Available online at www.sciencedirect.com
ScienceDirect
Journal of Applied Logic 6 (2008) 172–191
JOURNAL OF APPLIED LOGIC
www.elsevier.com/locate/jal

Agent strands in the action language $nC+$

Robert Craven *, Marek Sergot

Department of Computing, Imperial College London, UK

Available online 29 June 2007

Abstract

The action language $C+$ of Giunchiglia, Lee, Lifschitz, McCain and Turner is a formalism for specifying and reasoning about the effects of actions and the persistence ('inertia') of facts over time. An 'action description' in $C+$ defines a labelled transition system of a certain kind. $nC+$ is an extended form of $C+$ designed for representing normative and institutional aspects of (human or computer) societies. The deontic component of $nC+$ provides a means of specifying the permitted (acceptable, legal) states of a transition system and its permitted (acceptable, legal) transitions. We present this component of $nC+$, motivating its details with reference to some small illustrative examples, and elaborate the formalism by allowing scope for norms governing individual agents. We describe the various kinds of investigation possible on the semantic structures which $nC+$ defines.

Keywords: Action languages; Transition systems; Deontic logic; Norm-governed agent systems; Formal specification

1. Introduction

The action language $C+$ [1] is a formalism for specifying and reasoning about the effects of actions and the persistence ('inertia') of facts over time, building on a general-purpose non-monotonic representation formalism called 'causal theories'. An 'action description' in $C+$ is a set of $C+$ rules which define a labelled transition system of a certain kind. Implementations supporting a wide range of querying and planning tasks are available, notably in the form of the 'Causal Calculator' CCALC [2]. $C+$ and CCALC have been applied successfully to a number of benchmark examples in the knowledge representation literature (see e.g. [3] and the CCALC website [2]). We have used it in our own work to construct executable specifications of agent societies (see e.g. [4,5]).

$nC+$ [6,7] is an extended form of $C+$ designed for representing normative and institutional aspects of (human or computer) societies. There are two main extensions. The first is a means of expressing 'counts as' relations between actions, also referred to as 'conventional generation' of actions. This feature will not be discussed in this paper. The second extension is a way of specifying the permitted (acceptable, legal) states of a transition system and its permitted (acceptable, legal) transitions. The aim of the paper is to present this component of $nC+$ and some simple illustrative examples. $nC+$ was called $(C+)^{++}$ in earlier presentations.

In previous versions of this work [8], our emphasis in describing and defining $nC+$ has been on the modelling of system behaviour from an external, 'bird's eye' perspective, that is to say, from the system designer's point of view. It could then be verified whether properties hold or not of the system specified (a process analogous to that described in

* Corresponding author.
E-mail addresses: rac101@doc.ic.ac.uk (R. Craven), mjs@doc.ic.ac.uk (M. Sergot).

1570-8683/$ – see front matter
doi:10.1016/j.jal.2007.06.007

[9,10], which concentrates on epistemic properties and communicative acts). In the current paper, we extend this work to allow the representation of norms governing an individual agent's actions, and show how this more expressive $nC+$ can be defined to determine a richer class of 'coloured' labelled transition systems, on which a number of interesting deontic properties may be verified.

We have three existing implementations of the $nC+$ language. The first employs the 'Causal Calculator' CCALC. As explained later in the paper, the required modifications to CCALC are minor and very easily implemented. The second implementation provides an 'event calculus' style of computation with $C+$ and $nC+$ action descriptions. Given an action description and a 'narrative'—a record of what events have occurred—this implementation allows all past states, including what was permitted and obligatory at each past state, to be queried and computed. The third implementation connects $C+$ and $nC+$ to model checking software. System properties expressed in temporal logics such as CTL can then be verified by means of standard model checking techniques (specifically the model checker NuSMV) on transition systems defined using the $nC+$ language. A small example is presented in [7]. We do not discuss the implementations further for lack of space, except to explain how the CCALC method works.

Related work. Some readers may see a resemblance between $nC+$ and John-Jules Meyer's Dynamic Deontic Logic [11], and other well known works based on 'modal action logics' generally (e.g. [12,13]). There are three fundamental differences. (1) $C+$ and $nC+$ are not variants of dynamic logic or modal action logic. They are languages for defining specific instances of labelled transition systems. Other languages—we refer to them as 'query languages'—can then be interpreted on these structures. Dynamic logic is one candidate, the query language in CCALC is another, but there are many other possibilities: each $C+$ or $nC+$ action description defines a Kripke-structure, on which a variety of (modal) query languages, including a wide range of deontic and temporal operators, can be evaluated. We do not have space to discuss any of these possibilities in detail. (2) The representation of action is quite different from that in dynamic logic and modal action logic. (3) There are important differences of detail, in particular concerning the interactions between permitted states and permitted transitions between states.

Moses and Tennenholtz [14] define 'artificial social systems': automata-based systems of multiple agents together with *social laws*. Their concept of a social law is that of a subset of the (joint) actions which agents may perform in a given state, and where the interest is in social laws which enable all agents to achieve their individual goals together. As with many other formalisms, artificial social systems take the structure of a transition system as given—by contrast, one of our purposes is to investigate languages suitable for the specification of such systems, together with their deontic properties.

The semantical devices employed in $nC+$—classification of states and transitions into green/red (good/bad, ideal/sub-ideal), violation constants, explicit names for norms, orderings of states according to how well they comply with these norms, names for individual agents and partitionings of actions into sets of those performed by a given agent—are all frequently encountered in the deontic logic literature. The novelty here lies, first, in the details of how they are incorporated into labelled transition systems, and second, in the way the $nC+$ language is used to define these structures.

Finally, $C+$ is a (recent) member of a family of formalisms called 'causal action languages' in the AI literature. Several groups have suggested encoding normative concepts in such formalisms. We have done so ourselves in other work (see e.g. [4,5,15]) where we have used both $C+$ and the 'event calculus' for this purpose. Leon van der Torre [16] has made a suggestion along similar lines, though using a different causal action language and a different approach. See also the discussion in [13]. One feature that distinguishes $C+$ from other AI action languages is that it has an explicit semantics in terms of transition systems. It thereby proves a bridge between AI formalisms and standard methods in other areas of computer science and logic. It is this feature that $nC+$ seeks to exploit.

2. The language $C+$

We begin with a concise, and necessarily rather dense, summary of the $C+$ language. Some features (notably 'statically determined fluents' and 'exogenous actions') are omitted for simplicity. There are also some minor syntactic and terminological differences from the version presented in [1]. See [6] for details.

A *multi-valued propositional signature* σ is a set of symbols called *constants*. For each constant c in σ there is a non-empty set $dom(c)$ of values called the *domain* of c. For simplicity, in this paper we will assume that each $dom(c)$ is finite and has at least two elements. An *atom* of a signature σ is an expression of the form $c = v$ where c is a constant

in σ and $v \in dom(c)$. A *formula* φ of signature σ is any propositional compound of atoms of σ. The expressions $\top$ and $\bot$ are 0-ary connectives, with the usual interpretation.

A *Boolean* constant is one whose domain is the set of truth values $\{\mathsf{t}, \mathsf{f}\}$. If p is a Boolean constant, p is shorthand for the atom $p = \mathsf{t}$ and $\neg p$ for the atom $p = \mathsf{f}$. Notice that, as defined here, $\neg p$ is an *atom* when p is a Boolean constant.

In $\mathcal{C}+$, the signature σ is partitioned into a set σ^{f} of *fluent constants* (also known as 'state variables' in other areas of Computer Science) and a set σ^{a} of *action constants*. A *fluent formula* is a formula whose constants all belong to σ^{f}; an *action formula* is a formula containing at least one action constant and no fluent constants.

An *interpretation* of a multi-valued signature σ is a function that maps every constant c in σ to some value v in $dom(c)$; an interpretation I *satisfies* an atom $c = v$, written $I \models c = v$, if $I(c) = v$. The satisfaction relation $\models$ is extended from atoms to formulas in accordance with the standard truth tables for the propositional connectives. We write $\mathrm{I}(\sigma)$ for the set of interpretations of σ.

Transition systems. Every $\mathcal{C}+$ action description D of signature $(\sigma^{\mathrm{f}}, \sigma^{\mathrm{a}})$ defines a labelled transition system $\langle S, \mathbf{A}, R\rangle$ where

- S is a (non-empty) set of *states*, each of which is an interpretation of the fluent constants σ^{f} of D; $S \subseteq \mathrm{I}(\sigma^{\mathrm{f}})$;
- $\mathbf{A}$ is a set of *transition labels*, also called *events*; $\mathbf{A}$ is the set of interpretations of the action constants σ^{a}, $\mathbf{A} = \mathrm{I}(\sigma^{\mathrm{a}})$;
- R is a set of labelled transitions, $R \subseteq S \times \mathbf{A} \times S$.

As usual, we say that ε is *executable* in a state s when there is a transition (s, ε, s') in R, and *non-deterministic* in s when there are transitions (s, ε, s') and (s, ε, s'') in R with $s' \neq s''$. A *path* or *run* of length m of the labelled transition system $\langle S, \mathbf{A}, R\rangle$ is a sequence $s_0\varepsilon_0 s_1 \cdots s_{m-1}\varepsilon_{m-1}s_m$ $(m \geqslant 0)$ such that $(s_{i-1}, \varepsilon_{i-1}, s_i) \in R$ for $i \in 1..m$.

It is convenient in what follows to represent a state by the set of fluent atoms that it satisfies, i.e., $s = \{f = v \mid s \models f = v\}$. A state is then a (complete, and consistent) set of fluent atoms. We sometimes say a formula φ 'holds in' state s or 'is true in' state s as alternative ways of saying that s satisfies φ.

Action constants in $\mathcal{C}+$ are used to name actions, attributes of actions, or properties of a transition as a whole. Since a transition label/event ε is an interpretation of the action constants σ^{a}, it is meaningful to say that ε satisfies an action formula α ($\varepsilon \models \alpha$). When $\varepsilon \models \alpha$ we say that the transition (s, ε, s') is a transition of type α. Moreover, since a transition label is an interpretation of the action constants σ^{a}, it can also be represented by the set of atoms that it satisfies.

