Causality, Responsibility and Blame in Team Plans

Natasha Alechina
University of Nottingham
nza@cs.nott.ac.uk

Joseph Y. Halpern*
Cornell University
halpern@cornell.edu

Brian Logan
University of Nottingham
bsl@cs.nott.ac.uk

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?

*Supported in part by NSF grants IIS-0534064, IIS-0812045, IIS-0911036, and CCF-1214844, and by AFOSR grants FA9550-08-1-0438, FA9550-09-1-0266, and FA9550-12-1-0040, and ARO grant W911NF-09-1-0281.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

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/blameworthy 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 question of 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^p-completeness of determining causality in general [11] and the Σ_2^p-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* M is a pair $(\mathcal{S}, \mathcal{F})$, where $\mathcal{S}$ is a *signature* that explicitly lists the endogenous and exogenous variables and characterises their possible values, and $\mathcal{F}$ is a function that associates a structural equation with each variable. A signature $\mathcal{S}$ is a tuple $(\mathcal{U}, \mathcal{V}, \mathcal{R})$, where $\mathcal{U}$ is a set of exogenous variables, $\mathcal{V}$ is a set of endogenous variables, and $\mathcal{R}$ associates with every variable $Y \in \mathcal{U} \cup \mathcal{V}$ a nonempty set $\mathcal{R}(Y)$ of possible values for Y (i.e., the set of values over which Y *ranges*). $\mathcal{F}$ associates with each endogenous variable $X \in \mathcal{V}$ a function denoted F_X such that $F_X : (\times_{U\in\mathcal{U}}\mathcal{R}(U)) \times (\times_{Y\in\mathcal{V}-\{X\}}\mathcal{R}(Y)) \to \mathcal{R}(X)$. 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 $M = (\mathcal{S}, \mathcal{F})$ results in a new causal model, denoted $M_{X\leftarrow x}$, which is identical to M, except that the equation for X in $\mathcal{F}$ is replaced by $X = x$.

Given a signature $\mathcal{S} = (\mathcal{U}, \mathcal{V}, \mathcal{R})$, a *primitive event* is a formula of the form $X = x$, for $X \in \mathcal{V}$ and $x \in \mathcal{R}(X)$. A *causal formula (over $\mathcal{S}$)* is one of the form $[Y_1 \leftarrow y_1, \dots, Y_k \leftarrow y_k]\varphi$, where

- φ is a Boolean combination of primitive events,
- $Y_1, \dots, Y_k$ are distinct variables in $\mathcal{V}$, and
- $y_i \in \mathcal{R}(Y_i)$.

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

Following [11, 14], we restrict attention here to what are called *acyclic* models. This is the special case where there is some total ordering $\prec$ of the endogenous variables (the ones in $\mathcal{V}$) such that if $X \prec Y$, then X is independent of Y, that is, $F_X(\vec{z}, y, \vec{v}) = F_X(\vec{z}, y', \vec{v})$ for all $y, y' \in \mathcal{R}(Y)$. If $X \prec Y$, 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 $\vec{u}$ for the exogenous variables in $\mathcal{U}$, there is a unique solution for all the equations: we simply solve for the variables in the order given by $\prec$.

A causal formula ψ is true or false in a causal model, given a context. We write $(M, \vec{u}) \models \psi$ if the causal formula ψ is true in causal model M given context $\vec{u}$. The $\models$ relation is defined inductively. $(M, \vec{u}) \models X = x$ 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 $\vec{u}$ (i.e., the unique vector of values for the exogenous variables that simultaneously satisfies all equations in M with the variables in $\mathcal{U}$ set to $\vec{u}$). The truth of conjunctions and negations is defined in the standard way. Finally, $(M, \vec{u}) \models [\vec{Y} \leftarrow \vec{y}]\varphi$ if $(M_{\vec{Y}=\vec{y}}, \vec{u}) \models \varphi$. Thus, $[\vec{Y} \leftarrow \vec{y}]\varphi$ is true in $(M, \vec{u})$ if φ is true in the model that results after setting the variables in $\vec{Y}$ to $\vec{y}$.

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 $\vec{X} = \vec{x}$, can be causes. What can be caused are arbitrary Boolean combinations of primitive events. Roughly speaking, $\vec{X} = \vec{x}$ is a cause of φ if, had $\vec{X} = \vec{x}$ 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. *$\vec{X} = \vec{x}$ is an* actual cause of φ in $(M, \vec{u})$ *if the following three conditions hold:*

AC1. $(M, \vec{u}) \models (\vec{X} = \vec{x})$ *and* $(M, \vec{u}) \models \varphi$.

