ABSTRACT

Many objectives can be achieved (or may be achieved more effectively) only by a group of agents executing a team plan. If a team plan fails, it is often of interest to determine what caused the failure, the degree of responsibility of each agent for the failure, and the degree of blame attached to each agent. We show how team plans can be represented in terms of structural equations, and then apply the definitions of causality introduced by Halpern [11] and degree of responsibility and blame introduced by Chockler and Halpern [3] to determine the agent(s) who caused the failure and what their degree of responsibility/blame is. We also prove new results on the complexity of computing causality and degree of responsibility and blame, showing that they can be determined in polynomial time for many team plans of interest.

Keywords

Causality; responsibility; blame; team plans

1. INTRODUCTION

Many objectives can be achieved (or may be achieved more effectively) only by a coalition or team of agents. In general, for the actions of the agents in the team to be successful in achieving the overall goal, their activities must be coordinated by a team plan that specifies which task(s) should be performed by each agent and when they should be performed. As with single-agent plans, team plans may fail to achieve their overall objective: for example, agents may fail to perform a task they have been assigned. When a failure occurs, the inter-dependencies between tasks in the team plan can make it difficult to determine which agent(s) are responsible for the failure: did the agent simply not perform the task it was assigned, or was it impossible to perform the task due to earlier failures by other agents? For example, suppose that a major highway upgrade does not finish by the deadline, causing significant traffic problems over a holiday weekend. Many agents may be involved in the upgrade, each executing steps in a large, complex team plan. Which agents are the causes of the work not being completed on time? To what extent are they responsible or to blame?

Determining which agents are responsible for the failure of a team plan is a key step in recovering from the failure, determining which commitments may have been broken [22] (and hence which sanctions should be applied), and whether agents should be trusted in the future [8]. Identifying those agents most responsible/blemeworthy for a plan failure is useful for (re)assigning tasks when recovering from the failure (e.g., we may prefer to exclude agents with a high degree of blame); if resources are limited, we may wish to focus attention on the agents most responsible for the failure (e.g., to discover the reasons for their failure/try to change their behaviour). However, there has been relatively little work in this area. Work in plan diagnosis has focussed on determining the causes of failures in team plans (e.g., [19, 25]); it typically has not considered the degree of responsibility of agents for the failure (an exception is the notion of primary and secondary failures in, e.g., [4, 18]). Another strand of work focusses on the problem of how to allocate responsibility and blame for non-fulfilment of group obligations (e.g., [1, 5, 9, 10, 15, 16]). However, the definitions of causality and responsibility used in these work do not always give answers in line with our intuitions (see, e.g., [12] for examples of what can go wrong).

In this paper, we present an approach to determining the degree of responsibility and blame of agents for a failure of a team plan based on the definition of causality introduced by Halpern [11] (which in turn is based on earlier definitions due to Halpern and Pearl [13, 14]). One advantage of using the Halpern and Pearl definition of causality is that, as shown by Chockler and Halpern [3], it can be extended in a natural way to assign a degree of responsibility to each agent for the outcome. Furthermore, when there is uncertainty about details of what happened, we can incorporate this uncertainty to talk about the degree of blame of each agent, which is just the expected degree of responsibility.

We show that each team plan gives rise to a causal model in a natural way, so the definitions of responsibility and blame can be applied without change. In addition, it turns out that the causal models that arise from team plans have a special property: the equations that characterise each variable are monotone, that is, they can be written as propositional formulas that do not involve negation. For such monotone models, causality for a monotone formula can be determined in polynomial time, while determining the degree of responsibility and blame is NP-complete. This contrasts with the $D^{\mu}$-completeness of determining causality in general [11] and the $\Sigma^{\mu}_{2}$-completeness of determining responsibility (a result proved here). For postcondition minimal plans (where preconditions of each step are established by a unique combination of previous steps), the causal models that arise have a further property: they are conjunctive; that is, the equations can be written as monotone conjunction (so that they have neither negations nor disjunctions). In this case,
both causality and degree of responsibility can be determined in polynomial time. These complexity results may be of independent interest. For example, conjunctive and monotone formulas are of great interest in databases; indeed, it has already been shown that for the causal models that arise with databases (which are even simpler than the conjunctive models that we consider here), computing causality for conjunctive formulas can be done in polynomial time [17]. (However the notion of causality considered by Meliou at al. is closer to the original Halpern-Pearl definition [13], and thus not quite the same as that considered here.) This reduction in complexity can be useful in many settings, for example, where causality, responsibility and blame must be determined at run-time.

The remainder of the paper is structured as follows. In Section 2 we recall the definitions of causality, responsibility and blame from [3, 11]. In Section 3 we define our notion of team plan, and in Section 4 we show how team plans can be translated into causal models. As noted above, the resulting causal models are monotone; in Section 5 we prove general results on the complexity of checking causality, degree of responsibility, and degree of blame for monotone and conjunctive causal models. We conclude in Section 6.

2. CAUSALITY, RESPONSIBILITY, AND BLAME

In this section we briefly review Halpern’s definitions of causality [11] and Chockler and Halpern’s definition of responsibility and blame [3]; see [3, 11] for further details and intuition. Much of the description below is taken from [11].

The Halpern and Pearl approach (hereafter HP) assumes that the world is described in terms of variables and their values. Some variables may have a causal influence on others. This influence is modelled by a set of modifiable structural equations. It is conceptually useful to split the variables into two sets: the exogenous variables, whose values are determined by factors outside the model, and the endogenous variables, whose values are ultimately determined by the exogenous variables. The structural equations describe how the outcome is determined.

