Robert Craven
Marek Sergot

Distant Causation in $\mathcal{C}+$

Abstract. The action language $\mathcal{C}+$ of Giunchiglia, Lee, Lifschitz, McCain and Turner is a high-level, logical formalism for the representation of domains involving action and change. However, one cannot directly express relationships which hold between states more than one time-step distant, or even say that one action determines another at the next time. We present $\mathcal{C}+_{timed}$, a generalization of $\mathcal{C}+$ which removes these limitations. As for $\mathcal{C}+$, translations to the language of causal theories are given. We also define a new kind of transition system called a 'run system' to provide a graphical semantics. Finally, we show how domains involving prohibition and permission can be modelled, by incorporating the ideas of another extension of $\mathcal{C}+$.

Keywords: Action languages, temporal reasoning, causation

1. Introduction

The action language $\mathcal{C}+$ of Giunchiglia, Lee, Lifschitz, McCain and Turner [GLL$^+$04] is a high-level formalism for describing how the properties of a system change in response to actions performed in the system.

However, a limitation of $\mathcal{C}+$ as it currently stands is that when writing causal laws one must refer only to states at most one time-step away from each other. It is also impossible directly to say that the performance of an action causes the performance of another action at the next time-step, or at some other time in the future. Yet the fact that an action description of $\mathcal{C}+$ can be viewed as shorthand for an infinite sequence of causal theories (a sequence indexed by the natural numbers), together with the fact that there is no limitation on the times to which the rules of those causal theories refer, suggests that one may expand $\mathcal{C}+$, removing the restriction on temporal distance.

There will be benefits. Currently, encoding domains in $\mathcal{C}+$ in which delays and deadlines play an important role is awkward at best and computationally expensive at worst. For example, suppose one wanted to say that 20 time-steps after a *set* action, a *switch* turned to *red*. The signature would at least contain the fluent constant *switch*, with $dom(switch) = \{green, red\}$, and an action constant *set*, with a Boolean domain. But how are the causal laws to be written? One might think of something along the following lines:

Special Issue on Reasoning about Action and Change
Edited by **Gerhard Brewka and Pavlos Peppas**

inertial *switch*,	$\mathit{timer}{=}1$ **if** $\top$ **after** *set*,
exogenous *set*,	$\mathit{timer}{=}2$ **if** $\top$ **after** $\mathit{timer}{=}1$,
default $\mathit{timer}{=}\mathit{none}$,	$\mathit{timer}{=}3$ **if** $\top$ **after** $\mathit{timer}{=}2$,
	$\vdots$
	$\mathit{timer}{=}20$ **if** $\top$ **after** $\mathit{timer}{=}19$,
	$\mathit{switch}{=}\mathit{red}$ **if** $\mathit{timer}{=}20$.

But this will not do. For what if one time-step after the *set* action, another *set* action is performed? Let us suppose that the idea is, not to reset the timer so that countdown begins anew, but rather to ensure that the *switch* is *red* at another time. It seems obvious that one will need a whole series of timer_i constants, governed by a series of laws which decides when each of them is to be started:

inertial *switch*,
exogenous *a*,
$\mathit{timer}_1{=}1$ **if** $\top$ **after** $\mathit{set} \wedge \mathit{timer}_1{=}\mathit{none}$,
$\mathit{timer}_1{=}2$ **if** $\top$ **after** $\mathit{timer}_1{=}1$,
$\vdots$
$\mathit{timer}_1{=}20$ **if** $\top$ **after** $\mathit{timer}_1{=}19$,
$\mathit{switch}{=}\mathit{red}$ **if** $\mathit{timer}_1{=}20$,
default $\mathit{timer}_1{=}\mathit{none}$,
$\mathit{timer}_2{=}1$ **if** $\top$ **after** $\mathit{set} \wedge \neg\mathit{timer}_1{=}\mathit{none} \wedge \mathit{timer}_2{=}\mathit{none}$,
$\mathit{timer}_2{=}2$ **if** $\top$ **after** $\mathit{timer}_2{=}1$,
$\vdots$
$\mathit{switch}{=}\mathit{red}$ **if** $\mathit{timer}_2{=}20$,
default $\mathit{timer}_2{=}\mathit{none}$,
$\mathit{timer}_3{=}1$ **if** $\top$ **after** $\mathit{set} \wedge \neg\mathit{timer}_1{=}\mathit{none} \wedge \neg\mathit{timer}_2{=}\mathit{none}$
$\wedge\, \mathit{timer}_3{=}\mathit{none}$,
$\vdots$
$\mathit{timer}_{21}{=}1$ **if** $\top$ **after** $\mathit{set} \wedge \neg\mathit{timer}_1{=}\mathit{none} \wedge \cdots \wedge \neg\mathit{timer}_{20}{=}\mathit{none}$
$\wedge\, \mathit{timer}_{21}{=}\mathit{none}$,

$$\vdots$$

default $timer_{21}{=}none$.

That will now give the results we wanted. But it is hardly a perspicuous representation of the behaviour of the system, and though one might invent macros to generate such causal laws, there will clearly be a computational penalty to pay in the implicit causal theories with which CCalc,[1] for example, operates.

In this paper we present a solution.

2. Preliminaries

2.1. Causal Theories

The language of *causal theories* [GLL$^+$04] can be used to formalize reasoning about action and change. It is mathematically simple, and can be seen as underlying $\mathcal{C}+$. Accordingly, a brief account is given here. The reader is referred to the cited paper for further details and examples.

Begin with a multi-valued propositional signature, composed of a set σ of *constants*, and for each constant $c \in \sigma$ a non-empty, finite set $dom(c)$, disjoint from σ and known as the *domain* of c. An *atom* of the signature is an expression $c{=}v$, where $c \in \sigma$ and $v \in dom(c)$. *Formulas* are constructed from the atoms using propositional connectives and a familiar syntax, with a *literal* as an expression A or $\neg A$, for atomic A. The expressions $\top$ and $\bot$ stand for $A \vee \neg A$ and $A \wedge \neg A$ respectively, with A being an arbitrary atom.

Further, *Boolean* constants are those whose domain is the set of truth values $\{\mathbf{t}, \mathbf{f}\}$, and a *Boolean* signature is, by extension, one all of whose constants are Boolean. If c is a Boolean constant, we often write c for $c{=}\mathbf{t}$, so that where our propositional signatures are restricted to be Boolean and we deal with no formula containing $\mathbf{f}$, we may reduce our syntax to that of standard propositional logic.

A *causal rule* is an expression of the form

$$F \Leftarrow G,$$

where F and G are formulas of the underlying, multi-valued propositional signature. Such expressions are related to the (almost) natural language statement "if G, then the fact that F is caused". A *causal theory* is a set of causal rules.

[1]`http://www.cs.utexas.edu/users/tag/cc/ccalc.html`.

