Re-checking Normative System Coherence

Thomas C. King, Virginia Dignum, and M. Birna van Riemsdijk

TU Delft, Delft, The Netherlands
{t.c.king-1, m.v.dignum, m.b.vanriemsdijk}@tudelft.nl

Abstract. Sets of related norms (normative systems) are likely to evolve due to changing goals of an organization or changing values of a society, this may introduce incoherence, such as the simultaneous prohibition and obligation of an action or a set of deadlocked duties. This paper presents a compositional framework that may be used for detecting whether normative systems are coherent by analysing traces of actions and their legality. Unlike other mechanisms for checking normative system coherence, the framework makes it possible to re-check just those parts of the system that have changed, without re-checking the entirety.

1 Introduction

Increasingly, Multi-Agent Systems (MAS) are applied to solving a diverse range of problems, benefiting from available heterogeneous agents by providing an open system to which they may join. Although agents may be asked to do one thing, their autonomy can lead to behaviour different from what is desired [14]. Consequently, organisations are used to direct and constrain agents into achieving particular goals, by giving them social norms that specify what an agent ought to do in a given context and sanctions to deter them from disobeying.

It is difficult to design sets of norms (normative systems) where satisfying some norms does not cause agents to violate others (known as coherence), hence the extensive research on identifying incoherent normative systems [1, 2, 4, 7, 10–13].

Normative systems may also change and evolve over time (through the addition, deletion and/or modification of norms). This can be due to changed goals of an organisation, changed values of society or existing norms being shown to be inadequate (e.g. first introducing a speed limit and then later increasing or decreasing the limit). So, it is also important to *re-check* them for coherence.

Yet, it is undesirable to do completely new checks on normative systems if only a small part has changed. For this reason, this paper focuses on a structured and compositional means of re-checking just those parts of normative systems that have changed.

It is assumed that the validity of the compositional semantics presented here depends on the expressiveness of the framework. Re-checking has not been specifically examined before, so this paper proposes the following concepts that are interesting enough to, at a minimum, have the expressiveness required to argue the compositional semantics are useful:

- *Logical relationships* between norms, for example in many cases it may be required to stipulate that an agent ought to do a or ought to do b.
- Norms with a *condition*, *consequence* and/or *deadline* that may be the condition, consequence and/or deadline of other norms. First described by López and Luck as *interlocking norms* [8], their existence means incoherence may arise due to deadlock. This complicates the check for coherence. Frameworks which consider interlocking norms include those proposed by López, Luck et al. [8, 9] and later by Jiang et al. [6].
- Secondary norms that may act as a *sanction* for "fixing" the violation of a primary norm. Sanctions express what ideally and sub-ideally ought to be done. For example 'a person ought not steal' may be what ideally ought to be done and 'if someone steals, they ought to pay a fine' is a sanction for violating the primary norm.

The results of this paper make it possible to determine if a normative system is coherent in a compositional way that may make use of checks on previous versions of a normative system. This is especially useful for checking changes to a normative system before they are implemented, that is, before run-time. The approach taken is to first formally define the key concepts in a conceptual framework, such as the legality of actions and how norms may be structured (Section 2). Section 3 is the main contribution of a framework for compositionally determining what may and may not be legally done with respect to a given normative system, it is here that one possible semantics of norms is also given. In Section 4 the framework examples are given for illustrating checking and re-checking a system, using a running example on a shoplifting offence with a potential fine. The results from using the compositional semantics may be used to check the coherence of a normative system, what makes a coherent normative system is discussed in Section 5. Section 6 gives the relevant work surrounding efficient norm coherence checking. In Section 7, conclusions and directions for future work are presented.

2 The Normative Conceptual Framework

This section follows the conceptual normative framework of Jiang et al. [6] with some minor syntax changes. The conceptual framework gives an abstract representation of social norms, the relationships between social norms and the legality of actions.

In the following, let the set A be the set of all agents with typical element a and Act be the set of all actions with typical element φ.

A normative trace is an alternating sequence of zero or more agent/action pairs and the legal state of the sequence up until that point (denoting the legality of the preceding agent/action pairs). For simplicity it is assumed each agent/action pair occurs at most once, consequently if A and Act are finite sets then the set of all possible traces for $(A \times Act)$ is also finite. The legal states stipulate whether the preceding sequence is compliant (c) but with the possibility for there to be a violation in the future, in violation (v) but may or may not become completely compliant (cnd) in the future, or completely compliant with no possibility of there being a violation in the future (cnd). Formally:

Definition 1. *(Normative Trace). A normative trace nt is a finite sequence of alternating elements of the form: $[l_0, (a, \varphi)_1, l_1, ..., (a, \varphi)_n, l_n]$ where $l_i \in \{cnd, c, v\}$, $(a, \varphi) \in (A \times Act)$ and $(a, \varphi)_j \neq (a, \varphi)_k$ for $0 \leq i \leq n$, $1 \leq j < k \leq n$.*

From here on the variables X, Y and Z will be used to denote agent/action pairs s.t. $X, Y, Z \in (A \times Act)$ and the variables g and l will be used to denote legal states s.t. $g, l \in \{cnd, c, v\}$.