Formally, a causal model is a pair (S, F), where S is a signature that explicitly lists the endogenous and exogenous variables and characterizes their possible values, and F is a function that associates a structural equation with each variable. A signature S is a tuple (U, V, R), where U is a set of exogenous variables, V is a set of endogenous variables, and R associates with every variable Y ∈ U ∪ V a nonempty set of values for Y (i.e., the set of values over which Y ranges). F associates with each endogenous variable X ∈ V a function denoted F_X such that X : ((×_{u∈U}(R(U))) × (×_{Y∈V−{X}}(R(Y))) → R(X). Thus, F_X defines a structural equation that determines the value of X given the values of other variables. Setting the value of some variable X to x in a causal model M = (S, F) results in a new causal model, denoted M_{X=x}, which is identical to M, except that the equation for X in F is replaced by X = x.

Given a signature S = (U, V, R), a primitive event is a formula of the form X = x, for X ∈ V and x ∈ R(X). A causal formula (over S) is one of the form [Y_1 ← y_1, ..., Y_k ← y_k]φ, where

- φ is a Boolean combination of primitive events,
- Y_1, ..., Y_k are distinct variables in V, and
- y_i ∈ R(Y_i).

Such a formula is abbreviated as [Y ← y]φ. The special case where k = 0 is abbreviated as φ. Intuitively, [Y_1 ← y_1, ..., Y_k ← y_k]φ says that φ would hold if Y_i were set to y_i, for i = 1, ..., k.

Following [11, 14], we restrict attention here to what are called acyclic models. This is the special case where there is some total ordering ∼ of the endogenous variables (the ones in V) such that if X ∼ Y, then X is independent of Y, that is, F_X(z, y, v) = F_X(z, y', v) for all y, y' ∈ R(Y). If X ∼ Y, then the value of X may affect the value of Y, but the value of Y cannot affect the value of X. If M is an acyclic causal model, then given a context, that is, a setting v for the exogenous variables in U, there is a unique solution for all the equations: we simply solve for the variables in the order given by ∼.

A causal formula ψ is true or false in a causal model, given a context. We write (M, v) |= ψ if the causal formula ψ is true in causal model M given context v. The |= relation is defined inductively. (M, v) |= X = x if the variable X has value x in the unique (since we are dealing with acyclic models) solution to the equations in M in context v (i.e., the unique vector of values for the endogenous variables that simultaneously satisfies all equations in M with the variables in U set to v). The truth of conjunctions and negations is defined in the standard way. Finally, (M, v) |= [Y ← y]φ if (M_{Y=y}, v) |= φ. Thus, [Y ← y]φ is true in (M, v) if φ is true in the model that results after setting the variables in Y to y.

With this background, we can now give the definition of causality. Causality, like the notion of truth discussed above, is relative to a model and a context. Only conjunctions of primitive events, abbreviated as X = x, can be causes. What can be caused are arbitrary Boolean combinations of primitive events. Roughly speaking, X = x is a cause of φ if, had X = x not been the case, φ would not have happened. To deal with many well-known examples, the actual definition is somewhat more complicated.

Definition 2.1. X = x is an actual cause of φ in (M, v) if the following three conditions hold:

AC1. (M, v) |= (X = x) and (M, v) |= φ.

AC2. There is a set W of variables in V and settings w of the variables in W and w of the variables in W such that (M, v) |= W = w and

(M, v) i.e. [X ← x', W ← w], ∀φ.

AC3. X is minimal; no subset of X satisfies conditions AC1 and AC2.

AC1 just says that for X = x to be a cause of φ, both X = x and φ have to be true. AC3 is a minimalization condition, which ensures that only the conjuncts of X = x that are essential are parts of a cause. AC2" (the "m" is for modified; the notation is taken from [11]) captures the counterfactual. It says that if we change the value of X from x to x', while possibly holding the values of the variables in some (possibly empty) set W fixed at their values in the current context, then φ becomes false. We say that (W, x') is a witness to X = x being a cause of φ in (M, v). If X = x is a cause of φ in (M, v) and X = x is a conjunct of X = x, then X = x is part of a cause of φ in (M, v).
models where the agents’ intentions to perform their tasks are determined by $U$. The equation for $Fin$ is $Fin = T_1 \land T_2$. If $t_1$ is not performed while $t_2$ is, $T_1 = 0$ is the cause of $Fin = 0$. If $T_1 = 0$ and $T_2 = 0$, then both together are the cause of $Fin = 0$. Indeed, let $u$ be the context where the two tasks are not performed. AC1 is satisfied since $(M, u) \models T_1 = 0 \land T_2 = 0 \land Fin = 0$. AC2 is satisfied since $(M, u) \models [T_1 \leftarrow 1, T_2 \leftarrow 1](Fin = 1)$. Moreover, flipping the value of just $T_1$ or $T_2$ alone does not change the outcome, so AC3 is satisfied. If the completion of the plan depended on $n$ tasks instead of two, and none of them were performed, the cause would consist of the $n$ non-performed tasks. We would like to say that each of the non-performed tasks was “less” of a cause of $Fin = 0$ than in the case when plan failure is due to a single task not being performed. The notion of degree of responsibility, introduced by Chockler and Halpern [3], is intended to capture this intuition. Roughly speaking, the degree of responsibility $X = x$ for $\phi$ measures the minimal number of changes and number of variables that have to be held fixed in order to make $\phi$ counterfactually depend on $X = x$. We use the formal definition in [12], which is appropriate for the modified definition of causality used here.

**Definition 2.2.** The degree of responsibility of $X = x$ for $\phi$ in $(M, \bar{u})$, denoted $det((M, \bar{u}), (X = x), \phi)$, is $0$ if $X = x$ is not a part of a cause of $\phi$ in $(M, \bar{u})$; it is $1/k$ if there exists a cause $\bar{X} = \bar{x}$ of $\phi$ and a witness $(\bar{W}, \bar{x})$ to $\bar{X} = \bar{x}$ being a cause of $\phi$ in $(M, \bar{u})$ such that (a) $X = x$ is a conjunct of $\bar{X} = \bar{x}$, (b) $|\bar{W}| + |\bar{X}| = k$, and (c) $k$ is minimal, in that there is no cause $\bar{X}_1 = \bar{x}_1$ for $\phi$ in $(M, \bar{u})$ and witness $(\bar{W}', \bar{x}_1)$ to $\bar{X}_1 = \bar{x}_1$ being a cause of $\phi$ in $(M, \bar{u})$ that includes $X = x$ as a conjunct with $|\bar{W}'| + |\bar{X}_1| < k$.