An *interpretation* of a multi-valued propositional signature σ is a function mapping every constant c to some $v \in dom(c)$; an interpretation X is said to *satisfy* an atom $c{=}v$ if $X(c) = v$, and in this case one may write $X \models c{=}v$. Standard structural recursions over the propositional connectives apply, and where Γ is a set of formulas of our propositional signature, $X \models \Gamma$ expresses that $X \models F$, for every F in Γ.

Now let Γ be a causal theory, and take X to be an interpretation of its underlying propositional signature. The *reduct* of Γ with respect to X is defined as

$$\Gamma^X \;=_{def}\; \{F \mid F \Leftarrow G \in \Gamma \text{ and } X \models G\}.$$

X is a model of the causal theory Γ, written $X \models_{\mathsf{C}} \Gamma$, if X is the unique model of Γ^X.

2.2. The action language $\mathcal{C}+$

2.2.1. Syntax

Now, a brief summary of the action language $\mathcal{C}+$; for fuller details and examples, consult the original presentation [GLL$^+$04].

As with the logic of causal theories, the language is based on a multi-valued propositional signature σ, with σ partitioned into a set σ^f of *fluent constants* and a set σ^a of *action constants.* Further, the fluent constants are partitioned into those which are *simple* and those which are *statically determined.* A *fluent formula* is a formula whose constants all belong to σ^f; an action formula has at least one action constant and no fluent constants.

A *static law*[2] is an expression of the form

$$F \textbf{ if } G,$$

where F and G are fluent formulas. An *action dymanic law* is an expression of the same form in which F is an action formula and G is a formula. A *fluent dynamic law* has the form

$$F \textbf{ if } G \textbf{ after } H,$$

where F and G are fluent formulæ and H is a formula, with the restriction that F must not contain statically determined fluents. *Causal laws* are static laws or dynamic laws, and an *action description* is a set of causal laws.

[2]The keyword **caused**, which in the original presentation of $\mathcal{C}+$ appears at the beginning of static and dynamic rules, is omitted; causal laws have no especial relationship with causation, although they can be used to model causal connections. (This latter point justifies the title of the paper.)

An action description D is said to be *definite* when

- the head of every causal law of D is either an atom or $\bot$, and
- no atom is the head of infinitely many causal laws of D.

2.2.2. Semantics and causal theories

The language $\mathcal{C}+$ can be viewed as a useful shorthand for the logic of causal theories, for to every action description D of $\mathcal{C}+$ and non-negative integer t, there corresponds a causal theory Γ_t^D. The signature of Γ_t^D contains constants $c[i]$, such that

- $i \in \{0, \dots, t\}$ and c is a fluent constant of the signature of D, or
- $i \in \{0, \dots, t-1\}$ and c is an action constant of the signature of D,

and the domains of such constants $c[i]$ are kept identical to those of their constituents c. The expression $F[i]$, where F is a formula, denotes the result of suffixing $[i]$ to every occurrence of a constant in F. The causal rules of Γ_t^D are:

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

for every static law in D and every $i \in \{0, \dots, t\}$, and for every action dynamic law in D and every $i \in \{0, \dots, t-1\}$;

$$F[i+1] \Leftarrow G[i+1] \wedge H[i],$$

for every fluent dynamic law in D and every $i \in \{0, \dots, t-1\}$; and

$$c[0]{=}v \Leftarrow c[0]{=}v,$$

for every simple fluent constant c and $v \in \mathit{dom}(c)$.

The semantics of action descriptions are defined in terms of labelled transition systems, using the translation into causal theories given above. Let us suppose we have an action description D, with signature composed of $\sigma^f \cup \sigma^a$. We will identify interpretations of the underlying propositional signature of D (which are defined in the same way as those for causal theories) with the sets of atoms they satisfy. Thus, where i is a non-negative integer and s an interpretation, we can write $s[i]$ for the result of suffixing $[i]$ to the constant in every atom made true by the interpretation (in symbols, $\{c[i]{=}v \mid s \models c{=}v\}$).

The vertices of the transition system defined by D are *states*: interpretations s of σ^f, such that $s[0]$ is a model of Γ_0^D. The edges of the transition system are triples (s, e, s'), where s and s' are interpretations of σ^f and e is an interpretation of σ^a, and such that $s[0] \cup e[0] \cup s'[1]$ is a model of Γ_1^D. These triples are known as *transitions*.

Let Γ_t^D be the causal theory generated from the action description D and non-negative integer t as described above. Let $s_0, \ldots, s_t$ be interpretations of σ^f and $e_0, \ldots, e_{t-1}$ be interpretations of σ^a. Then using the notation above, we can represent interpretations of the signature of Γ_t^D in the form

$$s_0[0] \cup e_0[0] \cup s_1[1] \cup e_1[1] \cup \cdots \cup e_{t-1}[t-1] \cup s_t[t]. \tag{1}$$

The following result holds.

Theorem 2.1. An interpretation of the signature of Γ_t^D is a model of Γ_t^D iff each triple (s_i, e_i, s_{i+1}), for $0 \leqslant i < t$, is a transition.

Proof. Proposition 8 of [GLL+04]. ■

Let D be an action description of $\mathcal{C}+$. A *run* of length t through this transition system is defined to be a sequence

$$(s_0, e_0, s_1, e_1, \ldots, e_{t-1}, s_t) \tag{2}$$

such that all triples (s_i, e_i, s_{i+1}), for $0 \leqslant i < t$, are members of the transition system.

Theorem 2.2. Let D be an action description and t any non-negative integer. Then the sequence (2) is a run of the transition system iff the interpretation (1) is a model of the causal theory Γ_t^D.

Proof. First, assume we have a run of the transition system of length t. Then every triple (s_i, e_i, s_{i+1}), for $0 \leqslant i < t$, is a transition, and so by Theorem 2.1 the interpretation (1) is a model of Γ_t^D.
Alternately, suppose that (1) is a model of the causal theory Γ_t^D. Then clearly, each triple (s_i, e_i, s_{i+1}), for $0 \leqslant i < t$, is a transition, and so the sequence (2) is a run of the transition system defined by D. ■

3. Times

One could change the syntax of $\mathcal{C}+$ as it stands in the following way. Let λ be conceived of as an operator which enables one to refer to the immediately preceding time. Fluent dynamic laws can then be written as

$$F \textbf{ if } G \wedge \lambda(H);$$

both action dynamic laws and static laws will be written in the form

$$F \textbf{ if } G,$$

which is the same as for $\mathcal{C}+$.

We will allow ourselves to nest the λ. In this extended language, signatures are defined as before for $\mathcal{C}+$, and causal laws have the form

$$F \textbf{ if } G,$$

where F is as before a formula of the signature, and G is given by

$$G \ ::= \ c{=}v \mid \top \mid \neg G \mid G_1 \wedge G_2 \mid G_1 \vee G_2 \mid \lambda(G). \tag{3}$$

We can write $\lambda\lambda(F)$ as $\lambda^2(F)$, and so on. We insist that if the left-hand side of a causal law contains a fluent constant, then no action constant should appear on the right-hand side outside the scope of any λ. Also, we stipulate that if there is a λ with index greater than 0 on the right-hand side of a law, then that law's left-hand side must not contain statically defined fluents. These restrictions are essentially inherited from $\mathcal{C}+$.