An action description D in $\mathcal{C}+$ is a set of *causal laws*, which are expressions of the following three forms. A *static law* is an expression:

$$F \text{ if } G \tag{1}$$

where F and G are fluent formulas. Static laws express constraints on states. A state s satisfies a static law (1) if $s \models (G \rightarrow F)$. A *fluent dynamic law* is an expression:

$$F \text{ if } G \text{ after } \psi \tag{2}$$

where F and G are fluent formulas and ψ is any formula of signature $\sigma^{\mathrm{f}} \cup \sigma^{\mathrm{a}}$. Informally, (2) states that fluent formula F is satisfied by the resulting state s' of any transition (s, ε, s') with $s \cup \varepsilon \models \psi$, as long as fluent formula G is also satisfied by s'. Some examples follow. An *action dynamic law* is an expression:

$$\alpha \text{ if } \psi \tag{3}$$

where α is an action formula and ψ is any formula of signature $\sigma^{\mathrm{f}} \cup \sigma^{\mathrm{a}}$. Action dynamic laws are used to express, among other things, that any transition of type α must also be of type α' (written α' if α), or that any transition from a state satisfying fluent formula G must be of type β (written β if G).

The $\mathcal{C}+$ language provides various abbreviations for common forms of causal laws. We will employ the following in this paper.

α causes F if G expresses that fluent formula F is satisfied by any state following the occurrence of a transition of type α from a state satisfying fluent formula G. It is shorthand for the dynamic law F if $\top$ after $G \wedge \alpha$. α causes F is shorthand for F if $\top$ after α.

nonexecutable α if G expresses that there is no transition of type α from a state satisfying fluent formula G. It is shorthand for the fluent dynamic law $\bot$ if $\top$ after $G \wedge \alpha$, or α causes $\bot$ if G.

inertial f states that values of the fluent constant f persist by default (by 'inertia') from one state to the next. It is shorthand for the collection of fluent dynamic laws $f = v$ if $f = v$ after $f = v$ for every $v \in dom(f)$.

Of most interest are *definite* action descriptions, which are action descriptions in which the head of every law (static, fluent dynamic, or action dynamic) is either an atom or the symbol $\bot$, and in which no atom is the head of infinitely many laws of D. We will restrict attention to definite action descriptions in this paper.

Causal theories. The language $\mathcal{C}+$ is presented in [1] as a higher-level notation for defining particular classes of theories in a general-purpose non-monotonic formalism called 'causal theories'. For present purposes the important points are these: for every (definite) action description D and non-negative integer m there is a natural translation from D to a causal theory Γ_m^D which encodes the paths of length m in the transition system defined by D; moreover, for every definite causal theory Γ_m^D there is a formula $comp(\Gamma_m^D)$ of (classical) propositional logic whose (classical) models are in 1–1 correspondence with the paths of length m in the transition system defined by D. Thus, one method of computation for $\mathcal{C}+$ action descriptions is to construct the formula $comp(\Gamma_m^D)$ from the action description D and then employ a (standard, classical) satisfaction solver to determine the models of $comp(\Gamma_m^D)$. This is the method employed in the 'Causal Calculator' CCALC.

A causal theory of signature σ is a set of expressions ('causal rules') of the form

$$F \Leftarrow G$$

where F and G are formulas of signature σ. F is the head of the rule and G is the body. A rule $F \Leftarrow G$ is to be read as saying that there is a cause for F when G is true (which is not the same as saying that G is the cause of F).

Let Γ be a causal theory and let X be an interpretation of its signature. The *reduct* Γ^X is the set of all rules of Γ whose bodies are satisfied by the interpretation X: $\Gamma^X =_{\text{def}} \{F \mid F \Leftarrow G \text{ is a rule in } \Gamma \text{ and } X \models G$. X is a *model* of Γ iff X is the unique model (in the sense of multi-valued signatures) of Γ^X.

Given a definite action description D in $\mathcal{C}+$, and any non-negative integer m, translation to the corresponding causal theory Γ_m^D proceeds as follows. The signature of Γ_m^D is obtained by time-stamping every fluent constant of D with non-negative integers between 0 and m and every action constant with integers between 0 and $m-1$: the (new) atom $f[i] = v$ represents that fluent $f = v$ holds at integer time i, or more precisely, that $f = v$ is satisfied by the state s_i of a path $s_0\varepsilon_0 \cdots \varepsilon_{m-1}s_m$ of the transition system defined by D; the atom $a[i] = v$ represents that action atom $a = v$ is satisfied by the transition ε_i of such a path. The domain of each timestamped constant $c[i]$ is the domain of c. In what follows, $\psi[i]$ is shorthand for the formula obtained by replacing every atom $c = v$ in ψ by the timestamped atom $c[i] = v$.

Now, for every static law F if G in D and every $i \in 0..m$, include in Γ_m^D a causal rule of the form

$$F[i] \Leftarrow G[i]$$

For every fluent dynamic law F if G after ψ in D and every $i \in 0..m-1$, include a causal rule of the form

$$F[i+1] \Leftarrow G[i+1] \wedge \psi[i]$$

And for every action dynamic law α if ψ in D and every $i \in 0..m-1$, include a causal rule of the form

$$\alpha[i] \Leftarrow \psi[i]$$

We also require the following 'exogeneity laws'. For every fluent constant f and every $v \in dom(f)$, include a causal rule:

$$f[0] = v \Leftarrow f[0] = v$$

And for every action constant a, every $v \in dom(a)$, and every $i \in 0..m-1$, include a causal rule:

$$a[i] = v \Leftarrow a[i] = v$$

It is straightforward to check [1] that the models of causal theory Γ_m^D, and hence the (classical) models of the propositional logic formula $comp(\Gamma_m^D)$, correspond 1–1 to the paths of length m of the transition system defined by the $\mathcal{C}+$ action description D. In particular, models of $comp(\Gamma_1^D)$ encode the transitions defined by D and models of $comp(\Gamma_0^D)$ the states defined by D.

3. *nC+*: Coloured transition systems

An action description of $n\mathcal{C}+$ defines a *coloured transition system*, which is a structure of the form $\langle S, \mathbf{A}, R, S_g, R_g\rangle$ where $\langle S, \mathbf{A}, R\rangle$ is a labelled transition system of the kind defined by $\mathcal{C}+$ action descriptions, and where the two new components are

- $S_g \subseteq S$, the set of 'permitted' ('acceptable', 'ideal', 'legal') states—we call S_g the 'green' states of the system;
- $R_g \subseteq R$, the set of 'permitted' ('acceptable', 'ideal', 'legal') transitions—we call R_g the 'green' transitions of the system.

We refer to the complements $S_{red} = S - S_g$ and $R_{red} = R - R_g$ as the 'red states' and 'red transitions', respectively. Semantical devices which partition states (and here, transitions) into two categories are familiar in the field of deontic logic. For example, Carmo and Jones [17] employ a structure which has both ideal/sub-ideal states and ideal/sub-ideal transitions (unlabelled). van der Meyden's 'Dynamic logic of permission' [18] employs a structure in which transitions, but not states, are classified as 'permitted/non-permitted'. van der Meyden's version was constructed as a response to problems of Meyer's 'Dynamic deontic logic' [11] which classifies transitions as 'permitted/non-permitted' by reference only to the state resulting from a transition. 'Deontic interpreted systems' [9] classify states as 'green'/'red', where these states have further internal structure to model the local states of agents in a multi-agent context. In all of these examples (and others) the task has been to find axiomatisations of such structures in one form of deontic logic or another. Here we are concerned with a different task, that of devising a language for *defining* coloured transition systems of the form described above.

A coloured transition system $\langle S, \mathbf{A}, R, S_g, R_g\rangle$ must further satisfy the following constraint, for all states s and s' in S and all transitions (s, ε, s') in R:

$$\text{if } (s, \varepsilon, s') \in R_g \text{ and } s \in S_g \text{ then } s' \in S_g \tag{4}$$

We refer to this as the *green–green–green* constraint, or *ggg* for short. (It is difficult to find a suitable mnemonic.) The *ggg* constraint (4) expresses a kind of *well-formedness* principle: a green (permitted, acceptable, legal) transition in a green (permitted, acceptable, legal) state always leads to a green (acceptable, legal, permitted) state. What is the rationale? Since we are here classifying both states and transitions into green/red, it is natural to ask whether there are any relationships between the classification of states and the classification of transitions between them. As observed previously by Carmo and Jones [17] any such relationships are necessarily quite weak. In particular, and *contra* the assumptions underpinning John-Jules Meyer's construction of Dynamic Deontic Logic [11], a red (unacceptable, non-permitted) transition can result in a green (acceptable, permitted) state. Indeed such cases are frequent: suppose that there are two different transitions, (s, ε_1, s') and (s, ε_2, s'), between a green or red state s and a green state s'. It is entirely reasonable that the transition (s, ε_1, s') is classified as green whereas (s, ε_2, s') is classified as red. (s, ε_1, s') might represent an action by one agent, for example, and (s, ε_2, s') an action by another. This situation cannot arise if the transition system has a tree-like structure in which there is at most one transition between any pair of states, but we do not want to restrict attention to transition systems of this form. Similarly, it is easy to encounter cases in which a green (acceptable, permitted) transition can lead sensibly to a red (unacceptable, non-permitted) state: not all green (acceptable, permitted) transitions from a red state must be such that they restore the system to a green state. Some illustrations will arise in the examples later. The only plausible relationship between the classification of states and the classification of transitions, as also noted by Carmo and Jones [17], is what we called the *ggg* constraint above, if we regard it (as we do) as a required property of any well-formed system specification. Since the *ggg* constraint is so useful for the applications we have in mind, we choose to adopt it as a feature of every coloured transition system.

Note that the *ggg* constraint (4) may be written equivalently as:

$$\text{if } (s, \varepsilon, s') \in R \text{ and } s \in S_g \text{ and } s' \in S_{red} \text{ then } (s, \varepsilon, s') \in R_{red} \tag{5}$$

Any transition from a green (acceptable, permitted) state to a red (unacceptable, non-permitted) state must itself be red, in a well-formed system specification.

