \vartheta \rangle$ and $\Gamma' = \langle C', \vartheta' \rangle$ are effective coalitional normative systems, Γ is better than Γ' if C is a proper subset of C' (that is $C \subset C'$) – we say Γ is more *compact* than Γ'. An CNS $\Gamma = \langle C, \vartheta \rangle$ is a *compact* CNS if and only if it is an effective CNS and there doesn't exist any effective CNS Γ' which is more compact than Γ, and in this case, we say C is a *minimal controllable coalition*.

In other words, the key idea of compactness is minimizing the amount of agents in the coalition in order to minimize the communication cost in the system.

4. COALITIONAL NORMATIVE POWER AND ITS LIMITATION

Intuitively, the *normative power* of a coalition C is manifested by its ability of changing properties in CGSs by implementing C-norms. But very naturally the class of C-norms is not omnipotent, for we can show that some properties are inevitably beyond the reach of all the C-norms. In other words, the normative power of a coalition has its *limitations*. It is interesting to show what exactly the limitation is.

Power Limitation Characterization: For all formulas $\varphi \in \mathcal{L}$, we say *φ's satisfaction cannot be established by (implementing) a C-norm* if and only if for all CNSs S, there doesn't exists any C-norm Γ, such that there is a $q \in Q$ satisfying $S, \Gamma^C_\varnothing, q \nvDash \varphi$ and $S, \Gamma, q \vDash \varphi$; we say *$\varphi$'s satisfaction cannot be avoided by (implementing) a C-norm* if and only if for all CNSs S, there doesn't exists any C-norm Γ, such that there is a $q \in Q$ satisfying $S, \Gamma^C_\varnothing, q \vDash \varphi$ and $S, \Gamma, q \nvDash \varphi$. Then the power limitation of the class of C-norm can be characterized by answering the following two questions:

1. *which fragment of $\mathcal{L}$ is the set of formulas whose satisfaction cannot be established by a C-norm*?

2. *which fragment of $\mathcal{L}$ is the set of formulas whose satisfaction cannot be avoided by a C-norm*?

We then define two fragments of the CO-ATL language, $\mathcal{L}^+_C$ and $\mathcal{L}^-_C$, which are generated by the grammars φ and ψ below respectively.

$$\varphi ::= p | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | \langle\!\langle C^+ \rangle\!\rangle \bigcirc \varphi | \langle\!\langle C^+ \rangle\!\rangle \Box \varphi | \langle\!\langle C^+ \rangle\!\rangle \varphi_1 \mathcal{U} \varphi_2 | \neg \psi$$

$$\psi ::= p | \psi_1 \wedge \psi_2 | \psi_1 \vee \psi_2 | \langle\!\langle C^- \rangle\!\rangle \bigcirc \psi | \langle\!\langle C^- \rangle\!\rangle \Box \psi | \langle\!\langle C^- \rangle\!\rangle \psi_1 \mathcal{U} \psi_2 | \neg \varphi$$

where $p \in \Pi$, $C \subseteq C^+ \subseteq Ag$, $\varnothing \subseteq C^- \subseteq Ag \setminus C$.

In the following, we will show that $\mathcal{L}^+_C$ and $\mathcal{L}^-_C$ are exactly the answers to the above two questions respectively. That is, by $\mathcal{L}^+_C$ and $\mathcal{L}^-_C$ we can *soundly* and *completely* characterize the the limitation of the normative power of coalition C.

Soundness and Completeness: First of all, we prove the following two lemmas, which implies that a C-norm cannot add any thing new to the strategic ability of a coalition that consists of a superset of C, and cannot avoid any strategic ability of a coalition that consists of a subset of $Ag \setminus C$.