It is easy to show that causal rules defined as above have the following *canonical form*:

$$F \textbf{ if } \lambda^{n_0}(G_0) \wedge \cdots \wedge \lambda^{n_k}(G_k), \tag{4}$$

where $G_0, \ldots, G_n$ are formulas of $\sigma^f \cup \sigma^a$ (i.e., formulas containing no λ), and

- $k \geqslant 0$;
- $(n_0, \ldots, n_k)$ is a strictly increasing sequence of non-negative integers;
- if F contains fluent constants and $n_0 = 0$, then G_0 contains no action constants;
- if F contains statically determined fluent constants, then there is no λ-index greater than 0 on the right-hand side.

We have been calling the $n_0, \ldots, n_k$ λ-*indices*; we say that a law (4) has a greatest λ-index of n_k. We also allow ourselves to drop any operator λ^0, and to write λ for λ^1 An *action description* is a set of causal laws. This extension of $\mathcal{C}+$ is called $\mathcal{C}+_{timed}$.

To every action description D (signature $\sigma^f \cup \sigma^a$) of $\mathcal{C}+_{timed}$ and non-negative integer t there corresponds a causal theory Γ^D_t. The signature of Γ^D_t consists of constants $c[i]$, where

- $i \in \{0, \dots, t\}$ and $c \in \sigma^f$, or
- $i \in \{0, \dots, t-1\}$ and $c \in \sigma^a$,

and we keep the domains of such constants $c[i]$ identical to those of their constituents c. By $F[i]$, where F is a formula, we denote the expression resulting from suffixing $[i]$ to every occurrence of a constant in F. The causal laws of Γ_t^D are:

$$F[i + n_k] \Leftarrow G_0[i + n_k - n_0] \wedge \cdots \wedge G_k[i + n_k - n_k],$$

(where the last conjunct is of course equal to $G_k[i]$) for every causal law in D and every

- $i \in \{0, \dots, t - n_k - 1\}$, if F contains an action constant,
- $i \in \{0, \dots, t - n_k\}$, otherwise;

we also include

$$c[0]{=}v \Leftarrow c[0]{=}v,$$

for every simple fluent constant c and $v \in dom(c)$. So much for the language definition and translation into the language of causal theories.

For the purpose of illustration, consider the following simple domain DS, with Boolean signature $\sigma^f = \{p\}$ (p is simple) and $\sigma^a = \{a\}$. The causal laws are:

$$\begin{array}{l} \textbf{inertial } p, \\ \textbf{exogenous } a, \\ p \textbf{ if } \lambda^2(a), \end{array}$$

where the following abbreviations hold, as in $\mathcal{C}+$:

$$\begin{array}{rcll} \textbf{inertial } c & \mapsto & c{=}v \textbf{ if } c{=}v \wedge \lambda(c{=}v), & \text{for all } v \in dom(c),\ (c \in \sigma^f) \\ \textbf{exogenous } c & \mapsto & c{=}v \textbf{ if } c{=}v, & \text{for all } v \in dom(c),\ (c \in \sigma^a) \end{array}$$

For time-index 2, the causal theory determined by the action description DS is:

$$\begin{array}{rcl@{\qquad}rcl} a[0]{=}\mathbf{t} & \Leftarrow & a[0]{=}\mathbf{t}, & p[1]{=}\mathbf{t} & \Leftarrow & p[1]{=}\mathbf{t} \wedge p[0]{=}\mathbf{t}, \\ a[0]{=}\mathbf{f} & \Leftarrow & a[0]{=}\mathbf{f}, & p[1]{=}\mathbf{f} & \Leftarrow & p[1]{=}\mathbf{f} \wedge p[0]{=}\mathbf{f}, \\ a[1]{=}\mathbf{t} & \Leftarrow & a[1]{=}\mathbf{t}, & p[2]{=}\mathbf{t} & \Leftarrow & p[2]{=}\mathbf{t} \wedge p[1]{=}\mathbf{t}, \\ a[1]{=}\mathbf{f} & \Leftarrow & a[1]{=}\mathbf{f}, & p[2]{=}\mathbf{f} & \Leftarrow & p[2]{=}\mathbf{f} \wedge p[1]{=}\mathbf{f}. \\ p[2]{=}\mathbf{t} & \Leftarrow & a[0]{=}\mathbf{t}, & & & \\ p[0]{=}\mathbf{t} & \Leftarrow & p[0]{=}\mathbf{t}, & & & \\ p[0]{=}\mathbf{f} & \Leftarrow & p[0]{=}\mathbf{f}, & & & \end{array}$$

One may find the models of this theory using the 'literal completion' method [GLL$^+$04]; CCalc can do this for us. A sample model of Γ_2^{DS} is shown in Figure 1.

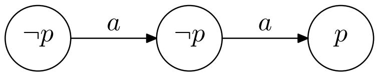

Figure 1. A model of Γ_2^{DS}

It is possible to imagine an add-on, or modification, to CCalc which compiles action descriptions of $\mathcal{C}+_{timed}$ into causal theories, and then computes the models; this would evidently not be a complex task.

4. Graphical Models

4.1. Run systems

It remains to decide what to do about the transition systems which were the original semantics for $\mathcal{C}+$. For whilst one could understand $\mathcal{C}+_{timed}$ simply as a means of writing in abbreviated form a family of causal theories, a very attractive feature of $\mathcal{C}+$ is that action descriptions can be shown to define labelled transition systems. These transition systems may afford a useful connection between $\mathcal{C}+$ and other formalisms for reasoning about actions, and of course they are useful in their own right as aids to visualisation. If we look at the transition systems defined by $\mathcal{C}+_{timed}$ theories, however, we see that important information is lacking.

Recall the means of calculating the transition system for an action description of $\mathcal{C}+$ described in Section 2.2.2. If we apply the definitions given there to action descriptions of $\mathcal{C}+_{timed}$, and calculate the transition system for the domain DS, the result is as shown in Figure 2. However, that is

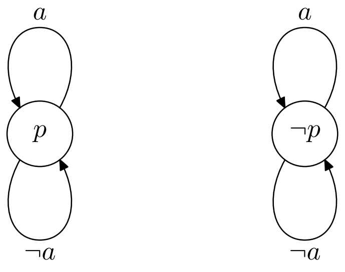

Figure 2. The (flawed) $\mathcal{C}+$-style transition system for $\mathcal{C}+_{timed}$ domain DS

clearly flawed, for the run of the system depicted in Figure 1 cannot be traced in the diagram. The reason for the failure is clear: the causal theory Γ_1^D used to find the transitions has not taken into account the law p **if** $\lambda^2(a)$ of our action description. In general, the theorem correlating paths through the transition system with models of causal theories (Theorem 2.2) does not hold for $\mathcal{C}+_{timed}$.