One can consider a range of other properties that we might require of a coloured transition system: that the transition relation must be serial, for example, or that there must be at least one green state, or that from every green state there must be at least one green transition, or that from every green state reachable from some specified initial state(s) there must be at least one green transition, and so on. The investigation of these, and other, properties is worthwhile but not something we undertake here. We place no restrictions on coloured transition systems, beyond the *ggg* constraint.

The language $n\mathcal{C}+$. To avoid having to specify separately which states and transitions are green and which are red, an $n\mathcal{C}+$ action description specifies those that are red and leaves the remainder to be classified as green by default. This is for convenience, and also to ensure that all states and transitions are classified completely and consistently. (One might ask why the defaults are not chosen to operate the other way round. It is very much more awkward to specify concisely what is green and allow the remainder to be red by default.)

Accordingly, the language $n\mathcal{C}+$ extends $\mathcal{C}+$ with two new forms of rules. A *state permission law* is an expression of the form

$$n\text{: not-permitted } F \text{ if } G \tag{6}$$

where n is an (optional) identifier for the rule and F and G are fluent formulas. not-permitted F is shorthand for the law not-permitted F if $\top$. An *action permission law* is an expression of the form

$$n\text{: not-permitted } \alpha \text{ if } \psi \tag{7}$$

where n is an (optional) identifier for the rule, α is an action formula and ψ is any formula of signature $\sigma^{\mathrm{f}} \cup \sigma^{\mathrm{a}}$. not-permitted α is a shorthand for the law not-permitted α if $\top$. We also allow oblig F as an abbreviation for not-permitted $\neg F$ and oblig α as an abbreviation for not-permitted $\neg\alpha$.[1]

Informally, in the transition system defined by an action description D, a state s is red whenever $s \models F \wedge G$ for a state permission law not-permitted F if G. All other states are green by default. A transition (s, ε, s') is red whenever $s \cup \varepsilon \models \psi$ and $\varepsilon \models \alpha$ for any action permission law not-permitted α if ψ. All other transitions are green, *subject to* the *ggg* constraint which may impose further conditions on the colouring of a given transition.

Let D be an action description of $n\mathcal{C}+$. D_{basic} refers to the subset of laws of D that are also laws of $\mathcal{C}+$. The coloured transition system defined by D has the states S and transitions R that are defined by its $\mathcal{C}+$ component, D_{basic}, and green states S_{g} and green transitions R_{g} given by $S_{\mathrm{g}} =_{\mathrm{def}} S - S_{\mathrm{red}}$, $R_{\mathrm{g}} =_{\mathrm{def}} R - R_{\mathrm{red}}$ where

$$S_{\mathrm{red}} =_{\mathrm{def}} \big\{ s \mid s \models F \wedge G \text{ for some law of the form (6) in } D \big\}$$

$$R_{\mathrm{red}} =_{\mathrm{def}} \big\{ (s, \varepsilon, s') \mid s \cup \varepsilon \models \psi,\ \varepsilon \models \alpha \text{ for some law of the form (7) in } D \big\} \cup \big\{ (s, \varepsilon, s') \mid s \in S_{\mathrm{g}} \text{ and } s' \in S_{\mathrm{red}} \big\}$$

The second component of the R_{red} definition ensures that the *ggg* constraint is satisfied. (The state permission laws not-permitted F if G and not-permitted $(F \wedge G)$ are thus equivalent in $n\mathcal{C}+$; we allow both forms for convenience.) It can be shown easily [6] that the coloured transition system defined in this way is unique and satisfies the *ggg* constraint. The definition of course does not guarantee that the coloured transition system satisfies any of the other possible properties that we mentioned earlier. If they are felt to be desirable in some particular application, they must be checked separately as part of the specification process. (These checks are easily implemented.)

The overall effect is thus:

- a state is green unless coloured red by some static permission law;
- a transition is red if it is coloured red by some action permission law, or by the *ggg* constraint; otherwise it is green.

That the colouring of transitions is dependent on the colouring of states should *not* be interpreted as a commitment to any philosophical position about the priority of the ought-to-be and the ought-to-do, and the derivability of one from the other. It is merely a consequence of, first, adopting the *ggg* constraint as an expression of the well-formedness of

[1] This does not raise the issue of 'action negation' as encountered in modal action logics. (See e.g. [13].) In $\mathcal{C}+$ and $n\mathcal{C}+$, α is not the name of an action but a formula expressing a property of transitions.

a system specification, and second, of choosing to specify explicitly what is red and letting green be determined by default.

Causal theories. Any (definite) action description of $n\mathcal{C}+$ can be translated to the language of (definite) causal theories, as follows. Let D be an action description and m a non-negative integer. The translation of the $\mathcal{C}+$ component D_{basic} of D proceeds as usual. For the permission laws, introduce two new fluent and action constants, status and trans respectively, both with possible values green and red. They will be used to represent the colour of a state and the colour of a transition, respectively.

For every state permission law n: not-permitted F if G and time index $i \in 0..m$, include in Γ_m^D a causal rule of the form

$$\mathsf{status}[i] = \mathsf{red} \Leftarrow F[i] \wedge G[i] \tag{8}$$

and for every $i \in 0..m$, a causal rule of the form

$$\mathsf{status}[i] = \mathsf{green} \Leftarrow \mathsf{status}[i] = \mathsf{green} \tag{9}$$

to specify the default colour of a state. A state permission rule of the form n: oblig F if G produces causal rules of the form $\mathsf{status}[i] = \mathsf{red} \Leftarrow \neg F[i] \wedge G[i]$.

For every action permission law n: not-permitted α if ψ and time index $i \in 0..m-1$, include in Γ_m^D a causal rule of the form

$$\mathsf{trans}[i] = \mathsf{red} \Leftarrow \alpha[i] \wedge \psi[i] \tag{10}$$

and for every $i \in 0..m-1$, a causal rule of the form

$$\mathsf{trans}[i] = \mathsf{green} \Leftarrow \mathsf{trans}[i] = \mathsf{green} \tag{11}$$

to specify the default colour of a transition. An action permission law of the form n: oblig α if ψ produces causal rules of the form $\mathsf{trans}[i] = \mathsf{red} \Leftarrow \neg\alpha[i] \wedge \psi[i]$.

Finally, to capture the *ggg* constraint, include for every $i \in 0..m-1$ a causal rule of the form

$$\mathsf{trans}[i] = \mathsf{red} \;\Leftarrow\; \mathsf{status}[i] = \mathsf{green} \wedge \mathsf{status}[i+1] = \mathsf{red} \tag{12}$$

It is straightforward to show [6] that models of the causal theory Γ_m^D correspond to all paths of length m through the coloured transition system defined by D, where the fluent constant status and the action constant trans encode the colours of the states and transitions, respectively.

The translation of $n\mathcal{C}+$ into causal theories effectively treats $\mathsf{status} = \mathsf{red}$ and $\mathsf{trans} = \mathsf{red}$ as 'violation constants'. Notice that, although action descriptions in $n\mathcal{C}+$ can be translated to causal theories, they cannot be translated to action descriptions of $\mathcal{C}+$: there is no form of causal law in $\mathcal{C}+$ which translates to the *ggg* constraint (12). However, implementation in CCALC requires only that the causal rules (8)–(12) are included in the translation to causal theories, which is a very simple modification.

In [8] we presented a refinement of the current approach, where instead of the binary classification of states as red or green, states in defined transition systems were ordered, depending on how well each complied with the state permission laws of an $n\mathcal{C}+$ action description. We also discussed possible generalisations of the *ggg* constraint. In the current paper, we keep to the more simple case of a classification of states as green or red.

4. Examples

The examples in this section are deliberately chosen to be as simple as possible, so that in each case we can show the transition system defined in its entirety. Other examples may be found in [6,7]. The first example illustrates the use of $n\mathcal{C}+$ in a typical (but very simple) system specification. The second is to motivate the more complicated account to come in Section 5.

Example *(File system).* I is some piece of (confidential) information. I, or material from which I can be derived, is stored in a file. Let x range over some set of agent names. Boolean fluent constants Kx represent that agent x has access to information I, that x 'knows' I. Boolean fluent constants Fx represent that x has read access to the file

containing I. If x has read access to the file (Fx) then x knows I (Kx). Fx is inertial: both Fx and $\neg Fx$ persist by default. $\neg Kx$ persists by default but once Kx holds, it holds for ever.

Suppose, for simplicity, that there are two agents, a and b. Suppose moreover that the file is the only source of information I, in the sense that if Kx holds for any x then either Fa or Fb holds. This does not change the essence of the example but it reduces the number of states and simplifies the diagrams.

There are two types of acts: Boolean action constants $read(x)$ represent that x is given read access to the file containing I. Boolean action constant *a tells b* represents that a communicates to b the information I (whether or not b knows it already), and *b tells a* that b communicates it to a. In this simple example there are no actions by which read access to the file is removed once it is granted.

We can represent the above as a definite action description as follows, for x ranging over a and b.

inertial Fx	$read(x)$ causes Fx
$\neg Kx$ if $\neg Kx$ after $\neg Kx$	*a tells b* causes Kb
Kx if $\top$ after Kx	*b tells a* causes Ka
	nonexecutable *a tells b* if $\neg Ka$
Kx if Fx	nonexecutable *b tells a* if $\neg Kb$
$\bot$ if $Kx \wedge \neg Fa \wedge \neg Fb$	nonexecutable $read(x)$ if Fx

Now suppose that a is permitted to know I, and b is not. We add the following law to the action description. (Ka is permitted by default.)

not-permitted Kb

The transition system defined by these laws is shown in Fig. 1. The label $read(a)$ stands for the transition $\{read(a), \neg read(b), \neg a \text{ tells } b, \neg b \text{ tells } a\}$; the label $read(b)$ stands for $\{\neg read(a), read(b), \neg a \text{ tells } b, \neg b \text{ tells } a\}$; and similarly for the labels *a tells b* and *b tells a*. The label $read(a), read(b)$ is shorthand for the transition label $\{read(a), read(b), \neg a \text{ tells } b, \neg b \text{ tells } a\}$. Reflexive arcs, corresponding to the 'null event' or to transitions of type *a tells b* and *b tells a* from state $\{Fa, Ka, \neg Fb, Kb\}$ to itself, are omitted from the diagram to reduce clutter. Also omitted from the diagram are transitions of type $read(a) \wedge a \text{ tells } b$, $a \text{ tells } b \wedge b \text{ tells } a$, etc. Again, this is just to reduce clutter.