Norms in a normative system have a deontic modality indicating whether they are an obligation or prohibition, permissions are not considered for simplicity:

Definition 2. *(Deontic Types). A deontic type d is a member of the set of deontic types $D = \{O, F\}$ where:*

- O - *Means that it is* obligatory *to carry out the action to which it applies.*
- F - *Means that it is* prohibited *to carry out the action to which it applies.*

Given the definition of deontic types, a norm may express that an agent is either obligated or forbidden to carry out an action under some (pre)condition before a deadline, if there is no (pre)condition or deadline, 'null' is used:

Definition 3. *(Norm). Let $d \in D$, $\rho \in (A \times Act)$, $\delta, \sigma \in (A \times Act) \cup \{null\}$ a norm is $n = (d(\rho) \leq \delta/\sigma)$ where:*

- *ρ is the agent/action pair which is obligatory or forbidden.*
- *δ is the non-temporal deadline of the norm.*
- *σ is the precondition of the norm.*

Norms may be related to other norms, in a norm net, via a logical connective. Such a relation is defined as a norm net in [6] and defined similarly here. This makes it possible to express different conditions of when a sequence of actions is legal with respect to two child nodes of a norm net:

- The sequence should be legal with both (AND) child nodes (you ought to do this and you ought to do that).
- The sequence should be legal with just one child node (OR) (you ought to do this or you ought to do that).
- The sequence either should be legal with the primary legislature or else (OE) it should be legal with the sanctioning legislature, but never both (ideally you ought to do this, if and only if you are not then you ought to do that).

Definition 4. *(Norm Net). Let n be a norm, a norm net NN is a formula in the following BNF grammar:*

$$NN ::= n \mid AND(NN, NN) \mid OR(NN, NN) \mid OE(NN, NN)$$

Norm nets are used to formalize an example of shoplifting where two agents are considered, a person and a security guard:

Example 1. $OE(n_1, n_2)$ A person ought not shoplift, $n_1 = F((p, \mathit{shoplift}) \leq \mathit{null}/\mathit{null})$. If a person tries to shoplift, then a security guard ought to give them a fine before they let them go $n_2 = O((s, \mathit{fine}) \leq (p, \mathit{let\_go})/(p, \mathit{shoplift}))$.

3 Compositional Semantic Framework

We define the semantics of a norm net as a set of normative traces. Coherence of a norm can then be defined as a property of this set of traces, e.g., at least one of the normative traces in the set has 'compliant' as its final legality, expressing that there is a way to satisfy all norms such that one ends up in a compliant state. In this section we define how to generate this set of traces.

The basic idea of how to generate the set of traces is to follow the compositional tree structure of norm nets. That is, we first define the semantics for obligations and prohibitions as a set of normative traces, which form the leaves of the tree (Section 3.1). Then we compose these sets of traces according to the tree structure of the norm net, taking into account the normative connectives in the nodes. For this we need to combine (i.e., interleave) the traces in the respective sets, computing the legality of the combined traces by combining the legality states of the constituent traces according to the normative connectives. We define how to combine legalities in Section 3.2 and define how to compose traces informally in Section 3.3 and formally in 3.4.

3.1 Norm Semantics

The general idea behind the framework as a whole is to produce a set of normative traces, $\mathit{traces}(NN)$, that expresses all those actions a norm net commands agents to do or not do and stipulates the legality of doing them. For example, given the norm that a security guard should fine a shoplifter before letting them go, $O((s, \mathit{fine}) \leq (p, \mathit{let\_go})/(p, \mathit{shoplift}))$, a possible set of normative traces is:

$$\{[c, (p, \mathit{shoplift}), c, (s, \mathit{fine}), \mathit{cnd}, (p, \mathit{let\_go}), \mathit{cnd}]$$
$$[c, (p, \mathit{shoplift}), c, (p, \mathit{let\_go}), v, (s, \mathit{fine}), v], [c]\}$$

However, these traces would suggest a norm is commanding the agent p to shoplift or do nothing, this is clearly not the case because shoplifting is a condition and not a consequence. We would not expect a security guard to fine someone unless the shoplifting offence occurred and we may therefore only wish to test whether fining is legal if we believe shoplifting ought to occur or will occur.

Therefore, from now on agent/action pairs which are things there is no reason to believe ought to be done, will be marked with $-$ as in $(a, \phi)^-$. We also introduce the concept that two agent/action pairs from different traces are loosely equal, $\approx$, if one agent/action is the same as the other regardless of if there is a marking. Furthermore, the concept of an agent/action pair being a member, $\in$, of a normative trace is also given.

Definition 5. *(Agent/Action Markings, Loose Equality, Membership and Ordering).*

- X *may have a marking* $^-$ *s.t.* $X = (a, \phi)^-$ *denotes that* X *is only found in the condition of a norm.*
- $X \approx Y$ *holds if* ($X = (a, \phi)^-$ *and* $Y = (a, \phi)^-$) *or* ($X = (a, \phi)^-$ *and* $Y = (a, \phi)$) *or* ($X = (a, \phi)$ *and* $Y = (a, \phi)^-$) *or* ($X = (a, \phi)$ *and* $Y = (a, \phi)$).
- $X \not\approx Y \leftrightarrow \neg(X \approx Y)$
- *Given a normative trace* nt *let* $X \in nt$ *denote that there is a* Y *in the normative trace* nt *s.t.* $X \approx Y$.
- $X \notin nt \leftrightarrow \neg(X \in nt)$
- *Given a normative trace* nt *and two agent action pairs* X *and* Y *let* $X <_{nt} Y$ *denote* $X, Y \in nt$ *and* X *occurs before* Y *in* nt.

