Inteligencia Artificial, 26(72) (2023), 60-80
doi: 10.4114/intartif.vol26iss72pp60-80

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

Exploring the Generality of Norms in Multi-Agent Systems

Jhonatan Alves[1,A], Jomi Fred Hübner[1,B], Jerusa Marchi[2,C]

[1]Department of Automation and Systems Engineering, Federal University of Santa Catarina, Florianópolis, 88040-900, Brazil
[A]jhonatan.alves@posgrad.ufsc.br
[B]jomi.hubner@ufsc.br
[2]Departament of Informatics and Statistics, Federal University of Santa Catarina, Florianópolis, 88040-900, Brazil
[C]jerusa.marchi@ufsc.br

Abstract: Norms are useful tools to regulate autonomous agents, and their generality is the focus of this paper. The generality of norms refers to the extent of behaviors the norms are capable of regulating. While very specific norms tend to be inefficient to avoid undesirable behaviors (since they are rarely activated), very general norms tend to limit excessively the options of the agents (since they are activated too often) hindering them to achieve the system goals. Therefore, a norm that efficiently regulates the agents should have a balanced generality, being neither too specific nor too general. Therefore, we consider that exploring the generality of norms is a fundamental key to obtaining efficient norms. However, the evaluation of their generality usually considers every behavior they regulate. Since it is likely an unfeasible task, in this paper, we investigate alternatives to estimate the norms generality from their syntactic characteristics. Based on these characteristics, we obtain different sequences of norms that vary approximately from the most specific to the most general. We assume thus that norms with a balanced generality are more easily found considering these orderings. Therefore, it is relevant to understand the impact of the syntactical characteristics in ordering the norms. In this context, we found out how different alternatives organize the norms space. This result is particularly useful for the development of algorithms for searching efficient norms that, through different strategies, may exploit how norms space is arranged and pruned.

Keywords: Normative Multi-agent Systems, Norms Generality, Goals Satisfiability, Conflicts Occurrence, Synthesis of Efficient Norms.

1 Introduction

Our social interactions are mediated by a vast and complex set of norms that exert a powerful influence on our decisions [3]. When we drive a car, hold a position in a company or play a game, for example, norms are present. In Multi-agent Systems, norms have been employed as a mechanism to regulate the society of autonomous agents [14] describing patterns of expected behaviors to promote, for example, coordination and cooperation between agents [1], [31], conflicts avoidance [24], reachability of system goals [28], modeling of electronic institutions [20], [9], among others. Generally, they are conceived in the form of rules that, in certain contexts, impose restrictions on the agents behavior indicating which actions are permitted, prohibited, or obligatory [6]. However, determining which norms effectively regulate the agents is a highly complex problem (NP-Complete) [32] that requires exploring a possibly large space [35].

ISSN: 1137-3601 (print), 1988-3064 (on-line)

Consider a traffic scenario constituted of a 3×3 grid and 3 agents, $\texttt{ag}_1$, $\texttt{ag}_2$ and $\texttt{ag}_3$, as shown in Figure 1.

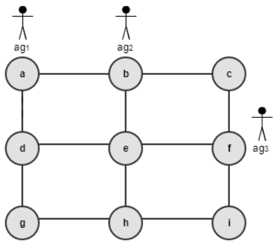

Figure 1: The 3 × 3 grid of traffic scenario.

The vertices of the grid represent locations while the edges represent paths that connect such locations. The agents $\texttt{ag}_1$, $\texttt{ag}_2$, and $\texttt{ag}_3$ are found initially on vertices `a`, `b`, and `f`, respectively, and the system goal is the agents reach a destination without getting into conflicts. A conflict, in turn, is characterized as a collision that occurs when an agent moves to a location occupied by another agent. The agents may choose to `move` between adjacent locations or keep on `idle` in their current locations. In this scenario, we may consider three different norms, n_1, n_2 and n_3, such that, n_1 prohibits just the agent $\texttt{ag}_1$ to move to occupied locations, n_2 prohibits any agent to move to occupied locations whereas n_3 prohibits any agent to move.

When comparing norms n_1 and n_2, we realize that norm n_2 regulates behaviors that norm n_1 also does, however, the opposite does not occur. The norm n_2 is thus more general than n_1, i.e., n_2 is applied to more circumstances than n_1. Similarly, when comparing norms n_2 and n_3, we realize that norm n_3 is more general than n_2. On the one hand, if we choose norm n_1 to regulate the agents, different collisions may occur since, if they wish, both agents $\texttt{ag}_2$ and $\texttt{ag}_3$ may move to occupied locations. This problem occurs because norm n_1 is too specific, constraining a few behaviors that, in turn, is insufficient to avoid conflicts. On the other hand, if our choice is norm n_3, no collision will occur, however, all agents will get stuck in their departure locations and, consequently, no destination will be reached. This problem occurs due to norm n_3 being too general, constraining a great number of behaviors, preventing the agents from performing the actions necessary for the system goal. Therefore, both norms n_1 and n_3 are not good options to regulate the agents.

A norm with a balanced generality (i.e., neither too specific nor too general) is necessary for regulating the agents with efficiency. In this sense, if we choose norm n_2, the agents will be able to move just between unoccupied locations. Thus, norm n_2 is the option capable to avoid conflicts and keep the system goal achievable. In this regard, we consider that exploring the *norms generality* is a fundamental key to finding efficient norms (i.e., norms that avoid conflicts and keep the system goal reachable). However, determining the norms generality may be a complex and unfeasible task that may require computing all the agents behaviors the norms regulate. Consequently, just a few works [2], [18], [24], [23] explored the norms generality in the synthesis of norms.

In [2] we have proposed an approach in which the generality of norms is calculated according to some selected syntactical characteristics of the norms specification. Then, considering the resulting generalities, we obtained a sequence of norms approximately from the most specific to the most general and verify the existence of a region of the sequence in which some norms are efficient. In this paper, we follow our initial ideas presented in [2] introducing new alternatives of syntactic assignments for calculating the generality of norms that, in turn, enable us to obtain different sequences of norms. Consequently, the region in which efficient norms are found in the resulting sequences is affected. In this direction, we aim to investigate *how different syntactic assignments impact the sequences of norms* and use this knowledge for building an algorithm for searching for efficient norms.

The remainder of this paper is structured as follows. In section 2 we present the basic concepts that underlie our approach. In section 3 we further explore the norms generality, introducing different generality assignments through a syntactic perspective. In section 4 we present some experiments where norms are organized by the previous assignments. Then, we discuss the results in order to answer our research question, and we point out some directions of how such assignments may be used in a search for efficient norms. In section 5 we discuss related work. Lastly, in section 6 we present our conclusions and future works.

2 Background

We start this section by introducing a restricted form of a first-order language, which is the base of our approach. We refer to the traffic scenario, introduced in the previous section, along the text to exemplify some concepts.

Definition 1 (Language) *Let* $\mathcal{L} = \langle \mathrm{Pred}^{\mathcal{L}}, \mathrm{Var}^{\mathcal{L}}, \mathrm{Const}^{\mathcal{L}}, \mathrm{Conec}^{\mathcal{L}} \rangle$ *be a first order language, such that,* $\mathrm{Pred}^{\mathcal{L}} = \{pred_1, \ldots, pred_i\}$ *is a set of predicate symbols,* $\mathrm{Var}^{\mathcal{L}} = \{v_1, \ldots, v_j\}$ *is a set of variable symbols,* $\mathrm{Const}^{\mathcal{L}} = \{c_1, \ldots, c_k\}$ *is a set of constant symbols, and* $\mathrm{Conec}^{\mathcal{L}} = \{\wedge\}$ *is a set of logical connectives.*

An atomic formula p is a predicate symbol associated with a finite set $Terms^p$ of terms. A term, in turn, is either a constant or a variable. We assume that terms described by distinct symbols represent distinct objects, variables symbol start with uppercase letters while constants symbol start with lowercase letters. Furthermore, a ground atomic formula is an atomic formula whose terms are all constants. On the other hand, an unground atomic formula is an atomic formula that has, at least, one variable. For convenience, we define a *strict unground atomic formula* as an unground atomic formula whose terms are just variables. We also refer to $\mathcal{L}^{gaf}$ and $\mathcal{L}^{uaf}$ as the sets of all ground and strict unground atomic formulas of $\mathcal{L}$, respectively. The well-formed formulas of $\mathcal{L}$ are given by the formation rules of the first-order logic [30] considering the connectives in $\mathrm{Conec}^{\mathcal{L}}$.

Example 1 *Consider that* $\{\mathtt{next},\mathtt{at}\} \subset Pred^{\mathcal{L}}$, $\{\mathtt{L_1},\mathtt{L_2}\} \subset Var^{\mathcal{L}}$, *and* $\{\mathtt{ag_1},\mathtt{a}\} \subset Const^{\mathcal{L}}$. *Through combining such symbols, we have as a strict unground atomic formula* $\mathtt{next}(\mathtt{L_1},\mathtt{L_2}) \in \mathcal{L}^{uaf}$ *to denote that an arbitrary location* $\mathtt{L_1}$ *of the grid is adjacent to another arbitrary location* $\mathtt{L_2}$. *Moreover, we have as a ground atomic formula* $\mathtt{at}(\mathtt{ag_1},\mathtt{a}) \in \mathcal{L}^{gaf}$ *to denote the agent* $\mathtt{ag_1}$ *is at location* $\mathtt{a}$ *of the grid.*

Definition 2 (System State) *A system state* $s = \{p_1, \ldots, p_i\}$ *is a set of ground atomic formulas. The set* $\mathcal{S} = \{s_1, \ldots, s_j\}$, *such that,* $\mathcal{S} \subseteq 2^{\mathcal{L}^{gaf}}$, *is the set of all system states.*

The system states are built according to the *closed-world assumption*, i.e., if a ground atomic formula is not explicitly included on a state, it does not hold in such state [25].

Definition 3 (Action) *An action is a tuple* $a = \langle \mathrm{Terms^a}, \mathrm{Pre^a}, \mathrm{Add^a}, \mathrm{Del^a} \rangle$, *such that,* $\mathrm{Terms^a} = \{v_1, \ldots, v_i\}$ *is a set of terms,* $\mathrm{Pre^a} = \{p_1, \ldots, p_j\}$ *is a set of preconditions that must hold for* a *to be executed,* $\mathrm{Add^a} = \{p_1, \ldots, p_k\}$ *is a set of addition effects that start to hold after the execution of* a, *and* $\mathrm{Del^a} = \{p_1, \ldots, p_l\}$ *is a set of deletion effects that does not hold after the execution of* a, *where* $\mathrm{Terms^a} \subset \mathrm{Var}^{\mathcal{L}}$ *and* $\mathrm{Pre^a}$, $\mathrm{Add^a}$, $\mathrm{Del^a} \subset \mathcal{L}^{uaf}$, *with* $\mathrm{Var}^{\mathcal{L}}$ *introduced in definition 1. The set* $\mathcal{A} = \{a_1, \ldots, a_m\}$ *is the set of all actions.*

Example 2 *Consider that* $\mathtt{move} \in \mathcal{A}$ *and it is described as* $\mathtt{move} = \langle\{\mathtt{Ag_1},\mathtt{L_1},\mathtt{L_2}\},\{\mathtt{at}(\mathtt{Ag_1},\mathtt{L_1}),\mathtt{next}(\mathtt{L_1},\mathtt{L_2})\}, \{\mathtt{at}(\mathtt{Ag_1},\mathtt{L_2})\},\{\mathtt{at}(\mathtt{Ag_1},\mathtt{L_1})\}\rangle$. *According to such action, for an agent* $\mathtt{Ag_1}$ *to move from a location* $\mathtt{L_1}$ *to a location* $\mathtt{L_2}$, $\mathtt{Ag_1}$ *must be at* $\mathtt{L_1}$ *which, in turn, must be* $\mathtt{next}$ *to* $\mathtt{L_2}$. *After* $\mathtt{move}$ *execution,* $\mathtt{Ag_1}$ *will be at location* $\mathtt{L_2}$ *and it will no longer be at location* $\mathtt{L_1}$.