Notice that transitions of type $read(b)$ are red because of the *ggg* constraint, except that $read(b)$ transitions come out to be green in states where Kb already holds. If the latter is felt to be undesirable, one could add another action permission law not-permitted $read(b)$, or a state permission law not-permitted Fb.

In a computerised system, b's access to information I would be controlled by the file access system. Naturally the file access system cannot determine whether b knows I: in practise, a specification of the computer system would simply say that $read(b)$ actions are nonexecutable, or simply that Fb is false. The latter can be expressed by adding

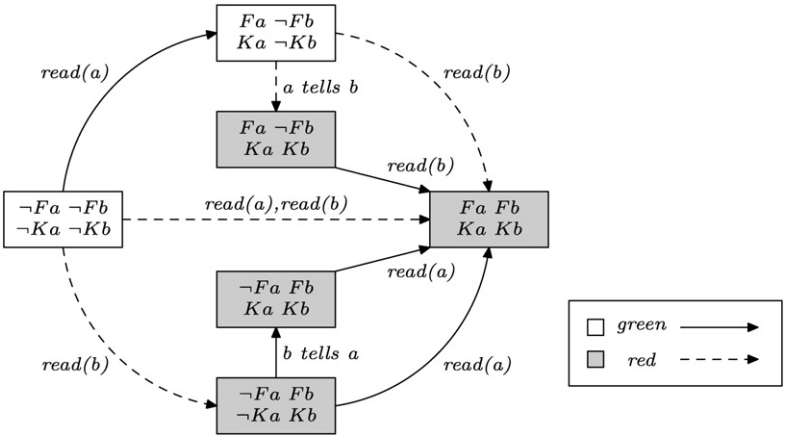

Fig. 1. Transition system for file access example.

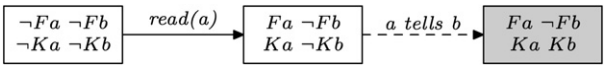

Fig. 2. Transition system for extended file access example.

the following static law to the action description:

$\bot$ if Fb

This eliminates all states in which Fb holds from the transition system. The transition system defined by this extended action description is depicted in Fig. 2. As usual, reflexive arcs are omitted from the diagram for clarity. Here, the action *read*(a) is under the control of the file access system, and *a tells b* is an action that can be performed by agent a. This difference is not explicit in the semantics of $\mathcal{C}+$ nor of $n\mathcal{C}+$ in the form we have introduced so far. Modifications to $n\mathcal{C}+$ to allow a greater scope for representing the actions of, and permission laws governing individual agents will follow, in the next section.

Example *(Rooms).* A second example concerns the specification of norm-governed interactions between agents acting independently. There are two categories of agents, male and female, who move around in a world of interconnecting rooms. The rooms are connected by doorways through which agents may pass (the precise topography, and number of rooms, can vary). Each doorway connects two rooms. Rooms can contain any number of male and female agents. The action constants in these domains will take the form $move(x) = p$, where x ranges over the agents in a particular example, and p typically ranges over a number of values representing directions in which agents can move, in addition to a value f: if a transition label (s, ε, s') has $\varepsilon \models move(a) = \mathsf{f}$, that is to be taken to represent that agent a does not move during that transition. A normative element is introduced by insisting that a female may not be present in a room alone with a male; such configurations are physically possible, and the transition systems defined by our action descriptions will include states representing them, but all such states will be coloured red.

Although this blueprint for action descriptions seems relatively simple, it shares essential features with a number of real-world domains, in which there are large numbers of interacting agents or components which may be in different states, and where some of those combinations of states are prohibited. (These real-world examples are *not* restricted to domains where agents perform physical actions.)

For the purposes of illustration, we shall consider a concrete instance of the example in which there are just two rooms, on the left and right, with one connecting door, and three agents, two males m_1 and m_2, and a female f_1. We have deliberately made the example simple in order to concentrate on the essentials of the action description and its deontic features, and so that we can depict the various states and transitions in their entirety. We will also insist that only one agent can move through the doorway at once; although this is a more significant restriction, it is both plausible (the doorways may be too narrow to let more than one agent pass through at once), and also enables us to depict the defined transition system in diagrammatic form with a minimum of clutter.

The signature of this instance of the 'rooms' domain contains simple fluent constants $loc(x) = l$ and $loc(x) = r$, where x ranges over m_1, m_2, f_1; $loc(m_1) = l$ is true when the male agent m_1 is in the left-hand room, $loc(m_2) = r$ is true when m_2 is in the right-hand room, and so on. The action constants are, in line with previous remarks, $move(x) = p$, where x ranges over the agents and p ranges over l, r, f. The causal laws are as follows:

inertial $loc(x)$

caused $loc(x) = p$ after $move(x) = p$ $\qquad (p \in \{l, r\})$

nonexecutable $move(x) = p$ if $loc(x) = p$ $\qquad (p \in \{l, r\})$

nonexecutable $move(x) = p$ if $move(y) \neq \mathsf{f}$ $\qquad (x \neq y, p \in \{l, r\})$

not-permitted$loc(m_1) = loc(f)$ if $loc(m_2) \neq loc(f)$

not-permitted$loc(m_2) = loc(f)$ if $loc(m_1) \neq loc(f)$

The last two laws clearly represent our insistence that it is undesirable that a male and a female should be alone in a room together. The action description is depicted in Fig. 3. We have not drawn the transitions from states to themselves, in order to keep the drawing clear; all such transitions are coloured green. Also, we have not included

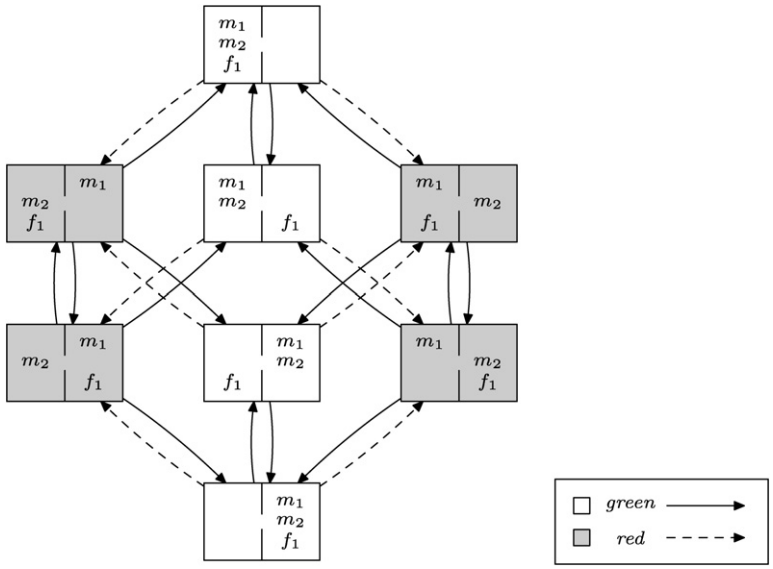

Fig. 3. A simple 'rooms' example.

labels for transitions. These can easily be deduced, for every arc in the diagram should have a label which makes precisely one $move(x) = p$ (for p one of l, r) action constant true; which action constant this is can in each case be seen from the components s and s' of the transition (s, ε, s'): for example, the (red) transition from the top-most state to the one immediately below and to the right $move(m_2) = r$, and therefore also $move(m_1) = \mathsf{f}$, $move(f_1) = \mathsf{f}$.

Given the permission laws in the action description, the semantics for $nC+$ determines the intended states as red, as can be seen from the diagram. There are no action permission laws, and so the red transitions are wholly determined by the *ggg* constraint: the red transitions are simply those where the system moves from a green state to one in which a male and a female are alone in a room together.

This latter feature of the transition system indicates a very inconvenient expressive paucity in $nC+$ as it currently stands. For consider, again, the transition from the top-most state, where all agents are in the left-hand room, to the state below and to the right of it, where m_1 and f_1 are left alone together after m_2 has exited to the right. In some sense, it is m_2 who has acted wrongly: he has left the room, leaving m_1 and f_1 alone together, in a configuration which thus violates the norms governing the system. On the other hand, if we remove the restriction that at most one agent can pass through the doorway at one time, it is far from clear which of the three agents, if any, acted wrongly when m_2 exited: it might have been m_2, or it might have been m_1, who should have followed m_2 out, or it might have been f_1, who should have followed m_2, or all of them collectively who acted wrongly, or none of them. The coloured transition systems we have defined, as they currently stand, do not have the capacity to represent that it is specifically one agent's actions rather than another's which must be marked as 'red'. There is no way to extract from, or represent in, the coloured transition system that a particular agent's 'strand' of the transition is sub-ideal, undesirable, and so on; indeed, there is no explicit concept of an individual agent in the semantics at all, something which surely must be introduced if we are to be able to reason effectively about the deontic properties of systems in which there are known to be large numbers of interacting agents.

5. Agent refinements

The language $nC+$ provides us with a means of representing when states and transitions satisfy, or fail to satisfy, a standard of legality, acceptability, desirability, and so on. Much can be said using the resources of this language. However, in representing systems in which there are a number of interacting agents (as with both of the simple domains depicted in the previous section), it is often essential to be able to speak about an individual agent's behaviour: in particular, about whether individual agents' actions are in the right or wrong—whether they are conforming to norms which govern specifically *their* behaviour. This is not possible using the resources of $nC+$ which we have introduced so far. The semantical structures in place are the labelled transition systems which describe behaviour, together with

colourings (red or green) of a whole transition and whole state: there is no scope for speaking of coloured *strands* of a transition, for a given agent.