AC2^m. *There is a set $\vec{W}$ of variables in $\mathcal{V}$ and settings $\vec{x}'$ of the variables in $\vec{X}$ and $\vec{w}$ of the variables in $\vec{W}$ such that $(M, \vec{u}) \models \vec{W} = \vec{w}$ and*

$$(M, \vec{u}) \models [\vec{X} \leftarrow \vec{x}', \vec{W} \leftarrow \vec{w}]\neg\varphi.$$

AC3. *$\vec{X}$ is minimal; no subset of $\vec{X}$ satisfies conditions AC1 and AC2^m.*

AC1 just says that for $\vec{X} = \vec{x}$ to be a cause of φ, both $\vec{X} = \vec{x}$ and φ have to be true. AC3 is a minimality condition, which ensures that only the conjuncts of $\vec{X} = \vec{x}$ that are essential are parts of a cause. AC2^m (the "m" is for modified; the notation is taken from [11]) captures the counterfactual. It says that if we change the value of $\vec{X}$ from $\vec{x}$ to $\vec{x}'$, while possibly holding the values of the variables in some (possibly empty) set $\vec{W}$ fixed at their values in the current context, then φ becomes false. We say that $(\vec{W}, \vec{x}')$ is a *witness* to $\vec{X} = \vec{x}$ being a cause of φ in $(M, \vec{u})$. If $\vec{X} = \vec{x}$ is a cause of φ in $(M, \vec{u})$ and $X = x$ is a conjunct of $\vec{X} = \vec{x}$, then $X = x$ is *part of a cause* of φ in $(M, \vec{u})$.

In general, there may be multiple causes for a given outcome. For example, consider a plan that requires performing two tasks, t_1 and t_2. Let M be a model with binary endogenous variables T_1, T_2, and Fin, and one exogenous variable U. $T_i = 1$ if task t_i is performed and 0 otherwise; $Fin = 1$ if the plan is successfully completed, and 0 otherwise; U determines whether the tasks were performed. (In what follows, we consider more sophisticated models where the agents' intentions to perform their tasks are determined by U.) The equation for Fin is $Fin = T_1 \wedge T_2$. 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 $(M, u) \models T_1 = 0 \wedge T_2 = 0 \wedge Fin = 0$. AC2^m 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 φ measures the minimal number of changes and number of variables that have to be held fixed in order to make φ 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 φ in $(M, \vec{u})$*, denoted* $dr((M, \vec{u}), (X = x), \varphi)$*, is 0 if* $X = x$ *is not part of a cause of* φ *in* $(M, \vec{u})$*; it is* $1/k$ *if there exists a cause* $\vec{X} = \vec{x}$ *of* φ *and a witness* $(\vec{W}, \vec{x}')$ *to* $\vec{X} = \vec{x}$ *being a cause of* φ *in* $(M, \vec{u})$ *such that (a)* $X = x$ *is a conjunct of* $\vec{X} = \vec{x}$*, (b)* $|\vec{W}| + |\vec{X}| = k$*, and (c)* k *is minimal, in that there is no cause* $\vec{X}_1 = \vec{x}_1$ *for* φ *in* $(M, \vec{u})$ *and witness* $(\vec{W}', \vec{x}_1')$ *to* $\vec{X}_1 = \vec{x}_1$ *being a cause of* φ *in* $(M, \vec{u})$ *that includes* $X = x$ *as a conjunct with* $|\vec{W}'| + |\vec{X}_1| < k$.

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 $(\mathcal{K}, \Pr)$, where $\mathcal{K}$ is a set of causal settings, that is, pairs of the form $(M, \vec{u})$, and $\Pr$ is a probability distribution over $\mathcal{K}$. We call such a pair an *epistemic state*. Note that once we have such a distribution, we can talk about the probability that $\vec{X} = \vec{x}$ is a cause of φ relative to $(\mathcal{K}, \Pr)$: it is just the probability of the set of pairs $(M, \vec{u})$ such that $\vec{X} = \vec{x}$ is a cause of φ in $(M, \vec{u})$. We also define the *degree of blame* of $X = x$ for φ to be the expected degree of responsibility:

DEFINITION 2.3. *The* degree of blame *of* $X = x$ *for* φ *relative to the epistemic state* $(\mathcal{K}, \Pr)$ *is*

$$\sum_{(M,\vec{u}) \in \mathcal{K}} dr((M, \vec{u}), X = x, \varphi) \Pr((M, \vec{u})).$$

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],[1] 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.