The general idea behind producing the set of traces for an obligation is that if there is a condition then we give the traces where the condition is met and we also give the possibility of doing nothing ($[c]$), because we do not believe the condition ought to be met and consequently no duties ought to arise. The set of traces should also convey the other possibilities, which are, carrying out the duty before any deadline that there may be (as ought to be done) or the deadline occurring before the duty is carried out (as ought not be done). Therefore, the set of traces for an obligatory norm n are:

Definition 6. *(Traces for Obligation). Let* $n = (O(\rho) \leq \delta/\sigma)$

If $\delta \neq null$ *and* $\sigma \neq null$ *then:*

$$\begin{aligned} traces(n) = \{&[c], [c, \sigma^-, c, \rho, cnd, \delta, cnd], [c, \sigma^-, c, \delta^-, v, \rho, v], \\ &[c, \delta^-, c, \sigma^-, v, \rho, v], [c, \delta, c, \rho, c, \sigma^-, v], \\ &[c, \rho, cnd, \sigma^-, cnd, \delta^-, cnd], [c, \rho, cnd, \delta^-, cnd, \sigma^-, cnd]\} \end{aligned}$$

If $\delta = null$ *and* $\sigma \neq null$ *then:*

$$traces(n) = \{[c], [c, \sigma^-, c, \rho, cnd], [c, \rho, cnd, \sigma^-, cnd]\}$$

If $\delta \neq null$ *and* $\sigma = null$ *then:*

$$traces(n) = \{[c, \rho, cnd, \delta^-, cnd], [c, \delta^-, v, \rho, v]\}$$

If $\delta = null$ *and* $\sigma = null$ *then:*

$$traces(n) = \{[c, \rho, cnd]\}$$

Like obligations, if a prohibition has a condition then the set of traces for it also includes doing nothing. Where prohibition differs is that we wish to generate traces that convey the prohibited action ought not be done. That is, the subject either ought to do nothing (in the case of no deadline) or see to it that a deadline occurs first.

We also note the following is true $(O(\rho) \leq \delta/\sigma) \equiv (F(\delta) \leq \rho/\sigma)$. This means the previous shoplifting example may be rephrased as "if someone shoplifts the security guard ought not let them go before giving them a fine". Formally, the traces for a prohibition n are:

Definition 7. *(Traces for Prohibition) Let* $n = (F(\rho) \leq \delta/\sigma)$

If $\delta \neq null$ *and* $\sigma \neq null$ *then:*

$$traces(n) = \{[c], [c, \sigma^-, c, \rho^-, v, \delta, v], [c, \sigma^-, c, \delta, cnd, \rho^-, cnd], [c, \delta, cnd, \sigma^-, cnd, \rho^-, cnd], [c, \delta, cnd, \rho^-, cnd, \sigma^-, cnd], [c, \rho^-, c, \sigma^-, v, \delta, v], [c, \rho^-, c, \delta, c, \sigma^-, v]\}$$

If $\delta = null$ *and* $\sigma \neq null$ *then:*

$$traces(n) = \{[c], [c, \sigma^-, c, \rho^-, v], [c, \rho^-, c, \sigma^-, v]\}$$

If $\delta \neq null$ *and* $\sigma = null$ *then:*

$$traces(n) = \{[c, \rho^-, v, \delta, v], [c, \delta, cnd, \rho^-, cnd]\}$$

If $\delta = null$ *and* $\sigma = null$ *then:*

$$traces(n) = \{[c], [c, \rho^-, v]\}$$

3.2 Connective Semantics

The general idea is that the legality of a sequence of actions with respect to a norm net may be composed from the legality with the norm net's child nodes using the semantics for the connectives defined here. OR and AND follow their counterparts in boolean logic. OE is given semantics for expressing, possibly cascading, sanctions. Sanctions 'fix' violations, e.g. given "a person ought not shoplift, or else you must pay a fine before a deadline.", paying a fine 'fixes' a shoplifting offence.

The semantics of sanctions gives rise to a three-valued legality system of compliance with no outstanding duties (cnd), compliance (c) and violation (v). The two compliance states distinguish between when there is compliance (c) with a sanction because the deadline has not passed (e.g. when we are waiting for a fine to be paid) or when there is compliance and no further duties (cnd) because it has been fulfilled (e.g. paying a fine before the deadline).

Given the aforementioned descriptions, the semantics for the connectives are defined in terms of the legality function and the following tables for each connective (to be interpreted in the same way as truth tables for a logic):

Definition 8. *(Legality Semantics). Let* $g, l \in \{cnd, c, v\}$*,* $conn \in \{AND, OR, OE\}$ *the following tables give the results of the function* $leg(conn, g, l)$*:*

$conn = AND$		l		
		cnd	**c**	**v**
g	**cnd**	cnd	c	v
	c	c	c	v
	v	v	v	v

$conn = OR$		l		
		cnd	**c**	**v**
g	**cnd**	cnd	cnd	cnd
	c	cnd	c	c
	v	cnd	c	v

$conn = OE$		l		
		cnd	***c***	***v***
g	***cnd***	v	c	cnd
	c	v	c	c
	v	cnd	v	v

The OE connective should be used in a particularly way. Firstly, the sanctioning norm should have a condition if we do not wish to consider the sanction may occur before the violation. Secondly, if the secondary norm is a prohibition used to 'fix' the primary norm then it should have a deadline.