That this greater scope for reference to the behaviour of an individual agent is desirable can easily be seen. Suppose, for example, that we are representing the workings of a bank: we may have a transition over which agent a withdraws £20 and agent b withdraws £30. If the transition is coloured red, we have no way of telling from the transition system whether it was a's action which was not permitted, b's action, or both, or indeed neither of their actions—it may have been some other factor in the environment, or a third agent. In this section we will refine our language and the graphical models determined by it, allowing the greater expressivity we desire, and letting us colour individual agents' strands of a transition.

The new semantical structure, which we will call a *coloured agent-stranded transition system*, is a tuple $\langle S, \mathbf{A}, R, S_g, R_g, Ag, strand, green\rangle$ where S is a set of states, $\mathbf{A}$ a set of transition labels/events, and R the labelled transitions between states, $R \subseteq S \times \mathbf{A} \times S$; S_g and R_g are the (globally) green states and transitions, as usual. Ag is a (finite) set of agent names. It is often conventional that one of the elements of Ag represents 'the environment'. We do not follow that convention: there are things we want to be able to say about actions of agents that are at best stilted if 'the environment' is treated as an agent like any other. In the present context an 'agent' in Ag could be a deliberative (human or computer) agent, or it could be a purely reactive component such as a simple computational unit or some other device. At the level of detail we are modelling here we do not make any distinction between these agents. *strand* is a function on $Ag \times \mathbf{A}$: $strand(x, \varepsilon)$ picks out from a transition label/event ε the component or 'strand' that corresponds to agent x's contribution to ε. We will write ε_x for $strand(x, \varepsilon)$. Given a transition (s, ε, s') and $x \in Ag$, ε_x may be thought of as the actions of agent x in the transition, where this does *not* imply that ε_x represents deliberate action, or action which has been freely chosen. As usual, ε_x may represent several concurrent actions by x, or actions with non-deterministic effects. ε_{env} denotes the actions that have occurred in the environment in a transition (s, ε, s'). Although 'the environment' is not treated here as an agent, we will want to be able to refer to actions that occur in the environment, outside the control of any agents in Ag.

Now, for each agent $x \in Ag$, there will be a set $green(x)$ of the green transitions for x: these are to be thought of as transitions where the actions of x have been in accordance with norms for x. For each agent x there will also be a set $red(x) = R - green(x)$ of those transitions in which the actions of x have failed to conform to x's norms. *green* is thus a function from the set of agents Ag to $\wp(R)$. $green(x)$ can be thought of as those transitions in which x's 'strand' is green, and $red(x)$ as the transitions where x's strand is red. When $(s, \varepsilon, s') \in green(x)$, or $(s, \varepsilon, s') \in red(x)$, we sometimes say that the transition (s, ε, s') is $green(x)$ or $red(x)$, respectively. We also say that ε_x is an action executable by agent x in a global state s when there is a transition (s, ε, s') in R, and that ε_x is $green(x)$ (resp., $red(x)$) in global state s when $(s, \varepsilon, s') \in green(x)$ (resp., $(s, \varepsilon, s') \in red(x)$).

We do not, at this stage, introduce more granularity into the colourings of states or consider norms which regulate the state of an individual agent. These are possible developments for further work. Our interest here is to study the norm-governed *behaviour* of agents, and how this may be related to the norms pertaining to the system as a whole. To that end, we will concentrate on the transitions which are used to represent agents' actions.

There is no analogue of the *ggg* constraint for the colouring of agent-specific strands of transitions. However, we do impose the following constraint: if (s, ε, s') is a transition in $green(x)$ (resp., $red(x)$), then every transition (s, ε', s'') in which x behaves in the same way as in (s, ε, s'), that is, every transition (s, ε', s'') with $\varepsilon'_x = \varepsilon_x$, must also be in $green(x)$ (resp., $red(x)$). In other words, for all transitions (s, ε, s') and (s, ε', s'') in R and all agents $x \in Ag$:

$$\text{if } \varepsilon_x = \varepsilon'_x \text{ then } (s, \varepsilon, s') \in green(x) \text{ iff } (s, \varepsilon', s'') \in green(x) \tag{13}$$

(and hence also $(s, \varepsilon, s') \in red(x)$ iff $(s, \varepsilon', s'') \in red(x)$ when $\varepsilon_x = \varepsilon'_x$). This reflects the idea that whether actions of agent x are in accordance with x's norms depends only on x's actions, not the actions of other agents, nor actions in the environment, or other extraneous factors: we might, with appropriate philosophical caution, think of this constraint as an insistence on the absence of 'moral luck'.

Notice that the constraint (13) covers the case where $\varepsilon = \varepsilon'$, that is to say, the case where there is a transition (s, ε, s') and another transition (s, ε, s'') with a different resulting state: ε is non-deterministic in the state s. Constraint (13) requires that, for every agent x, both of these transitions are coloured the same way irrespective of their resulting states. This is how we want it to be. For an agent x, there is no difference whether the results of its actions in a given state are not fully determined because x's own actions are non-deterministic, or because actions by other agents acting

independently determine what the resulting state will be, or whether that depends also on actions that take place in the environment.

We also have an optional *coherence constraint* relating the colouring of agent-specific strands to the colouring of transitions as a whole. The colouring of a transition as (globally) red represents that the system as a whole fails to satisfy the required standard of acceptability, legality, desirability represented by the global green/red colouring. In many settings it is then natural to say that if any one of the system components (agents) fails to satisfy its standards of acceptability, legality, desirability, then so does the system as a whole: if a transition is $red(x)$ for some agent x then it is also (globally) red. Formally, the transition system $\langle S, \mathbf{A}, R, S_{\mathrm{g}}, R_{\mathrm{g}}, Ag, strand, green\rangle$ satisfies the local-global coherence constraint whenever, for all agents $x \in Ag$,

$$red(x) \subseteq R_{\mathrm{red}}. \tag{14}$$

The coherence constraint (14) is optional and not appropriate in all settings. We will adopt it in the examples discussed below. Notice though, that even if the coherence constraint is adopted, it is possible that a transition has all its agent-specific strands coloured $green(x)$ for every agent x and still itself be coloured globally red. We will give some examples presently.

There are now two separate lines of development we can pursue.

1. Given a coloured agent-stranded transition system of the kind just defined, to what extent is it possible to determine from the colouring of its global transitions the implied colourings of the agent-specific strands of its transitions, or in other words, to derive from a specification of global system norms the implied agent-specific norms that govern an individual agent x's own actions? This would require us to formulate the conditions under which it is an agent x's actions ε_x that are, in some appropriate sense, responsible for, or the cause of, a system transition (s, ε, s')'s being coloured (globally) red. Such conditions can be formulated but raise a number of further questions, and will not be discussed here. We leave that for a separate paper.
2. We specify for every agent x in Ag the norms specific to x that govern x's individual actions: some subset of the actions executable by x in a given global state will be designated as $green(x)$ and the others as $red(x)$. If we want to think in terms of agent protocols, a transition (s, ε, s') is designated as $green(x)$ when x's actions ε_x in system state s comply with agent x's local protocol. We then specify, separately, system norms which constrain various combinations of actions by individual agents, or other interactions of interest, by classifying global system transitions and global system states as globally red or green. So we have two separate layers of specification: (i) norms specific to agents governing their individual actions, and (ii) norms governing system behaviour as a whole. We are interested in examining the relationships, if any, between these two separate layers. We might be interested in verifying, for example, that all behaviour by agent x compliant with the norms for x guarantees that the system avoids globally red states, or produces only globally green runs, or always recovers from a global red state to a global green state, and so on. This is the setting we have in mind for discussion in this paper. We will present some concrete examples below.

The required modifications to the language $n\mathcal{C}+$ are very straightforward. First we partition the action constants σ^{a} into those that represent actions by agents in Ag, and a (possibly empty) category of action constants representing actions in the environment:

$$\sigma^{\mathrm{a}} = \bigcup_{x \in Ag} \sigma^{\mathrm{a}}_x \cup \sigma^{\mathrm{a}}_{\mathrm{env}}$$

where σ^{a}_x are the action constants representing actions by agent x and $\sigma^{\mathrm{a}}_{\mathrm{env}}$ is a disjoint, and possibly empty, set of action constants representing actions in the environment. (In other work we sometimes also include a further set of action constants $\sigma^{\mathrm{a}}_{\mathrm{ext}}$, representing properties we may wish to ascribe, globally, to system transitions. We make no use of $\sigma^{\mathrm{a}}_{\mathrm{ext}}$ in the following examples, and will not mention it again in the paper.)

The transition labels/events ε in a transition system defined by $n\mathcal{C}+$ are interpretations of the action constants, and are represented by the set of action atoms satisfied by ε. The strand ε_x for an agent x of a transition label ε is therefore simply the subset of atoms in ε that belong to σ^{a}_x, and the strand $\varepsilon_{\mathrm{env}}$ representing actions in the environment is the subset of atoms in ε that belong to $\sigma^{\mathrm{a}}_{\mathrm{env}}$. In the files example of Section 4, for instance, the set of agents $Ag = \{a, b, file\_system\}$. The Boolean action constants *a tells b* and *b tells a* represent actions by agents a and b,

respectively, and Boolean action constants $read(a)$ and $read(b)$ represent actions by the file system whereby a and b are granted read-access to the file, respectively. There are no actions in the environment in this example. In the transition label/event

$$\varepsilon = \{read(a), read(b), a\ tells\ b, \neg b\ tells\ a\}$$

representing a transition in which the file system simultaneously grants read access to a and b while a tells b and b does not tell a, has strands $\varepsilon_{file\_system} = \{read(a), read(b)\}$, $\varepsilon_a = \{a\ tells\ b\}$, and $\varepsilon_b = \{\neg b\ tells\ a\}$.