Accordingly, we broaden our conception and define a kind of transition system called a *run system.* These have two differences with the subclass of transition systems defined by $\mathcal{C}+$ action descriptions. First, states (understood as models of Γ_0^D) may be represented by more than one vertex. Secondly, out of each set of vertices which represent the same state, a privileged *initial* vertex will be marked. (In the following, we make use of the notation $\mathrm{I}(\sigma^f)$ to stand for the set of interpretations of σ^f, and similarly for σ^a.)

DEFINITION 4.1. A *(labelled) run system* of a signature $\sigma^f \cup \sigma^a$ is any graph G with vertices $V(G)$ and edges $E(G)$, such that:

- $V(G) \subseteq \mathrm{I}(\sigma^f) \times \mathbb{N}$,
- $E(G) \subseteq V(G) \times \mathrm{I}(\sigma^a) \times V(G)$.

Any $(s, 0) \in V(G)$ is called an *initial* vertex.

In contrast with the transition systems defined by $\mathcal{C}+$ action descriptions, vertices are pairs of states and natural numbers (the use of $\mathbb{N}$ here is arbitrary, and is simply a means of allowing more than one vertex to be associated with a given state). Where (s, n) is a vertex of a run system, we call s the *state component*, and if $((s, n), e, (s', n'))$ is an edge of a run system, then we call (s, e, s') the *transition component.*

We say that a run system G *represents* an action description D of $\mathcal{C}+_{timed}$ when, for all $t \geqslant 0$, paths of length t through G starting at an initial vertex correspond to models of the causal theory Γ_t^D. For any action description of $\mathcal{C}+_{timed}$ there clearly exists at least one run system representing it; the correspondence is clearly not $1{-}1$.

In diagrams, we will represent the initial states by circling them twice, and we will not include the n component of our vertices (s, n). As an illustration of what we are working towards, a run system for the domain DS is shown in Figure 3. It can clearly be seen that paths through this run system beginning at one of the twice-circled vertices correspond to models of the causal theories generated by DS.

We now introduce the notion of a 'commitment', which will be of central importance in the generation of run systems from action descriptions of $\mathcal{C}+_{timed}$.

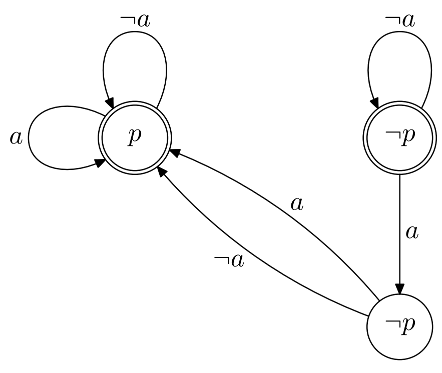

Figure 3. Run system for the simple action description DS

4.2. Commitments

The idea is that as a system makes runs, it may accrue commitments, which express that something should be true in the future, if certain other things are true. (Thus we use 'commitment' here as a technical term, unrelated to the concept encountered in, for example, multi-agent systems.) Commitments have the same syntax as causal rules, to which they have a very close relationship.

Figure 4 represents a model of the causal theory Γ_2^{DS} (the same model shown in Figure 1). That model, as we know, can be partitioned into sets

$$s_0[0] \cup e_0[0] \cup \cdots \cup s_2[2],$$

and in our diagram the vertices are the s_i, and the edges (s_i, e_i, s_{i+1}). To the right of each vertex, a set of commitments has been drawn. These commitments stem from the law p **if** $\lambda^2(a)$ in the action description DS. The first thing to notice is that vertices which encode the same interpretation of σ^f may be labelled by different sets of commitments: this is true of the first and second vertices in the diagram. From this flows the fact that in the run system depicted in Figure 3, there are two vertices which are labelled with $\neg p$.

In commitments, the time-stamp is to be understood as relative to the current state—i.e., relative to the one which is labelled by the commitment. Thus the fact that the second vertex in our diagram is labelled by the commitment $p[2] \Leftarrow a[0]$ means that if a is performed at the outgoing edge, then 2 units into the future, p must be true.

Starting at the top and moving down through the diagram, we see that if a is performed on the outgoing edge of the first state, p must be true

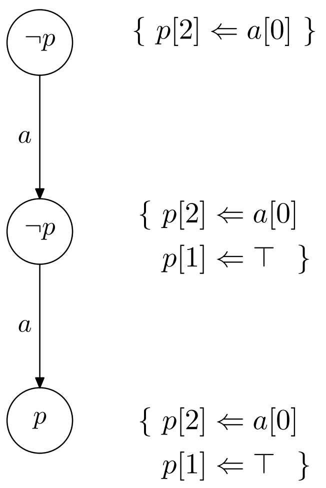

Figure 4. A model of Γ_2^{DS}, marked with commitments

two time-steps later. At the next state this is also true; further, since a *was* performed between states one and two, p is now definitely constrained to be true at the third state. The third state itself is labelled by the same commitments as the second. So it goes.

In general, there are two ways in which a state in such a diagram may come to be labelled by a commitment: because of the nature of the state itself, and because a commitment has been inherited from a previous state.

Consider a model of some causal theory Γ_t^D. If a state s_j of that model is labelled with the commitment

$$F[n] \Leftarrow G_0[n_0] \wedge \cdots \wedge G_k[n_k]$$

(where here $n \geqslant n_0$), then this should be taken to mean that if

$$\begin{gathered} s_{j+n_k} \cup e_{j+n_k} \models G_k, \\ \vdots \\ s_{j+n_0} \cup e_{j+n_0} \models G_0, \end{gathered}$$

then we must have $s_{j+n} \cup e_{j+n} \models F$.

We define a function from commitments to sets of commitments, which will be used in specifying how these commitments change from one state to another. Let x be the commitment

$$F[n] \Leftarrow G_0[n_0] \wedge \cdots \wedge G_k[n_k].$$

The value of $cmt(x)$ will be

$$\{\ F[n-1] \Leftarrow G_0[n_0-1] \wedge \cdots \wedge G_k[n_k-1]\ \}$$

if $n-1>0$, or if $n-1=0$ and F is an action atom; in this case we omit any conjuncts of the right-hand side which have time-stamps less than 0. Otherwise, the value of $cmt(x)$ is the empty set. Thus the function *cmt* has as its domain the set of commitments of some signature, and has as its range the set containing singletons of commitments, together with the empty set. As an example, we refer back to the domain DS:

$$\begin{aligned} cmt(p[2] \Leftarrow a[0]) &= \{p[1] \Leftarrow \top\}, \\ cmt(p[1] \Leftarrow \top) &= \emptyset, \end{aligned}$$

By convention, if S is a set of commitments, we set

$$cmt(S) = \bigcup_{x \in S} cmt(x).$$

4.3. Generation of run systems

Let D be an action description of $\mathcal{C}+_{timed}$, with signature $\sigma^f \cup \sigma^a$. Assume the laws of D are in canonical form, and so have the structure

$$F \textbf{ if } \lambda^{n_0}(G_0) \wedge \cdots \wedge \lambda^{n_k}(G_k) \tag{5}$$

for some sequence of natural numbers $(n_0, \ldots, n_k)$. Let $\overline{D} \subseteq D$ be the set of all those laws of D which have a maximum λ-index greater than 1, or else have this index equal to 1 and an action atom as F. Thus $\overline{D}$ contains those laws of D which could not be expressed in $\mathcal{C}+$, and $D-\overline{D}$ is, despite its odd syntax, a $\mathcal{C}+$ action description (with the same signature).