This definition of responsibility assumes that everything relevant about the facts of the world and how the world works is known. In general, there may be uncertainty both about the context and about the causal model. The notion of blame takes this into account. We model an agent’s uncertainty by a pair $(K, \Pr)$, where $K$ is a set of causal settings, that is, pairs of the form $(M, \bar{u})$, and $\Pr$ is a probability distribution over $K$. We call such a pair an epistemic state. Note that once we have such a distribution, we can talk about the probability that $\bar{X} = \bar{x}$ is a cause of $\phi$ relative to $(K, \Pr)$: it is just the probability of the set of pairs $(M, \bar{u})$ such that $\bar{X} = \bar{x}$ is a cause of $\phi$ in $(M, \bar{u})$. We also define the degree of blame of $X = x$ for $\phi$ to be the expected degree of responsibility:

**Definition 2.3.** The degree of blame of $X = x$ for $\phi$ relative to the epistemic state $(K, \Pr)$ is

$$
\sum_{(M, \bar{u}) \in K} det((M, \bar{u}), (X = x), \phi) \Pr((M, \bar{u})).
$$

### 3. TEAM PLANS

In this section, we define the notion of team plan. Our definition is essentially the same as that used in much of the work in multiagent planning and work in plan diagnosis [19, 25], except that we explicitly record the assignment of agents to primitive tasks. It thus encompasses partial order causal link plans [24], primitive task networks [6], and the notion of team plan used in [9, 10], where a team plan is constrained to be a sequence of possibly simultaneous individual actions.

As is standard in planning literature (e.g., plans and planning problems relative to a planning domain description; however, for simplicity, we assume that the domain is described using propositional rather than first order logic. A planning domain is a tuple $D = (II, T, pre, post)$, where $II$ is a set of atomic propositions, $T$ is the set of tasks possible in the domain, and $pre$ and $post$ are functions from $T$ to subsets of $II \cup \{\neg p : p \in II\}$. For each $t \in T$, $pre(t)$ specifies the preconditions of $t$ (the set of literals that must hold before $t$ can be executed), and $post(t)$ specifies the postconditions of $t$ (the effects of executing $t$).

A planning problem $\mathcal{G}$ is defined relative to a planning domain, and consists of an initial or starting situation and a goal. The initial situation and goal are specified by the distinguished tasks $Start$ and $Finish$ respectively. $post(Start)$ is the initial state of the environment, and $Finish$ has the goal as its preconditions and no postconditions.

Given a planning problem, a team plan consists of a set of tasks $T \subseteq T \cup \{Start, Finish\}$, an assignment of agents to tasks that specifies which agent is going to perform each task in $t \in T \setminus \{Start, Finish\}$, and a partial order $\prec$ specifying the order in which tasks in $T$ must be performed. If $t \prec t'$, whichever agent is assigned to $t$ must get done before $t'$ started. $\prec$ is “minimally constraining” in the sense that every linearization $\prec'$ of tasks compatible with $\prec$ achieves the goal (in a sense we make precise below). We assume that the agents desire to achieve the goal of the team plan and have agreed to the assignment of tasks; we define causality and responsibility relative to a team plan.

**Definition 3.1.** A team plan $P$ over a planning domain $D$ and problem $\mathcal{G}$ is a tuple $P = (T, Ag, \prec, \alpha)$, where

- $\{Start, Finish\} \subseteq T \subseteq T \cup \{Start, Finish\}$ is a finite set of tasks;
- $Ag$ is a finite set of agents;
- $\prec$ is an acyclic transitive binary relation on $T$ such that $Start \prec t \prec Finish$ for all tasks $t \in T \setminus \{Start, Finish\}$;
- $\alpha$ is a function that assigns to each task in $T \setminus \{Start, Finish\}$ an agent $a \in Ag$ (intuitively, $\alpha(t)$ is the agent assigned to execute task $t$; $Start$ is executed automatically), such that $Finish$ is executable, that is, the goal specified by $\mathcal{G}$ is achieved (in a sense made precise in Definition 3.2).

Given a task $t$ and a precondition $\ell$ of $t$, a task $t'$ is a clobberer of $t$ (or the precondition $\ell$ of $t$) if $\ell \prec t'$ (where $\prec$ denotes $\neg p$ if $\ell = p$ and $p$ if $\ell = \neg p$).

**Definition 3.2.** Given a team plan $P = (T, Ag, \prec, \alpha)$, a task $t' \in T$ establishes literal $\ell$ for a task $t \in T$ if $t' \in prec(t)$, $\ell \in post(t')$, $t' \prec t$, and for every task $t'' \in T$ that clobbers $t$, either $t'' \prec t$ or $t' \prec t''$. A set $S \subseteq T$ of tasks is an establishing set for task $t \in T$ if and only if $S$ is a minimal set that establishes all literals $\ell \in prec(t)$. $P$ achieves the goal specified by $\mathcal{G}$ if each task $t \in T \cup \{Finish\}$ has an establishing set in $T$.

It is easy to check that if $P$ achieves the goal and $\prec'$ is a linear order on tasks that extends $\prec$ (so that $t \prec t'$ implies $t \prec' t'$), all tasks have their preconditions established at the point when they are executed. This justifies the claim that the constraints in $\prec$ capture all the ordering information on tasks that is needed.

We call a team plan postcondition minimal if there is a unique minimal establishing set for each task $t \in T$. Most planning algorithms construct plans that approximate postcondition minimal
plans, since they add only one task for each precondition to be achieved. However, since they typically do not check for redundancy, the resulting plan may contain several tasks that establish the same precondition \( t \) of some task \( t \).