We now introduce a new form of agent-specific permission law, as follows:

$$n : \mathsf{not\text{-}permitted}(x)\ \alpha\ \mathsf{if}\ \ \psi \tag{15}$$

where α is any formula containing only action constants from σ^{a}_x, and where ψ, as usual, is any formula of $\sigma^{\mathrm{f}} \cup \sigma^{\mathrm{a}}$. n is an optional identifier for the law. $\mathsf{oblig}(x)\ \alpha$ can be used as a shorthand for $\mathsf{not\text{-}permitted}(x)\ \neg\alpha$.

The rest is as one might expect. The coloured agent-stranded transition system defined by an action description D of agent-centric $n\mathcal{C}+$ has states S, transitions R, (globally) green states S_{g}, and (globally) green transitions R_{g} just as for $n\mathcal{C}+$, and, for every agent $x \in Ag$, $green(x) =_{\mathrm{def}} R - red(x)$ where

$$red(x) =_{\mathrm{def}} \{(s, \varepsilon, s') \mid s \cup \varepsilon \models \psi,\ \varepsilon \models \alpha \text{ for some law of the form (15) in } D\}$$

There is no *ggg* constraint for agent-specific strands; the optional coherence constraint (14) is easily added if required. Translation of agent-centric $n\mathcal{C}+$ action descriptions to causal theories, computation methods, and so on, proceed exactly as for $n\mathcal{C}+$. We omit the details since they are straightforward.

Notice that there is a difference between an agent-specific permission law of the form (15) and the $n\mathcal{C}+$ permission law

$$n : \mathsf{not\text{-}permitted}\ \alpha\ \mathsf{if}\ \ \psi \tag{16}$$

even when α is a formula of σ^{a}_x. The former colours only the x-specific strand of a transition; the latter colours the global transition as a whole, and expresses a system norm not an agent-specific norm. If we choose to adopt the coherence constraint (14), however, we can view a law of the form (15) where α is a formula of σ^{a}_x as shorthand for both (16) and the law (15).

Henceforth, the remainder of the discussion is more general. It pertains to coloured agent-stranded transition systems *in general*, not just to those that are defined by means of an agent-centric $n\mathcal{C}+$ action description.

6. Example

In order to illustrate some of the distinctions we are now able to articulate using the agent-stranded refinement of coloured transition systems, we consider in this section an extended version of the 'rooms' example of Section 4. As before, we have two categories of agents, male and female, say, and some configuration of interconnecting rooms between which the agents can move. In any transition, an agent can move to an adjoining room or stay where it is, and, in the first instance, we will suppose that any number of agents can move at a time, even through the same connecting doorway. We also have a system norm to the effect that any state in which a male agent and a female agent are alone in a room is (globally) red. This part is just as before. We now extend the example with some agent-specific norms. As a concrete example (one of many that could be devised) let us attempt to specify an (imperfect) protocol for recovery from red system states: whenever a male agent and a female agent are alone in a room, anywhere, every male agent is required to move to the room to its left (if there is one), and every female agent is required to move to the room to its right (if there is one). More precisely: in any state s of the system in which there is a male agent and a female agent alone in a room, for every male agent x, the action of moving to the room on its left is $green(x)$, the action of staying where it is when there is no room to its left is $green(x)$, and any other move by x is $red(x)$. And similarly for female agents, but with 'left' replaced by 'right'. Further (let us suppose) in a global state s of the system where there is not a male agent and a female agent alone in a room, for any agent x, a move by x in any direction, including staying where it is, is $green(x)$. Thus, the agents are free to move around from room to room, but if ever the system enters a red global state, their individual norms or protocols require them to move to the left or right as the case may be; once the system re-enters a green global state they are free to move around again.

The precise mechanism by which agents detect that there is a male agent and a female agent alone in a room somewhere is not modelled at this level of detail. We will simply assume that there is some such mechanism—a klaxon sounds, or a suitable message is broadcast to all agents—the details do not matter for present purposes. Similarly, we are not modelling here how an agent determines which way to move. In a more detailed representation, we could model an agent's internal state, its perceptions of the environment in which it operates, how it determines where to move, and the mechanism by which it perceives that there is a male agent and a female agent alone in a room. We will not do so here: the simpler model is sufficient for present purposes. Evidently, the agent-specific norms described above are easily expressed using $n\mathcal{C}+$ permission laws of the form (15); we omit the details.

6.1. Fully compliant behaviour

As suggested above, we might now be interested in examining the relationship between system norms and individual agent-specific norms—in the present example, for instance, to determine whether the agent-specific norms expressed by the $green(x)$ specification do have the desired effect of guaranteeing recovery from a red system state to a green system state. Given a coloured agent-stranded transition system representing the system norms and individual norms, defined by an $n\mathcal{C}+$ action description or by some other means, we focus attention on those paths of the transition system that start at a red system state, and along which every agent always acts in accordance with its protocol: we consider only those paths in which every transition is $green(x)$ for each of the agents x. A natural property to look for is whether all such paths eventually pass through a green system state; if this property holds, it indicates that the agent-centric protocols are doing a good job in ensuring that systems in violation of their global system norms eventually recover to a green state, assuming (as we are) that all agents follow their individual protocols correctly. (It ought to be noted that there is a further natural requirement: in the case where the system is initially in a red system state s, there should be at least one transition $(s, \varepsilon, s') \in R$. Otherwise, the requirement that all paths starting at s eventually reach a green system state would be vacuously satisfied.)

In particular applications, it might not be a reasonable assumption to make that agents always act in accordance with their individual protocols; this might be for several reasons. Sometimes physical constraints in the environment being modelled prevent joint actions in which all agents act well; in other circumstances, and noteworthy especially because we have in mind application areas in multi-agent systems, agents may not comply with the norms that govern them because it is more in their interests not to comply. In the latter case, penalties are often introduced to try and coerce agents into compliance, and it would clearly be possible to introduce more detail into our action descriptions in order to study such penalty mechanisms. We leave that discussion to one side, however, as it is tangential to the current line of enquiry.

We now move to the 'rooms' example in particular, and study what happens when we assume that all agents are acting in accordance with their individual protocols. It is clear that the effectiveness of our protocol (if in a red state, males move to the left when possible, females move to the right when possible) in guaranteeing that the system will eventually reach a green state, depends on the topography of rooms and connecting doors. However, let us assume that there is a finite number of rooms, each room has at least one connecting room to its left or one to its right, and that there are no cycles in the configuration, in the sense that if an agent continues moving in the same direction it will never pass first out of, then back into, the same room. Under these circumstances, and removing the restriction on how many agents can pass through a door at the same time, it is intuitive that there is always a recovery, in the sense defined, from every red system state. Since all agents act in accordance with their protocols, every male will move to the left (if it can), and every female will move to the right (if it can). If the resulting system state is not green, they will move again. Eventually, in the worst case, the males and females will be segregated in separate rooms, which is a green system state.

Of course, we cannot guarantee that having reached a green system state, the agents will not re-enter a red state: the individual protocols only dictate how agents should behave when the system is globally red. Once the system has recovered, the agents may mingle again. It is easy to imagine how we might use a model-checker to verify this and similar properties on coloured agent-stranded transition systems; we will not discuss the details in this paper.

6.2. Non-compliant behaviours

One must be careful not to assume that if an agent x fails to comply with its individual norms—if some transition $(s, \varepsilon, s') \in red(x)$—then it must be that agent x acted wilfully, perhaps to seek some competitive advantage, or

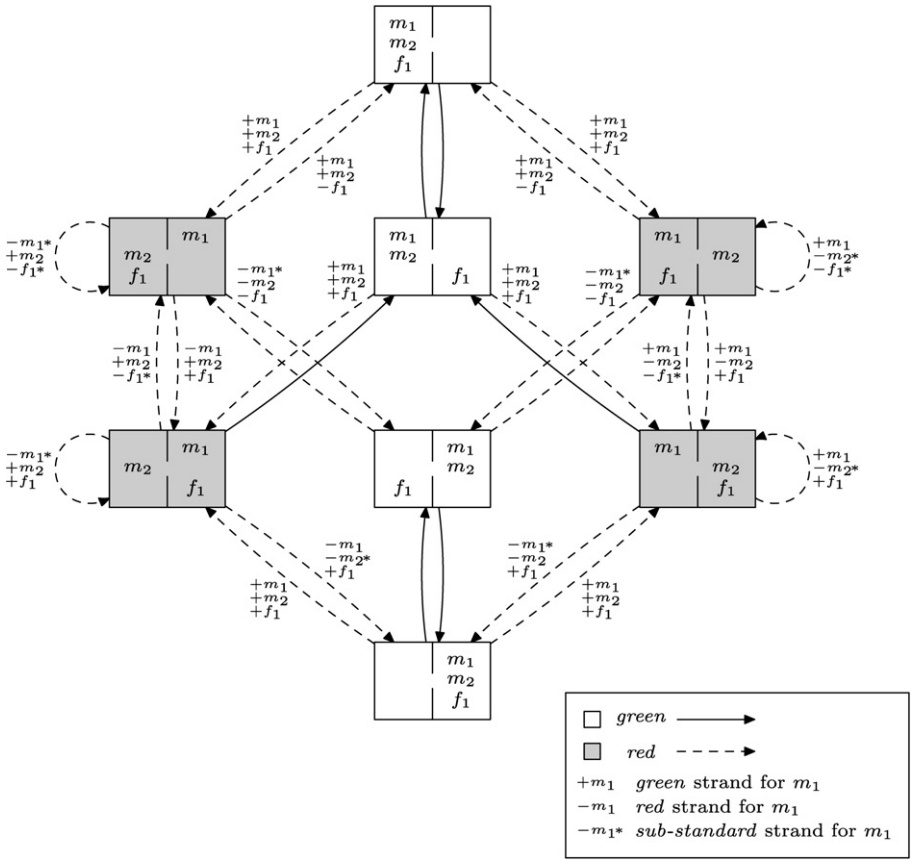

Fig. 4. Transitions without annotations have all three stands coloured green. Reflexive arcs on green nodes, where no agent moves, are omitted from the diagram: they are all globally green, and each agent strand is green. (The concept of a *sub-standard* strand is explained in Section 6.3.)

carelessly, or if it is a simple reactive device, that its constructors failed to implement it correctly. This may be so, but an agent may also fail to comply with its norms because of factors beyond its control, because it is prevented from complying by the actions of other agents, or by extraneous factors in the environment. To illustrate: suppose we modify the 'rooms' example so that now it is impossible for more than one agent to pass through the same doorway at the same time. All other features, including the specification of system norms and agent-specific norms, remain as before. Clearly the situation can now arise where several agents are required by their individual norms to pass through the same doorway; at most one of them can comply, and if one does comply, the others must fail to comply.