Let *init* be the set

$$\{s \mid s[0] \models \Gamma_0^D\}.$$

init can be seen as being the set of states (a notion inherited from $\mathcal{C}+$) of our system. Let the set com_D contain those commitments

$$F[n_k] \Leftarrow G_0[n_k-n_0] \wedge \cdots \wedge G_k[n_k-n_k]$$

such that there is a law of form (4) in $\overline{D}$ (the last conjunct here is of course simply $G_k[0]$).

Run systems, it will be recalled, have as their vertices pairs consisting of a state and a natural number. Before systems of this kind can be generated, we use sets of commitments as the second components. So, we make a graph G_D, which has as its set of vertices those pairs (s, c), where s is as usual an interpretation of σ^f, and where $c \subseteq \bigcup_{n \geqslant 0} cmt^n(com_D)$ and $com_D \subseteq c$. The set $\bigcup_{n \geqslant 0} cmt^n(com_D)$ includes all commitments which could label any state.

For the edges, we need to introduce a function, which will model how commitments change over time as a consequence of the preceding state of, and actions performed in, a transition. We know that all commitments which can label a state are contained in

$$\bigcup_{n \geqslant 0} cmt^n(com_D);$$

commitments in general have the form

$$F[n] \Leftarrow G_0[n_0] \wedge \cdots \wedge G_k[n_k].$$

The function introduced is $trans_{com}$, which has the domain $\mathrm{I}(\sigma^f) \cup \mathrm{I}(\sigma^a) \cup \bigcup_{n \geqslant 0} cmt^n(com_D)$, where, if x is a commitment:

$$trans_{com}(s, e, x) = \begin{cases} \emptyset & \text{if } n_k = 0 \text{ and } s \cup e \not\models G_k \\ cmt(x) & \text{otherwise.} \end{cases}$$

Clearly, we want commitments to persist (their time-stamps decremented) through a transition, only when those formulas on the right-hand side of the commitment which relate to the transition are satisfied. In our example we have that

$$\begin{aligned} trans_{com}(\{\neg p\}, \{a\}, p[2] \Leftarrow a[0]) &= cmt(p[2] \Leftarrow a[0]), \\ &= \{p[1] \Leftarrow \top\}. \end{aligned}$$

Thus we ensure the presence of the right commitments. By convention, where c is a set of commitements, we let

$$trans_{com}(s, e, c) =_{def} \bigcup_{x \in c} trans_{com}(s, e, x).$$

So, the edges of our graph G_D will be those triples $((s, c), e, (s', c'))$, where

- $s \in init$,
- $c \subseteq \bigcup_{n \geqslant 0} cmt^n(com_D)$,
- $com_D \subseteq c$,
- $s[0] \cup e[0] \cup s'[1] \models_{\mathsf{C}} \Gamma_1^D \cup c$,
- $c' = com_D \cup trans_{com}(s, e, c)$.

Now we are engaged in a graph-theoretic problem: we must determine members of $V(G_D)$ that can be reached along a path beginning at a vertex of G_D of the form (s, com_D), for some $s \in init$. Remove from $V(G_D)$ those vertices which *cannot* be so reached, and remove all edges of G_D which are incident with a vertex which has been removed. Call the result G^*_D.

All that remains is to rename the vertices of G^*_D, so that the sets of commitments are replaced by natural numbers. Thus we rename (s, c) to (s, n), with $n \in \mathbb{N}$, in such a way that if (s, c) and (s, c') are two distinct vertices of G^*_D, and their replacements are (s, n), (s, n'), then $n \neq n'$. Where the commitment part c of a vertex is com_D, we rename that vertex to $(s, 0)$.

The result is a run system for the $\mathcal{C}+_{timed}$ domain D. Call it Run_D.

A transition system T, of the type defined by a $\mathcal{C}+$ action description, is essentially a run system in which there are only initial states, so that $V(T) = \{(s, 0) \in V(T)\}$.

It remains, of course, to prove that this method of generating run systems works. The desired result states that given an action description D of $\mathcal{C}+_{timed}$, paths through the system Run_D of length t which begin at an initial vertex $(s, 0)$ correspond to models of the causal theory Γ^D_t. In other words, we need to show that Run_D represents the domain D.

THEOREM 4.2. Let D be an action description of $\mathcal{C}+_{timed}$ with signature $\sigma^f \cup \sigma^a$, and t a non-negative integer. Then, where $s_0, \ldots, s_t$ are interpretations of σ^f, and $e_0, \ldots, e_{t-1}$ interpretations of σ^a, we have that

$$s_0[0] \cup e_0[0] \cup \cdots \cup s_t[t]$$

is a model of Γ^D_t if and only if

$$((s_0, 0), e_0, (s_1, n_1), \ldots, e_{t-1}, (s_t, n_t))$$

is a path through Run_D, for some $n_1, \ldots, n_t \in \mathbb{N}$.

PROOF. Follows closely the method above for the generation of G^*_D, and relies on splitting the statement

$$s_0[0] \cup \cdots \cup s_t[t] \models_{\mathsf{C}} \Gamma^D_t$$

into a series of statements

$$s_i[i] \cup e_i[i] \cup s_{i+1}[i+1] \models_{\mathsf{C}} \Delta^D_{\mathcal{M},i},$$

for $0 \leqslant i < t$ and relevant sets $\Delta^D_{\mathcal{M},i}$, where $\mathcal{M}$ is the model $s_0[0] \cup \cdots \cup s_t[t]$. Details omitted. ■

4.4. An example generation

Let us work through the simple action description DS of $\mathcal{C}+_{timed}$, in order to illustrate the procedures described above. The domain is Boolean, has signature $\sigma^f \cup \sigma^a$, and contains the laws

$$\begin{aligned}&\textbf{inertial}\ p,\\&\textbf{exogenous}\ a,\\&p\ \textbf{if}\ \lambda^2(a).\end{aligned}$$

Thus we have $\overline{DS} = \{p\ \textbf{if}\ \lambda^2(a)\}$.