Definition 4 (Action Instance) *An action instance* a *is a copy of an action* $a_i \in \mathcal{A}$, *such that,* $\mathrm{Term^a} \subset \mathrm{Const}^{\mathcal{L}}$ *is a set of constants and* $\mathrm{Pre^a}$, $\mathrm{Add^a}$, $\mathrm{Del^a} \subset \mathcal{L}^{gaf}$ *are sets of ground atomic formulas, with* $\mathrm{Const}^{\mathcal{L}}$ *and* $\mathcal{A}$ *introduced in definitions 1 and 3, respectively. The set* $\mathcal{A}' = \{a_1, \ldots, a_j\}$ *is the set of all action instances.*

The set $\mathcal{A}'$ can be obtained by replacing the variables of actions in $\mathcal{A}$ by all constants in $\mathrm{Const}^{\mathcal{L}}$. Moreover, the signature of an action is a concatenation between its identifier, an open parenthesis, the action's arguments, and a closing parenthesis. For example, the signature of action $\mathtt{move}$, introduced in example 2, is $\mathtt{move}(\mathtt{Ag_1},\mathtt{L_1},\mathtt{L_2})$. In the next examples, we will refer to both actions and action instances through their signatures.

Definition 5 (State-Transition Function) *A state-transition function* $\gamma : S \times \mathcal{A}' \to \mathcal{S}$ *describes to which state* $s \in \mathcal{S}$ *the system goes when an agent executes an action* $a \in \mathcal{A}'$ *over an state* $s_i \in \mathcal{S}$, *such that, if* $Pre^a \subseteq s_i$, *then* $s = (s_i - Del_a) \cup Add_a$, *with* $\mathcal{S}$ *and* $\mathcal{A}'$ *introduced in definitions 2 and 4, respectively.*

Example 3 *Consider the system state illustrated by Figure 1. It is described as* $s_i = \{\mathtt{at}(\mathtt{ag_1},\mathtt{a}),\mathtt{at}(\mathtt{ag_2},\mathtt{b}), \mathtt{at}(\mathtt{ag_3},\mathtt{f}), \mathtt{next}(\mathtt{a},\mathtt{b}),\mathtt{next}(\mathtt{a},\mathtt{d}),\ldots,\mathtt{next}(\mathtt{h},\mathtt{i}),\mathtt{next}(\mathtt{f},\mathtt{i})\}$[1], *with* $s_i \in \mathcal{S}$. *Also, consider the action* $\mathtt{move}(\mathtt{ag_1},\mathtt{a},\mathtt{d}) \in \mathcal{A}'$ *whose preconditions, addition and deletion effects are, respectively, stated by* $Pre^{\mathtt{move}} = \{\mathtt{at}(\mathtt{ag_1},\mathtt{a}),\mathtt{next}(\mathtt{a},\mathtt{d})\}$, $Add^{\mathtt{move}} = \{\mathtt{at}(\mathtt{ag_1},\mathtt{d})\}$, *and* $Del^{\mathtt{move}} = \{\mathtt{at}(\mathtt{ag_1},\mathtt{a})\}$. *Since* $Pre^{\mathtt{move}} \subseteq s_i$, *if the agent* $\mathtt{ag_1}$ *executes such action over state* s_i, *the system changes to state* $s = \{\mathtt{at}(\mathtt{ag_1},\mathtt{d}),\mathtt{at}(\mathtt{ag_2},\mathtt{b}),\mathtt{at}(\mathtt{ag_3},\mathtt{f}),\mathtt{next}(\mathtt{a},\mathtt{b}),\mathtt{next}(\mathtt{a},\mathtt{d}),\ldots,\mathtt{next}(\mathtt{h},\mathtt{i}), \mathtt{next}(\mathtt{f},\mathtt{i})\}$, *with* $s \in \mathcal{S}$.

Another important concept of our work is the concept of *conflict*. Generally, conflicts are defined according to certain particular characteristics of the areas in which they are considered. However, most of such definitions share that a conflict may be seen, in essence, as "*an undesirable or critical situation which should be avoided*" [16]. In MAS, there exist different types of conflicts, and they may arise from different sources as resource scarcity or task interdependencies, for example [33]. In this work, we consider that a conflict is a situation that causes negative impacts on the system, as a result of the agents executions, compromising the system goal reachability.

[1] For simplicity, we omitted some atomic formulas that represent adjacent locations.

Definition 6 (Conflict) *A conflict* $c = \{p_1, \ldots, p_i\}$ *in the system is a set of strict unground atomic formulas that should ideally not hold. The set* $\mathcal{C} = \{c_1, \ldots, c_j\}$*, such that,* $\mathcal{C} \subseteq 2^{\mathcal{L}^{uaf}}$*, is the set of all types of conflicts that may occur during the system execution.*

Example 4 *A collision between agents is one type of conflict that may occur on the grid. It may be described as* $c = \{\mathtt{at}(\mathtt{Ag}_1, \mathtt{L}_1),\ \mathtt{at}(\mathtt{Ag}_2, \mathtt{L}_1)\}$*. Thus, according to c, a collision occurs when two distinct agents* $\mathtt{Ag}_1$ *and* $\mathtt{Ag}_2$ *are at the same location* $\mathtt{L}_1$*.*

Definition 7 (Conflict Instance) *Given a conflict* $c \in \mathcal{C}$*,* $\mathcal{I}^c = \{c'_1, \ldots, c'_i\}$ *is a set of instances of c, with* $\mathcal{C}$ *introduced in definition 6. An instance* $c' \in \mathcal{I}^c$ *is a copy of conflict c whose atomic formulas are ground, such that,* $c' \subset \mathcal{L}^{gaf}$*.*

Example 5 *An instance of conflict* c_1*, from example 4, is* $c' = \{\mathtt{at}(\mathtt{ag}_1, \mathtt{a}), \mathtt{at}(\mathtt{ag}_2, \mathtt{a})\}$ *which denotes that both agents* $\mathtt{ag}_1$ *and* $\mathtt{ag}_2$ *are at location* $\mathtt{a}$*.*

The conflict instances may be obtained by a unification process [29], as in logic programming [7], between the strict unground atomic formulas of a conflict and the ground atomic formulas of a system state.

Definition 8 (Conflict State) *A system state* $s \in \mathcal{S}$ *is said to be a conflict state if* $\exists c \in \mathcal{C}\ \exists c' \in \mathcal{I}^c$*:* $c' \subset s$*, with* $\mathcal{S}$*,* $\mathcal{C}$*, and* $\mathcal{I}$ *introduced in definitions 2, 6, and 7, respectively.*

There exist different types of goal in MAS [34]. In this work, the system goal is defined as declarative, i.e., it describes a desired situation (state of affairs) that should be reached by the agents [21].

Definition 9 (System Goal) *The system goal* $\mathcal{G} = \{p_1, \ldots, p_i\}$ *is a set of ground atomic formulas, such that,* $\mathcal{G} \subset \mathcal{L}^{gaf}$*.*

Definition 10 (Goal State) *A system state* $s_\mathcal{G} \in \mathcal{S}$ *is said to be a goal state if* $\mathcal{G} \subseteq s_\mathcal{G}$*, with* $\mathcal{S}$ *and* $\mathcal{G}$ *introduced in definition 2 and 9, respectively. The set* $S_\mathcal{G} = \{s_\mathcal{G} \in \mathcal{S} \mid \mathcal{G} \subseteq s_\mathcal{G}\}$ *is the set of all goal states.*

Example 6 *Consider again the system state s from example 3. Consider also the system goal for the traffic scenario is* $\mathcal{G} = \{\mathtt{at}(\mathtt{ag}_1, \mathtt{d}), \mathtt{at}(\mathtt{ag}_2, \mathtt{b}), \mathtt{at}(\mathtt{ag}_3, \mathtt{f})\}$*. Therefore, the state s is a goal state since* $\mathcal{G} \subseteq s$*.*

To achieve the system goal, the agents should reach a goal state without reaching a conflict state. Different mechanisms have been proposed to help the agents to solve conflicts [17, 27]. In this scenario, a well-employed mechanism is norms [12]. In this work, norms are employed as expectations of which behaviors the agents must exhibit to avoid conflict states in order to preserve the reachability of the system goal. Although there exist different types of norms in MAS, we employ the regulative norms [5].

Definition 11 (Norm) *A regulative norm* $n = \varphi^n \underset{\sigma^n}{\rightarrow} a^n$ *is a rule that applies a deontic operator* $\sigma^n \in \{prh, obl, perm\}$ *over an action* $a^n \in \mathcal{A}$ *when both activation context* $\varphi^n = \{p_1, \ldots, p_i\}$ *and preconditions of* a^n *hold, such that, prh, obl, perm stand for prohibition, obligation and permission, respectively, with* $\varphi^n \subseteq \mathcal{L}^{uaf}$ *and* $\mathcal{A}$ *introduced in definition 3.*

Example 7 *Consider the conflict introduced in example 4. A norm to regulate the agents avoiding collisions and keeping them safe to reach their destinations may be described as* $n = \{\mathtt{at}(\mathtt{Ag}_2,\mathtt{L}_2)\} \underset{prh}{\rightarrow} \mathtt{move}(\mathtt{Ag}_1, \mathtt{L}_1, \mathtt{L}_2)$*. In natural language, n may be read as "if an agent* $\mathtt{Ag}_2$ *is at a location* $\mathtt{L}_2$*, then another agent* $\mathtt{Ag}_1$ *is prohibited to move from some location* $\mathtt{L}_1$ *to a location* $\mathtt{L}_2$*".*

Definition 12 (Norm Instance) *Given a norm n,* $\mathcal{I}^n = \{n'_1, \ldots, n'_i\}$ *is a set of instances of n, where an instance* $n' = \varphi^{n'} \underset{\sigma^{n'}}{\rightarrow} a^{n'}$ *is a copy of n, such that,* $\varphi^{n'} \subseteq \mathcal{L}^{gaf}$ *is a set of ground atomic formulas,* $a^{n'} \in \mathcal{A}'$ *is an action instance and* $\sigma^{n'}$ *is a deontic operator, with* $\mathcal{A}'$ *and n introduced in definitions 4 and 11, respectively.*

Example 8 *Consider the norm n from example 7. An instance of n is* $n' = \{\mathtt{at}(\mathtt{ag}_2,\mathtt{b})\} \underset{prh}{\rightarrow} \mathtt{move}(\mathtt{ag}_1, \mathtt{a}, \mathtt{b})$ *that prohibits agent* $\mathtt{ag}_1$ *to* move *from location* a *to location* b *if agent* $\mathtt{ag}_2$ *is at* b*.*

As well as it occurs with conflict instances, norm instances can be obtained by a unification process between the strict unground atomic formulas of a norm and the ground atomic formulas of a system state. Therefore, the norm is regulating the agents if there exists a norm instance for such state.

Definition 13 (Norm Activation) *Given a system state $s \in \mathcal{S}$ and a norm n, n is active in s, denoted by $n \rhd s$, if $\exists n' \in \mathcal{I}^n$, such that, $\varphi^{n'} \cup Pre^{a^{n'}} \subset s$, with $\mathcal{S}$, $Pre^{a^{n'}}$, n, and $\mathcal{I}^n$ introduced in definitions 2, 3, 11, and 12, respectively.*

If $n \rhd s$, then there exists a possibility to the system transits to a conflict state. In order to avoid such state, the norm n may prohibit ($\varphi^n = prh$), obligate ($\varphi^n = obl$), or permit ($\varphi^n = perm$) the execution of action a^n in s.

Example 9 *Consider the system states s and s_i from example 3. Consider also the norm n and its instance n' from examples 7 and 8, respectively. Such norm is active in state s_i since $\varphi^{n'} \cup Pre^{a^{n'}} \subset s_i$, then the norm n is regulating the agent $\mathtt{ag_1}$ through its instance n'. However, since $\varphi^{n'} \cup Pre^{a^{n'}} \not\subset s$, no agent is being regulated by norm n' in state s. Moreover, consider a norm $n_i = \{\ \} \underset{prh}{\rightarrow} \mathtt{move(Ag_1, L_1, L_2)}$. Since the activation context of n_i is empty, such a norm becomes active in every system state where the preconditions of* `move` *holds. One instance of n_i is $n_i' = \{\ \} \underset{prh}{\rightarrow} \mathtt{move(ag_1, a, b)}$. Since $Pre^{a^{n_i'}} \subset s$, the agent $\mathtt{ag_1}$ is prohibited to* `move` *from* `a` *to* `b` *independently of agent $\mathtt{ag_2}$ is at* `b`*.*

During their reasoning cycle, the agents consider analyzing the existence of norms active in the system states to be aware of which actions they are constrained to execute.