[1]In their approach to identifying causes, Witteveen et al. [25] assume that tasks are executed as soon as possible, consistent with the order on tasks; we do not assume this.

As is standard in planning literature, e.g., [24, 25], we define 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 $\mathcal{D} = (\Pi, \mathcal{T}, pre, post)$, where Π is a set of atomic propositions, $\mathcal{T}$ is the set of tasks possible in the domain, and pre and $post$ are functions from $\mathcal{T}$ to subsets of $\Pi \cup \{\neg p : p \in \Pi\}$. For each $t \in \mathcal{T}$, $pre(t)$ 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 $T \subseteq \mathcal{T} \cup \{Start, Finish\}$, 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 t done before t' is 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 $\mathcal{P}$ *over a planning domain* $\mathcal{D}$ *and problem* $\mathcal{G}$ *is a tuple* $\mathcal{P} = (T, Ag, \prec, \alpha)$*, where*

- $\{Start, Finish\} \subseteq T \subseteq \mathcal{T} \cup \{Start, Finish\}$ *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* $t \in T \setminus \{Start, Finish\}$*;*
- α *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 ℓ of t, a task t' is a *clobberer* of t (or the precondition ℓ of t) if $\sim\ell \in post(t')$ (where $\sim\ell$ denotes $\neg p$ if $\ell = p$ and p if $\ell = \neg p$).

DEFINITION 3.2. *Given a team plan* $\mathcal{P} = (T, Ag, \prec, \alpha)$*, a task* $t' \in T$ establishes literal ℓ *for a task* $t \in T$ *if* $\ell \in prec(t)$*,* $\ell \in post(t')$*,* $t' \prec t$*, and for every task* $t'' \in T$ *that clobbers* ℓ*, 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* $\ell \in prec(t)$*.* $\mathcal{P}$ 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 $\mathcal{P}$ 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 $\{t_1, \ldots, t_n\}$ 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 ℓ of some task t.

As an illustration, consider the plan $\mathcal{P}_1 = (T_1, Ag_1, \prec, \alpha_1)$, where $T_1 = \{Start, Finish, t_1, t_2\}$, t_1 is laying cables for traffic signals (under the road surface), t_2 is surfacing the road, $Ag_1 = \{a_1, a_2\}$, $\prec = Start \prec t_1 \prec t_2 \prec Finish$, $\alpha_1(t_1) = a_1$, and $\alpha_1(t_2) = a_2$. The goal $prec(Finish) = \{c, s\}$, where c stands for 'cables laid' and s for 'road surfaced'. $post(Start) = \{\neg c, \neg s\}$; $prec(t_1) = \{\neg s\}$ (since cables are laid under the surface); $post(t_1) = \{c\}$; $prec(t_2) = \emptyset$; and $post(t_2) = \{s\}$. This plan is accomplishes its goal; the preconditions of $Finish$ are established by $\{t_1, t_2\}$, while the precondition of t_1 is established by $Start$. 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 $\prec$ to be executed after t_1. Note that the plan $\mathcal{P}_1$ 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 $\mathcal{P} = (T, Ag, \prec, \alpha)$ determines a causal model $M_\mathcal{P}$ 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 $est(t)$ and the set $clob(t)$. The set $est(t)$ consists of all the establishing sets for task t. The assumption that the plan accomplishes its goal ensures that, for all tasks t, $est(t) \neq \emptyset$.

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

For each task $t \in T$, we have variables $en(t)$ for 't is enabled', $in_a(t)$ for 'agent $a = \alpha(t)$ intends to do task t', and $pf(t)$ for 't is performed'. $en(t)$ is true if all the tasks in one of the establishing sets S of t are performed, and no t' such that $(s, t') \in clob(t)$ and $s \in S$ is performed after s (i.e., s is not clobbered). (We typically omit $en(t)$ from the causal model if $est(t)$ is empty, since $en(t)$ is vacuously true in this case.) In order for t to be performed, it has to be enabled and the agent assigned the task has to actually decide to perform it; the latter fact is captured by the formula $in_a(t)$. For example, even if the roadbed has been laid and it is possible to surface the road (so the road-surfacing task is enabled), if the road-surfacing contractor does not show up, the road will not be surfaced. $in_a(t)$ depends only on the agent a. $pf(t)$ is true if both $en(t)$ and $in_a(t)$ are true, where $a = \alpha(t)$. Finally, for each pair (s, t') in $clob(t)$, we have a variable $nc(s, t', t)$, which stands for 't' is *not* executed between s and t'.