The causal theory Γ_0^{DS} is evidently

$$\begin{aligned}p[0] &\Leftarrow p[0],\\ \neg p[0] &\Leftarrow \neg p[0],\end{aligned}$$

so that the set *init* contains the two states

$$\{p\} \quad \text{and} \quad \{\neg p\}.$$

We also have com_{DS} as the singleton containing the commitment $p[2] \Leftarrow a[0]$. As was noted previously, we have

$$\begin{aligned}cmt(p[2] \Leftarrow a[0]) &= \{p[1] \Leftarrow \top\},\\ cmt(p[1] \Leftarrow \top) &= \emptyset,\end{aligned}$$

so that

$$\bigcup_{n\geqslant 0} cmt^n(com_{DS}) = \{p[2] \Leftarrow a[0], p[1] \Leftarrow \top\}.$$

We now make the graph G_{DS}. The vertices of G_{DS} are the pairings of members of *init* with those subsets of $\bigcup_{n\geqslant 0} cmt^n(com_{DS})$ which also contain com_{DS}, namely:

$$\begin{aligned}&(\{p\},\{p[2] \Leftarrow a[0])\}),\\&(\{p\},\{p[2] \Leftarrow a[0],\ p[1] \Leftarrow \top\}),\\&(\{\neg p\},\{p[2] \Leftarrow a[0]\}),\\&(\{\neg p\},\{p[2] \Leftarrow a[0],\ p[1] \Leftarrow \top\}),\end{aligned}$$

To find the edges for our graph, we first need the causal theory Γ_1^{DS}. This has the laws:

$$\begin{aligned}p[1] &\Leftarrow p[1] \wedge p[0],\\ \neg p[1] &\Leftarrow \neg p[1] \wedge \neg p[0],\\ a[0] &\Leftarrow a[0],\end{aligned}$$

$$\neg a[0] \Leftarrow \neg a[0],$$
$$p[0] \Leftarrow p[0],$$
$$\neg p[0] \Leftarrow \neg p[0].$$

The triples $((s, c), e, (s', c'))$ which satsify the constraints (given in the previous section) on edges of G_{DS} will not be calculated explicitly; they are represented in Figure 5, which shows G_{DS}. In the diagram, the vertices

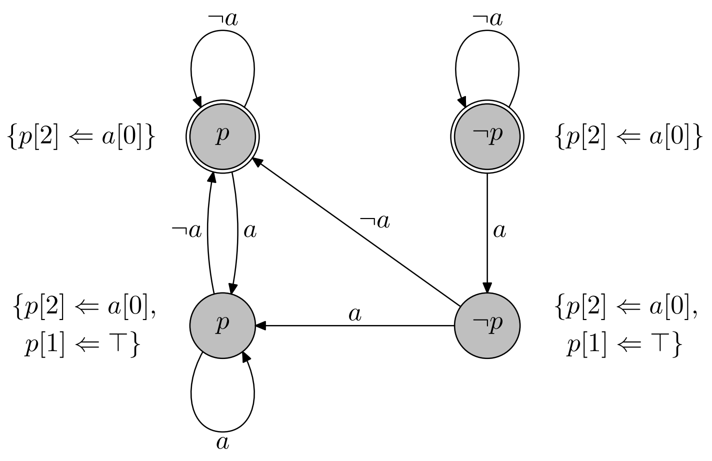

Figure 5. The graph G_{DS}, with states and associated commitments

(s, c) have been depicted so that the components s are shown inside circles, with the commitments c shown adjacently and outside. Vertices of the form (s, com_{DS}) have been circled twice, and the circles which are shaded are the ones which can be reached from a path starting from the double-circled vertices. (In the current case these are all the vertices in the graph, but in general that need not be so.) Thus G^*_{DS} is the subgraph of the graph represented in Figure 5 which consists of shaded vertices only, and whose edges are all those which are not incident to an unshaded vertex. In our example we happen to have $G_{DS} = G^*_{DS}$. After replacing the commitments by natural numbers, we arrive at the run system whose vertices are

$$(\{p\}, 0), (\{p\}, 1), (\{\neg p\}, 0), (\{\neg p\}, 1),$$

and whose edges are

$$((\{p\}, 0), \{a\}, (\{p\}, 1)), \qquad ((\{\neg p\}, 0), \{a\}, (\{\neg p\}, 1)),$$
$$((\{p\}, 0), \{\neg a\}, (\{p\}, 0)), \qquad ((\{\neg p\}, 0), \{\neg a\}, (\{\neg p\}, 0)),$$

$$((\{p\},1),\{a\},(\{p\},1)), \qquad ((\{\neg p\},1),\{a\},(\{p\},1)),$$
$$((\{p\},1),\{\neg a\},(\{p\},0)), \qquad ((\{\neg p\},1),\{\neg a\},(\{p\},0)),$$

This is Run_{DS}, and diagramatically it would be like Figure 5, but with the shading and commitments absent. It is clear that paths through this run system—beginning at a double-circled vertex and of length t—correspond to models of the causal theory Γ_t^{DS}.

4.5. Reduction

There is an obvious structural difference between Figures 3 and 5. We gave Figure 3 as an example of what we were aiming for in producing a run system for the simple domain DS, but if that is correct than we have missed our target. The graph which we actually generated is larger (more vertices, more edges) than other graphs which are also run systems for the same domain. It is clear that in general there will be many run systems representing each $\mathcal{C}+_{timed}$ action description (indeed, it is easy to see there will be infinitely many).

The question naturally arises, whether there is a way of reducing run systems to find a graph which is least or minimal according to some appropriate measure, but which still represents the generating domain of $\mathcal{C}+_{timed}$. It seems that cardinality of the sets of vertices and edges of a run system would be the appropriate measure. Thus given two run systems G_D and G'_D which both represent the domain described by the action description D, we define

$$G_D <_r G'_D \quad \text{if} \quad \begin{cases} |V(G_D)| < |V(G'_D)|, & \text{or} \\ |V(G_D)| = |V(G'_D)| \wedge |E(G_D)| < |E(G'_D)|. \end{cases}$$

We proceed intuitively. Reduction might be thought to involve identifying vertices which were previously different, 'collapsing' a set of vertices onto a single representative of that set. Clearly, for any two vertices (s,n) and (s',n') in a set which is reducible, we must have $s = s'$. Assume G_D is a run system for the $\mathcal{C}+_{timed}$ domain D, and that there is $S \subseteq V(G_D)$, such that

- for all (s,n), (s',n') in S, we have $s = s'$;
- if $(s_0,n_0) \in S$ and $((s_0,n_0),e,(s_1,n_1)) \in E(G_D)$, then for all $(s'_0,n'_0) \in S$, we have $((s'_0,n'_0),e,(s_1,n_1)) \in E(G_D)$.

Then we might collapse the members of S onto a single representative, as follows. If $(s,0) \in S$—an initial state—then we choose that vertex as the representative, otherwise we choose any member of S. Let the chosen member of S be (s^*, n^*). Then, we remove each edge $((s_0, n_0), e, (s, n))$, for $(s, n) \in S$, from G_D, and replace it by the edge $((s_0, n_0), e, (s^*, n^*))$. We also remove each edge $((s, n), e, (s_0, n_0))$, where $(s, n) \in S$, and replace it by $((s^*, n^*), e, (s_0, n_0))$. We then remove the vertices $S - \{(s^*, n^*)\}$ from G_D. This procedure is continued until there are no more sets S satisfying the properties given above.

A single reduction step clearly leaves the paths through the run system unchanged.

This kind of reduction will cope with all of the run systems we have studied so far, but we have as yet no proof of completeness: that is, that it always reduces a run-system to an $\leqslant_r$-minimal counterpart.