LEMMA 4. *Given an arbitrary CGS S, a state q_0 in S and an arbitrary C-norm Γ. If C^+ is a set of agents satisfying $C \subseteq C^+ \subseteq Ag$ and F_{C^+} is a Γ-conformant C^+-strategy, then there is a C^+-strategy F'_{C^+} such that*

$$out(q_0, F'_{C^+}) = out_\Gamma(q_0, F_{C^+}).$$

PROOF. Always, we can define $F'_{C^+} = F_{C^+}$. And in all states q, after the agents in C^+ selected a C^+-action, the available joint actions and available Γ-conformant joint actions for the agents in $Ag \setminus C^+$ are the same, that is, all the joint actions in $D_{Ag \setminus C^+}(q)$, as Γ put no constraint on the agents in $Ag \setminus C^+$. $\square$

LEMMA 5. *Given a CGS S, a state q_0 in S and an arbitrary C-norm Γ. If C^- is a set of agents satisfying $\varnothing \subseteq C^- \subseteq Ag \setminus C$ and F_{C^-} is a C^--strategy, then there is an Γ-conformant C^--strategy F'_{C^-} such that*

$$out_\Gamma(q_0, F'_{C^-}) \subseteq out(q_0, F_{C^-}).$$

PROOF. As Γ actually cannot put any constraints on the behavior of the agents in C^-, all the C^--strategies are Γ-conformant joint strategies for C^-. So, we can define $F'_{C^-} = F_{C^-}$. And in all states q, for the agents in $Ag \setminus C^-$, the set of Γ-conformant joint actions is a subset of the joint actions, because of the effect of Γ. $\square$

Then, soundness of our characterization can be established by the following theorem.

THEOREM 6. *Given an arbitrary CGS S, an arbitrary C-norm Γ and an arbitrary state q in S. Then*

1. *$\forall \varphi \in \mathcal{L}^+_C$, we have $S, \Gamma, q \vDash \varphi \Rightarrow S, \Gamma^C_\varnothing, q \vDash \varphi$.*

2. *$\forall \psi \in \mathcal{L}^-_C$, we have $S, \Gamma^C_\varnothing, q \vDash \psi \Rightarrow S, \Gamma, q \vDash \psi$.*

PROOF. By induction on the structure of φ and ψ. For the case of propositions the conclusion trivially hold. For the other cases, suppose for all $\varphi \in \mathcal{L}^+_C$ and $\psi \in \mathcal{L}^-_C$ the conclusion holds. Then the satisfaction for the cases of $\varphi_1 \wedge \varphi_2$, $\psi_1 \wedge \psi_2$, $\varphi_1 \vee \varphi_2$, $\psi_1 \vee \psi_2$, $\neg\varphi$ and $\neg\psi$ are immediate.

Moreover, for all $C \subseteq C^+ \subseteq Ag$ and $\varnothing \subseteq C^- \subseteq Ag \setminus C$:

$S, \Gamma, q \vDash \langle\!\langle C^+ \rangle\!\rangle \bigcirc \varphi \Rightarrow$ (by the CO-ATL semantics) there is a Γ-conformant C^+-strategy F_{C^+} such that for all Γ-conformant q-computations $\lambda \in out_\Gamma(q, F_{C^+})$, $S, \Gamma, \lambda[1] \vDash \varphi \Rightarrow$ (by the induction hypothesis) for all $\lambda \in out_\Gamma(q, F_{C^+})$, $S, \Gamma^C_\varnothing, \lambda[1] \vDash \varphi \Rightarrow$ (by *lemma 4* and *proposition* 3) there is a F'_{C^+} such that for all $\lambda' \in out(q, F'_{C^+})$ we have $S, \lambda'[1] \vDash_{ATL} \varphi \Rightarrow$ (by the ATL semantics and *proposition* 3) $S, \Gamma^C_\varnothing, q \vDash \langle\!\langle C^+ \rangle\!\rangle \bigcirc \varphi$.

$S, \Gamma^C_\varnothing, q \vDash \langle\!\langle C^- \rangle\!\rangle \bigcirc \psi \Rightarrow$ (by *proposition* 3 and the ATL semantics) there is a C^--strategy F_{C^-} such that for all $\lambda \in$

$out(q, F_{C^-})$, $S, \lambda[1] \vDash_{ATL} \psi \Rightarrow$ (by *lemma 5*) there is a Γ-conformant C^--strategy F'_{C^-} for all $\lambda' \in out_\Gamma(q, F'_{C^-})$ we have $S, \lambda'[1] \vDash_{ATL} \psi \Rightarrow$ (by *proposition* 3 and the induction hypothesis) for all $\lambda' \in out_\Gamma(q, F'_{C^-})$ we have $S, \Gamma, \lambda'[1] \vDash \psi$ $\Rightarrow$ (by the CO-ATL semantics) $S, \Gamma, q \vDash \langle\langle C^- \rangle\rangle \bigcirc \psi$.