Consider again the example plan $\mathcal{P}_1$ from Section 3. The causal model for $\mathcal{P}_1$ has the variables $pf(Start)$, $en(t_1)$, $in_{a_1}(t_1)$, $pf(t_1)$, $in_{a_2}(t_2)$, $pf(t_2)$, $en(Finish)$, $pf(Finish)$, and $nc(Start, t_2, t_1)$. (Note that we omit $en(Start)$ and $en(t_2)$ because $Start$ and t_2 have no preconditions.) $nc(Start, t_2, t_1)$ is true if t_2 is performed after t_1 and false if t_2 is performed before t_1. $en(t_1)$ is true if $pf(Start)$ is true and $nc(Start, t_2, t_1)$ is true.

More precisely, a team plan $\mathcal{P} = (T, Ag, \prec, \alpha)$ determines causal model $M_\mathcal{P} = ((\mathcal{U}_\mathcal{P}, \mathcal{V}_\mathcal{P}, \mathcal{R}_\mathcal{P}), \mathcal{F}_\mathcal{P})$ as follows:

- $\mathcal{U}_\mathcal{P} = \{U_{a,t} : t \in T, a = \alpha(t)\} \cup \{U_{nc(s,t',t)} : s, t', t \in T, (s, t') \in clob(t)\}$. Intuitively, $U_{a,t}$ and $U_{nc(t',s,t)}$ determine the value of $in_a(t)$ and $nc(t', s, t)$, respectively.

- $\mathcal{V}_\mathcal{P} = \{en(t) : t \in T\} \cup \{pf(t) : t \in T\} \cup \{in_a(t) : t \in T, a = \alpha(t)\} \cup \{nc(s, t', t) : s, t', t \in T, (s, t') \in clob(t)\}$. Note that $|\mathcal{V}_\mathcal{P}| \leq |T|^3 + 3|T|$.

- $\mathcal{R}_\mathcal{P}(X) = \{0, 1\}$ for all variables $X \in \mathcal{U}_\mathcal{P} \cup \mathcal{V}_\mathcal{P}$ (i.e., all variables are binary).

- $\mathcal{F}_\mathcal{P}$ is determined by the following equations:
  $in_a(t) = U_{a,t}$
  $nc(s, t', t) = U_{nc(s,t',t)}$
  $pf(t) = en(t) \wedge in_a(t)$ (where $t \in T$ and $a = \alpha(t)$)
  $en(t) = \bigvee_{S \in est(t)} (\bigwedge_{s \in S} pf(s) \wedge \bigwedge_{(s,t') \in clob(t)} nc(s, t', t))$.

It should be clear that $M_\mathcal{P}$ captures the intent of the team plan $\mathcal{P}$. In particular, it is easy to see that the appropriate agents performing their tasks results in $\mathcal{P}$ accomplishing its goal iff $(M_\mathcal{P}, \vec{u}) \models pf(Finish)$, 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 $M_\mathcal{P}$ is polynomial in the size of $\mathcal{P}$ if $\mathcal{P}$ 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 $est(t)$ 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 $pf(Finish)$ = 0. As we suggested earlier, we are interested only in causes that involve formulas of the form $in_a(t) = 0$. We refer to variables of the form $in_a(t)$ as the variables *controlled* by agent a.