Definition 14 (Efficient Norm) *Given a conflict $c \in \mathcal{C}$ and a norm $n \in \mathcal{N}$, n is said to be efficient if: i) $\forall s \in \mathcal{S}\ \exists a_1, a_2, a_3, \ldots, a_i \in \mathcal{A}'$: $n \rhd s \rightarrow s_1 = \gamma(s, a_1) \wedge s_2 = \gamma(s_1, a_2) \wedge s_3 = \gamma(s_2, a_3) \wedge \ldots \wedge s_{\mathcal{G}} = \gamma(s_{i-1}, a_i)$ (for each system state where the norm is active, there exists a sequence of action instances that leads the system to a system goal), with $s_{\mathcal{G}} \in \mathcal{S}_{\mathcal{G}}$ and $i \geq 1$; and ii) $\forall s \in \mathcal{S}\ \nexists s' \in \mathcal{S}\ \nexists c' \in I^c$: $n \rhd s \rightarrow c' \not\subset s'$ (for each system state where the norm is active, there is no state where the conflict occurs), where $\mathcal{S}$, $\mathcal{A}'$, $\mathcal{C}$, $\mathcal{G}$, and $\mathcal{S}_{\mathcal{G}}$ are introduced in definitions 2, 4, 6, 9, and 10, respectively.*

Example 10 *Consider again the conflict c and norm n from examples 4 and 7, respectively. The norm n may be considered an efficient norm since no collision occurs when it is regulating the agents and to each system state where it is active there exists a sequence of actions $a_1, a_2, a_3, \ldots, a_i$ for the agents reach their destinations.*

Definition 15 (Goal Reachability) *The system goal $\mathcal{G}$ is said to be reachable if for all conflict in $\mathcal{C}$ there exists, at least, one efficient norm to avoid it, with $\mathcal{C}$ and $\mathcal{G}$ introduced in definitions 6 and 9, respectively.*

Therefore, when executing their plans, if the agents comply with the efficient norms, the system goal keeps reachable.

Definition 16 (Norms Space) *The norms space $\mathcal{N} = \{n_1, \ldots, n_i\}$ is a set constituted by all possible norms of a given domain, such that, each tuple $(\varphi^n, a^n, \sigma^n) \in (2^{\mathcal{L}^{uaf}} \times \mathcal{A} \times \{prh, obl, perm\})$ constitutes a norm $n = \varphi^n \underset{\sigma^n}{\rightarrow} a^n$, with $\mathcal{A}$ and n introduced in definitions 3 and 11, respectively.*

The norms space is constituted by norms that vary both the actions they regulate and the atomic formulas that form their activation contexts. Such contexts, in turn, vary their length from 0 (as the norm n_i from example 9) up to $|\mathcal{L}^{uaf}|$.

Lastly, we may define what a Normative Multi-agent System is according to our perspective.

Definition 17 (Normative Multi-agent System) *Let $nMAS = \langle Ag, \mathcal{S}, s_0, \mathcal{A}', \mathcal{C}, \mathcal{G}, \mathcal{N} \rangle$ be a normative Multi-agent System, such that, $Ag = \{ag_1, \ldots, ag_i\}$ stand for a set of agents while $\mathcal{S}$, $\mathcal{A}'$, $\mathcal{C}$, $\mathcal{G}$, and $\mathcal{N}$ stand for, respectively, system states, actions instances, conflicts, system goal, and norms according to definitions 2, 4, 6, 9, and 16, respectively. Moreover, $s_0 \in S$ represents the initial system state.*

In the next section, we start discussing the problem of determining the generality of norms. Then, we introduce an approach to assign the generality to norms through a syntactic perspective.

3 The Generality of Norms

As previously discussed in section 1, exploring the generality of norms is a fundamental key for finding efficient norms. In the context of this work, the generality of norms refers to the set of system states where the norms are active. We formalize such concept in definition 18. We also introduce the *more general than* operator as a mechanism to compare norms according to their generality.

Definition 18 (Norms Generality) *Given a norm $n \in \mathcal{N}$, $Sa^n = \{s \in \mathcal{S} \mid n \rhd s\}$ is the set of system states in which the norm n is active, with $\mathcal{S}$, n, $\rhd$, and $\mathcal{N}$ introduced in definitions 2, 11, 13, and 16, respectively. Given two distinct norms $n_i, n_j \in \mathcal{N}$, the norm n_i is more general than n_j, denoted by $n_i \underset{g}{>} n_j$, if $Sa^{n_j} \subset Sa^{n_i}$, $Sa^{n_i} \not\subset Sa^{n_j}$, $a^{n_i} = a^{n_j}$, and $\sigma^{n_i} = \sigma^{n_j}$.*

The relation *is more general than* between norms results in an irreflexive, asymmetric, and transitive partial order $\{(n_j,n_i) \in \mathcal{N} \times \mathcal{N} \mid n_i \underset{g}{>} n_j\}$ on set $\mathcal{N}$. Obtaining such order is a semantic problem that requires: i) building the set $\mathcal{S}$ of system states (which may be huge); building the set Sa^n for each norm $n \in \mathcal{N}$; and iii) obtaining the pair of norms where one norm is more general than the other. However, this is a complex and possibly unfeasible task that needs comparing all norms.

To deal with this problem, we follow our initial ideas presented in [2] where the generality of norms is calculated according to some selected syntactical characteristics of the norms specification. In this work, we introduce new alternatives of syntactic assignments for calculating the generality of norms that, in turn, enable us to obtain different sequences of norms. Consequently, the region in which efficient norms are located in the resulting arrangements is affected. In this direction, we aim to investigate *how different syntactic assignments impact obtaining sequence of norms* to use this knowledge in the building of a search algorithm for efficient norms. In the following, we present the concepts and definitions that underlie our improved approach starting with the definition of generality value.

Definition 19 (Generality Value) *A generality value is a real number that represents the generality of a norm.*

For calculating the generality values, we propose the concept of *generality function*.

Definition 20 (Generality Function) *A generality function $f : \mathcal{N} \to \mathbb{R}^*_+$ is a function that assigns a generality value $f(n)$ to a norm $n \in \mathcal{N}$ according to certain syntactic characteristics of norm, with $\mathcal{N}$ introduced in definition 16.*

According to definition 11, the syntactic characteristics of norms are the *predicates*, *variables*, *atomic formulas*, *deontic operators*, and *actions*. While some characteristics do not impact the generality of norms, as the actions and deontic operator, the occurrences of certain characteristics, such as the atomic formulas and variables, may influence the extent of agents behaviors the norms regulate. Consider, for example, the norms $n = \{\texttt{at}(\texttt{Ag}_2,\texttt{L}_2)\} \underset{prh}{\to} \texttt{move}(\texttt{Ag}_1,\texttt{L}_1,\texttt{L}_2)$ and $n_i = \{\ \} \underset{prh}{\to} \texttt{move}(\texttt{Ag}_1,\texttt{L}_1,\texttt{L}_2)$ from examples 7 and 9, respectively, such that, $n_i \underset{g}{>} n$. The former norm has 1 atomic formula and 4 variables, while the later norm has 0 atomic formulas and 3 variables. Although this is a simple example, we might consider that more specific norms tend to have a larger number of such characteristics than more general norms. Therefore, to calculate the generality values, a generality function may consider the number of atomic formulas and variables. For this, we additionally defined the concept of *auxiliary function*. An auxiliary function gives the occurrence of distinct elements of a certain syntactic characteristic. Since the number of deontic operators and actions is always 1, we do not define a specific function to count them. We define two auxiliary functions, namely, *formulas* and *variables*. The former function counts the norms unground atomic formulas in the activation contexts, while the latter function counts the number of variables.

Definition 21 (Formulas function) *formulas: $\mathcal{N} \to \mathbb{Z}_+$ is a function that gives the occurrence of atomic formulas of a norm $n \in \mathcal{N}$, such that, formulas(n) $= |\varphi^n|$, with $\mathcal{N}$ introduced in definition 16.*

Definition 22 (Variables function) *variables: $\mathcal{N} \to \mathbb{Z}_+$ is a function that gives the occurrence of variables of a norm $n \in \mathcal{N}$, such that, variables(n) $= |(\bigcup_{p \in \varphi^n \wedge t \in Terms^p \wedge t \in Var} t) \bigcup Terms^{a^n}|$, with $Terms^{a^n}$ and $\mathcal{N}$ introduced in definitions 3 and 16, respectively.*

The auxiliary functions may be combined in different ways to build the generality functions. Therefore, the generality values of norms may vary according to the generality functions and certain norms may be assigned to the same generality value when they have the same occurrence of syntactic characteristics that are being exploited. Considering that, a more general norm must have a greater generality value than a more specific norm, we remark the following desirable property of generality functions.

Property 1 (Increasing Generality Function) *A generality function f is said to be increasing if $\forall n_i \in \mathcal{N}\ \forall n_j \in \mathcal{N} : n_i \underset{g}{>} n_j \to f(n_i) > f(n_j)$, with $\mathcal{N}$ and f introduced in definitions 16 and 20, respectively.*

Definition 23 (Generality Set) *Given a generality function f, a generality set $G_k = \{n \in \mathcal{N} \mid f(n) = k\}$ is a set constituted by the norms whose generality value is k which, in turn, is assigned by function f, with k and f introduced in definitions 19 and 20, respectively.*

Definition 24 (Generality Sets Sequence) *Given a set of generality sets $G^G = \{G_i, G_j, \ldots, G_m\}$, we say that $O = (G_i, G_j, \ldots, G_m)$ is an increasing sequence of generality sets from set G^G if $i < j < \ldots < m$. We also say that G^{min}, G^{cen}, and G^{max} are, respectively, the lower limit, the central and the upper limit generality sets of O if $G^{min} = G_i$, $G^{cen} = G_k$ and $G^{max} = G_m$ with k being the median of $(i, j, \ldots, m)$.*

Definition 25 (Contiguous Generality Sets) *Given a set of generality sets G^G and an increasing sequence O over it, we say that two generality sets G_i and G_j, with $\{G_i, G_j\} \subset G^G$, are contiguous in O, represented by $G_i * G_j$, if $i < j$ and there is no other generality set $G_k \in G^G$, such that, $i < k < j$, where both G^G and O are introduced in definition 24.*

The assignment of generality values to norms results in an organization of the norms space into an increasing sequence of generality sets. Through such sets, the norms are ordered approximately from the most specific to the most general, such that, as the norms become more general, the occurrence of syntactic characteristics the generality functions exploit decreases, while the norms generality value increases. Moreover, the number of generality sets in the resulting sequences, the size of such sets as well as their generality value vary according to the generality functions.

Definition 26 (Close Norms) *Given two norms, n_i and n_j, with $\{n_i, n_j\} \subset \mathcal{N}$, we say that n_i and n_j are close norms, represented by $n_i * n_j$, if $n_i \underset{g}{>} n_j$ and there is no other norm $n_k \in \mathcal{N}$, such that, $n_i \underset{g}{>} n_k \underset{g}{>} n_j$.*

Depending on the generality function, close norms are mapped to non-contiguous generality sets. Such functions distribute the norms into more generality sets, obtaining thus bigger sequences. In this context, we remark the following property of generality functions about the mapping of close norms into generality sets.

Property 2 (Contiguous Generality Function) *A generality function f is said to be contiguous if $\forall n_i \in \mathcal{N}$ $\forall n_j \in \mathcal{N} : n_i * n_j \rightarrow G_{k_1} * G_{k_2}$, where $k_1 = f(n_i)$, $k_2 = f(n_j)$, $n_i \in G_{k_1}$, and $n_j \in G_{k_2}$, with $\mathcal{N}$ and f introduced in definitions 16 and 20, respectively.*

Example 11 *Consider two sets of generality sets $G_1^G = \{G_{0.22}, G_{0.50}, G_{0.80}\}$ and $G_2^G = \{G_{0.18}, G_{0.33}, G_{0.61}, G_{0.97}\}$ obtained, respectively, by applying two distinct generality functions, f_1 (which exploits atomic formulas) and f_2 (which exploits variables), over a norms space $\mathcal{N} = \{n_1, \ldots, n_7\}$. Consider also two sequences of generality sets, O_1 and O_2, obtained on sets G_1^G and G_2^G, respectively. Figure 2 illustrates such sequences, where the generality sets are represented by vertical ellipses whose identifiers are in blue and the occurrences of syntactic characteristics the functions exploit are on the top of the ellipses. Moreover, the red edges represent* is more general than *relations between close norms.*

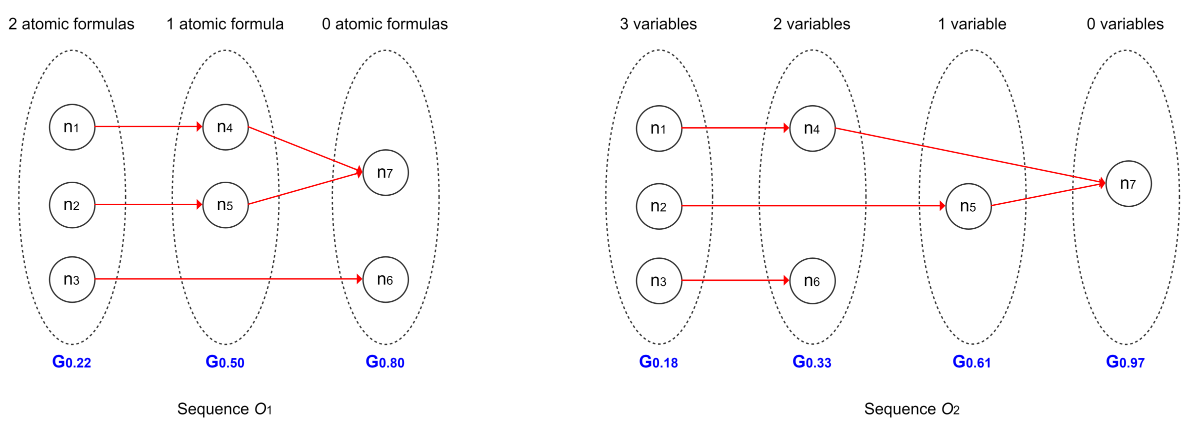

Figure 2: An example of two sequences of generality sets.