As an illustration, consider the plan \( P_1 = (T_1, \alpha_1, \alpha_2) \), where \( T_1 = \{ \text{Start, Finish, t1, t2} \} \) is laying cables for traffic signals (under the road surface), \( \alpha_1 = \{ a_1, a_2 \} \), \( \prec_1 = \text{Start} < t_1 < t_2 < \text{Finish} \), and \( \alpha_1(t_1) = \alpha_2(t_2) = a_2 \). The goal \( \text{prec} (\text{Finish}) = \{ c, s \} \), where \( c \) stands for ’cables laid’ and \( s \) for ’road surfaced’. \( \text{post} (\text{Start}) = \{ \neg s, \neg c \}; \text{prec} (t_t) = \{ \neg s \} \) (since cables are laid under the surface); \( \text{post}(t_1) = \{ c \}; \text{prec}(t_2) = \emptyset; \) and \( \text{post}(t_2) = \{ s \} \). This plan accomplishes its goal; the preconditions of \( \text{Finish} \) are established by \{ t_1, t_2 \}, while the precondition of \( t_1 \) is established by \( \text{Start} \). Note that \( t_2 \) is a clobberer of \( t_1 \) because it undoes the precondition \( \neg s \) of \( t_1 \). For this reason, \( t_2 \) is required by \( \prec_1 \) to be executed after \( t_1 \). Note that the plan \( P_1 \) is postcondition minimal.

4. TRANSLATING TEAM PLANS TO CAUSAL MODELS

In this section, we apply the definitions of causality, responsibility, and blame given in Section 2 to the analysis of team plans. We start by showing that a team plan \( P = (T, \alpha, \prec_\alpha) \) determines a causal model \( M_P \) in a natural way. The preconditions of a task are translated as endogenous variables, as well as whether the agent intends to perform it. Whatever determines whether the agent intends to perform the task is exogenous. The structural equations say, for example, that if the agent intends to perform a task \( t \) and all its preconditions hold, then the task is performed.

For each task \( t \in T \), we compute the set \( \text{est}(t) \) and the set \( \text{clob}(t) \). The set \( \text{est}(t) \) consists of all the establishing sets for task \( t \). The assumption that the plan accomplishes its goal ensures that, for all tasks \( t \), \( \text{est}(t) \neq \emptyset \).

The set \( \text{clob}(t) \) contains all pairs \( \{s, t'\} \) where \( s \in S \) for some \( S \in \text{est}(t) \), \( s \) establishes some precondition \( t \) of \( t \), and \( t' \) is a clobberer of \( t \).

For each task \( t \in T \), we have variables \( \text{en}(t) \) for ‘\( t \) is enabled’, \( \text{in}(t) \) for ‘agent \( a = \alpha(t) \) intends to do task \( t \)’, and \( \text{pf}(t) \) for ‘\( t \) is performed’. \( \text{en}(t) \) is true if all the tasks in one of the establishing sets \( S \) of \( t \) are performed, and \( \text{pf}(t) \) is true if both \( \text{en}(t) \) and \( \text{in}(t) \) are true, where \( a = \alpha(t) \). Finally, for each pair \( \{s, t'\} \) in \( \text{clob}(t) \), we have a variable \( \text{nc}(s, t', t) \), which stands for ‘\( t' \) is not executed between \( s \) and \( t' \).

Consider again the example plan \( P_1 \) from Section 3. The causal model for \( P_1 \) has the variables \( \text{pf} (\text{Start}), \text{en}(t_1), \text{in}(t_1), \text{pf}(t_1), \text{in}(t_2), \text{pf}(t_2), \text{en}(\text{Finish}), \text{pf}(\text{Finish}) \), and \( \text{nc} (\text{Start}, t_2, t_1) \). (Note that we omit \( \text{en}(\text{Start}) \) and \( \text{en}(t_2) \) because \( \text{Start} \) and \( t_2 \) have no preconditions.) \( \text{nc}(\text{Start}, t_2, t_1) = 1 \) if \( t_2 \) is performed after \( t_1 \) and \( t_2 \) is performed before \( t_1 \). \( \text{en}(t_1) \) is true if \( \text{pf}(\text{Start}) \) is true and \( \text{nc}(\text{Start}, t_2, t_1) = 1 \).

More precisely, a team plan \( P = (T, \alpha, \prec_\alpha) \) determines causal model \( M_P = (V_P, \prec_P, R_P, \text{F}_P) \) as follows:

- \( U_P = \{ U_{\alpha(a)} : a \in \alpha(t) \} \) \cup \{ U_{\text{nc}(s, t'; t)} : s, t', t \in T, \{s, t'\} \in \text{clob}(t) \} \). Intuitively, \( U_{\alpha(a)} \) and \( U_{\text{nc}(s, t'; t)} \) determine the value of \( \text{in}(a) \) and \( \text{nc}(t', s, t) \), respectively.
- \( V_P = \{ \text{en}(t) : t \in T \} \cup \{ \text{pf}(t) : t \in T \} \cup \{ \text{in}(a) : a \in \alpha(t) \} \cup \{ \text{nc}(s, t', t) : s, t', t \in T, \{s, t'\} \in \text{clob}(t) \} \). Note that \( |V_P| \leq |T| + |3|T| \).
- \( \text{R}_P(X) = \{0, 1\} \) for all variables \( X \in U_P \cup V_P \) (i.e., all variables are binary).
- \( \text{F}_P \) is determined by the following equations:
  \[ \text{en}(t) = U_{\alpha(a)} \]
  \[ \text{nc}(s, t', t) = U_{\text{nc}(s, t'; t)} \]
  \[ \text{pf}(t) = \text{en}(t) \land \text{in}(a(t)) \]
  \[ \text{en}(t) = \bigvee_{S \in \text{est}(t)} \left( \bigwedge_{s \in S} \text{pf}(s) \land \bigwedge_{\{s, t'\} \in \text{clob}(t)} \text{nc}(s, t', t) \right) \]

It should be clear that \( M_P \) captures the intent of the team plan \( P \). In particular, it is easy to see that the appropriate agents performing their tasks results in \( P \) accomplishing its goal iff \( (M_P, \vec{u}) \models \text{pf}(\text{Finish}) \), where \( \vec{u} \) is the context where the corresponding agents intend to perform their actions and no clobbering task is performed at the wrong time (i.e., between the establishing of the precondition they clobber and the execution of the task requiring the precondition).