Again, in order to keep diagrams of the transition system small enough to be shown in full, we will consider just the case of two interconnecting rooms, and three agents, m_1, m_2, and f_1, of whom the first two are male and the last is female. Fig. 4 shows the coloured agent-stranded transition system for this version of the example. We have adopted here the local-global coherence constraint (14) which is why some transitions that were globally green in the version of Section 4 are now globally red. Nothing essential in what follows depends on this. Transition labels are omitted from the diagram: since at most one agent can move at a time, they are obvious from looking at the states. Annotations on the arcs indicate the colourings of the three agent strands for each transition; where arcs have no such annotation all strands are $green(x)$ for each of the three agents x. Omitted from the diagram are reflexive arcs from the green system states to themselves, representing transitions in which no agent moves. These transitions are all globally green, and therefore also (given local-global coherence) $green(x)$ for each agent x. The significance of the asterisks in some of the annotations will be explained presently.

One can see from the diagram that the system exhibits the following kinds of behaviour, among others.

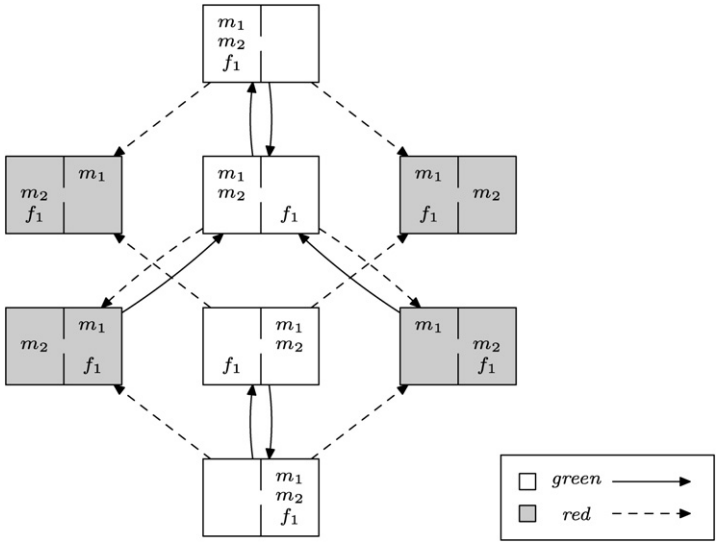

Fig. 5. System behaviour if all three agents comply with their individual norms. The colours of transition strands are not shown: all are green. Reflexive arcs on green nodes are omitted from the diagram.

- There are transitions whose strands are $green(x)$ for all three agents x but which are nevertheless globally red (all those from a green system state to a red system state). This is because the individual norms do not constrain agents' actions in green system states, as discussed earlier.
- There are globally green transitions from red system states to green system states (such as the one from the state at the lower right of the diagram in which m_2 moves to the left and m_1 and f_1 stay where they are). These are transitions in which all three agents are able to comply with their individual norms. In this example, though not necessarily in other versions with more elaborate room configurations and more agents, such transitions always recover from a red system state to a green system state.
- There are also globally red transitions in which at least one agent fails to comply with its individual norms and that lead from a red system state to a green system state (such as the one from the state at the lower right of the diagram in which m_1 moves to the right and m_2 and f_1 stay where they are). These transitions recover from a red system state to a green system state but not in accordance with the individual norms for agents.
- There are globally red transitions, such as the one from the state at the upper right of the diagram in which m_1 moves to the right, and f_1 and m_2 stay where they are, in which no agent complies with its individual norms.
- And as the example is designed to demonstrate, there are globally red transitions where one agent complies with its individual norms but in doing so makes it impossible for one or both of the others to comply with theirs. For example, in the red system state at the upper right of the diagram, where m_1 and f_1 are in the room on the left and m_2 is on the room on the right, there is no transition in which both m_2 and f_1 can comply with their individual norms.

In this modified version of the example, what are the possible system behaviours in the case where all agents do comply with their individual norms? Fig. 5 shows the transition system that results if all $red(x)$ transitions are discarded, for all three agents x. The diagram confirms that when there is a constraint preventing more than one agent from moving through a doorway at a time, the system can enter a state from which there is no transition unless at least one agent fails to comply with its individual norms. In the diagram, these are the two red system states where the female agent f_1 is in the left-hand room with a male.

6.3. Sub-standard behaviours

The example is designed to demonstrate several different categories of non-compliant agent behaviour. We pick out one for particular attention. Consider the state in which m_1 and f_1 are in the room on the left and m_2 is in the room on the right. (This is the red system state at the upper right of the diagram.) Because of the constraint on moving through

the doorway, it is not possible for all three agents to comply with their individual norms. But suppose that each agent behaves in such a way that it will comply with its individual norms *in as much as it can*. A purely reactive agent, let us suppose, is programmed in such a way that it will attempt to act in accordance with its individual norms though it may not always succeed if something prevents it. A deliberative agent (human or computer) incorporates its individual norms in its decision-making procedures and takes them into account when planning its actions: it will always attempt to act in accordance with its individual norms but again may be unsuccessful. If all agents in the system behave in this way, then there are two possible transitions from the red system state in which m_1 and f_1 are on the left and m_2 is on the right: either f_1 succeeds in moving to the right in accordance with its individual norms, or m_2 succeeds in moving to the left in accordance with its. The third possible transition from this system state, in which every agent stays where it is, can be ignored: it can only occur if no agent attempts to act in accordance with its individual norms, and this, we are supposing, is not how the agents behave. The exact mechanism which determines which of the two agents m_2 and f_1 is successful in getting through the doorway is not represented at the level of detail modelled here. At this level of detail, all we can say is that one or other of the agents m_2 and f_1 will pass through the doorway but we cannot say which.

Similarly, in the red system state at the lower right of the diagram, in which m_1 is on the left and m_2 and f_1 are on the right, we can ignore the transition in which m_1 moves to the right, if m_1's behaviour is such that it always attempts to comply with its individual norms. The transition in which f_1 moves to the left can also be ignored, if f_1's behaviour is to attempt to comply with its individual norms. And the transition in which m_2 stays where it is can be ignored, if m_2's behaviour is to attempt to comply with its individual norms. This leaves just one possible transition, in which m_2 attempts to move to the left; this will succeed because the other two agents will not act in such a way as to prevent it. (We are tempted to refer to this kind of behaviour as behaviour in which every agent 'does the best that it can' to comply with its individual norms. The term has too many unintended connotations, however, and so we avoid it.)

We are not suggesting, of course, that agents *always* behave in this way, only that there are circumstances where they do, or where it can be reasonably assumed that they do, and where we are interested in examining what system behaviours result in that case.

We now make these ideas more precise. We will say that x's behaviour ε_x in a transition (s, ε, s') is *sub-standard* if the transition is $red(x)$ and, had x acted differently in state s while all other agents, and the environment, acted in the same way they did, the transition from system state s could have been $green(x)$: x could have acted differently and complied with its individual norms. Formally, let *sub-standard* be a function from the set of agents Ag to $\wp(R)$. For every agent $x \in Ag$ and every transition $(s, \varepsilon, s') \in R$:

$$
\begin{aligned}
(s, \varepsilon, s') \in \textit{sub-standard}(x) \text{ iff } & (s, \varepsilon, s') \in \textit{red}(x) \text{ and there exists} \\
& (s, \varepsilon', s'') \in \textit{green}(x) \text{ such that } \varepsilon'_{\text{env}} = \varepsilon_{\text{env}} \text{ and} \\
& \varepsilon'_y = \varepsilon_y \text{ for every agent } y \in Ag - \{x\}
\end{aligned} \tag{17}
$$

Notice that the definition allows for the possibility of actions in the environment. It is easy to imagine other versions of the example where an agent may be unable to act in accordance with its individual norms not because of the actions of other agents but because of extraneous factors in the environment. (Suppose, for instance, that an agent is unable to move to the room on the left while it is raining.) And here is a reason why we prefer not to treat 'the environment' as a kind of agent: we do not want to be talking about *sub-standard* behaviours of the environment, or of agents preventing the environment from acting in accordance with its individual norms. In this respect at least, 'the environment' is a very different kind of agent from the others. Notice finally that the definition does not need to refer to the possibility of non-deterministic transitions: the 'absence of moral luck' constraint (13) makes that unnecessary.

Alternatively, as another way of looking at it, we could say that a $red(x)$ transition (s, ε, s') is *unavoidably* $red(x)$ if, for every transition $(s, \varepsilon', s'') \in R$ such that $\varepsilon'_{\text{env}} = \varepsilon_{\text{env}}$ and $\varepsilon'_y = \varepsilon_y$ for every agent $y \in Ag - \{x\}$, we have that $(s, \varepsilon', s'') \in red(x)$. This is closer to the informal discussion above. It is easy to see that every $red(x)$ transition is *sub-standard*(x) if and only if it is not *unavoidably* $red(x)$. Indeed, every $red(x)$ transition is either *sub-standard*(x) or *unavoidably* $red(x)$, but not both.