Although the generality relations represented in the sequences O_1 and O_2 is the same, the generality sets of such sequences are different. The former sequence is smaller and is composed by 3 generality sets, while the latter is composed by 4 generality sets. Considering the sequence O_1, it has $G^{min} = G_{0.22}$, $G_{cen} = G_{0.50}$, and $G_{max} = G_{0.80}$, where $G_{0.22} * G_{0.50}$ whereas not $G_{0.22} * G_{0.80}$. Except for norms n_3 and n_6, the others close norms are mapped to contiguous generality sets. On the other hand, considering the sequence O_2, it has $G^{min} = G_{0.18}$, G_{cen}

$= G_{0.47}$, and $G_{max} = G_{0.97}$, where the generality value of set G_{cen} is the median of generality values of generality sets of set G_2^G (recall definition 24). Except for norms n_2 and n_5, n_3 and n_6, and n_4 and n_7, the others close norms are mapped to continuous generality sets.

At this point, we have introduced the essential definitions to build the generality functions. A generality function combines the auxiliary functions to exploit one or two syntactic characteristics of norms to obtain an increasing sequence of generality sets. To make things clearer about how such functions behave, we firstly present a norms space in example 12 and, as the functions are introduced, we show the sequences of generality sets the generality functions obtain when applied on such space. Moreover, the representation schema of the resulting sequences is similar to the one in example 11.

Example 12 *Consider the norms space $\mathcal{N}$ for the traffic scenario is constituted by the following 10 norms that, in turn, vary the occurrence of atomic formulas and variables:*

$$\begin{aligned}
n_1 &= \{\mathtt{at(Ag_2,L_2)}, \mathtt{next(L_1,L_2)}\} \underset{prh}{\rightarrow} \mathtt{move(Ag_1,L_1,L_2)}\\
n_2 &= \{\mathtt{at(Ag_2,L_2)}, \mathtt{waiting(Ag_1,L_2)}\} \underset{prh}{\rightarrow} \mathtt{move(Ag_1,L_1,L_2)}\\
n_3 &= \{\mathtt{next(L_1,L_2)}, \mathtt{waiting(Ag_1,L_2)}\} \underset{prh}{\rightarrow} \mathtt{move(Ag_1,L_1,L_2)}\\
n_4 &= \{\mathtt{at(Ag_2,L_2)}\} \underset{prh}{\rightarrow} \mathtt{move(Ag_1,L_1,L_2)}\\
n_5 &= \{\mathtt{next(L_1,L_2)}\} \underset{prh}{\rightarrow} \mathtt{move(Ag_1,L_1,L_2)}\\
n_6 &= \{\mathtt{waiting(Ag_1)}\} \underset{prh}{\rightarrow} \mathtt{move(Ag_1,L_1,L_2)}\\
n_7 &= \{\ \} \underset{prh}{\rightarrow} \mathtt{move(Ag_1,L_1,L_2)}\\
n_8 &= \{\mathtt{at(Ag_2,L_2)}, \mathtt{next(L_1,L_2)}\} \underset{prh}{\rightarrow} \mathtt{idle(Ag_1,L_1)}\\
n_9 &= \{\mathtt{at(Ag_1,L_1)}\} \underset{prh}{\rightarrow} \mathtt{idle(Ag_1,L_1)}\\
n_{10} &= \{\ \} \underset{prh}{\rightarrow} \mathtt{idle(Ag_1,L_1)}
\end{aligned}$$

Definition 27 (Generality Function NGI) *The generality function NGI: $\mathcal{N} \rightarrow\ ]0,1]$ exploits the number of atomic formulas of norms, according to definition 21, to assign a generality value to norms as follows:*

$$NGI(n) = \frac{1}{1 + formulas(n)} \tag{1}$$

In the sequence of generality sets, as the generality values grow, the occurrence of atomic formulas decreases. Therefore, the fewer atomic formulas a norm has, the greater is its generality value. Thereby, such sequence starts with the norms that have the greatest number of atomic formulas and ends up with the norms that have the lowest number of such syntactic characteristic.

Example 13 *The Figure 3 illustrates the sequence O_1 obtained by applying the generality function NGI on set $\mathcal{N}$.*

The sequence O_1 is constituted by three generality sets, $G_{0.33}$, $G_{0.50}$, and G_1, whose norms have, 2, 1, and 0 atomic formulas, respectively. Moreover, we have that $G^{min} = G_{0.33}$, $G^{cen} = G_{0.50}$, and $G^{max} = G_1$. According to function NGI, the most specific norms of $\mathcal{N}'$ are those that form generality set $G_{0.33}$, while the most general norms are those that form generality set G_1.

Definition 28 (Generality Function NGJ) *The generality function NGJ: $\mathcal{N} \rightarrow\ ]0,2]$ exploits the number of variables of norms, according to definition 22, to assign a generality value to norms as follows:*

$$NGJ(n) = \left\{ \frac{1}{1 + variables(n)} \right. \tag{2}$$

In the sequence of generality sets, as the generality values grow, the occurrence of variables decreases. Therefore, the fewer variables a norm has, the greater is its generality value. Thereby, such sequence starts with the norms that have the greatest number of variables and ends up with the norms that have the lowest number of such syntactic characteristic.

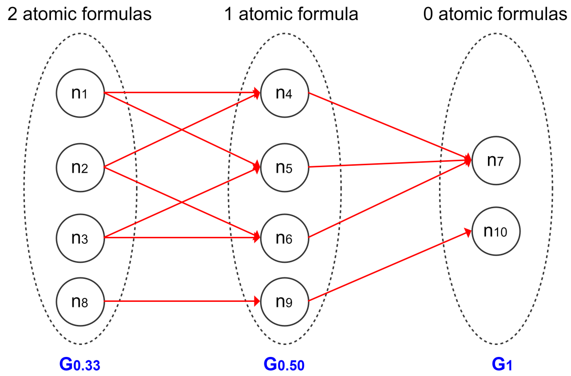

Figure 3: Sequence O_1 obtained from space $\mathcal{N}$.

Example 14 *The Figure 4 illustrates the sequence O_2 obtained by applying the generality function NGJ on set $\mathcal{N}$.*

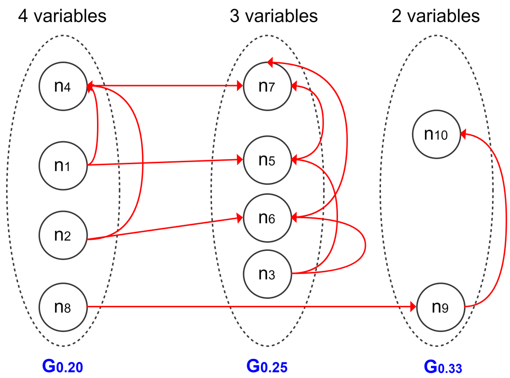

Figure 4: Sequence O_2 obtained from space $\mathcal{N}$.

The sequence O_2 is constituted by 3 generality sets, $G_{0.20}$, $G_{0.25}$, and $G_{0.33}$, whose norms have, 4, 3, and 2 variables, respectively. Moreover, we have that $G^{min} = G_{0.20}$, $G^{cen} = G_{0.25}$, and $G^{max} = G_{0.33}$. The close norms that have the same number of variables are assigned to the same generality value. It occurs with the norms n_1 and n_4 and norms n_2 and n_4 that are in the generality set $G_{0.20}$. Also, it occurs with the norms n_3 and n_5 and norms n_3 and n_6 that are in the generality set $G_{0.25}$. Since some norms of a generality relation are assigned to the same generality value, this function does not present the desirable property 1. Moreover, according to function NGJ, the most specific norms of $\mathcal{N}$ are those mapped to set $G_{0.20}$, while the most general norms are those mapped to the set $G_{1.33}$.

Definition 29 (Generality Function NGK) *The generality function NGK: $\mathcal{N} \rightarrow]0, w]$, where $w \in \mathbb{R}^*_+$ (the positive real numbers), exploits the number of both atomic formulas and variables of norms, according to definitions 21 and 22, respectively, to assign a generality value to norms as follows:*

$$NGK(n) = \frac{1}{formulas(n) + \dfrac{1 + variables(n)}{w}} \tag{3}$$

In the sequence of generality sets, as the generality values grow, the occurrence of both atomic formulas and variables decreases. Therefore, the fewer atomic formulas and variables a norm has, the greater is its generality value. Thereby, such sequence starts with the norms that have both the greatest number of atomic formulas and variables and ends up with the norms that have both the lowest number of such syntactic characteristics. Moreover, in respect to norms with the same number of atomic formulas, the parameter w increases the generality value of norms with a lower number of variables.

Example 15 *The Figure 5 illustrates the sequence O_3 obtained by applying the generality function NGK on set $\mathcal{N}$ with w = 10.*

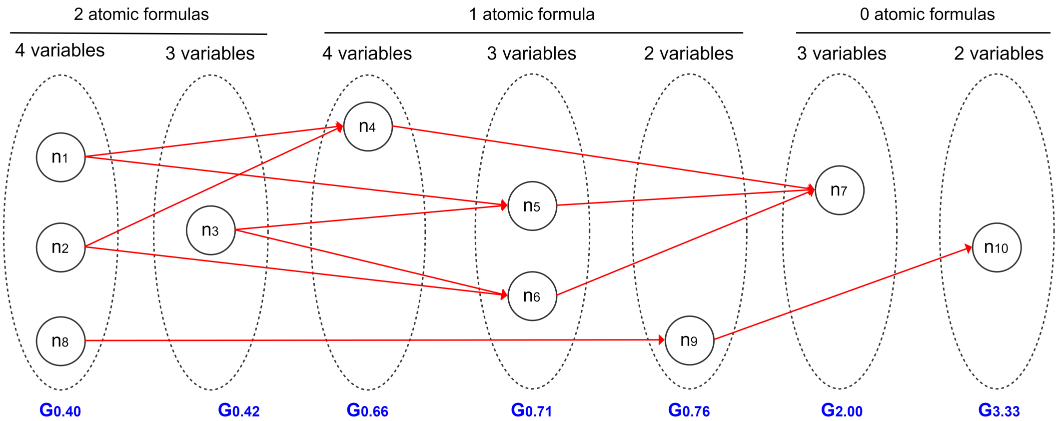

Figure 5: Sequence O_3 obtained from the space $\mathcal{N}$.

The sequence O_3 is larger than those of the previous functions. It is constituted by 7 generality sets, $G_{0.40}$, $G_{0.42}$, $G_{0.66}$, $G_{0.71}$, $G_{0.76}$, $G_{2.00}$, and $G_{3.33}$. Such sets, in turn, are on average smaller than those of functions NGI and NGJ. Moreover, we have that G^{min} = $G_{0.40}$, G^{cen} = $G_{0.71}$, and G^{max} = $G_{3.33}$. In such sequence, the number of atomic formulas decreases from 2 to 0, while the number of variables of norms with the same number of atomic formulas may decrease from 4 to 2. According to function NGK, the most specific norms of $\mathcal{N}$ are those in the generality set $G_{0.40}$, while the most general norms are those in the generality set $G_{3.33}$.