Without a deadline for a sanctioning prohibition, compliance with it will not fix the primary norm because the subject of the prohibition may yet violate it. For example, if someone steals and consequently they ought not visit a shop again, but there is no deadline on the shopping ban, then there is no way of telling whether by not visiting the shop they have fixed the shoplifting offence or they are merely postponing violating the ban (and so the shoplifting offence would never be fixed by observing the ban). As Governatori et al. put it, legislators need to use deadlines for sanctions to be enforced and to represent a hierarchy of what ought to be done [3].

3.3 Informal Compositional Semantics

Previously, the set of traces that express what an individual norm stipulates should and should not be done were defined. When combining individual norms through a connective, the set of traces for the resulting norm net should describe those things that the norm net as a whole says should and should not be done.

Interleaving traces The basic idea of combining two sets of normative traces through a connective is to do a pairwise interleaving of the traces in the two sets and compute the legality of the resulting traces by applying the connective semantics on the legality of the constituent traces. That is, we compute interleavings for all combinations of traces from the two sets (preserving the ordering of agent/action pairs), where after a sequence of agent/action pairs each legal state in an interleaved trace is composed from the legal state of each contributory trace that comes after the same sequence agent/action pairs and before any agent/action pairs yet to occur in the interleaving. For example, consider two normative traces $[c, (p_1, eat), v, (p_1, think), v]$ and $[c, (p_1, work), cnd, (p_1, rest), cnd]$ and assuming we want to combine them through an OR connective. Then we get the following set of interleaved traces:

$$
\begin{aligned}
\{&[c, (p_1, eat), v, (p_1, think), v, (p_1, work), cnd, (p_1, rest), cnd],\\
&[c, (p_1, eat), v, (p_1, work), cnd, (p_1, think), cnd, (p_1, rest), cnd],\\
&[c, (p_1, eat), v, (p_1, work), cnd, (p_1, rest), cnd, (p_1, think), cnd],\\
&[c, (p_1, work), cnd, (p_1, eat), cnd, (p_1, think), cnd, (p_1, rest), cnd],\\
&[c, (p_1, work), cnd, (p_1, eat), v, (p_1, rest), cnd, (p_1, think), cnd],\\
&[c, (p_1, work), cnd, (p_1, rest), cnd, (p_1, eat), cnd, (p_1, think), cnd]\}
\end{aligned}
$$

The resulting set expresses what should and should not be done in the composed system.

Minimality In the example above, the agent/action pairs occurring in the traces are disjoint. If these are (partly) overlapping, we need to take several additional considerations into account when composing the traces. We start by considering the case of one overlapping agent/action pair as in the following example:

Example 2. $NN = AND(n_1, n_2)$. A person ought to eat before they go out $n_1 = O((p, eat) \leq (p, go\_out)/null)$ and they ought not drink before they eat $n_2 = F((p, drink) \leq (p, eat)/null)$.

$$traces(n_1) = \{[c, (p, eat), cnd, (p, go\_out), cnd], [c, (p, go\_out), v, (p, eat), v]\}$$
$$traces(n_2) = \{[c, (p, eat), cnd, (p, drink), cnd], [c, (p, drink), v, (p, eat), v]\}$$

We can see that both $traces(n_1)$ as well as $traces(n_2)$ refer to p eating. Thus both of these sets have something to say about whether eating could lead to a violation: the traces of $traces(n_1)$ express that eating before going out is okay, but they state nothing about drinking, whilst the traces for norm n_2 stipulate that eating leads to a compliant state if it is before drinking.

We interleave traces with an overlapping agent/action pair such that the ordering of agent/action pairs is preserved, and the overlapping pair occurs only once in the trace, in accordance with the definition of normative traces. Intuitively, we strive for a kind of "minimality" of traces that still allows us to derive conclusions concerning coherence of the norm net. Thus when composing traces from $[c, (p, eat), cnd, (p, go\_out), cnd]$ and $[c, (p, eat), cnd, (p, drink), cnd]$, the resulting set of traces is:

$$\{[c, (p, eat), cnd, (p, go\_out), cnd, (p, drink), cnd],$$
$$[c, (p, eat), cnd, (p, drink), cnd, (p, go\_out), cnd]$$

The legality of the first trace after eating is composed from the legality after eating in the contributing traces. Whilst the last legality is composed from the legality after eating and going out in the first contributory trace and the legality after eating and drinking in the second contributory trace.

Compatibility Now we consider the case of composing two traces that have multiple overlapping agent/action pairs, which induces a second aspect to take into consideration. These overlapping pairs are either in the same order or in a different order in the two traces. If they are in a different order, they consider different cases. For example, $[c, (p, eat), cnd, (p, go\_out), cnd]$ and $[c, (p, go\_out), v, (p, eat), v]$. Intuitively, these traces express properties of different situations that cannot be considered jointly, i.e., the traces are not *compatible.* Thus we do not compute interleavings for incompatible traces.