Our causal model abstracts away from pre- and postconditions of tasks, and concentrates on high level ‘establishing’ and ‘clobbering’ links between them. This is standard practice in planning; see, for example, [24]. We also abstract away from the capabilities of agents: our model implicitly assumes that agents are able to perform the tasks assigned to them. All we require is that the preconditions of the task hold and that the agent intends to perform it.

The size of \( M_P \) is polynomial in the size of \( P \) if \( P \) is postcondition minimal or we treat the maximal number of preconditions of any task in the plan as a fixed parameter (if there are at most \( k \) preconditions of a task, then \( \text{est}(t) \) has size at most \( 2^k \)). Note that all equations are monotone: there are no negations. Moreover, the only disjunctions in the equations come from potentially multiple ways of establishing preconditions of some tasks. Thus, for postcondition minimal plans the formulas are conjunctive.

Having translated team plans to causal models, we can apply the definitions of Section 2. There may be several causes of \( \text{pf}(\text{Finish}) \) = 0. As we suggested earlier, we are interested only in causes that involve formulas of the form \( \text{in}(a) = 0 \). We refer to variables of the form \( \text{in}(a) \) as the variables controlled by agent \( a \).

DEFINITION 4.1. Agent \( a \)'s degree of responsibility for the failure of plan \( P \) (i.e., for \( \text{pf}(\text{Finish}) = 0 \) in \( (M_P, \vec{u}) \), where \( M_P \) is the causal model determined by a team plan \( P \) = 0 if none of the variables controlled by agent \( a \) is part of a cause of \( \text{pf}(\text{Finish}) = 0 \) in \( (M_P, \vec{u}) \); otherwise, it is the maximum value \( m/k \) such that there exists a cause \( \vec{X} = \vec{x} \) of \( \text{pf}(\text{Finish}) = 0 \) and a witness \( \vec{W}, \vec{x} \) to \( \vec{X} = \vec{x} \) being a cause of \( \text{pf}(\text{Finish}) = 0 \) in \( (M_P, \vec{u}) \) with \( |\vec{X}| + |\vec{W}| = k \), and agent \( a \) controls \( m \) variables in \( \vec{X} \).

Intuitively, agent \( a \)'s responsibility is greater if it failed to perform a greater proportion of tasks. The intentions of agents in our setting are determined by the context. Although the intention of some agents can be inferred from observations (e.g., if a task \( t \) assigned to agent \( a \) was performed, then \( \text{in}(a) \) must hold), in some cases, we do not know whether an agent intended to perform a task. In general, there will be a set of contexts consistent with the information that we are given. If we are able to define a probability distribution over this set, we can then determine the degree of blame.
In determining this probability, we may want to stipulate that, unless we have explicit evidence to the contrary, the agents always intend to perform their tasks (so that the agents who we assigned to perform tasks that were not enabled are not to blame).

To show that our approach gives an intuitive account of responsibility and blame for plan failures, we briefly outline some simple scenarios involving the example plan $P_3$ and its corresponding causal model $M_{P_3}$. Assume that the context $u$ is such that $en(t_1) = 1$, $nc(Start, t_2, t_1) = 1$, $pf(t_1) = 1$, $pf(t_2) = 0$, and $pf(\text{Finish}) = 0$. We cannot observe the values of $in_{a_1}(t_1)$ and $in_{a_2}(t_2)$, but from $pf(t_1) = 0$ we can conclude that $in_{a_1}(t_1) = 1$, and, from the fact that $pf(t_2) = in_{a_2}(t_2)$ (since $t_2$ is always enabled), we can conclude that $in_{a_2}(t_2) = 0$. Then the cause of $pf(\text{Finish}) = 0$ is $in_{a_2}(t_2) = 0$, and the degree of both responsibility and blame of agent $a_2$ is 1. (Note that $pf(t_2) = 0$ is also a cause of $pf(\text{Finish}) = 0$, but we are interested only in causes involving agents' intentions.) So far, the analysis is the same as in plan diagnosis: we identify a minimal set of 'faulty components' (unwilling agents) such that, had they functioned correctly, the failure would not have happened.

For a more complex example of responsibility and blame, consider a slightly extended plan $P_3$, which is like $P_1$, but has an extra task $t_0 < t_1$ that establishes $t_1$: $en(t_1) = pf(t_0)$. $t_0$ is enabled and assigned to $a_2$. Suppose the context is $en(t_0) = 1$, $nc(t_0, t_2, t_1) = 1$, $pf(t_0) = 0$, $pf(t_1) = 0$, $pf(t_2) = 0$, and $pf(\text{Finish}) = 0$. As before, $in_{a_2}(t_0) = 0$ and $in_{a_2}(t_2) = 0$ are parts of the cause of $pf(\text{Finish}) = 0$. However, we cannot observe $in_{a_1}(t_1)$; since $t_1$ was not enabled and not performed, we cannot say whether agent $a_1$ was willing to perform it. In the context $u_1$ where $a_1$ was willing, the cause of $pf(\text{Finish}) = 0$ is just $\{in_{a_2}(t_0) = 0, in_{a_2}(t_1) = 0\}$ and the degree of responsibility of $a_1$ is 0. In the context $u_2$ where $a_1$ was not willing, the cause is $\{in_{a_2}(t_0) = 0, in_{a_1}(t_1) = 0, in_{a_2}(t_2) = 0\}$ and $a_1$'s degree of responsibility is $1/3$. If we assign probability 1 to $u_1$, then the blame attached to $a_1$ is 0.