Analogously, we can prove that the conclusion hold for $\langle\langle C^+ \rangle\rangle \Box \varphi$, $\langle\langle C^+ \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$, $\langle\langle C^- \rangle\rangle \Box \psi$, and $\langle\langle C^- \rangle\rangle \psi_1 \mathcal{U} \psi_2$. $\Box$

Finally, we can justify the completeness of our characterization by the following theorem.

THEOREM 7. *(1) If $\varphi \notin \mathcal{L}^+_C$, then the satisfaction of φ can be established by a C-norm. (2) If $\varphi \notin \mathcal{L}^-_C$, then the satisfaction of φ can be avoided by a C-norm.*

PROOF. We are only going to prove the first part of this theorem, as the proof for the second part is similar.

To show the satisfaction of all the formulas in $\mathcal{L} \setminus \mathcal{L}^+_C$ can be established by implementing a C-norm. Our method is mainly based on constructing the required concurrent game structure for each such formula.

Let $A \subseteq Ag$ and $A \cap C \neq C$ (i.e., A is not a C^+). For all formulas of the form $\langle\langle A \rangle\rangle \gamma$ where γ is of the form $\bigcirc \varphi$, $\Box \varphi$, or $\varphi_1 \mathcal{U} \varphi_2$, and $\varphi, \varphi_1, \varphi_2$ are arbitrary CO-ATL formulas, we can construct a concurrent game structure S with a state q in its state space, satisfying $S, q \nvDash_{ATL} \langle\langle A \rangle\rangle \gamma$, but $S, q \vDash_{ATL} \langle\langle A \cup C \rangle\rangle \gamma$. So, in such S, agents in the set $A \cup C$ have a joint strategy $F_{A \cup C}$ such that all computations in $out(q, F_{A \cup C})$ satisfy ψ. According to [5, 11], $F_{A \cup C}$ can be a set of "*memory-free*" strategies that map states to $A \cup C$-actions. We construct the C-norm Γ to restrict the joint actions of the agents in set $A \cap C$ to only those are consistent with $F_{A \cup C}$. Then, by Γ, we have $S, \Gamma, q \vDash \langle\langle A \rangle\rangle \gamma$. That is, the satisfaction of $\langle\langle A \rangle\rangle \gamma$ can be established by implementing a C-norm. Proving the result "if $\varphi \notin \mathcal{L}^-_C$ then the satisfaction of $\neg \varphi$ can be established by implementing a C-norm" can be transformed to proving "if $\varphi \notin \mathcal{L}^-_C$ then the satisfaction of φ can be avoided by a implementing a C-norm".

Moreover, it is straightforward to prove that if the satisfaction of φ can be established by a implementing C-norm, then the satisfaction of $\varphi \wedge \varphi'$, $\varphi \vee \varphi'$, $\langle\langle A \rangle\rangle \bigcirc \varphi$, $\langle\langle A \rangle\rangle \Box \varphi$, $\langle\langle A \rangle\rangle \varphi' \mathcal{U} \varphi$, $\langle\langle A \rangle\rangle \varphi \mathcal{U} \varphi'$, $\neg \langle\langle A \rangle\rangle \bigcirc \varphi$, $\neg \langle\langle A \rangle\rangle \Box \varphi$, $\neg \langle\langle A \rangle\rangle \varphi' \mathcal{U} \varphi$, and $\neg \langle\langle A \rangle\rangle \varphi \mathcal{U} \varphi'$, where $A \subseteq Ag$ and φ' is an arbitrary CO-ATL formula, can also be established by implementing a C-norm. $\Box$

Interesting Corollaries: It is easy to show that all the CO-ATL formulas are beyond the normative power of an empty coalition, and the power limitation characterization for NS given by [14] is not complete.

COROLLARY 8. *(1) The limitation of the normative power of coalition $\varnothing$ can be characterized by $\mathcal{L}^+_\varnothing$ and $\mathcal{L}^-_\varnothing$, where both $\mathcal{L}^+_\varnothing$ and $\mathcal{L}^-_\varnothing$ are the class of CO-ATL formulas generated by the following grammar φ (i.e., all the CO-ATL formulas):*

$$\varphi ::= p | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | \langle\langle A \rangle\rangle \bigcirc \varphi | \langle\langle A \rangle\rangle \Box \varphi | \langle\langle A \rangle\rangle \varphi_1 \mathcal{U} \varphi_2 | \neg \varphi$$

where $p \in \Pi$, and $\varnothing \subseteq A \subseteq Ag$.