4.6. Example—Reagan and Gorbachev

As a second illustration, consider the following version of a familiar example from the field of deontic logic [Bel87]. (In deontic logic it is used in the study of 'contrary-to-duty' reasoning; we exploit it for different purposes here.)

If Reagan is told crucial strategic information, then Gorbachev must also be told, unless Gorbachev knows already; and vice versa. Once someone is told, then they know. We model this in $\mathcal{C}+_{timed}$ using a Boolean domain, with $\sigma^f = \{k_r, k_g\}$ and $\sigma^a = \{r, g\}$. k_r represents that Reagan knows, and r is the action of telling Reagan; similarly for g and Gorbachev. The causal laws for this domain, D_{rg}:

inertial k_g,	k_g **if** $\lambda(g)$,
inertial k_r,	k_r **if** $\lambda(r)$,
exogenous r,	g **if** $\neg k_g \wedge \lambda(r)$,
exogenous g,	r **if** $\neg k_r \wedge \lambda(g)$.

Notice that the two bottom-right laws above could not have been expressed directly in $\mathcal{C}+$. Noteworthy is the fact that we only insist that Gorbachev (for example) should be told if an action of telling Reagan is performed; if the system starts in a state where there is disparity of knowledge, no telling need take place. A run system for this domain is shown in Figure 6.

5. Interaction with $(\mathcal{C}+)^{++}$

Say an agent promises to perform an action for another agent at some specified time in the future. If the agent fails to perform the promised action, we may wish to signal this as a breach of contract, regulation, or statement

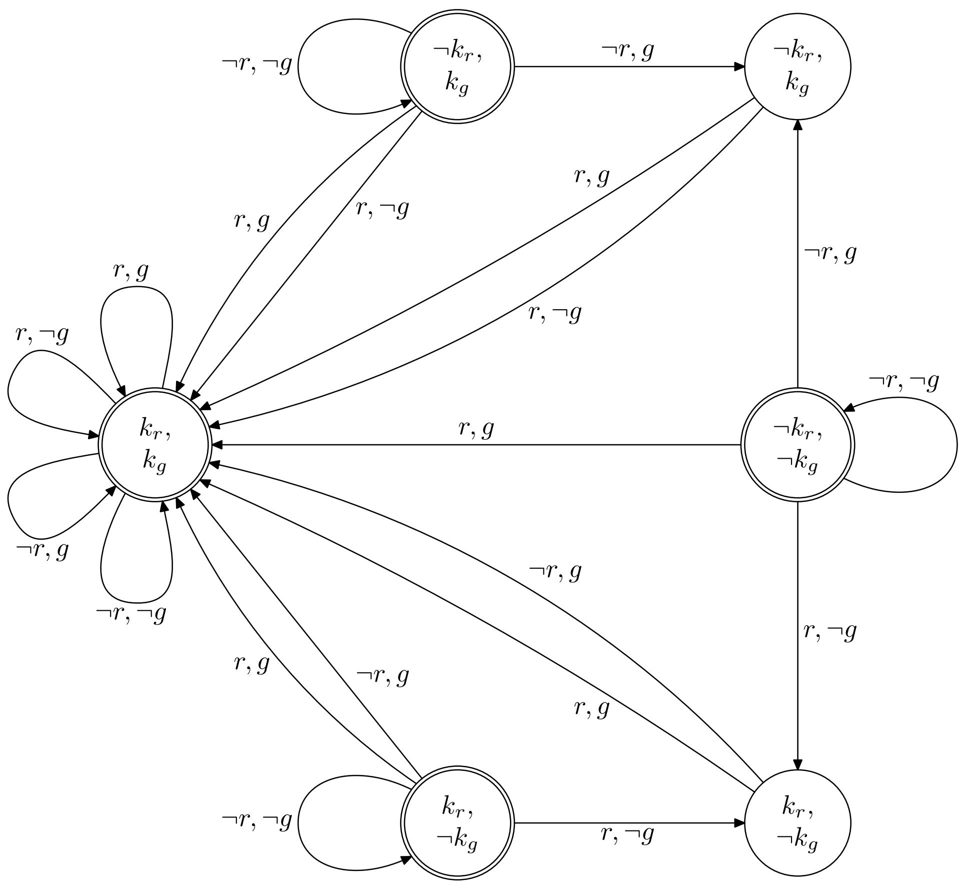

Figure 6. Reagan and Gorbachev

of assurance. This can be achieved using $(\mathcal{C}+)^{++}$ [Ser04], an extended form of the language $\mathcal{C}+$, which allows us to specify which actions and states are not permitted, and thus, implicitly, which actions and states *are* permitted.

However, the motivating example for $\mathcal{C}+_{timed}$ surfaces here in much the same form. For if the promise above is for an action to be performed 20 time-steps in the future, a proliferation of agents and promises would make our logical model complex and computationally inefficient. This argues in favour of a marriage of $(\mathcal{C}+)^{++}$ and $\mathcal{C}+_{timed}$.

5.1. The language $(\mathcal{C}+)^{++}$

The language $(\mathcal{C}+)^{++}$ provides two main extensions to $\mathcal{C}+$. The first is a means of expressing 'counts as' relations [JS96] between actions, also referred to as 'conventional generation' of actions. This will not be the focus of attention in this paper. The second is a way of specifying the permitted (acceptable, legal) states of a transition system and its permitted (acceptable, legal) actions and runs.

In the second extension, *state permission laws* and *action permission laws* are introduced. The former have the form

$$\textbf{not-permitted } F,$$

where F is a fluent formula; an action permission law has the form

$$\textbf{not-permitted } F \textbf{ if } G \textbf{ after } H,$$

where F is an action formula, G is a fluent formula, and H is a formula. The transition system defined by an action description D of $(\mathcal{C}+)^{++}$ is the transition system defined as for $\mathcal{C}+$, by ignoring the permission laws in D. Colours are then added to this system: a state s is coloured red when $s \models F$ for some state permission law in D; all other states are coloured green. A transition (s, e, s') is coloured red if $s \cup e \models H$, $s' \models G$, and $e \models F$. All other transitions are coloured green by default, except where this would contravene the 'green-green-green' constraint:

given a transition (s, e, s'), if s and e are green, then so is s'.

In these exceptions to the default, the transitions are coloured red.

Notice that the transition system here is defined using the pure $\mathcal{C}+$ part of an action description D, i.e. without reference to permission laws. The permission laws don't alter the 'structure' of the transition system, they just add colour.

5.2. The language $(\mathcal{C}+)^{++}_{timed}$

We supplement the language of $\mathcal{C}+_{timed}$ with *permission laws*, which have the form

$$\textbf{not-permitted } F \textbf{ if } G.$$

Here, F is a formula, and G is a formula which can contain nested λ or $\top$, given as in (3). Further restrictions match those from Section 3: a permission law is simply a law of $\mathcal{C}+_{timed}$ with the prefix **not-permitted**. Where D is an action description of $(\mathcal{C}+)^{++}_{timed}$, we say that the $\mathcal{C}+_{timed}$*-component* of D is that subset of its laws which do not contain the keyword **not-permitted**. Laws of $(\mathcal{C}+)^{++}_{timed}$ can be given a canonical form just as for $\mathcal{C}+_{timed}$.