The diagram of the transition system for this example shown earlier in Fig. 4 shows the sub-standard transitions for each agent. They are those in which the transition annotations are marked with an asterisk. For example, in the red system state at the upper right of the diagram, where m_1 and f_1 are on the left and m_2 is on the right, the transition in which all three agents stay where they are is *sub-standard*(m_2), because there is a $green(m_2)$ transition from this

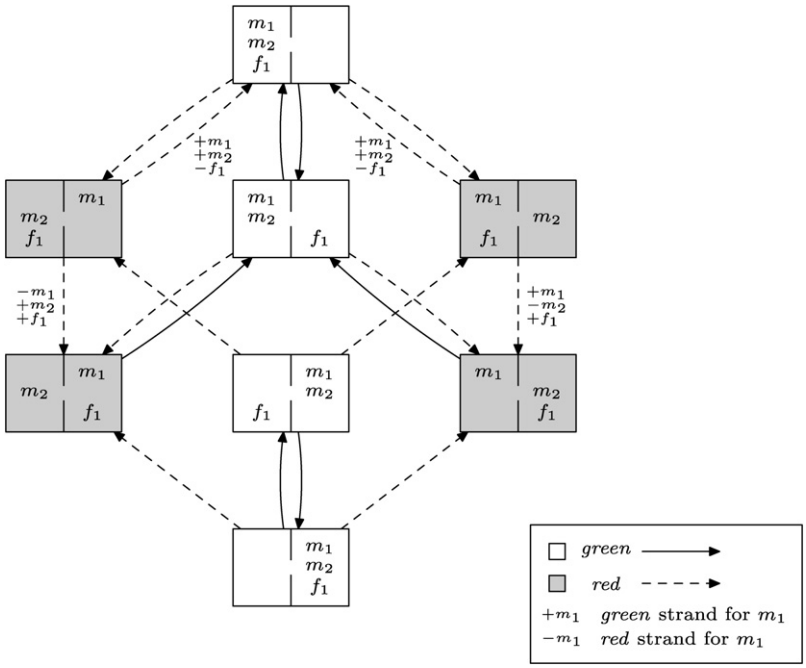

Fig. 6. System behaviour if all three agents comply with their individual norms, in as much as they can. Transitions without annotations have all three strands coloured green. Reflexive arcs on green nodes are omitted from the diagram.

state in which m_1 and f_1 act in the same way and m_2 acts differently, namely the transition in which m_1 and f_1 stay where they are and m_2 moves to the left in accordance with its individual norms. Similarly, the transition from that same system state, in which m_1 moves to the right and m_2 and f_1 stay where they are, is *sub-standard*(m_1) because the transition where all three agents stay where they are is *green*(m_1). And likewise for the other transitions marked as sub-standard in the diagram. The *red*(x) transitions not marked as *sub-standard*(x) are unavoidably *red*(x).

Suppose we wish to examine what system behaviours result if all three agents comply, in as much as they can, with their individual norms, in other words, if we disregard those transitions which are sub-standard for any of the three agents x. The result is shown in Fig. 6.

Many other variations of the example could be examined in similar fashion. If female agents are more reliable than male agents, for instance, we might be interested in examining what system behaviours result when there is never sub-standard behaviour by females though possible sub-standard behaviour by males.

As a final remark, notice that what is sub-standard or unavoidably *red*(x) for an agent x can depend on *normative* as well as *physical* constraints. Suppose (just for the sake of an example) that there is another individual norm for m_1 to the effect that it should never stay in a particular room (say, the room on the left) but should move out immediately if it enters it: a transition in which m_1 stays in the room on the left is *red*(m_1), in every system state, red or green. With this additional constraint, some of the transitions that were globally green are now globally red because of the local-global coherence constraint (assuming we choose to adopt it). But further, the transition from the red system state at the upper right of the diagram in Fig. 4, in which m_1 moves to the right and m_2 and f_1 stay in the room on the right, was previously *sub-standard*(m_1). It is no longer *sub-standard*(m_1): there is now no *green*(m_1) transition from this state when m_2 and f_1 stay where they are.

Clearly, in this example, if m_1 is in the room on the left in a red system state, it has conflicting individual norms: one requiring it to move to the right, and one requiring it to stay where it is. It cannot comply with both, so neither action is *sub-standard*(m_1); both are *unavoidably red*(m_1). How m_1 should resolve this conflict is an interesting question but not one that we intend to consider here. It is also a question that only has relevance when m_1 is a deliberative agent which must reason about what to do. If m_1 is a purely reactive device, then its behaviour in this case could perhaps be predicted by examining its program code. Both of these possibilities are beyond the level of detail of agent and system behaviours modelled in this paper.

7. Conclusion

We presented the main elements of the language $n\mathcal{C}+$, leaving out its treatment of 'counts as' relations between actions [6] which we did not discuss and a refinement which deals with 'graded transition systems' where system states are classified according to how well they satisfy system norms [8]. We focused instead on a new agent-centric refinement of $n\mathcal{C}+$ which allows us to speak about an individual agent's actions and the individual norms which govern them in addition to norms which govern system behaviour as a whole. There are really two separate aspects to the presentation. The first concerns the varieties of coloured transition systems that we use as a semantic device: for the original version of $n\mathcal{C}+$, a simple green/red colouring of system states and system transitions, and for the agent centric version, a more elaborate structure which picks out the agent-specific strands in each system transition and allows these to be coloured independently to represent individual norms. The second aspect concerns the use of $n\mathcal{C}+$ as a convenient formalism for defining these structures succinctly and encoding them in a computational representation. These methods, and the associated computational tools, are inherited from the language $\mathcal{C}+$ with minor modification. We did not discuss in this paper the range of query languages that could be used with these semantic structures. A wide variety of deontic and branching-time temporal operators, in various combinations, can already be defined and evaluated on the simple coloured transition systems. The possibilities are even greater for coloured agent-stranded transition systems. It would be interesting to examine, for example, how to adapt recent proposals for deontic query languages based on CTL [19] in this connection.

We used a simple example of agents moving around rooms to illustrate how the finer structure of agent-stranded transition systems allows us to distinguish several different categories of non-compliant agent behaviour. We focused in particular on what we called 'unavoidably red' agent behaviour, and how to investigate what system behaviours result when agents can be assumed to 'do the best that they can' to comply with their individual norms, in the sense that they never perform what we called 'sub-standard' actions. Clearly there are further distinctions that could be investigated. We also mentioned, but did not pursue, the question of whether it is possible to derive agent-specific individual norms from system norms. This raises a number of new questions, however, which deserve fuller discussion and which we therefore leave for a separate paper.

It might be felt that our main 'rooms' example is too simple to be taken seriously as representative of real-world domains. We deliberately chose the simplest configuration of rooms and agents that allowed us to make the points we wanted to make, while still being able to be depicted in their entirety. The example works just as well with more rooms, more than two categories of agents, and a wider repertoire of actions that the agents are able to perform. Generally, the issues we have addressed arise whenever we put together a complex system of interacting agents, acting independently, whose individual behaviours are subject to norms or protocols, and where we wish to impose further system norms to regulate their possible interactions.

References

[1] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, H. Turner, Nonmonotonic causal theories, Artificial Intelligence 153 (2004) 49–104.
[2] CCALC: http://www.cs.utexas.edu/users/tag/cc.
[3] V. Akman, S.T. Erdoğan, J. Lee, V. Lifschitz, H. Turner, Representing the Zoo World and the Traffic World in the language of the Causal Calculator, Artificial Intelligence 153 (2004) 105–140.
[4] A. Artikis, M.J. Sergot, J. Pitt, Specifying electronic societies with the Causal Calculator, in: F. Giunchiglia, J. Odell, G. Weiss (Eds.), Agent-Oriented Software Engineering III, in: LNCS, vol. 2585, Springer, 2003, pp. 1–15.
[5] A. Artikis, M.J. Sergot, J. Pitt, An executable specification of an argumentation protocol, in: Proc. 9th International Conference on Artificial Intelligence and Law (ICAIL'03), Edinburgh, ACM Press, 2003, pp. 1–11.
[6] M.J. Sergot, $(\mathcal{C}+)^{++}$: An action language for modelling norms and institutions, Technical Report 2004/8, Dept. of Computing, Imperial College London (2004).
[7] M.J. Sergot, Modelling unreliable and untrustworthy agent behaviour, in: B. Dunin-Keplicz, A. Jankowski, A. Skowron, M. Szczuka (Eds.), Monitoring, Security, and Rescue Techniques in Multiagent Systems. Advances in Soft Computing, Springer, 2005, pp. 161–178.
[8] M.J. Sergot, R. Craven, The deontic component of action language $n\mathcal{C}+$, in: Proc. Deon'06, in: LNAI, vol. 4048, pp. 222–237.
[9] A. Lomuscio, M.J. Sergot, Deontic interpreted systems, Studia Logica 75 (1) (2003) 63–92.
[10] A. Lomuscio, M.J. Sergot, A formalisation of violation, error recovery, and enforcement in the bit transmission problem, Journal of Applied Logic 2 (2004) 93–116.
[11] J.J.C. Meyer, A different approach to deontic logic: Deontic logic viewed as a variant of dynamic logic, Notre Dame Journal of Formal Logic 29 (1) (1988) 109–136.
[12] T. Maibaum, Temporal reasoning over deontic specifications, in: J.J.C. Meyer, R.J. Wieringa (Eds.), Deontic Logic in Computer Science: Normative System Specification, John Wiley & Sons, Chichester, England, 1993, pp. 141–202.

[13] J. Broersen, Modal action logics for reasoning about reactive systems, PhD thesis, Vrije Universiteit Amsterdam, 2003.
[14] Y. Moses, M. Tennenholtz, Artificial social systems, Computers and AI 14 (6) (1995) 533–562.
[15] A. Artikis, J. Pitt, M.J. Sergot, Animated specification of computational societies, in: C. Castelfranchi, W.L. Johnson (Eds.), Proc. 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02), Bologna, ACM Press, 2002, pp. 1053–1062.
[16] L. van der Torre, Causal deontic logic, in: Proceedings of the Fifth Workshop on Deontic Logic in Computer Science (Deon2000), 2000, pp. 351–367.
[17] J. Carmo, A.J.I. Jones, Deontic database constraints, violation and recovery, Studia Logica 57 (1) (1996) 139–165.
[18] R. van der Meyden, The dynamic logic of permission, Journal of Logic and Computation 6 (3) (1996) 465–479.
[19] T. Agotnes, W. van der Hoek, J. Rodríguez-Aguilar, C. Sierra, M. Wooldridge, On the logic of normative systems, in: Proceedings of the 20th International Joint Conference on Artificial Intelligence, 2007, pp. 1175–1180.