## 5. THE COMPLEXITY OF CAUSALITY FOR MONOTONE MODELS

A causal model is **monotone** if all the variables are binary and all the equations are monotone (i.e., are negation-free propositional formulas). A monotone model is **conjunctive** if all the equations are conjunctive (i.e., they involve only conjunctions; no negations or disjunctions). As we have seen, the causal models that are determined by team plans are monotone; if the team plans are postcondition minimal, then the causal models are also conjunctive.

In this section we prove general results on the complexity of checking causality, degree of responsibility, and degree of blame for monotone and conjunctive models. We first consider the situation for arbitrary formulas. Recall that the complexity class $D^P$ consists of languages $L$ such that $L = L_1 \cap L_2$, where $L_1$ is in NP and $L_2$ is in co-NP [21].

### Theorem 5.1

(a) [11] Determining if $\vec{X} = \vec{\bar{x}}$ is a cause of $\varphi$ in $(M, \vec{u})$ is $D^P$-complete.

(b) Determining if $X = x$ is part of a cause of $\varphi$ in $(M, \vec{u})$ is $\Sigma_2^P$-complete.

(c) Determining if $X = x$ has degree of responsibility at least $1/k$ is $\Sigma_2^P$-complete.

**Proof:** Part (a) was proved by Halpern [11].

For part (b), first note that the problem is clearly in $\Sigma_2^P$: we simply guess $\vec{X}$, $\vec{\bar{x}}$, $\vec{\bar{\bar{x}}}$, and $\vec{W}$, where $X = x$ is a conjunct of $\vec{X} = \vec{\bar{x}}$, compute $\vec{w}$ such that $(M, \vec{u}) \models \vec{W} = \vec{w}$ (in general, checking whether $(M, \vec{u}) \models \psi$ is easily seen to be in polynomial time in acyclic models, assuming that the ordering $\prec$ on variables is given, or can be easily computed from presentation of the equations), check that $(M, \vec{u}) \models [\vec{X} \leftarrow \vec{\bar{x}}, \vec{W} \leftarrow \vec{w}] \models \varphi$, and check that there is no $\vec{Y} \subset \vec{X}$, setting $\vec{y}$ of the variables in $\vec{Y}$, and set $\vec{W}'$ such that $(M, \vec{u}) \models [\vec{Y} \leftarrow \vec{y}, \vec{W} \leftarrow \vec{w}] \models \neg \varphi$, where $\vec{w}'$ is such that $(M, \vec{u}) \models \vec{W}' = \vec{w}'$.

For $\Sigma_2^P$-hardness, we adapt arguments used by Aleksandrowicz et al. [2] to show that checking whether a formula satisfies AC1 and AC2 is $\Sigma_2^P$-complete.

Recall that to show that a language $L$ is $\Sigma_2^P$-hard, it suffices to show that we can reduce determining if a closed quantified Boolean formula (QBF) of the form $\exists \vec{x} \forall \vec{y} \varphi'$ is true (the fact that it is closed means that all the variables in $\varphi'$ are contained in $\vec{x} \cup \vec{y}$) to checking if a string $s \in L$. [23]. Given a closed QBF $\varphi = \exists \vec{x} \forall \vec{y} \varphi'$, we construct a causal formula $\psi$, a causal model $M$, and context $\vec{u}$ such that $\varphi$ is true iff $A = 0$ is part of a cause of $\psi$ in $(M, \vec{u})$.

We proceed as follows: we take $M$ to be a model with endogenous variables $\mathcal{V} = A \cup X^0 \cup X^2 \cup Y$, where for each variable $x \in \mathcal{X}$, there are corresponding variables $X^0_x \in \mathcal{X}^0$ and $X^2_x \in \mathcal{X}^2$, and for each variable $y \in \mathcal{Y}$ there is a corresponding variable $Y_y \in \mathcal{Y}$, and a single exogenous variable $U$. All the variables are binary. The equations are trivial: the value of $U$ determines the values of all variables in $\mathcal{V}$. Let $u$ be the context where all the variables in $\mathcal{V}$ are set to 0. Let $\varphi'$ be the causal formula that results from replacing all occurrences of $x$ and $y$ in $\varphi'$ by $X^1_x$ and $Y_y$, respectively. Let $\psi$ be the formula $\psi_1 \lor (\psi_2 \land \psi_3)$, where

$$\psi_1 = (\forall s \in \mathcal{V}(X^2_s = X^1_s))^2;$$

$$\psi_2 = A = 0 \lor \neg(Y = \vec{1});$$

$$\psi_3 = (A = 1) \lor \varphi'.$$

We now show that $A = 0$ is part of a cause of $\psi$ in $(M, u)$ iff $\varphi$ is true. First suppose that $\varphi$ is true. Then there is an assignment $\tau$ to the variables in $\vec{x}$ such that $\forall \vec{y} \varphi'$ is true given $\tau$. Let $\vec{x}'$ be the subset of variables in $\vec{x}$ that are set to true in $\tau$, let $\vec{X}'$ be the corresponding subset of $\vec{X}^1$ and let $\vec{X}''$ be the complementary subset of $\vec{X}^0$ (so that if $x \in \vec{x}'$ is false according to $\tau$, then the corresponding variable $X^0_x$ is in $\vec{X}''$). Note that for each variable $x \in \vec{x}$, exactly one of $X^0_x$ and $X^2_x$ is in $\vec{X}^1 \cup \vec{X}''$. We claim that $A = 0 \land \vec{X}' = \vec{0} \land \vec{X}'' = \vec{0} \land Y = \vec{1}$ is a cause of $\psi$ in $(M, u)$. Clearly $(M, u) \models A = 0 \land \psi$ (since $(M, u) \models \psi_1$). It is immediate from the definitions of $\psi_2$ and $\psi_3$ that

$$(M, u) \models [\vec{A} \leftarrow 1, \vec{X}' \leftarrow \vec{I}, \vec{X}'' \leftarrow \vec{I}, \vec{Y} \leftarrow \vec{1}](-\psi_1 \land \neg \psi_2),$$

so

$$(M, u) \models [\vec{A} \leftarrow 1, \vec{X}' \leftarrow \vec{I}, \vec{X}'' \leftarrow \vec{I}, \vec{Y} \leftarrow \vec{1}] \psi.$$
Finally, we must show that if $A = 0$ is part of a cause of $\psi$ in $(M, u)$ then $\exists \exists \forall \exists \forall \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exists \exist...
The model $M$ is clearly a trivial monotone model. This completes the proof of part (d).