Consider the example Boolean action description DP, whose signature is given by $\sigma^f = \{p\}$ and $\sigma^a = \{a\}$, and whose causal laws are

$$\begin{array}{l}\textbf{inertial } p,\\ \textbf{exogenous } a,\\ p \textbf{ if } \lambda(a),\\ \textbf{not-permitted } p \textbf{ if } \lambda^2(a).\end{array}$$

Translation to causal theories proceeds just as one would expect, given the translation for $(\mathcal{C}+)^{++}$ described in [Ser04]. This is a matter of introducing constants *status* and *trans* into the causal theories Γ^D_t which correspond to $(\mathcal{C}+)^{++}_{timed}$ action descriptions D; the domain of each constant is $\{green, red\}$. The purpose of *status* is to describe the deontic status of states, and *trans* that of transitions. Appropriate causal rules then govern the colouring of states and transitions in accordance with the default behaviour and 'green-green-green' constraint.

What of the run systems? The means of generating these using causal theories and commitments, as described in Section 4, proceeds as before. This will give us the run system shown in Figure 7, for DP. This surely

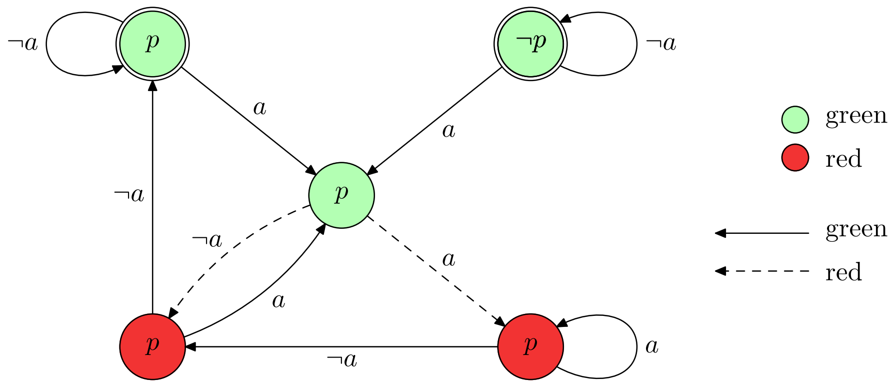

Figure 7. Run system for $(\mathcal{C}+)^{++}_{timed}$ domain DP

corresponds to what had been intended in the action description.

Notice however that the graph does not simply represent a colouring of the run system defined by the $\mathcal{C}+_{timed}$-component of the action description DP. Thus we cannot say of $(\mathcal{C}+)^{++}_{timed}$ that adding permission laws simply adds colour to the states and transitions—there may be a radical restructuring of the original run system, and in a certain sense this makes our models *deontically non-local*: the colour of states and transitions need not depend only on the interpretations (ignoring the values of the constants *status* and *trans*) which constitute those states and transitions. We can spell this out. Where x is a set of atoms, let $pure(x)$ be

$$x - \{c{=}v \mid (c = status \vee c = trans) \wedge v \in dom(c)\}.$$

Then, in our run system we may have vertices (s, n) and (s', n'), where $pure(s) = pure(s')$, but (s, n) and (s', n') are coloured differently. Similarly for transitions: it is possible to have two edges $((s_1, n_1), e_1, (s'_1, n'_1))$ and $((s_2, n_2), e_2, (s'_2, n'_2))$ where $pure(s_1) = pure(s_2)$, $pure(e_1) = pure(e_2)$ and $pure(s'_1) = pure(s'_2)$, but where the edges have different colours. We have both kinds of non-locality in the shown run-system for DP.

6. Conclusion

We have presented $\mathcal{C}+_{timed}$, a natural generalization of the action language $\mathcal{C}+$ [GLL⁺04].

Syntactically, $\mathcal{C}+_{timed}$ extends the laws of $\mathcal{C}+$ by adding a λ-operator allowing reference to past states and transitions: it thus removes the restriction to the immediately preceding state and transition. Semantically, it generalizes the transition systems defined by $\mathcal{C}+$ action descriptions. We have called the new kind of transition system a run system. The transition systems defined in $\mathcal{C}+$ are a special case where there is only one vertex for each state, and every vertex is initial. We have also shown how action descriptions of $\mathcal{C}+_{timed}$ are a shorthand for causal theories; using $\mathcal{C}+_{timed}$ we can make use of more of the language of causal theories, whilst retaining the key property: models of the causal theories are in correspondence with paths through the run system defined by the $\mathcal{C}+_{timed}$ action description. Computationally, action descriptions of $\mathcal{C}+_{timed}$ are much more efficient than attempted encodings of the same domains in $\mathcal{C}+$. The task of implementation is easy, through an adaptation of CCalc.

We have defined a means of generating a run system from a $\mathcal{C}+_{timed}$ action description. A further step of reduction is sometimes necessary to give the most compact run systems. Though our reduction steps are sound, we do not yet know whether they are complete, that is, whether they are guaranteed to result in run systems which are minimal according to the order we introduced. Further work will explore this question. We are also interested in seeing whether the original generation of the run systems can be optimised so as to obviate reduction.

We concluded the paper with a brief review of $(\mathcal{C}+)^{++}_{timed}$, which merged the extensions of $\mathcal{C}+_{timed}$ with those of $(\mathcal{C}+)^{++}$ [Ser04], allowing us to represent domains where permission and prohibition play a role.

We are also developing a *run language*, a temporal logic to describe properties of paths through transition systems or run systems defined by languages such as $\mathcal{C}+$, $\mathcal{C}+_{timed}$ and $(\mathcal{C}+)^{++}_{timed}$. This will enable us to express properties of runs which even $\mathcal{C}+_{timed}$ cannot capture.

References

[Bel87] Belzer, M., 'Legal reasoning in 3-D', in *Proceedings of the First International Conference on Artificial Intelligence and Law*, ACM Press, 1987, pp. 155–163.

[GLL+04] Giunchiglia, E., J. Lee, V. Lifschitz, N. McCain, and H. Turner, 'Nonmonotonic causal theories', *Artificial Intelligence*, 153:49–104, 2004.

[JS96] Jones, A., and M. Sergot, 'A formal characterisation of institutional power', *Journal of the IGPL*, 4(3):429–445, 1996.

[Ser04] Sergot, M., 'The language $(\mathcal{C}/\mathcal{C}+)^{++}$', in J. Pitt, (ed.), *The Open Agent Society*. Wiley, 2004. To appear.

Robert Craven
Department of Computing,
Imperial College London,
180 Queen's Gate,
London SW7 2BZ,
United Kingdom
`rac101@doc.ic.ac.uk`

Marek Sergot
Department of Computing,
Imperial College London,
180 Queen's Gate,
London SW7 2BZ,
United Kingdom
`mjs@doc.ic.ac.uk`