(2) The limitation of the normative power of coalition Ag can be characterized by $\mathcal{L}^+_{Ag}$ and $\mathcal{L}^-_{Ag}$, which are the classes of CO-ATL formulas generated by the following grammars φ and ψ respectively:

$$\varphi ::= p | \varphi_1 \wedge \varphi_2 | \varphi_1 \vee \varphi_2 | \langle\langle Ag \rangle\rangle \bigcirc \varphi | \langle\langle Ag \rangle\rangle \Box \varphi | \langle\langle Ag \rangle\rangle \varphi_1 \mathcal{U} \varphi_2 | \neg \psi$$

$$\psi ::= p | \psi_1 \wedge \psi_2 | \psi_1 \vee \psi_2 | \langle\langle \rangle\rangle \bigcirc \psi | \langle\langle \rangle\rangle \Box \psi | \langle\langle \rangle\rangle \psi_1 \mathcal{U} \psi_2 | \neg \varphi$$

where $p \in \Pi$.

As the set of NSs is a strict subset of the class of Ag-norms. We can conclude that $\mathcal{L}^+_{Ag}$ and $\mathcal{L}^-_{Ag}$ soundly characterize the limitation of the power of NSs. Although the completeness result doesn't hold for $\mathcal{L}^+_{Ag}$ and $\mathcal{L}^-_{Ag}$ with respect to NSs, we are sure that $\mathcal{L}^+_{Ag}$ and $\mathcal{L}^-_{Ag}$ is a more comprehensive characterization for the power limitation of NSs compared to $\mathcal{L}^e$ and $\mathcal{L}^u$ given by [14], because $\mathcal{L}^e \subset \mathcal{L}^+_{Ag}$ and $\mathcal{L}^u \subset \mathcal{L}^-_{Ag}$.

Moreover, we can compare the normative power of different coalitions. We say coalition C_1 is more powerful than coalition C_2 if and only if $\mathcal{L}^+_{C_1} \subseteq \mathcal{L}^+_{C_2}$ and $\mathcal{L}^-_{C_1} \subseteq \mathcal{L}^-_{C_2}$. Then immediately we can show that for two arbitrary coalitions C_1 and C_2 if $C_1 \subseteq C_2$ then C_2 is more powerful than C_1. Hence, with respect to normative power, $\varnothing$ is the weakest coalition and Ag is the strongest coalition.

5. COMPLEXITY

5.1 Basic Computational Problems

The basic computational problems related to CNSs may include *checking whether a* CNS *is effective*, *checking whether there is an effective* CNS, and *finding an effective* CNS. We formalize them respectively as follows:

CNS EFFECTIVENESS:
Given: CGS S, CNS Γ and objective o.
Question: Is Γ effective for o?

CNS FEASIBILITY:
Given: CGS S, coalition C and objective o.
Question: Is there a C-norm which is effective for o?

CNS SYNTHESIS:
Given: CGS S, coalition C and objective o.
Output: A C-norm Γ that is effective for o.

Note that, similar problems have been proposed for NS in [14]. It has been established that the effectiveness problem for NS is in PTIME and the feasibility problem for NS is NP-complete. For CNS, we have the following results.

THEOREM 9. CNS EFFECTIVENESS *is* PTIME*-complete, and can be solved in time $\mathcal{O}(m \cdot n \cdot l)$ for a* CGS *with m transitions, and an objective of length n, where the max length of the* CO-ATL *formulas in the objective is bounded by l.*

PROOF. To see whether an objective is effective we can firstly determine the effectiveness of all the atomic objectives which requires time $\mathcal{O}(m \cdot l \cdot n)$, then the remain work equals verifying an assignment for a Boolean formula, which requires time $\mathcal{O}(n)$. So the overall time complexity is $\mathcal{O}(m \cdot l \cdot n)$. Thus this problem is in PTIME. With respect to the lower bound, PTIME-hardness is trivial, for verifying $S, \Gamma, q \vDash \varphi$ can be directly reduced to verifying $S \dagger \Gamma \rightsquigarrow \langle q, \varphi \rangle$. $\Box$

While CNS EFFECTIVENESS is tractable, CNS FEASIBILITY is possibly intractable according to the following theorem. Note that, our result is based on the ATL assumption that the agent number, i.e., k, is a constant. But the state number $|Q|$, and the max available action number of every agent in every state, i.e., d, are considered as variables.