DEFINITION 4.1. *Agent a's* degree of responsibility for the failure of plan *$\mathcal{P}$ (i.e., for $pf(Finish) = 0$ in $(M_\mathcal{P}, \vec{u})$, where $M_\mathcal{P}$ is the causal model determined by a team plan $\mathcal{P}$) is 0 if none of the variables controlled by agent a is part of a cause of $pf(Finish) = 0$ in $(M_\mathcal{P}, \vec{u})$; otherwise, it is the maximum value m/k such that there exists a cause $\vec{X} = \vec{x}$ of $pf(Finish) = 0$ and a witness $(\vec{W}, \vec{x}')$ to $\vec{X} = \vec{x}$ being a cause of $pf(Finish) = 0$ in $(M_\mathcal{P}, \vec{u})$ 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 $in_a(t)$ 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 $\mathcal{P}_1$ and its corresponding causal model $M_{\mathcal{P}_1}$. 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 $pf(Finish) = 0$. We cannot observe the values of $in_{a_1}(t_1)$ and $in_{a_2}(t_2)$, but from $pf(t_1) = 1$ 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 $pf(Finish) = 0$ 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 $pf(Finish) = 0$, 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 $\mathcal{P}_2$, which is like $\mathcal{P}_1$, but has an extra task $t_0 \prec t_1$ 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 $pf(Finish) = 0$. As before, $in_{a_2}(t_0) = 0$ and $in_{a_2}(t_2) = 0$ are parts of the cause of $pf(Finish) = 0$. 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 $pf(Finish) = 0$ is just $\{in_{a_2}(t_0) = 0, in_{a_2}(t_2) = 0\}$ 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 $\vec{X} = \vec{1}$ is a cause of φ in $(M, \vec{u})$ is D^p-complete*

(b) Determining if $X = x$ is part of a cause of φ in $(M, \vec{u})$ is Σ_2^p-complete

(c) Determining if $X = x$ has degree of responsibility at least $1/k$ is Σ_2^p-complete.

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

For part (b), first note that the problem is clearly in Σ_2^p: we simply guess $\vec{X}$, $\vec{x}$, $\vec{x}'$, and $\vec{W}$, where $X = x$ is a conjunct of $\vec{X} = \vec{x}$, 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 $(M, \vec{u}) \models [\vec{X} \leftarrow \vec{x}', \vec{W} \leftarrow \vec{w}]\neg\varphi$, 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 $(M, \vec{u}) \models [\vec{Y} \leftarrow \vec{y}', \vec{W} \leftarrow \vec{w}']\neg\varphi$, where $\vec{w}'$ is such that $(M, \vec{u}) \models \vec{W}' = \vec{w}'$.

For Σ_2^p-hardness, we adapt arguments used by Aleksandrowicz et al. [2] to show that checking whether a formula satisfies AC1 and AC2^m is Σ_2^p-complete.

Recall that to show that a language L is Σ_2^p-hard, it suffices to show that we can reduce determining if a closed quantified Boolean formlua (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 φ' are contained in $\vec{x} \cup \vec{y}$) to checking if a string $\sigma \in L$ [23]. Given a closed QBF $\varphi = \exists\vec{x}\,\forall\vec{y}\,\varphi'$, we construct a causal formula ψ, a causal model M, and context $\vec{u}$ such that φ is true iff $A = 0$ is part of a cause of ψ in $(M, \vec{u})$.

We proceed as follows: we take M to be a model with endogenous variables $\mathcal{V} = A \cup \vec{X}^0 \cup \vec{X}^1 \cup \vec{Y}$, where for each variable $x \in \vec{x}$, there are corresponding variables $X_x^0 \in \vec{X}^0$ and $X_x^1 \in \vec{X}^1$, and for each variable $y \in \vec{y}$ there is a corresponding variable $Y_y \in \vec{Y}$, 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 $\bar{\varphi}'$ be the causal formula that results from replacing all occurrences of x and y in φ' by $X_x^1 = 1$ and $Y_y = 1$, respectively. Let ψ be the formula $\psi_1 \vee (\psi_2 \wedge \psi_3)$, where

- $\psi_1 = (\bigvee_{x \in \vec{x}} (X_x^0 = X_x^1))$;[2]
- $\psi_2 = A = 0 \vee \neg(\vec{Y} = \vec{1})$;
- $\psi_3 = (A = 1) \vee \bar{\varphi}'$.

We now show that $A = 0$ is part of a cause of ψ in (M, u) iff φ is true. First suppose that φ is true. Then there is an assignment τ to the variables in $\vec{x}$ such that $\forall\vec{y}\,\varphi'$ is true given τ. Let $\vec{x}'$ be the subset of variables in $\vec{x}$ that are set to true in τ, 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 τ, then the corresponding variable X_x^0 is in $\vec{X}''$). Note that for each variable $x \in \vec{x}$, exactly one of X_x^0 and X_x^1 is in $\vec{X}' \cup \vec{X}''$. We claim that $A = 0 \wedge \vec{X}' = \vec{0} \wedge \vec{X}'' = \vec{0} \wedge \vec{Y} = \vec{0}$ is a cause of ψ in (M, u). Clearly $(M, u) \models A = 0 \wedge \psi$ (since $(M, u) \models \psi_1$). It is immediate from the definitions of ψ_2 and ψ_3 that