Definition 30 (Generality Function NGF) *The generality function NGF: $\mathcal{N} \rightarrow]0, 1 + |\mathrm{Vars}_{\mathcal{L}}|]$, where $\mathrm{Vars}_{\mathcal{L}}$ corresponds to the set of variables of language $\mathcal{L}$ (recall definition 1), also exploits the number of both atomic formulas and variables of norms, according to definitions 21 and 22, respectively, to assign a generality value to norms as follows:*

$$NGF(n) = \frac{1}{formulas(n) + \frac{1}{1+variables(n)}} \tag{4}$$

This function works similarly to the function NGK. However, as the generality values grow, the occurrence of atomic formulas decreases whereas the occurrence of variables increases. It is resulting from the inversion of the second fraction in the denominator of NGK's formula, where the weight w is fixed to 1, which derived the function NGF. Thereby, the sequence of generality sets starts with the norms that have both the greatest number of atomic formulas and the lowest number of variables and ends up with the norms that have both the lowest number of atomic formulas and the greatest number of variables. Since the greatest number of variables a norm may have is $|\mathrm{Vars}_{\mathcal{L}}|$, the the greatest generality value a norm may be assigned to is $1 + |\mathrm{Vars}_{\mathcal{L}}|$, which is resulting of substituting variables(n) by $|\mathrm{Vars}_{\mathcal{L}}|$ in the denominator 1+variables(n) of function NGF.

Example 16 *The Figure 6 illustrates the sequence O_4 obtained by applying the generality function NGF on set $\mathcal{N}$.*

The major difference between the sequences O_3 and O_4 is that, in the latter, the generality sets are arranged from the norms with the smallest number of variables to the norms with the biggest number of such syntactic characteristic. The sequence O_4 is also constituted by 7 generality sets, $G_{0.44}$, $G_{0.45}$, $G_{0.75}$, $G_{0.80}$, $G_{0.83}$, $G_{3.03}$, and $G_{4.00}$, that are on average also smaller than those of functions NGI and NGJ. Moreover, we have that G^{min} = $G_{0.44}$, G^{cen} = $G_{0.80}$, and G^{max} = $G_{4.00}$. According to function NGF, the most specific norms of $\mathcal{N}$ are those in the set $G_{0.44}$, while the most general norms are those in the set $G_{4.00}$.

Table 1 compares the presented generality functions according to three criteria: i) having property 1 (yes or no); ii) having property 2 (yes or no); and iii) the number of generality sets (3 or 7).

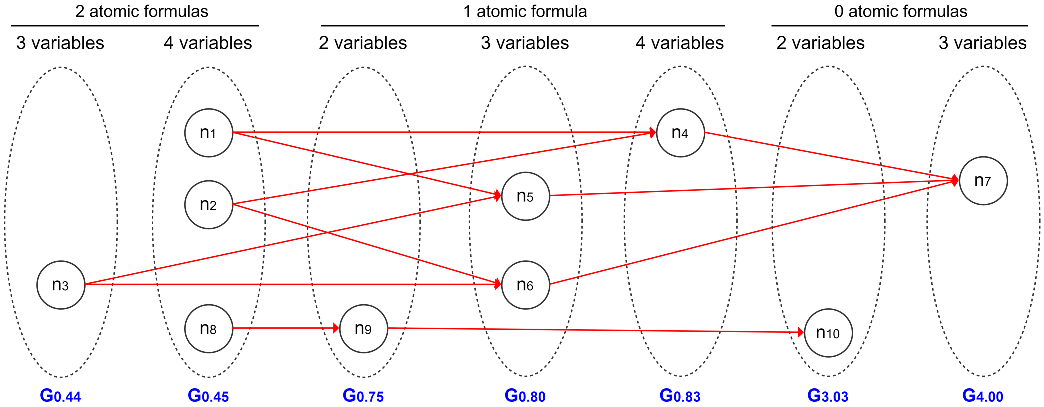

Figure 6: Order O_4 obtained from the space $\mathcal{N}$.

Criteria	NGI	NGJ	NGK	NGF
Having property 1	yes	no	yes	yes
having property 2	yes	no	no	no
Number of generality sets	3	3	7	7

Table 1: comparison between the generality functions.

The function NGI is the only one to map close norms to contiguous generality sets. Since the norms space is constituted by norms whose activation contexts vary from 0 to $|\mathcal{L}^{uaf}|$ (recall definition 16), disregarding the norms with the greatest number of atomic formulas, for each norm n_i in the space there exists a norm n_j, such that, $n_i * n_j$ and *formulas(*n_i*)* = *formulas(*n_j*)* - 1. Since such norms differ by one atomic formula, the function NGI, which just exploits this syntactic characteristic, makes them part of generality sets that are contiguous in the resulting sequence. As it might be seen, in the sequence O_1, except for the respective set G^{min}, the norms of a given generality set generalize the norms of the predecessor set, such that, they have one atomic formula of difference in their activation contexts. On the other hand, there is no guarantee that there exists two close norms that have one variable of difference among themselves. This way, we have that *variables(*n_i*)* = *variables(*n_j*)* - m, where $m \geq 0$ and $m \in \mathbb{N}$ (the natural numbers). When $m = 0$, such norms have the same number of variables, then the function NGJ, which just exploits this syntactic characteristic, assigns the same generality value to them. Thus, the function NGJ may not be considered an increasing generality function. As it might be seen, there are some close norms in the sequence O_2 that have the same number of variables and are in the same generality set.

Since the functions NGK and NGF exploit both the number of variables and atomic formulas they make close norms part of distinct generality sets, which may be contiguous or not (due to the difference between their number of variables and atomic formulas). Moreover, the functions NGK and NGF are more prone to obtain a bigger set of generality sets G^G (recall definition 24) than functions NGI and NGJ. Therefore, the functions NGK and NGF make a fewer number of norms be part of a larger number of generality sets in the resulting sequences. As it will be shown in the next section, such a difference in the resulting sets G^G has a significant impact on the experiments, mainly when we analyze the generality sets where efficient norms are found. In this regard, we present the following definition, which formalizes the existence of the *efficient generality sets*.

Definition 31 (Efficient Generality Set) *Given a generality set G_k and an efficient norm $n \in \mathcal{N}$, we call G_k as an efficient set, denoted by G^{apex}, if $n \in G_k$, with n, $\mathcal{N}$, and G_k introduced in definitions 14, 16, and 24, respectively.*

In the sequences O_1, O_2, O_3, and O_4, the respective sets G^{apex} are the generality sets $G_{0.50}$, $G_{0.20}$, $G_{0.66}$ and $G_{0.83}$. Such sets, in turn, have the norm $n_4 \in \mathcal{N}'$, which is an efficient norm, as explained in example 10.

In the next section, we introduce three scenarios where we perform some experiments in order to answer our research question. For this, we obtain an organization of the norms space to each scenario according to the generality functions NGI, NGJ, NGK, and NGF. Thereby, we may analyze how the scheme of grouping the norms into different generality sets, which vary in number and size, impacts the results of experiments and placement of efficient norms in the resulting organizations of the norms spaces. We finish the section by discussing the results of the scenarios and pointing out some directions of how the knowledge about the organization of the norms space may be useful for building an algorithm for searching for efficient norms.

4 Tests and Results

To answer our research question, we developed a system in Java for creating norms, simulating Normative Multi-agent Systems (nMAS), and summarizing the results. Initially, the system builds a space of *prohibitive norms* for a given application scenario to evaluate their efficiency in avoiding conflicts and keeping the system goal reachable when regulating the agents (the evaluation of other regulative norms as *obligations* and *permissions* will be considered for future work). Then, the nMAS is run for all tuples $(s_0,\mathcal{G},c,n) \in (S_0 \times \mathcal{G}^* \times \mathcal{C} \times \mathcal{N})$, where S_0 is a set of initial states, $\mathcal{G}^*$ is a set of system goals, and $\mathcal{N}$ and $\mathcal{C}$ are introduced in definitions 6 and 16, respectively. This way, we obtain the possible executions of the nMAS for all norms. Secondly, the system generates a sequence of generality sets using the functions NGI, NGJ, NGK, and NGF as introduced in the definitions 27, 28, 29, and 30, respectively. In the third step, for each sequence of generality sets, the system summarizes the results of the nMAS executions.

During the nMAS execution, as the initial configuration changes, the agents behavior may vary, and different situations may lead to conflicts in the system. The execution of nMAS can result in: (i) system goal satisfied; (ii) conflict occurrence; or (iii) timeout. In the first case, the norm avoids conflicts and the agents remain able to perform the actions necessary for the system goal. In the second case, the norm does not effectively regulate the agents and conflicts arise. Lastly, in the third case, the norm may excessively regulate the agents, causing them to avoid conflicts, however, making the system goal unreachable.

During the summarization of results, to each generality set, the system obtains a rate (average) of satisfiability (system goals satisfied), conflicts occurrence, and timeout considering the results associated with the norms in such sets. Moreover, when analyzing the rates of satisfiability, we aim to identify the generality sets that contain efficient norms. This way, we may analyze if there exists a region in the sequences where the sets G^{apex} (recall definition 31) are more likely to be found. In this context, we specify 3 regions of the sequences of generality sets for comparison purposes, namely, *region* G^{cen}, *region* G^{min}, and *region* G^{max}. The first region includes the set G^{cen} and those around it. The second region includes the set G^{min} and the next generality set. Lastly, the third region includes the set G^{max} and the predecessor generality set.

For the sets S_0 and $\mathcal{G}^*$ a systematic sampling [22] is employed with a confidence level of 95% and an error margin of 5%. We also defined the parameter w of function NGK as 10 for all scenarios, and we assume that the agents comply with the norms (there is no norm violation). Moreover, in the following graphics, the x-axis corresponds to sequences of generality sets, where the x values correspond to the generality values k of the generality sets (recall definition 23) and the y-axis corresponds to rates of satisfiability (blue lines), conflicts (red lines), and timeout (yellow lines). We obtained 4 graphics per scenario, one for each sequence. The generality function from which a sequence was obtained is identified next to the y-axis. The scenarios were tested on a computer using an Intel Core i5 5200U processor running at 2.2GHz, with Ubuntu 16.10 64-bit as the operating system and 8 GB of RAM.

4.1 Traffic Scenario

We implemented a traffic scenario adapted from [10]. Such scenario is similar to the one introduced in section 1 and is illustrated in Figure 7. However, the vertices of the current grid are changed to represent *locations* or *tunnels*, while edges still represent paths that connect such vertices. The grid has 2 tunnels, $\texttt{t}_1$ and $\texttt{t}_2$, that are represented by blue squares, and 7 locations, from `a` to `g`, that are represented by gray circles. In turn, the tunnels entrance and exit are represented, respectively, by entry and depart edges. Moreover, there are 5 agents on the grid, $\texttt{ag}_1$ to $\texttt{ag}_5$, that intend to reach their destinations executing their traversal plans without colliding. The agents may choose to `move` between adjacent vertices or keep on `idle` in their current locations. After entering into a tunnel, the agent is required to move out in its next turn (it is neither possible to enter a tunnel from its exit nor keep on idle inside a tunnel).

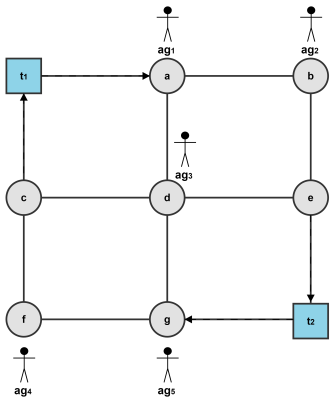

Figure 7: An extension of 3×3 grid from section 1.

In the initial states, the agents are placed in their departure locations, while in the system goals, they are placed in their destinations (tunnels are neither departure nor destinations). The agents, in turn, make use of *Manhattan distance* to estimate the best plan to reach the system goal. Let *Vertices* and *Agents* be, respectively, the sets of grid vertices and agents on it, then, in the worst case, each agent has $2^{|\ Vertices|}$ steps to finish its plan and thus the system timeout in $|Agents| * 2^{|\ Vertices|}$ steps. The tests consisted of 225 initial states, 225 goals, and 14 norms. Thus, the nMAS was executed 708.705 times. The results of this scenario are illustrated in Figure 8. In such Figure (as well as in Figures 9 and 10 from the next two scenarios) we inserted a vertical dotted line in the middle of x-axes (i.e., the sequences) to *highlight* the distance from sets G^{cen} to sets G^{apex}.