THEOREM 10. CNS FEASIBLITY *is* NP*-complete, even for concurrent game structures with only one agent.*

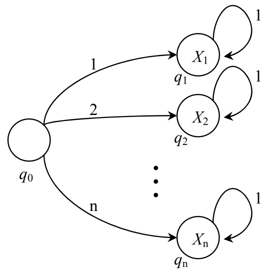

Figure 2: The reduction from SAT to CNS FEASIBILITY

PROOF. Membership in NP can be seen by the following nondeterministic algorithm:
(1) guess a C-norm Γ;
(2) verify that Γ is effective for o.
Since step (1) can be done in non-deterministic polynomial time $\mathcal{O}(d^k \cdot |Q|)$, and step (2) requires only polynomial time.
To see NP-hardness we reduce SAT to it. Given a SAT instance $\phi(x_1, ..., x_n)$. We create a CGS S depicted as Figure 2. That is, $S_\phi = \langle k, Q, \Pi, \pi, d, \delta\rangle$ where $k = 1$; $Q = \{q_0, ..., q_{n+1}\}$; $\Pi = \{X_1, ..., X_n\}$; $\pi(q_i) = \{X_i\}$ for all $i \in \{1, ..., n\}$; $d_1(q_0) = \{1, ..., n\}$, $d_1(q_i) = \{1\}$ for all $i \in \{1, ..., n\}$; and $\delta(q_0, i) = q_i$ for all $i \in \{1, ..., n\}$, $\delta(q_i, 1) = q_i$ for all $i \in \{1, ..., n\}$. Let ϕ^* be the result of systematically substituting for every Boolean variable x_i in ϕ the CO-ATL expression $\langle\!\langle 1\rangle\!\rangle \bigcirc X_i$. Then it is easy to see that ϕ is satisfiable if and only if there is a $\{1\}$-norm which is effective for the objective $\langle q_0, \phi^*\rangle$. □

The synthesis problem for CNS is a function problem that requires an answer more elaborate than "yes" or "no".

THEOREM 11. CNS SYNTHESIS *is* FNP*-complete, even for concurrent game structures with only one agent.*

PROOF. Membership in FNP: Let L be the language for the CNS FEASIBILITY problem. By Theorem 10 we know that for all string x, to decide whether $x \in L$ is NP-complete. And we can define a relation R_L such that $R_L(x, y)$ if and only if $x \in L$ and y is an output of CNS SYNTHESIS given the instance x. It is easy to check that R_L is polynomial-time decidable and polynomially balanced.
To see FNP-hardness we reduce FSAT to it. The reduction is similar to that of Theorem 10. Given a boolean formula $\phi(x_1, ..., x_n)$ we can create the CGS S_ϕ and the CO-ATL formula φ^*. Then we can see that $x_1, ..., x_n$ satisfy ϕ if and only if the CNS $\Gamma = \langle\{1\}, \vartheta\rangle$, where $\vartheta(q_0) = \{i | 0 \leq i \leq n \textit{ and } x_i = 0\}$, $\vartheta(q_i) = \varnothing$ for all $i \in \{1, ..., n\}$, is effective for the objective $\langle q_0, \phi^*\rangle$. □

It is easy to see that the synthesis problem for NS is also FNP-complete. So now we can conclude that the effectiveness, feasibility and synthesis problems of CNS are no more complex than the corresponding problems of NS.

5.2 Complexity of Minimality Checking

The problem of Checking whether a CNS is a minimal CNS is a basic problem related to the concept of minimality.

<u>MINIMAL CNS CHECKING</u>:
Given: CGS S, CNS Γ and objective o.
Question: Is Γ a minimal CNS for o?