The argument that determining if $X = 0$ is a part of a cause of $\neg \varphi$ is NP-complete is almost identical. In particular, essentially the same argument as that above shows that $\neg \varphi$ is a satisfiable propositional formula iff $X = 0$ is part of a cause of $\neg \varphi^+$ in $(M, u')$, where $M$ is as above and $u'$ is the context where all variables in $\mathbb{V}$ get value 0.

Part (e) follows easily from part (d). To show that checking if the degree of responsibility of $X = 1$ for $\varphi$ is at least $1/k$ is in NP, given $k$, we guess a cause $\bar{X} = \bar{1}$ that includes $X = 1$ as a conjunct and has $k$ or fewer conjuncts. As observed above, the fact that $\bar{X} = \bar{1}$ is a cause of $\varphi$ in $(M, \bar{u})$ can be confirmed in polynomial time.

For the lower bound, using the notation of part (d), if the propositional formula $\varphi$ mentions $n$ primitive propositions, say $x_1, \ldots, x_n$, then we claim that $X = 1$ has degree of responsibility at least $1/(n + 1)$ for $\varphi^+$ in $(M, u)$ iff $\neg \varphi$ is satisfiable. As observed above, if $\neg \varphi$ is not satisfiable, then $X = 1$ is not a cause of $\neg \varphi^+$, and hence has degree of responsibility 0. On the other hand, if $\neg \varphi$ is satisfiable, then as shown above, $X = 1$ is part of a cause $\bar{Z}^+ = 1$ for $\varphi$ in $(M, u)$. Since $|\bar{Z}^+| = n + 1$, it follows that the degree of responsibility of $X = 1$ for $\varphi$ is at least $1/(n + 1)$. (It is not hard to show that it is in fact exactly $1/(n + 1).$)

The argument for showing that checking if the degree of responsibility of $X = 0$ for $\neg \varphi$ is at least $1/k$ is NP-complete is essentially identical; we leave details to the reader.

Again, it follows that the problem of computing the degree of responsibility of $X = x$ for $\varphi$ in $(M, \bar{u})$ is in $FNP^{\log n}$ (a little more effort in the spirit of [3, Theorem 4.3] shows that it is $FNP^{\log n}$-complete), while the problem of computing the degree of blame of $X = x$ for $\varphi$ relative to an epistemic state $(K, Pr)$ is in $FNP^{\log n}$.

We can do even better in conjunctive models.

**Theorem 5.3.** If $M$ is a conjunctive causal model, $\varphi$ is a conjunctive formula, and $(K, Pr)$ is an epistemic state where all the causal models in $K$ are conjunctive, then the degree of responsibility of $\bar{X} = \bar{1}$ for $\varphi$ (resp., $\bar{X} = \bar{0}$ for $\neg \varphi$) in $(M, \bar{u})$ can be computed in polynomial time, as can the degree of blame of $\bar{X} = \bar{1}$ for $\varphi$ (resp., $\bar{X} = \bar{0}$ for $\neg \varphi$) relative to $(K, Pr)$.

**Proof:** It is easy to check that $\bar{X} = \bar{1}$ is a cause of the conjunctive formula $\varphi$ in $(M, \bar{u})$, where $M$ is a conjunctive causal model, iff $\bar{X}$ is a singleton and $(M, \bar{u}) \models [X = 0] \neg \varphi$. (This means $X = 1$ is a “but-for” cause, in legal language.) Thus, $X = 1$ has degree of responsibility 1 for $\varphi$. It is clearly easy to determine if $X = 1$ is a but-for cause of $\varphi$ and find all the causes of $\varphi$ in polynomial time in this case. It follows that the degree of responsibility and degree of blame of $\bar{X} = \bar{1}$ can also be computed in polynomial time.

In the case of degree of responsibility of $\bar{X} = \bar{0}$ for $\neg \varphi$, observe that for a conjunctive formula $\varphi$, there is exactly one cause of $\neg \varphi$ in $(M, \bar{u})$: the one containing all conjuncts of the form $Y = 0$. It is easy to check whether $X = 0$ is part of that single cause, and if it is, then its degree of responsibility is $1/k$, where $k$ is the number of variables which have value 0. Similarly, it is easy to compute degree of blame in polynomial time.

Since the causal models that are determined by team plans are monotone, the upper bounds of Theorem 5.2 apply immediately to team plans (provided that we fix the maximal number of literals.
...whether $X$ in a precondition); similarly, Theorem 5.3 applies to team plans. It is possible that the causal models that arise from team plans have additional structure that makes computing whether $X = 1$ is part of a cause of $\varphi$ easier than it is for arbitrary monotone causal models, and similarly for responsibility. As the following result shows, this is not the case.

**Theorem 5.4.** Determining whether $im_a(t) = 0$ is part of a cause of $\neg pf(Finish)$ in $(M, \alpha)$, where $M$ is the causal model determined by a team plan $P$, is NP-complete, as is determining whether the degree of responsibility of agent $a$ for $\neg pf(Finish)$ is at least $m/k$.

**Proof:** As we observed, the upper bound for determining whether $im_a(t) = 0$ is part of a cause follows from part (d) of Theorem 5.2.