Maximality Finally we identify one more case in which we do not compute interleavings. Take the following example:

$$S1 = \{[c, W, cnd, X, v], [c, W, cnd]\} \quad S2 = \{[c, W, cnd, X, cnd, Y, cnd]\}$$

In identifying traces in S1 with which we can combine the trace $[c, W, cnd, X, cnd, Y, cnd]$ from S2, one may expect that this trace should be combined with all traces from S1. However, if it is combined with $[c, W, cnd]$ then although the resulting trace would take into account what S1 says about performing W, it would not take into account what it says about performing W and then X, therefore the result would not be composed of all of the 'facts' stated by S1. Thus, the idea is to take only "maximal" traces, where maximal means that a trace should only be combined with another if there does not exist another trace in the same set that says more about the trace with which it is being combined.

3.4 Formal Compositional Semantics

The formal semantics are given in terms of the informal requirements outlined in the previous section. We wish to only interleave those traces with the same ordering of agent/action pairs, a symmetric relation $compatible(nt^1, nt^2)$ is defined for traces that meet the **compatibility** requirement:

Definition 9. *(Compatible Normative Traces). For two normative traces nt^1 and nt^2, $compatible(nt^1, nt^2)$ holds iff:*

$$\forall X \forall Y (X <_{nt^1} Y) : X <_{nt^2} Y$$

The idea behind the semantics of interleaving compatible traces is to first create a triple, $\langle nt^1, nt^2, result\rangle$, of two compatible normative traces nt^1 and nt^2 and an empty trace, $result$, that will become an interleaving of the two. Then, a system of transition rules is repeatedly applied to this triple, taking the first elements off nt^1 and nt^2, adding them to the result. After an action is added to the result, so is a legal state composed from the legal states of the last actions added from nt^1 and nt^2. This is done until nt^1 and nt^2 are empty and thus the trace $result$ is an interleaving of the two.

To meet the requirement of **minimality** the transition system should 'merge' agent/action pairs from the traces if they are the same, rather than add the same agent/action pair twice. However, agent/action pairs may be the same yet have different markings. An operation is defined to only maintain markings signifying an action is a condition if both agent/action pairs have the marking:

Definition 10. *(Composing Markings). Let X and Y be two agent/action pairs with the same agent and action (a, ϕ). The function $comp(X, Y)$ is defined as:*

$$\begin{aligned} &comp(X, Y) = \\ &(a, \phi)^-, \textit{ iff } X = (a, \phi)^- \textit{ and } Y = (a, \phi)^- \\ &\quad (a, \phi), \textit{ otherwise} \end{aligned}$$

Each of the following rules of the transition system are just for a single step of the interleaving operation. Traces are merged with respect to a connective, thus the transition rules include a connective in their definitions $c \in \{AND, OR, OE\}$.

The following transition rule defines how to progress with the interleaving if the next agent/action pair in both traces being interleaved is the same (as in the first condition). We do this by merging the agent/action pairs and adding them with the correct markings to the result (performed by the second condition). The last condition expresses that the new legal state is composed from the legalities that occur in each trace after the agent/action pair that is being added.

$$\frac{\langle [l_0, X_1, l_1, \mathit{left\_seq}],\ \ [g_0, Y_1, g_1, \mathit{right\_seq}],\ \ [\mathit{result}]\rangle \quad X_1 = Y_1 \quad Z = \mathit{comp}(X_1, Y_1) \quad l' = \mathit{leg}(c, l_1, g_1)}{\langle [l_1, \mathit{left\_seq}],\ \ [g_1, \mathit{right\_seq}],\ \ [\mathit{result}, Z, l']\rangle}\ \mathit{Merge}$$

The next transition rule defines how to progress if the next agent/action pairs in the left and right traces are different (stipulated by the first condition) and thus a choice must be made to add one of them to the interleaving result (this rule is for choosing the agent/action pair from the left trace). The second condition states that this choice can only be made if the next action in the left trace is not found somewhere else in the right, this stops the same agent/action pair being added again (preserving **minimality**). The final rule composes the new legal state for the interleaving from the legality of the agent/action pair in the left trace being added and the legality of the last agent/action pair added from the right trace. Thus, the new legality takes into account what both traces being interleaved say about the sequence of actions up until that point.

$$\frac{\langle [l_0, X_1, l_1, \mathit{left\_seq}],\ \ [g_0, Y_1, g_1, \mathit{right\_seq}],\ \ \mathit{result}\rangle \quad X_1 \not\approx Y_1 \quad X_1 \notin \mathit{right\_seq} \quad l' = \mathit{leg}(c, l_1, g_0)}{\langle [l_1, \mathit{left\_seq}],\ \ [g_0, Y_1, g_1, \mathit{right\_seq}],\ \ [\mathit{result}, X_1, l']\rangle}\ \mathit{Arbitrary\ Choice\ 1}$$

The final transition rule defines how to progress if one trace only contains a legal state (in which case we wish to add the remaining agent/action pairs from the other trace). This may be because a normative trace $[c]$ is being interleaved with a longer trace, or because all of the agent/action pairs from one trace have been added. Here the rule is given for when the right trace only has a legal state, where the first and only condition composes the new legality state in the same way as the aforementioned arbitrary choice rule.

$$\frac{\langle [l_0, X_1, l_1, \mathit{left\_seq}],\ \ [g_0],\ \ \mathit{result}\rangle \quad l' = \mathit{leg}(c, l_1, g_0)}{\langle [l_1, \mathit{left\_seq}],\ \ [g_0],\ \ [\mathit{result}, X_1, l']\rangle}\ \mathit{Exhausted\ Choices\ 1}$$

The transition system $\Sigma(c)$ where $c \in \{AND, OR, OE\}$ is defined as consisting of the rules above, the rule symmetric to the rule 'Arbitrary Choice 1', the rule symmetric to the rule 'Exhausted Choices Trace 1' (left out for brevity) and the variable c in each rule substituted with the value of c in $\Sigma(c)$.