THEOREM 12. MINIMAL CNS CHECKING *is co-*NP*-complete.*

PROOF. We can show that the complement problem to MINIMAL CNS CHECKING is NP-complete. That is, given a CGS S, a CNS Γ, and an objective o, determining whether there is a CNS such that $\Gamma' \prec \Gamma$ and Γ' is effective for o. Note that, an arbitrary CNS FEASIBILITY instance is an instance of this problem that taking Γ to be all the transitions in the CGS. So this problem subsumes CNS FEASIBILITY and thus is NP-hard. And the membership in NP is trivial. □

5.3 Complexity of Compactness Checking

With respect to compactness, there are two basic decision problems, that is, *deciding whether a* CNS *is a compact* CNS, and *deciding whether a coalition is a minimal controllable coalition.* We define them formally as follows:

<u>COMPACT CNS CHECKING</u>:
Given: CGS S, cns Γ and objective o.
Question: Is Γ a compact CNS for o?

<u>MINIMAL CONTROLLABLE COALITION CHECKING (MCC)</u>:
Given: CGS S, coalition C and objective o.
Question: Is C a minimal controllable coalition for o?

THEOREM 13. COMPACT CNS CHECKING *is co-*NP*-complete.*

PROOF. The problem complement to COMPACT CNS CHECKING is as follows: given a CGS S, a CNS $\Gamma = \langle C, \vartheta\rangle$ and an objective o, is it true that Γ is not effective for o or there is a C' such that $C' \subset C$ and there is an effective C'-norm for o? We can show this problem is NP-complete: NP-hardness is immediately, for it subsumes CNS FEASIBILITY; and since the amount of agents is a constant, the amount of subsets of C is bounded by a constant. So we can guess a CNS for every subset of C respectively in nondeterministic polynomial-time, and then verify that every CNS is an effective CNS in polynomial-time. This establishes the NP upper bound . □

The problem of deciding whether a coalition is a minimal controllable coalition seems a harder problem, we can show that it is a problem that complete for the class DP[6].

THEOREM 14. MINIMAL CONTROLLABLE COALITION CHECKING *is* DP*-complete.*

PROOF. Membership in DP can be seen from the following algorithm using an oracle for CNS FEASIBILITY:
(1) query the oracle to see whether C is effective for o;
(2) query the oracle to see whether there is an agent $a \in C$ such that $C \setminus \{a\}$ is effective for o;
(3) if step (1) returns "yes" and step (2) returns "no" then return "yes", otherwise return "no".
To prove DP-hardness we reduce SAT-UNSAT [9] to it. Given a SAT-UNSAT instance $(\phi(x_1, ..., x_m), \phi'(y_1, ..., y_n))$, we can create a CGS S depicted as Figure 3.
Let Ψ be the formula $\langle\!\langle\rangle\!\rangle \bigcirc (P_X \wedge \phi^*) \vee (P_Y \wedge \phi'^*)$, where ϕ^* is the result of systematically substituting for every Boolean variable x_i in ϕ the CO-ATL expression $\langle\!\langle 1, 2\rangle\!\rangle \bigcirc X_i$, and ϕ'^* is the result of systematically substituting for every Boolean variable y_i in ϕ' the CO-ATL expression $\langle\!\langle 1, 2\rangle\!\rangle \bigcirc Y_i$. Then we can prove that (ϕ, ϕ') is a "yes" instance of SAT-UNSAT if

[6] DP is a complexity class "between" NP and PSPACE, and consists of all languages that are intersections of a language in NP and a language in co-NP (see [9] for the details of DP).

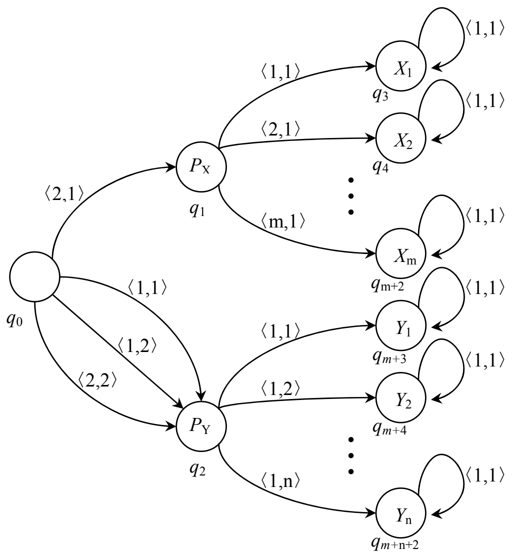

Figure 3: The reduction from SAT-UNSAT **to** MCC

and only if $\{1,2\}$ is a minimal controllable coalition for the objective $\langle q_0, \Psi \rangle$.