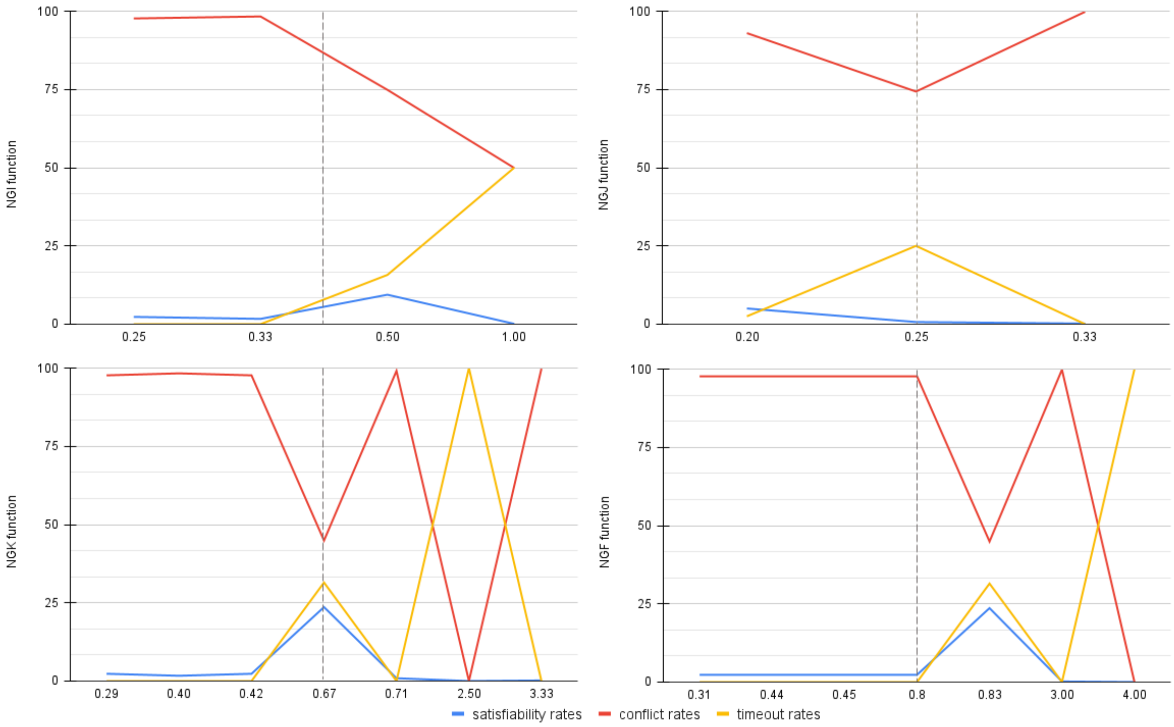

Figure 8: Results For the Grid Scenario.

In general, it can be seen that the conflict rates (red lines) are higher than other rates for most values of the x-axes. This is mainly due to the fact that most norms are not able to prevent collisions between agents. When agents are regulated by such norms, the execution of nMAS tends to lead to conflicts. However, as norms become more general, they exert more control over agents, which helps to reduce collisions and consequently the conflict rates. In this sense, we observe that conflict rates gradually decrease, even if only slightly. However, they decrease sharply around the middle of the x-axes. This is mainly due to the fact that the generality sets $G_{0.50}$, $G_{0.20}$, $G_{0.67}$, and $G_{0.83}$ of the functions NGI, NGJ, NGK, and NGF, respectively, are formed by the norm $n = \{\mathtt{at(Ag_2,L_2)}\} \underset{prh}{\rightarrow} \mathtt{move(Ag_1,L_1,L_2)}$ (introduced in example 7). As agents have enough time to complete their plans and collisions do not occur when agents are regulated by this norm, the success rates of the previous sets are positively affected by the number of executions of nMAS in which the system goals are achieved.

Norm n has success, conflict, and timeout rates of 90%, 0%, and 10%, respectively. Although certain executions of nMAS with such a norm result in a timeout[2], we consider n to be an efficient norm because its timeout rate is relatively small and it does not allow conflicts. Moreover, it is likely that, given the initial state and system goals considered in the experiments where the system timeout, agents can satisfy the system goals by choosing alternative paths in the grid when they are regulated by norm n. In this context, according to definition 31, the sets $G_{0.50}$, $G_{0.20}$, $G_{0.67}$, and $G_{0.83}$ correspond to the sets G^{apex} of the sequences to which they belong. However, the sets G^{apex} are also composed of inefficient norms, and consequently their success rates are negatively affected by the poor performance of the system when the agents are regulated by such norms. Therefore, the success rates of the sets G^{apex} do not reach 100%.

Since the functions NGK and NGF tend to obtain sequences with more generality sets than functions NGI and NGJ (as discussed at section 3), the success rates of the G^{apex} sets from sequences obtained with NGK and NGF are higher than the success rates of the sets G^{apex} from sequences obtained with NGI and NGJ. In particular, the function NGJ obtains the sequence whose set G^{apex} has the lowest success rate. This is essentially because in such a sequence, most of the norms are part of its set G^{apex}, which negatively affects the success rate of such a set due to the inefficient norms that form it.

As for the timeout rates (yellow lines), we note that they present a gradual but slightly increase in all domains. However, around the middle of the x-axes, the timeout rates increase more sharply and reach their highest values in the region G^{max}. This is largely due to the fact that as norms become more general, they may prohibit agents from performing certain actions required to achieve the system goals. Consequently, when agents are regulated by very general norms, the execution of nMAS tends to timeout.

When analyzing the positions of the sets G^{apex} along the x-axes, we found out that there is a region where such sets can be found: the region G^{cen}. To verify whether the sets G^{apex} tend to be found in such a region, in the following sections, we examine larger and more complex scenarios looking for where the efficient norms are found in the resulting sequences.

4.2 Security Scenario

In this scenario, an organization shares with its employees a set of documents. Such documents are classified as public or confidential while the employees has low or high access credentials according to the roles they play in the organization. The documents are available in a repository of a local server, which is accessible from any computer of the organization's internal network. To open a document an employee must provide his credentials to access the repository. However, due to data protection and privacy issues (as theft and information leakage), it is expected that only employees with high credential open confidential documents (no other mechanism to prevent a data breach, unless the norms, are implemented by the organization). Thus, when an employee with low credential opens a document of that type, a conflict occurs. Although opening a confidential document by those employees does not make part of the system goal, it can make part of the individual goal of some of them.

Essentially, the employees behavior consists of entering the repository, trying to open the documents allocated to them by the system goal or those related to their individual goals (when they exist), and then leaving the repository. We assume that 30% of employees with low credentials have goals that are not coherent with those of the system (they intend to open confidential documents). In their turn, these employees decide with a 50% chance which goal (individual or overall) to satisfy. The set S_0 is constituted by initial states that vary the classifications of employees and documents. Moreover, the employees are logged out to the repository in all initial states. On the other hand, the set $\mathcal{G}^*$ is constituted by system goals that vary the documents the employees must open. Also, such goals specify that after opening the target documents, the employees must log out of the repository.

Each employee has 2 + (*MaxDocs* $\times$ *Chance*) steps to satisfy the system goal, where *MaxDocs* denotes the maximum number of documents an employee may have access to, *Chance* denotes a percentage increase in the number of attempts to open such documents, and the number 2 represents the actions of entering and leaving the repository. We also defined that *MaxDocs* = 3 and *Chance* = 1.5. Moreover, there are 30 employees in the organization and 30 documents in the repository. Let *Employees* be the set of employees, thus the system timeout is $|Employees| \times [2 + (MaxDocs \times Chance)]$. The tests consisted of 245 initial states, 245 goals and 388 norms. Thus, the system was executed around 2.3×10^7 times. Figure 9 illustrates the results for this scenario.

[2]Timeout occurs when, for example, an agent gets stuck in a corner of the grid because other agents occupying adjacent vertices do not give way.

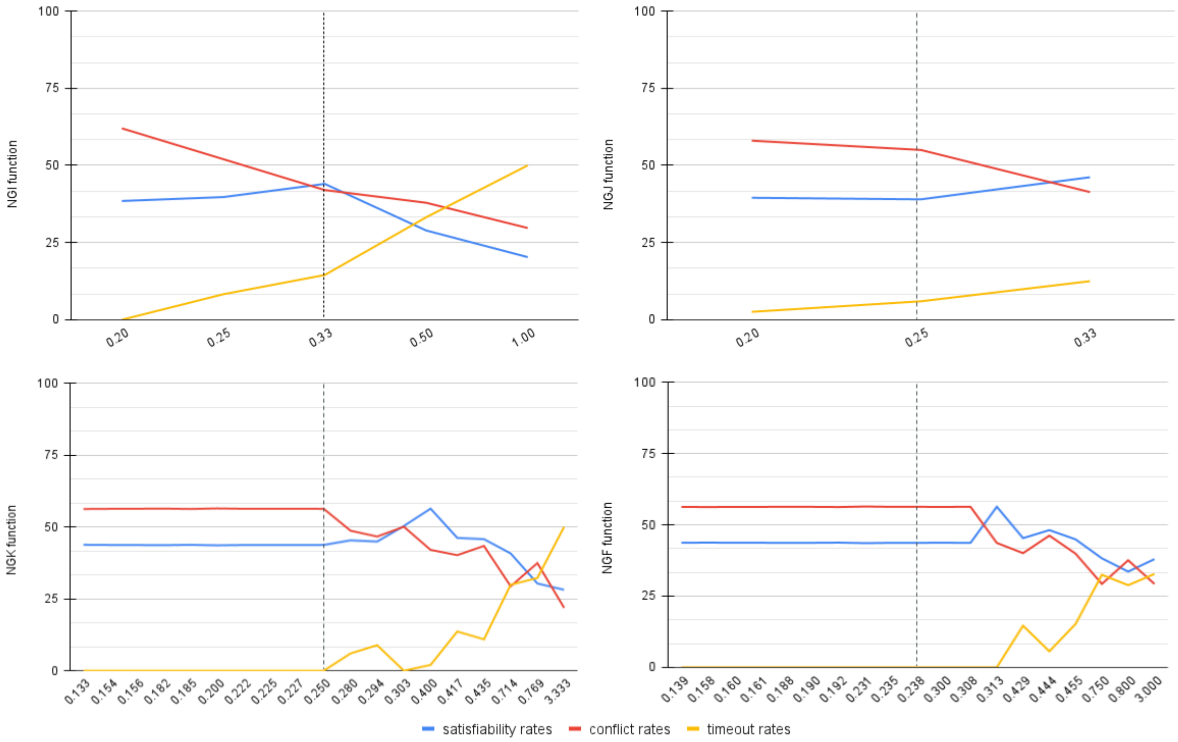

Figure 9: Results For the Security Scenario.

As in the previous scenario, it can be observed that conflict rates (red lines) are higher than other rates for most values of the x-axes. This is primarily due to the fact that most norms are unable to prevent employees with low credentials from accessing confidential documents. Thus, when employees are regulated by such norms, the execution of nMAS tends to lead to conflicts. The conflict rates gradually decrease, and around the middle of the x-axes they decrease even more. It can be observed that the sets $G_{0.33}$, $G_{0.33}$, $G_{0.400}$, and $G_{0.429}$ of functions NGI, NGJ, NGK, and NGF, respectively, have the highest success rates among the sequences to which they belong. This happens because such sets are formed by the norm $n' = \{\texttt{lowCredential}(\texttt{Ag}_1), \texttt{confidential}(\texttt{Doc}_1)\} \underset{p}{\rightarrow} \texttt{open}(\texttt{Ag}_1, \texttt{Doc}_1)$, which prohibits an employee $\texttt{Ag}_1$ from opening a document $\texttt{Doc}_1$ if his credential is low and $\texttt{Doc}_1$ is confidential. By adhering to the norm n', employees with low credentials do not cause conflicts, and since there is enough time to achieve the system goals, the success rate of the previous sets are positively affected by the good performance of the system when it runs with the norm n'. In turn, the norm n' has success, conflict, and timeout rates of 100%, 0%, and 0%, respectively. Since in all experiments the system goals are achieved and no conflicts happens when the norm n' regulates the employees, we consider n' to be an efficient norm. In this context, the sets $G_{0.33}$, $G_{0.33}$, $G_{0.400}$ and $G_{0.429}$ correspond to the sets G^{apex} of sequences to which they belong. However, as in the previous scenario, the sets G^{apex} also consist of inefficient norms, and consequently their success rates are negatively affected by the poor performance of the system when the employees are regulated by such norms. Therefore, the success rates of the sets G^{apex} also do not reach 100%.