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

so

$$(M, u) \models [\vec{A} \leftarrow 1, \vec{X}' \leftarrow \vec{1}, \vec{X}'' \leftarrow \vec{1}, \vec{Y} \leftarrow \vec{1}]\neg\psi.$$

Thus, AC1 and AC2^m hold. It suffices to prove AC3. So suppose that there is some subset $\vec{Z}$ of $\vec{A} \cup \vec{X}' \cup \vec{Y}$ and a set $\vec{W}$ such that $(M, u) \models [\vec{Z} \leftarrow \vec{1}, \vec{W} = \vec{w}]\neg\psi$, where $(M, u) \models \vec{W} = \vec{w}$. Since $(M, u) \models \vec{W} = \vec{0}$, it must be the case that $\vec{w} = \vec{0}$, so $(M, u) \models [\vec{Z} \leftarrow \vec{1}]\neg\psi$. Clearly we must have $\vec{Z} \cap (\vec{X}^0 \cup \vec{X}^1) = \vec{X}' \cup \vec{X}''$, for otherwise $(M, u) \models [\vec{Z} \leftarrow \vec{1}]\psi_1$ and $(M, u) \models [\vec{Z} \leftarrow \vec{1}]\psi$. $(M, \vec{u}) \models [\vec{Z} \leftarrow \vec{1}]\psi$. so $(M, u) \models [\vec{Z} \leftarrow \vec{1}]\psi_3$. We must have $A \in \vec{Z}$, since otherwise $(M, u) \models [\vec{Z} \leftarrow \vec{1}](A = 0)$, so

[2] $X_x^0 = X_x^1$ is an abbreviation for the causal formula $(X_x^0 = 0 \wedge X_x^1 = 0) \vee (X_x^0 = 1 \wedge X_x^1 = 1)$.

$(M,u) \models [\vec{Z} \leftarrow \vec{1}]\psi_2$, and thus $(M,\vec{u}) \models [\vec{Z} \leftarrow \vec{1}]\psi_2$, and thus $(M,\vec{u}) \models [\vec{Z} \leftarrow \vec{1}]\psi$. We also must have $\vec{Y} \subseteq \vec{Z}$, for otherwise $(M,u) \models [\vec{Z} \leftarrow \vec{1}]\neg(\vec{Y} = \vec{1})$, and again $(M,u) \models [\vec{Z} \leftarrow \vec{1}]\psi_2$ and $(M,u) \models [\vec{Z} \leftarrow \vec{1}]\psi$. Thus, $\vec{Z} = A \cup \vec{X}' \cup \vec{X}'' \cup \vec{Y}$, and AC3 holds.

Finally, we must show that if $A = 0$ is part of a cause of ψ in (M,u) then $\exists \vec{x} \forall \vec{y}\, \varphi'$ is true. So suppose that $A = 0 \wedge \vec{Z} = \vec{0}$ is a cause of ψ in (M,u), where $\vec{Z} \subseteq \mathcal{V} - \{A\}$. We must have $(M,u) \models [A \leftarrow 1, \vec{Z} \leftarrow 1]\neg\psi$, which means that $(M,u) \models [\vec{Z} \leftarrow 1]\neg\psi_1$. Thus, for each $x \in \vec{x}$, $\vec{Z}$ must contain exactly one of X_x^0 and X_x^1. We must also have

$$(M,u) \models [A \leftarrow 1, \vec{Z} \leftarrow 1](\neg\psi_2 \vee \neg\psi_3).$$

Since $(M,u) \models [A \leftarrow 1, \vec{Z} \leftarrow 1](A = 1)$, we have $(M,u) \models [A \leftarrow 1, \vec{Z} \leftarrow 1]\psi_3$, so $(M,u) \models [A \leftarrow 1, \vec{Z} \leftarrow 1]\neg\psi_2$. It follows that $\vec{Y} \subseteq \vec{Z}$.