For the $\Rightarrow$ direction, we can define a $\{1,2\}$-norm that delete all the transitions from q_0 to q_2, and delete some transitions started from q_1 to make Ψ satisfied in state q_0. But for any $\{1\}$-norm or $\{2\}$-norm the transitions from q_0 to q_2 cannot be completely deleted at the same time, and since ϕ' is unsatisfiable, Ψ cannot be true in state q_0.

For the $\Leftarrow$ direction, the existence of effective $\{1,2\}$-norms requires ϕ or ϕ' is satisfiable. And the fact of Ψ cannot be satisfied in state q_0 by implementing any $\{1\}$-norm or $\{2\}$-norm means ϕ' is unsatisfiable, otherwise we can delete the transition from state q_0 to state q_1 and delete some transitions started from state q_2 by a $\{2\}$-norm to make Ψ satisfied in state q_0. So, ϕ is satisfiable and ϕ' is unsatisfiable. $\square$

6. CONCLUSIONS AND FUTURE WORK

We have proposed the framework for coalitional normative systems in this paper. Three aspects of theoretical work have been done: firstly, we have extended the semantics of ATL and proposed *Coordinated* ATL (CO-ATL) to support the formalizing of CNSs; secondly, we have proved that the limitation of the normative power of an arbitrary coalition C can be soundly and completely characterized by the CO-ATL fragments $\mathcal{L}_C^+$ and $\mathcal{L}_C^-$; and thirdly, we have established the computational complexity of some key problems related to coalitional normative systems.

One opportunity for further research is to more systematically investigate the related computational complexity. As our current results are built on the conventional assumption that the amount of agents k is a constant. So it may be interesting to study the complexity when k is considered as a variable (as in [8]). Another possible further research is modeling the problem of finding an optimal CNS as an optimization problem (as the work in [2]).

7. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant No.60503021,60721002 and 60875038; the Science and Technology Support Foundation of Jiangsu Province under Grant No.BE2009142 and BE2010180; and the Scientific Research Foundation of Graduate School of Nanjing University under Grant No.2011CL07. We are grateful to the reviewers for their helpful comments.

8. REFERENCES

[1] T. Ågotnes, W. van der Hoek, M. Tennenholtz, and M. Wooldridge. Power in normative systems. In *Proceedings of AAMAS-09*, pages 145–152, 2009.

[2] T. Ågotnes and M. Wooldridge. Optimal social laws. In *Proceedings of AAMAS-10*, pages 667–674, 2010.

[3] T. Ågotnes, M. Wooldridge, and W. van der Hoek. Normative system games. In *Proceedings of AAMAS-07*, pages 876–883, 2007.

[4] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of FOCS-97*, pages 100–109, 1997.

[5] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[6] D. Fitoussi and M. Tennenholtz. Minimal social laws. In *Proceedings of AAAI-98*, pages 26–31, 1998.

[7] D. Fitoussi and M. Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence*, 119:61–101, 2000.

[8] W. Jamroga and J. Dix. Do agents make model checking explode (computationally)? In *Multi-Agent Systems and Applications IV (LNAI 3690)*, 2005.

[9] C. H. Papadimitriou. The complexity of facets (and some facets of complexity). In *Proceedings of STOC-82*, pages 255–260, 1982.

[10] M. Pauly and M. Wooldridge. Logic for mechanism design-a manifesto. In *Proceedings of the 2003 Workshop on Game Theory and Decision Theory in Agent Systems (GTDT-2003)*, 2003.

[11] P. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Computer Science*, 85(2):82–93, 2004.

[12] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings of AAAI-92*, pages 276–281, 1992.

[13] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. In *P.E. Agre and S.J.Rosenschein (Eds.),Computational Theories of Interaction and Agency*, pages 597–618, MIT Press,Cambridge, MA, 1996.

[14] W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness feasibility, and synthesis. *Synthese*, 156:1–19, 2007.

[15] W. van der Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164:81–119, 2005.

[16] M. Wooldridge, T. Agotnes, P. E. Dunne, and W. van der Hoek. Logic for automated mechanism design - a progress report. In *Proceedings of AAAI-07*, pages 9–16, 2007.

[17] M. Wooldridge and W. van der Hoek. On obligations and normative ability:towards a logical analysis of the social contract. *J. of Applied Logic*, 3:396–420, 2005.