The set of traces for a norm net NN, $\mathit{traces}(NN)$, may be composed from the sets of traces for the child nodes of NN. The idea is to take all those compatible pairs of traces for the child nodes and produce all interleavings of them by applying the rules of $\Sigma(c)$ until all possibilities are exhausted. However, the requirement for **maximality** should be observed such that a trace on the left side should not be interleaved with a trace on the right if there is another trace on the right with more information for the resulting interleaved trace and vice versa. We approach this problem by defining the concept of subsumption and only interleave those traces that are not subsumed by others. If given three compatible traces nt^1, nt^2 and nt^3, nt^1 has all of the agent action pairs in nt^2 and some additional pairs found in nt^3, we say nt^1 subsumes nt^2 with respect to nt^3. A predicate $subsume(nt^1, nt^2, nt^3)$ is defined for such a relationship:

Definition 11. *(Normative Trace Subsumption). Let:*

nt^1, nt^2 and nt^3 be normative traces. $subsume(nt^1, nt^2, nt^3)$ holds iff:

$$\begin{aligned}
&compatible(nt^1, nt^2) \wedge compatible(nt^1, nt^3) \wedge compatible(nt^2, nt^3)\\
&\wedge \forall X \in nt^2, nt^3, \exists Y \in nt^1 : X \approx Y\\
&\wedge \exists X \in nt^1, nt^3, \forall Y \in nt^2 : X \not\approx Y
\end{aligned}$$

For two sets of normative traces NT^1 and NT^2, the pairs of normative traces that are compatible but not subsumed by other traces in the same set are in the set $maximal(NT^1, NT^2)$:

Definition 12. *(Maximality Set). Let NT^1, NT^2 be two sets of normative traces. The set of pairs of traces $maximal(NT^1, NT^2)$ is defined as:*

$$\begin{aligned}
maximal(NT^1, NT^2) = \{\langle nt^1, nt^2\rangle : (&\exists nt^1, \forall nt^{1\prime} \in NT^1, \exists nt^2, \forall nt^{2\prime} \in NT^2)\\
&\big[compatible(nt^1, nt^2) \wedge \neg subsume(nt^{1\prime}, nt^1, nt^2)\\
&\wedge \neg subsume(nt^{2\prime}, nt^2, nt^1)\big]\}
\end{aligned}$$

Performing the interleavings for an entire tree produces the full set of traces for a norm net (see Algorithm 1). If the traces for a particular node are already computed (i.e. cached), then they may be re-used so long as the node has not changed. Thus *re-checking* of a norm net avoids a check on the entire structure.

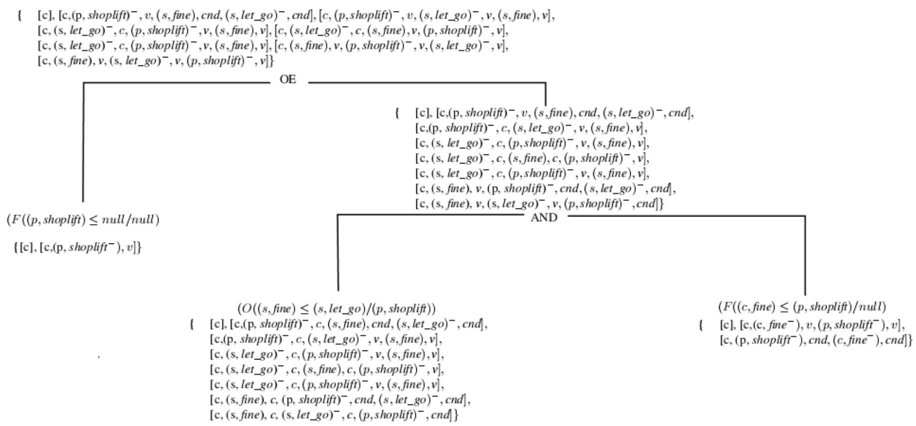

Fig. 1. The results of a compositional computation on a normative system for a shoplifting offence

---

Algorithm 1 ComputeTraces(*NormNet*)

Require: *NormNet* is a norm net in the grammar of *NN*
Ensure: The set of all traces for *NormNet*
  if *NormNet* is a norm **then**
    $traces \leftarrow traces(NormNet)$
  else *NormNet* is a norm net $c(NN_1, NN_2)$ where $c \in \{AND, OR, OE\}$
    if the traces of *NormNet* are cached **then**
      $traces \leftarrow cached(NormNet)$
    else
      $LT \leftarrow$ ComputeTraces(NN_1)
      $RT \leftarrow$ ComputeTraces(NN_2)
      $traces \leftarrow \emptyset$
      for all $\langle lt, rt\rangle \in maximal(LT, RT)$ **do**
        $traces \leftarrow traces \cup$ all possible values for *result* produced by applying $\Sigma(c)$ to $\langle lt, rt, result\rangle$
      end for
    end if
  end if
  return traces

---

4 Examples

The framework is illustrated by formalising Example 3, an extended version of the example on shoplifting and a security guard's responsibilities.