Moreover, since functions NGK and NGF obtain sequences with more sets with fewer norms on average, the sets G^{apex} of such sequences have higher success rates than the sets G^{apex} of sequences obtained by functions NGI and NGJ. Moreover, since the function NGJ obtains the smallest sequence, the generality sets of its sequence have more norms on average. Therefore, such a sequence has the set G^{apex} with the lowest success rate.

The timeout rates (yellow lines) start in the region G^{cen} of the sequences obtained by the functions NGK, NGF and NGI, while the timeout rate starts in the region G^{min} in the sequence obtained by the function NGI. Moreover, the timeout rates in all sequences reach the highest value in the region G^{max}.

When analyzing the position of the sets G^{apex} in the sequences of this scenario, we can see that there are basically two regions where such sets can be found. In the sequences obtained by the functions NGI and NGJ, the sets G^{apex} belong to the region of G^{cen} (although in the sequence obtained by the function NGJ they are also part of the region G^{max}). In contrast, in the sequences obtained by the functions NGK and NGF, the sets G^{apex} lie between the regions G^{cen} and G^{max}.

4.3 Medical Scenario

In this scenario, an outpatient service is offered to a set of patients who look for medical care. The service is provided in a medical center consisting of a waiting room and a medical office. The waiting room contains a password panel and three rows of four connected chairs. Due to COVID-19 [15] and social distancing policies, when two patients have a sit next to each other a conflict situation occurs (we assume the patients are wearing face masks and respecting the basic sanitary protocols). The behavior of patients consists of entering the waiting room (where they receive automatically a service password), choosing a chair to sit in, and waiting for the medical appointment. The choice of chairs is random, and if the norm prevents the patients from sitting down, they may choose to wait for the appointment while standing inside the clinic or outside. Due to the fear of contamination by COVID-19 in small and closed environments, the patients have an 80% of preference for the second option.

The panel shows a new password in an interval t, and the patient owner of the current password enters the medical office and left it on his next turn. Let $Patients$ be the set of patients, thus the system timeout in $|Patients| \times t + 4$ steps, where the value 4 represents the number of actions necessary for the last patient to satisfy the system goal when he is seated in a chair (i.e., get up from the chair, enter and leave the medical office, and finally leave the medical center). It is defined that $t = 2 \times |Patients|$ and $|Patients| = 12$. The set S_0 is constituted by only one initial state specifying that all patients are outside the medical center. The set $\mathcal{G}^*$, in turn, is constituted by only one system goal specifying that all agents must receive medical attendance and, after that, leave the medical center (in order to avoid agglomerations inside it). Moreover, the tests consisted of 6150 norms, thus the system was executed 6150 times. Figure 10 illustrates the results of this scenario.

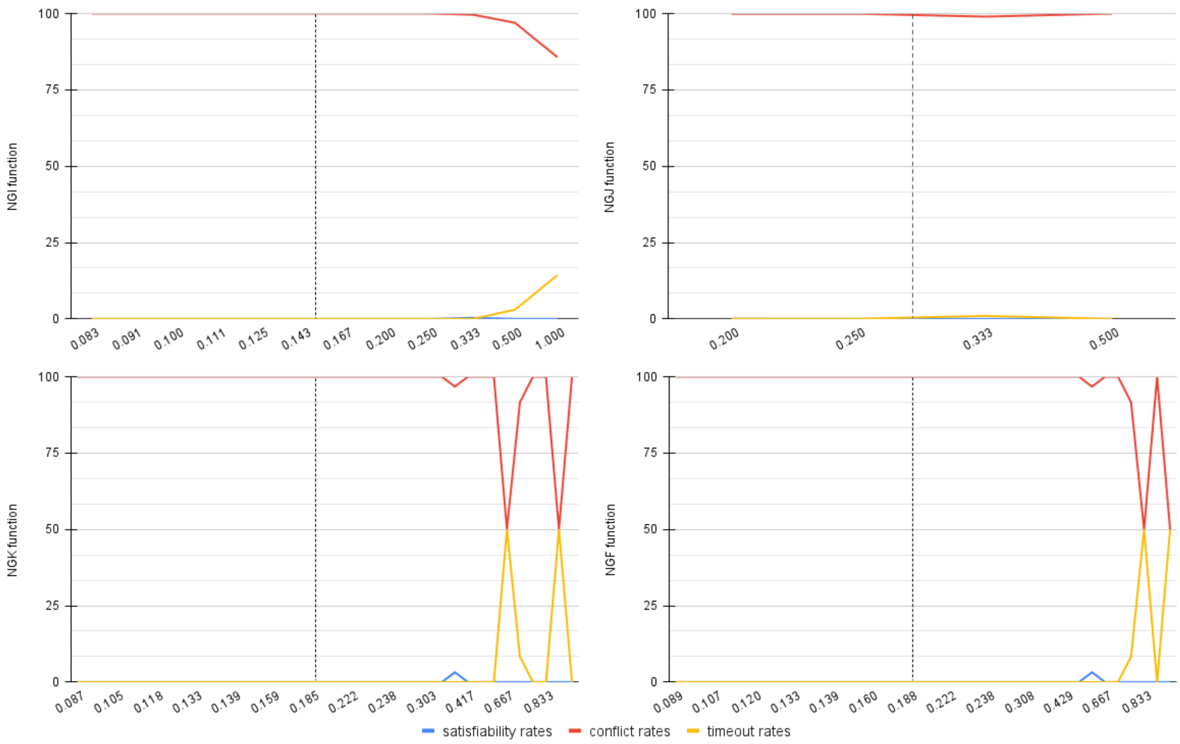

Figure 10: Results For the Medical Care Scenario.

In this scenario, it is also observed that conflict rates (red lines) are higher than other rates for most values of the x-axes, which is due to the fact that most norms cannot prevent patients from sitting next to each other. Thus, when patients are regulated by such norms, the execution of nMAS tends to lead to conflicts. However, in sequences of functions NGI, NGK, and NGF, the conflict rate decreases more sharply near the region G^{max}, while the timeout rate increases more sharply near such a region. On the other hand, neither a decrease in the conflict rate nor an increase in the timeout rate is visually evident in the sequence of the function NGJ.

In this scenario[3], the norm that efficiently regulates the patients is $n'' = \{\texttt{sitting}(\texttt{Ag}_2, \texttt{C}_2), \texttt{next}(\texttt{C}_2, \texttt{C}_1)\} \underset{p}{\rightarrow}$

[3]The sequences obtained by the functions NGK and NGF have 39 generality sets, however, due to lack of space, most generality sets are omitted.

$\texttt{sit}(\texttt{Ag}_1, \texttt{C}_1)$, which prohibits a patient $\texttt{Ag}_1$ from sitting on a chair $\texttt{C}_1$ if another patient $\texttt{Ag}_2$ is sitting on a chair $\texttt{C}_2$, which in turn is next to $\texttt{C}_1$. The norm n'' has success, conflict, and timeout rates of 100%, 0%, and 0%, respectively. Since all executions of nMAS with norm n'' result in the system goals being achieved, we consider n'' to be an efficient norm.

The generality sets $G_{0.333}$, $G_{0.200}$, $G_{0.400}$ and $G_{0.455}$ of the functions NGI, NGJ, NGK and NGF, respectively, correspond to the sets G^{apex} of this scenario. Since the current norms space is larger than that of the grid and security scenarios, the generality sets of the current sequences are on average much bigger. Therefore, the sets G^{apex} of all current sequences are negatively affected by the great number of inefficient norms that form them, more than in the previous scenarios. This contributes to the satisfiability rate of sets G^{apex} of the functions NGI and NGJ is not visible in the corresponding graphs and the satisfiability rate of sets G^{apex} of the functions NGK and NGF is much lower than in the previous scenarios. When analyzing the location of the sets G^{apex}, we found that the efficient norm lies between the regions G^{cen} and G_{max} in sequences obtained by the functions NGI, NGK, and NGF. However, in the sequence obtained by the function NGJ, the set G^{apex} is located in the region G^{min}.

4.4 Discussion

Our approach organizes the norms space by placing norms into different sequences of generality sets. Through these sets, the norms are ordered approximately from the most specific to the most general. In the resulting sequences, as the norms get further away from set G^{cen} to the left, the norms start to have more occurrences of syntactic characteristics that the generality functions explore, which makes them hardly activate. As a result, norms on the left side of a sequence tend to have a higher conflict rate. In contrast, as the norms get further away from set G^{cen} to the right, they start to have less occurrences of syntactic characteristics that the generality functions explore, which makes them activate more easily. Consequently, the norms on the right side of sequences tend to have a higher timeout rate. Thus, an efficient norm is unlikely to be in the regions G^{min} and G^{max}, since it should have a balanced generality to regulate agents efficiently. This means that the norm should be neither too specific nor too general so that it can be easily activated without over-regulating the agents. Although this reasoning leads us to believe that the set G^{apex} is likely to appear in the region G^{cen}, both its position and satisfiability rate are differently affected by the syntactic characteristics exploited by the generality functions.

In order to answer our research question *how different syntactic assignments impact the sequences of norms*, Table 2 summarizes the results from the experiments by generality function according to four evaluation criteria: *Sets* - the number of generality sets; *Dist* - the distance between the sets G^{apex} and G^{cen} (given in number of generality sets); *Pos* - the position of sets G^{apex} in the sequences obtained by the generality functions (to the *left* or *right* of sets G^{cen} or *center* when $G^{apex} = G^{cen}$); and *Sat* - the satisfiability rates of sets G^{apex}.

	NGI				NGJ				NGK				NGF			
Scenario	Sets	Dist	Sat	Pos	Sets	Dist	Sat	Pos	Sets	Dist	Sat	Pos	Sets	Dist	Sat	Pos
Traffic	4	2	9.4%	*right*	3	1	5%	*left*	7	0	23.6%	*center*	7	1	23.6%	*right*
Security	5	0	45%	*center*	3	1	41.1%	*right*	19	4	56.3%	*right*	19	4	56.3%	*right*
Medical	12	4	0.35%	*right*	4	2	0.03%	*left*	39	11	3.2%	*right*	39	14	3.2%	*right*

Table 2: Comparison between the results from the experiments.

For all scenarios, the functions NGK and NGF obtained the greatest number of generality sets and sets G^{apex} with the greatest satisfiability rate. Except for the traffic scenario, where the set G^{apex} is in the *middle* of the sequence obtained by function NGK, both functions NGK and NGF obtained sequences that have their set G^{apex} on their *right* side. On the other hand, the sequences obtained by function NGJ have the set G^{apex} with the smallest satisfiability rate. Such sets are on the *left* or *right* side of the corresponding sequences. Lastly, the sequences obtained by function NGI have a set G^{apex} with satisfiability rate greater than those of sequences obtained by function NGJ, however, lower than those of sequences obtained by functions NGK and NGF. The sequences obtained by function NGI have the set G^{apex} on their *middle* or *right* side. Moreover, concerning the distance between the sets G^{apex} and G^{cen}, in the sequences obtained by functions NGI and NGJ, they are on average[4] 1.33 and 2 apart, respectively. In the sequences obtained by functions NGK and NGF, such sets are, respectively, on average 2.6 and 3 apart.

According to these results, we can conclude that functions that exploit *two syntactic characteristics of the norms* are more prone to obtain sequences in which the set G^{apex} has a higher satisfiability rate compared to functions that exploit only one of these characteristics. In this way, a generality set that contains an efficient norm has a greater satisfiability rate when obtained by the functions NGK and NGF than when obtained by the functions NGI and NGJ. We also can conclude that functions that exploit *only one syntactic characteristic of the norms* tend to obtain sequences in which the set G^{apex} is closer to the middle than functions that exploit two

[4]The averages are obtained by adding the values in the column Dist and dividing the result by the number of scenarios.

of such characteristics. Since the fewer syntactic characteristics a function exploits, the smaller is the resulting sequence, then it is possible that there is a smaller number of generality sets between G^{cen} and G^{apex} in smaller sequences than in larger sequences.

Moreover, considering the identified regions where the efficient norms are located, then it is possible that just exploring the variables is not enough to place such norms into those regions in the sequences. Moreover, the function NGJ is not an increasing generality function (recall property 1), then it may place close norms into the same generality set. For instance, if all norms have the same number of variables, they are mapped to an unique generality set. Since sets do not have order, when applying the function NGJ over a norms space, it is possible that no sequence is obtained. Therefore, considering these two characteristics of function NGJ, the sequences that it obtains are probably not good organizations for the norms space. Since the sequences obtained by functions NGI, NGK, and NGF have efficient norms from the region G^{cen} to the right, and this trend is observed as the scenarios become more complex and the size of the sequences increases, it is possible that functions that exploit, at least, the number of atomic formulas tend to position efficient norms between the regions G^{cen} and G^{max}.