For the lower bound, recall that it is already NP-hard to compute whether $X = 0$ is part of a cause of $\neg \psi$ in a trivial monotone causal model, where $\psi$ has the form $\neg (\varphi \vee X = 1)$, $\varphi$ is a formula in DNF whose variables are contained in $\{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$, and $\psi$ is the formula that results from replacing each disjunct $X_i = 1 \lor Y_j = 1$ in $\varphi$ by $W_i = 1$, and let $\psi'$ be the formula that results from replacing each occurrence of $\psi$ in $\varphi$ by $\psi'$. Clearly, $\varphi'$ is monotone.

We construct a team plan $P = (T, A, \prec, \alpha)$ with $T = \{Start, \text{Finish}, t_1, t_2, \ldots, t_m\}$, $A = \{a_1, \ldots, a_n\}$; that is, besides $Start$ and $Finish$, there is a task corresponding to each variable in $\psi'$. The only non-trivial ordering conditions are $t_1, t_2, \ldots, t_m \prec t_{m+1}$. Take $Ag = \{a_t : t \in T \setminus \{Start, Finish\}\}$ and $\alpha$ such that each task $t$ in $T \setminus \{Start, Finish\}$ is associated with agent $a_t$.

Finally, we define $pre$ and $post$ so that $clob(t) = \emptyset$ for all actions $t$, $est(t_{m+1}) = \emptyset$, and $est(t_i) = 0$ for $i = 1, \ldots, m$, and $est(Finish) = \{E_{a_1}, \ldots, E_{a_n}, \{t_1, t_2, \ldots, t_m\}\}$, where $E_{a_t}$ consists of the tasks $t_1, t_2, \ldots, t_m$ such that $X_i$ and $Y_j$ appear in $\alpha_t$, together with $t_{m+1}$. This ensures that the equation for $pf(Finish)$ looks like $\neg \varphi$, except each variable $Z \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$ is replaced by $im_{a_t}(t_{m+1})$.

Consider the causal model $M$. We claim that $X = 0$ is part of a cause of $\neg \varphi$ in $(M, \alpha)$, where $\alpha$ sets all endogenous variables to 0, if $im_{\alpha_t}(t_{m+1}) = 1$ is a cause of $\neg pf(Finish)$ in $(M, \alpha, \alpha')$, where $\alpha'$ is such that $im_a(t) = 0$ for all tasks $t \in T \setminus \{Start, Finish\}$. Suppose that $X = 0$ is part of a cause of $\neg \varphi$ in $(M, \alpha')$. Then there exists some $\tilde{V} \subseteq \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$ such that $\tilde{V} = \emptyset \land X = 0$ is a cause of $\neg \varphi'$. The corresponding conjunction $\left(\forall V \subseteq \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\} : \neg pf(Finish) \land \neg \varphi(V) \rightarrow \alpha(V) \rightarrow \neg pf(Finish)\right)$ is a cause of $\neg pf(Finish)$. Conversely, suppose that $im_{\alpha_t}(t_{m+1}) = 0$ is a part of a cause of $\neg pf(Finish)$ in $(M, \alpha')$. Thus, there exists a set $\tilde{V}$ such that $\tilde{V} = \emptyset \land im_{\alpha_t}(t_{m+1}) = 0$ is a cause of $\neg pf(Finish)$ in $(M, \alpha')$. Note that $im_{\alpha_t}(t_{m+1}) \notin \tilde{V}$ for $i = 1, \ldots, n$. For it is easy to see that $(M, \alpha') = \left[im_{\alpha_t}(t_{m+1}) \rightarrow 1\right] pf(Finish)$, so AC3 would be violated if $im_{\alpha_t}(t_{m+1}) \in \tilde{V}$. The same holds true if $en(t_{m+1}) \in \tilde{V}$ if $pf(t_{m+1}) \in \tilde{V}$ or $en(t_{m+1}) \in \tilde{V}$. Next note that if $en(t) \in \tilde{V}$ then it can be replaced by $im_{\alpha_t}(t_Z)$, for $Z \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$, and similarly for $pf(t_{m+1})$. That is, if $\tilde{V}'$ is the set obtained after doing this replacement, then $\tilde{V}' \land X = 0$ is a cause of $\neg pf(Finish)$ iff $\tilde{V}' \land X = 0$ is a cause of $\neg pf(Finish)$. The upshot of this discussion is that, without loss of generality, we can take $\tilde{V}$ to be a subset of $\{im_{\alpha_t}(t) : Z \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}\}$. It now easily follows that if $\tilde{V}'$ is the corresponding subset of $\{im_{\alpha_t}(t) : Z \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}\}$, then $\tilde{V}' = \emptyset \land X = 0$ is a cause of $\neg \varphi'$ in $(M, \alpha)$. This completes the proof for part of a cause.

The argument for the degree of responsibility is similar to Theorem 5.2. For the upper bound, we guess a cause where the proportion of $a$-controlled variables with value 0 is greater or equal to $m/k$. Then we can check in polynomial time that it is indeed a cause of $\neg pf(Finish)$. The lower bound follows from the previous argument (for the special case when $m = 1$) and the degree of responsibility of an agent $a_t$ is the same as the degree of responsibility of $im_a(t)$, as in Theorem 5.2.

### 6. CONCLUSIONS

We have shown how the definitions of causality, responsibility and blame from [11] can be used to give useful insights in the context of team plans. We also showed that the resulting problems are tractable: causality for team plans can be computed in polynomial time, while the problem of determining the degree of responsibility and blame is NP-complete; for postcondition minimal plans, the degree of responsibility and blame can be computed in polynomial time. We can extend our model with external events (or actions by an environment agent) without increase in complexity. We chose not to consider events here, as we are concerned only with allocating responsibility and blame to agents (rather than to the environment). In future work, we would like to consider a richer setting, where agents may be able to perform actions that decrease the probability of plan failure due to external events.

The epistemic perspective of the paper is that of an outside observer rather than the agents. In future work we plan to model agents reasoning about the progress of plan execution, which would involve their beliefs about what is happening and who is to blame for the failure of the plan.

### REFERENCES