Example 3. (Shoplifting). If a person p shoplifts, they should be fined by a security guard s before the shoplifter leaves the shop. The security guard should not fine the person p before they shoplift. This is formalised as a norm net $NN_1 =$

$OE(n_1, NN_2)$, where $n_1 = F((p, \textit{shoplift}) \leq \textit{null}/\textit{null})$, $NN_2 = AND(n_3, n_4)$, $n_3 = O((s, \textit{fine}) \leq \textit{leave\_}/\textit{null})$ and $n_4 = F((s, \textit{fine}) \leq (p, \textit{shoplift})/\textit{null})$

We see in Figure 1 that compliance is possible with the system. Using Example 4 we demonstrate that a partial re-check on the system may also be performed if the norm net is revised. In this example, we assume the duties of two criminals are expressed as obligations and criminals in the organisation's norm net structure, this assumption is made so that we can illustrate the system.

Example 4. (Criminals). As before, shoplifters should be fined, the agents p and s are governed by the norm net NN_1 from the previous shoplifting norm net with an OE connective. However, both the person p and the security guard s are now criminals working together, we may represent this as norms in the system (in reality such information would be private), $NN_3 = AND(NN_1, NN_4)$, $NN_4 = AND(n_5, n_6)$, $n_5 = O((p, \textit{shoplift}) \leq \textit{null}/\textit{null})$ and $n_6 = F((s, \textit{fine}) \leq \textit{null}/\textit{null})$.

The norm net NN_3 in the new example gives us the traces:

$$\{[c, (p, \textit{shoplift}), c], [c, (s, \textit{fine})^-, v, (p, \textit{shoplift}), v],$$
$$[c, (p, \textit{shoplift}), c, (s, \textit{fine})^-, v]\}$$

Now we may take the traces computed for the previous version of the normative system and re-use them for computing the result of conjoining these new norms for a norm net $AND(NN_1, textitNN_3)$. As a consequence it is clear that the person p and the security guard s cannot carry out both their duties as employees and as criminals:

$$\{[c, (p, \textit{shoplift}), v, (s, \textit{let\_go})^-, v, (s, \textit{fine}), v]$$
$$[c, (p, \textit{shoplift}), v, (s, \textit{fine}), v, (s, \textit{let\_go}), v], ...\}$$

Furthermore, it appears that there is no opportunity to do nothing, in all cases something has been commanded and in all cases it is also prohibited.

The previous example is on a straightforward conflict between an obligation and a prohibition, but using Example 5 we can also see that the formalism is sufficient for detecting deadlock:

Example 5. $AND(n_1, n_2)$. Consider two people (p_1 and p_2) and the norm that you ought to be the first to apologise, $n_1 = O(p_1, \textit{apologise}) \leq (p_2, \textit{apologise})/\textit{null})$ and $n_2 = O(p_2, \textit{apologise}) \leq (p_1, \textit{apologise})//\textit{null})$.

The set of traces for this example show that it is not possible for both people to comply with the system, because it obligates one thing to be done before another whilst also obligating the opposite of this:

$$\{[c, (p_1, \textit{apologise}), v, (p_2, \textit{apologise}), v], [c, (p_2, \textit{apologise}), v, (p_1, \textit{apologise}), v]\}$$

These examples show the framework can be used to detect conflicts between obligations and prohibitions and deadlock, and for a norm net to be re-checked for these properties.

However, we note that in one example there was a marking signifying that we do not expect the security guard to let someone go before fining them because they have not been told to. Whether this is a good notion to have in the trace is unclear, it does not seem harmful but nor is it particularly useful. We also note a drawback of this approach, namely that the ordering of actions is irrespective of whether they can really happen, thus, it would be advisable to ignore traces that consist of a sequence of events that are impossible.

5 What is a Coherent Norm Net?

We say there is coherence if a certain level of compliance is possible under some conditions, but what level of compliance and the conditions that should be assumed to be true is not necessarily clear. We do not aim to solve the problem of giving a 'one size fits all' definition, but instead give the general notion and argue the framework supports many definitions, for brevity we leave out formal definitions of the properties.

In terms of the problems in defining coherence, many are the same as those presented by Hansen et al. [5]. These are not repeated here for reasons of brevity, but generally they encompass the problem of determining what elements of a normative system should be assumed to be in the same context, the facts that should be simultaneously assumed (affecting what duties are simultaneously active) and whether conditions should be considered for conflict with consequences.

Another problem is determining what level of compliance a normative system requires to be deemed coherent, particularly given that sanctions may fix violations. It may be the case that a system is coherent if there is at least one trace that ends up being compliant. This would imply the legislator considered sanctions to be mitigating costs for the violation of a norm. In such a case violations are allowable if they can always be 'fixed'.

Alternatively, it may be the case that sanctions merely act to deter too many malign agents from violating norms that are costly to the organization, but not so costly to the point of discouraging too many agents from joining the organization (as may be the case if they are mitigating costs).

6 Related Work

In the area of checking normative systems for particular properties, this paper appears to be the first to examine the *re-checking* of normative systems. Therefore, there is little directly relevant work.