We intend to use the knowledge about the identified regions of the efficient norms as a heuristic to guide a search for efficient norms. Firstly, we may organize the norms space according to a given generality function. Secondly, the search may start in the set G^{cen}, traversing the space to the next sets according to a visiting strategy until the set G^{apex} is found. This way, we can prune part of the space in case an efficient norm is found in the region of G^{cen}. Although different strategies may be employed, a good strategy may depend on the generality functions. Since the function NGJ tends to obtain smaller sequences with a set G^{apex} that can be present on either their left or right sides, covering both sides alternatively may improve the performance of the search in such sequences. Therefore, the strategy may apply a *zigzag* pattern starting in the set G^{cen}, turning to the right, then to the left until reaching the sets G^{max} and G^{min}. For the sequences obtained by functions NGI, NGK, and NGF, a possible good strategy may start *from center to upper and lower limits*, such that, it firstly visits the set G^{cen} until reach the set G^{max}, then proceed directly to the previous set of G^{cen} going towards the set G^{min}.

The search in sequences obtained by functions NGK and NGF possibly perform better than in sequences obtained by functions NGI and NGJ. This hypothesis takes into consideration that the total number of norms to be analyzed between the sets G^{cen} and G^{apex} may be lower in application scenarios constituted by smaller generality sets. As an example, consider again the sequences from examples 13 to 16. Consider also the two previous visiting strategies, *Zigzag* and *From center to upper and lower limits*, to find the efficient norm n_4. Table 3 presents the number of norms analyzed, in the worst case, between the sets G^{cen} and G^{apex}.

Criteria	O_1	O_2	O_3	O_4
Zigzag	4	10	4	3
From center to upper and lower limits	4	10	6	3

Table 3: number of norms analyzed in the worst case.

According to Table 3, for both strategies, the search presents the best result for sequence O_4 (obtained by function NGF) and the worst result for sequence O_2 (obtained by function NGJ). This simple example shows that it is possible to reduce the number of norms analyzed when the search occurs in a sequence with a larger number of generality sets that, in turn, are composed of a smaller number of norms.

5 Related Works

Although norms are widely employed in MAS as a regulatory mechanism, most works depend on the designer expertise to norms be created [26, 4, 11, 8, 13]. However, on complex systems, the manual creation of norms may be an infeasible task and without a thorough analysis of how the norms impact on the agents behavior, different problems may arise during their execution as unforeseen conflicts or inability to fulfill the system goals. As previously discussed, exploring the norms generality is a fundamental key to find efficient norms. However, the norms generality is considered by a few works. In the following, we discuss three works from the literature.

In [18] and [19], the authors propose a method to obtain *minimal* norms. An efficient norm n is said to be minimal if there is no other efficient norm n_i that imposes fewer restrictions than n. Therefore, a minimal norm provides more freedom for the agents to choose their actions. However, the authors hard-wire the norms into the agents, such that, for each agent state, a minimal norm prohibits the smallest set of actions. This way of creating norms may be unmanageable for systems that have a great number of agents with a great number of states. Moreover, changes in the system specifications may require that agents are reprogrammed.

In [24], the authors propose a method to organize norms in a network of generalization relations, where the nodes represent norms and the edges connect specific norms to general norms. During the system execution, as conflicts occur, norms are created and inserted into the network, which is initially empty. The network is updated until it contains a subset of norms that is efficient in avoiding conflicts (the reachability of the system goal is not considered). However, the creation of norms starts with the most specific. Moreover, all specific norms need to be first obtained so that more general norms are obtained as well. Therefore, many norms may be created until a norm with a balanced generality is found. In the context of our work, this approach is similar to applying a search for efficient norms in sequences of generality sets starting in G^{min}.

In [23], the previous authors improve the method to obtain a network of norms through an ontology that represents a taxonomy of terms. In the same way as before, norms are created to avoid conflicts as they occur (the reachability of the system goal is neither considered). However, when two or more norms are potentially generalizable, a new norm is created as a copy of such norms replacing the terms of the atomic formulas for more general terms according to the ontology. This way, there is no need that all specific norms are provided. Although such an approach may decrease the occurrence of conflicts more than in [24], it also needs to start the search by the most specif norms.

We believe that through our approach, where we assume that there exists a region where possible efficient norms are placed, the number of attempts to find them may be fewer than in the previous approaches. However, our work is the first step in such a direction, and we intend to further explore such hypotheses in future works.

6 Conclusion

In this work, we propose an approach for exploring the generality of norms through a syntactic perspective. To achieve this, we introduced some alternatives that exploit the syntactic characteristics of norms and organize the norms space through different sequences where the norms are ordered approximately from the most specific to the most general. Considering that norms that are too specific tend to regulate a few behaviors of agents, which may result in them being insufficient to avoid conflict, and norms that are too general tend to over-regulate the agents, which may prevent the agents from performing the actions necessary for the system goal, we argue that a norm to be efficient should have a balanced generality, i.e., it should be neither too specific nor too general. Therefore, according to this assumption, efficient norms are likely not located in the extremes of the sequences. However, the position of an efficient norm in a sequence is affected by the syntactic characteristics exploited in the calculus of the generality of norms.

As reported in our experiments, alternatives that only explore the variables of norms do not tend to position efficient norms in a specific region of a sequence. In such sequences, efficient norms are found on both the left and right sides. For this reason, we argue that such alternatives are likely not suitable options for organizing the norms space. On the other hand, alternatives that explore at least the atomic formulas of norms tend to position efficient norms in sequences from the middle to the right. For this reason, we also argue that such alternatives turn out to be more appropriate options to organize the norms space.

We also discussed that these results may be used as a heuristic to guide a search for efficient norms. Thus, according to the alternative used to obtain a sequence of norms, the heuristic may indicate in which region a search may start. Therefore, part of the norms space may be pruned in case an efficient norm is found around such a region. As future works, we plan to: i) explore scenarios where other regulative norms, such as permissions and obligations, are necessary to regulate the agents' society with efficiency; ii) characterize a heuristic considering the previous results; iii) develop search strategies for efficient norms that use the heuristic; and iv) identify which alternative to obtain a sequence of norms and strategy may improve the search for efficient norms.

References

[1] W. Alshabi, S. Ramaswamy, M. Itmi, and H. Abdulrab. Coordination, cooperation and conflict resolution in multi-agent systems. In Tarek Sobh, editor, *Innovations and Advanced Techniques in Computer and Information Sciences and Engineering*, pages 495–500, Dordrecht, 2007. Springer Netherlands.

[2] J. Alves, J. F. Hübner, and J. Marchi. The impact of norms generality on mas goal. In *Proceedings of the 15th Workshop-School on Agents, Environments, and Applications - WESAAC*, pages 83–94, Rio de Janeiro, Brazil, 2021.

[3] K. G. Binmore. *Natural Justice*. Oxford scholarship online. Economics and finance module. Oxford University Press, 2005.

[4] G. Boella and L. Lesmo. Deliberate normative agents. 05 2001.

[5] G. Boella and L. van der Torre. Regulative and constitutive norms in normative multiagent systems. In *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning*, KR'04, pages 255–265. AAAI Press, 2004.

[6] G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computation and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems*, 12(2-3):71–79, oct 2006.

[7] M. Bramer. *Logic Programming with Prolog.* Springer Publishing Company, Incorporated, 2nd edition, 2014.

[8] N. Bulling and M. Dastani. Verifying normative behaviour via normative mechanism design. pages 103–108, 01 2011.

[9] H. L. Cardoso and E. Oliveira. Institutional reality and norms: Specifying and monitoring agent organizations. *International Journal of Cooperative Information Systems*, 16:67–95, 03 2007.

[10] G. Christelis and M. Rovatsos. Automated norm synthesis in an agent-based planning environment. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, volume 1 of *AAMAS '09*, pages 161–168, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems.

[11] R. Conte and C. Castelfranchi. Understanding the effects of norms in social groups through simulation. *Artificial Societies: The Computer Simulation of Social Life*, 01 2006.

[12] N. Criado, E. Argente, and V. Botti. Open issues for normative multi-agent systems. *AI Commun.*, 24:233–264, 01 2011.

[13] F. J. P. da Cunha, T. F. M. Sirqueira, M. L. Viana, and C. J. P. de Lucena. Extending bdi multiagent systems with agent norms. *International Journal of Computer and Information Engineering*, 12(5):302 – 309, 2018.

[14] M. Dastani, J. Dix, H. Verhagen, and S. Villata. Normative Multi-Agent Systems (Dagstuhl Seminar 18171). *Dagstuhl Reports*, 8(4):72–103, 2018.

[15] I. J. B. do Nascimento, N. Cacic, H. M. Abdulazeem, T. C. von Groote, U. Jayarajah, I. Weerasekara, M. A. Esfahani, V. T. Civile, A. Marusic, A. Jeroncic, N. C. Junior, T. P. Pericic, I. Zakarija-Grkovic, S. M. M. Guimarães, N. L. Bragazzi, M. Bjorklund, A. Sofi-Mahmudi, M. Altujjar, M. Tian, D. M. C. Arcani, D. P. O'Mathúna, and M. S. Marcolino. Novel coronavirus infection (covid-19) in humans: A scoping review and meta-analysis. *Journal of Clinical Medicine*, 9(4), 2020.

[16] M.A. Eggert, M. Eggert, and W. Falzon. *The Resolving Conflict Pocketbook.* Management Pocketbooks Series. Management Pocketbooks, 2004.

[17] L. Ekenberg. Detecting conflicts in multi-agent systems. In C. Zhang and D. Lukose, editors, *Multi-Agent Systems: Methodologies and Applications.* Lecture Notes in Artificial Intelligence 1286, Springer-Verlag, 1997.

[18] D. Fitoussi and M. Tennenholtz. Minimal social laws. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, pages 26–31, 1998.

[19] D. Fitoussi and M. Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence*, 119(1):61–101, 2000.

[20] A. García-Camino, P. Noriega, and J. Rodríguez-Aguilar. Implementing norms in electronic institutions. pages 667–673, 01 2005.

[21] K. V. Hindriks, F. S. De Boer, W. van der Hoek, and J. Meye. Agent programming with declarative goals. 01 2001.

[22] N. Khan. Critical review of sampling techniques in the research process in the world. 2020.

[23] J. Morales, M. López-Sánchez, J. Rodríguez-Aguilar, M. Wooldridge, and W. Vasconcelos. Minimality and simplicity in the on-line automated synthesis of normative systems. In *AAMAS*, 2014.

[24] J. Morales, M. Lopez-Sanchez, J. A. Rodriguez-Aguilar, M. Wooldridge, and W. Vasconcelos. Automated synthesis of normative systems. In *Proc. of AAMAS 2013*, pages 483–490, 2013.

[25] I. Pratt. *Closed World Assumptions*, pages 65–84. Macmillan Education UK, London, 1994.

[26] T. Ågotnes and M. Wooldridge. Optimal social laws. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, page 667â674, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

[27] S. Resmerita and M. Heymann. Conflict resolution in multi-agent systems. volume 3, pages 2537 – 2542 Vol.3, 01 2004.

[28] M. Riad and F. Golpayegani. Run-time norms synthesis in multi-objective multi-agent systems. *CoRR*, abs/2105.00124, 2021.

[29] J. A. Robinson. Computational logic: The unification computation. *Machine intelligence*, 6:63–72, 1971.

[30] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 edition, 2010.

[31] F. P. Santos. Social norms of cooperation in multiagent systems. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, page 1859â1860, Richland, SC, 2017. International Foundation for Autonomous Agents and Multiagent Systems.

[32] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 73(1):231–252, 1995.

[33] C. Tessier, L. Chaudron, and H. Müller, editors. *Conflicting Agents: Conflict Management in Multi-Agent Systems*. Kluwer Academic Publishers, USA, 2001.

[34] M. B. van Riemsdijk, M. Dastani, and M. Winikoff. Goals in agent systems: a unifying framework. pages 713–720, 01 2008.

[35] T. Yang, J. Hao, Z. Meng, and Z. Wang. *Norm Emergence in Multiagent Systems*, chapter Chapter 8, pages 179–206. 2018.