Let ν be a truth assignment such that $\nu(x)$ is true iff $X_x^1 \in \vec{Z}$. We claim that ν satisfies $\forall \vec{y}\, \varphi'$. Once we show this, it follows that $\varphi = \exists \vec{x} \forall \vec{y}\, \varphi'$ is true, as desired. Suppose, by way of contradiction, that ν does not satisfy $\forall \vec{y}\, \varphi'$. Then there exists a truth assignment ν' that agrees with ν on the assignments to the variables in $\vec{x}$ such that ν' satisfies $\neg\varphi'$. Let $\vec{Y}'$ be the subset of $\vec{Y}$ corresponding to the variables $y \in \vec{y}$ that are true according to ν'. Then if $\vec{Z}'$ is the result of removing from $\vec{Z}$ all the variables in $\vec{Y}$ that are not in $\vec{Y}'$, we have that $(M,u) \models [\vec{Z}' \leftarrow \vec{1}](\neg\psi_1 \wedge \neg\psi_3)$, so $(M,u) \models [\vec{Z}' \leftarrow \vec{1}]\neg\psi$. Thus, $A = 0 \wedge \vec{Z} = 0$ is not a cause of ψ (it does not satisfy AC3), giving us the desired contradiction.

Part (c) is almost immediate from part (b). Again, it is easy to see that checking whether $X = x$ has degree of responsibility in $(M,\vec{u})$ at least $1/k$ is in Σ_2^p: we simply guess $\vec{X}$, $\vec{x}$, $\vec{x}'$, and $\vec{W}$ such that $X = x$ is a conjunct of $\vec{X} = \vec{x}$ and $|\vec{X}| + |\vec{W}| \leq k$, and confirm that $\vec{X} = \vec{x}$ is a cause of φ in $(M,\vec{u})$ with witness $(\vec{x}', \vec{W})$.

To show that $X = x$ has degree of responsibility in $(M,\vec{u})$ at least $1/k$ is Σ_2^p-hard, given an arbitrary formula $\varphi = \exists \vec{x} \forall \vec{y}\, \varphi'$. Note that it follows from part (b) that $A = 0$ has degree of responsibility at least $\frac{1}{|\vec{x}| + \vec{y} + 1}$ for the formula ψ as constructed in part (b) iff φ is true. The result follows. ∎

It now follows that by doing binary search we can compute the degree of responsibility of $X = x$ for φ with $\log(|\varphi|)$ queries to a Σ_2^p oracle, and, as in [3], that the complexity of computing the degree of responsibility is in $\mathrm{FP}^{\Sigma_2^P[\log n]}$, where for a complexity class A, $\mathrm{FP}^{A[\log n]}$ consists of all functions that can be computed by a polynomial-time Turing machine with an A-oracle which on input x asks a total of $O(\log |x|)$ queries [20]. (Indeed, it is not hard to show that it is $\mathrm{FP}^{\Sigma_2^P[\log n]}$-complete; see [3].) Similarly, the problem of computing the degree of blame is in $\mathrm{FP}^{\Sigma_2^P[n]}$.[3]

As we now show, checking causality in a monotone model for formulas φ or $\neg\varphi$, where φ is monotone, is significantly simpler. For team plans, we are interested in determining the causes of $\neg pf(Finish)$ (why was the plan not completed); $pf(Finish)$ is clearly monotone. Say that a causal model is *trivial* if the equations for the endogenous variables involve only exogenous variables (so there are no dependencies between endogenous variables).

[3]We can characterise the complexity of computing the degree of blame by allowing parallel (non-adaptive) queries to an oracle (see [3]); we omit this discussion here.

THEOREM 5.2. *Suppose that M is a monotone causal model and φ is a monotone formula.*

(a) If $(M,\vec{u}) \models \varphi$, then we can find $\vec{X}$ such that $\vec{X} = \vec{1}$ is a cause of φ in $(M,\vec{u})$ in polynomial time.

(b) If $(M,\vec{u}) \models \neg\varphi$, then we can find $\vec{X}$ such that $\vec{X} = \vec{0}$ is a cause of $\neg\varphi$ in $(M,\vec{u})$ in polynomial time.

(c) Determining if $\vec{X} = \vec{1}$ is a cause of φ (resp., $\vec{X} = \vec{0}$ is a cause of $\neg\varphi$) in $(M,\vec{u})$ can be done in polynomial time.

(d) Determining if $X = 1$ is a part of a cause of φ (resp., $X = 0$ is part of a cause of $\neg\varphi$) in $(M,\vec{u})$ is NP-complete; NP-hardness holds even if M is a trivial monotone causal model and φ has the form $\psi \wedge (\varphi' \vee X = 1)$, where φ' is a monotone formula in DNF whose variables are contained in $\{X_1,\ldots,X_n,Y_1,\ldots,Y_n\}$ and ψ is the formula $(X_1 = 1 \vee Y_1 = 1) \wedge \ldots \wedge (X_n = 1 \vee Y_n = 1)$.