The most relevant work is that from which the semantic framework is derived, loosely the work of López, Luck et al. on interlocking norms [8, 9]. More specifically, the work of Jiang et al. [6] which focuses on a normative framework and in particular different contexts for norms. In their work, they provide a conceptual normative framework for interlocking norms. The conceptual framework does not have semantics, instead these are given when norms expressed in the framework are mapped to Coloured Petri Nets, operationalising the system in the process. Their Coloured Petri Nets produce compliance traces that are similar to the normative traces in this paper.

Outside of this, there is some relevant work on the efficient checking (but not re-checking) of normative system coherence. A common approach is to use first-order unification, where conflict is detected if two norms with the same consequence may be unified and they have opposing modalities (i.e. obligation and prohibition) [7, 10–13]. This is a suitable mechanism if efficient algorithms are used, but it does not consider the advantages of using previous checks for re-checking. Furthermore, such work has not considered the increased computational complexity interlocking norms would cause.

Finally, since the efficiency of checking and re-checking a normative system depends on its structure, loosely related research on efficient structuring follows the same motivation as this paper. In their work on Defeasible Deontic Logic, Governatori and Rotolo [4] provide rewrite rules for placing normative systems in a normal form, removing redundancies in the process and identifying conflicts thereafter. Although normal forms are not directly relevant to the work in this paper, it is important for the general goal of efficient coherence checks and checking for changes in normative systems (where normal forms may aid in equivalence checks).

7 Conclusions

This paper has given a novel, compositional, approach to checking the coherence of a normative system. This was achieved by giving the semantics to norms with terms of normative traces, connectives in terms of their legality and semantics for interleaving normative traces compositionally such that the full set of traces to be checked may be generated.

This is not applied to a system of unrelated norms, but instead systems of interlocking norms which increase the complexity. Thus, this paper argues that the framework is particularly invaluable for such complex systems of interrelated norms. Not just for the checking of normative system's coherence, but re-checking any changes made by making use of cached traces of previous checks. We leave a formal analysis of the time complexity of the proposal for future work that may give algorithms for all of the operations defined.

Two topics for future work are identified, namely examining the definition of coherence further and defining how to change a normative system and how to apply the work here to just those parts of a system have changed, such that it is re-checked efficiently. The topic of coherence has been discussed already, in terms of system change we expect it to be in a similar vein to that of Governatori et al. [4] on optimal structures of normative systems and normal forms that may be used for equivalence checks.

References

1. G. Boella, G. Pigozzi, and L. van der Torre. Normative framework for normative system change. In *Proceedings of AAMAS 2009*, pages 169–176, 2009.

2. G. Governatori and P. D. Giusto. Modifying Is Better Than Deleting: A New Approach To Base Revision. *AI* IA 99*, pages 145–154, 1999.
3. G. Governatori, J. Hulstijn, and A. Rotolo. Characterising deadlines in temporal modal and defeasible logic. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence*, pages 486–496, 2007.
4. G. Governatori and A. Rotolo. How Do Agents Comply with Norms? In *Dagstuhl Seminar Proceedings*, pages 488–491. IEEE, 2009.
5. J. Hansen, G. Pigozzi, and L. van der Torre. Ten philosophical problems in deontic logic. In *NORMAS*, Dagstuhl, Germany, 2007.
6. J. Jiang, H. Aldewereld, V. Dignum, and Y.-H. Tan. Norm Contextualization. In *Proceedings of COIN'12*, 2012.
7. M. J. Kollingbaum, W. W. Vasconcelos, A. García-Camino, and T. J. Norman. Managing conflict resolution in norm-regulated environments. *ESAW 2007*, 2007.
8. F. L. y. López and M. Luck. Modelling norms for autonomous agents. In *Proceedings of The Fourth Mexican Conference on Computer Science*, pages 238–245. IEEE Computer Society, 2003.
9. F. L. y. López, M. Luck, and M. D'Inverno. A normative framework for agent-based systems. *Computational and Mathematical Organization Theory*, 12(2-3):227–250, 2006.
10. E. Lupu and M. Sloman. Conflicts in policy-based distributed systems management. *IEEE Transactions on Software Engineering - Special Issue on Inconsistency Management*, 25(6):852–869, 1999.
11. A. Uszok, J. Bradshaw, R. Jeffers, N. Suri, P. Hayes, M. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott. KAoS policy and domain services: Toward a description-logic approach to policy representation, deconfliction, and enforcement. In *Proceedings of IEEE Policy 2003*, pages 93–98, Los Amitos, CA, 2003. IEEE Computer Society.
12. A. Uszok, J. M. Bradshaw, J. Lott, M. Breedy, L. Bunch, P. Feltovich, M. Johnson, and H. Jung. New Developments in Ontology-Based Policy Management: Increasing the Practicality and Comprehensiveness of KAoS. *2008 IEEE Workshop on Policies for Distributed Systems and Networks*, pages 145–152, June 2008.
13. W. Vasconcelos, M. J. Kollingbaum, and T. J. Norman. Resolving conflict and inconsistency in norm-regulated virtual organizations. In *Proceedings of AAMAS '07*, volume 5, pages 632–639, New York, New York, USA, 2007. ACM Press.
14. M. Wooldridge. *An Introduction to Multi-Agent Systems*. Wiley, 2002.