(e) Determining if $X = 1$ has degree of responsibility at least $1/k$ for φ (resp., $X = 0$ has degree of responsibility at least $1/k$ for $\neg\varphi$) in $(M,\vec{u})$ is NP-complete. NP-hardness holds even if M is a trivial monotone causal model and φ has the form $\psi \wedge (\varphi' \vee X = 1)$, where φ' is a formula in DNF whose variables are contained in $\{X_1,\ldots,X_n,Y_1,\ldots,Y_n\}$ and ψ is the formula $(X_1 = 1 \vee Y_1 = 1) \wedge \ldots \wedge (X_n = 1 \vee Y_n = 1)$.

Proof: For part (a), let $X_1,\ldots,X_k$ be all the variables that are 1 in $(M,\vec{u})$. Clearly, only $X_i = 1$ for $i = 1,\ldots,k$ can be part of a cause of φ in $(M,\vec{u})$ (since M and φ are monotone). Let $\vec{X}^0 = \{X_1,\ldots,X_k\}$. Clearly, $(M,\vec{u}) \models [\vec{X}^0 \leftarrow \vec{0}]\neg\varphi$. Define $\vec{X}^j$ for $j > 0$ inductively by taking $\vec{X}^j = \vec{X}^{j-1} - \{X_j\}$ if $(M,\vec{u}) \models [\vec{X}^j - \{X_j\} \leftarrow 0]\neg\varphi$, and $\vec{X}^j = \vec{X}^{j-1}$ otherwise. The construction guarantees that $(M,\vec{u}) \models [\vec{X}^k \leftarrow 0]\neg\varphi$, and that $\vec{X}^k$ is a minimal set with this property. Thus, $\vec{X}^k = \vec{1}$ is a cause of φ in $(M,\vec{u})$.

For part (b), we proceed just as in part (a), except that we switch the roles of φ and $\neg\varphi$ and replace 0s by 1s. We leave details to the reader.

For part (c), to check that $\vec{X} = \vec{1}$ is a cause of φ, first check if $(M,\vec{u}) \models (\vec{X} = \vec{1}) \wedge \varphi$. (As observed above, this can be done in polynomial time.) If so, then AC1 holds. Then check if $(M,\vec{u}) \models [\vec{X} \leftarrow \vec{0}]\neg\varphi$. If not, $\vec{X} = \vec{1}$ is not a cause of φ in $(M,\vec{u})$, since AC2^m fails; the fact that M and φ are monotone guarantees that for all sets $\vec{W}$, if $(M,\vec{u}) \models \vec{W} = \vec{w}$ and $(M,\vec{u}) \models [\vec{X} \leftarrow \vec{0}]\varphi$, then $(M,\vec{u}) \models [\vec{X} \leftarrow \vec{0}, \vec{W} \leftarrow \vec{w}]\varphi$. (Proof: Suppose that $W' \in \vec{W}$. If $(M,\vec{u}) \models W' = 1$, then, because M and φ are monotone, $(M,\vec{u}) \models [\vec{X} \leftarrow \vec{0}, W' \leftarrow 1]\varphi$. On the other hand, if $(M,\vec{u}) \models W' = 0$, then the fact that M is monotone guarantees that $(M,\vec{u}) \models [\vec{X} \leftarrow \vec{0}](W' = 0)$, so $(M,\vec{u}) \models [\vec{X} \leftarrow \vec{0}, W' \leftarrow 0]\varphi$.[4]) For AC3, suppose that $\vec{X} = \{X_1,\ldots,X_k\}$. Let $\vec{X}_{-i}$ consist of all variables in $\vec{X}$ but X_i. Since M and φ are monotone, it is necessary and sufficient to show that $(M,\vec{u}) \models [\vec{X}_{-i} \leftarrow \vec{0}]\varphi$ for all $i = 1,\ldots,k$. Clearly, if any of these statements fails to hold, then AC3 does not hold. On the other hand, if all these statements hold, then AC3 holds. This gives us a polynomial-time algorithm for checking if $\vec{X} = \vec{1}$ is a cause of φ in $(M,\vec{u})$. The algorithm for checking that $\vec{X} = \vec{0}$ is a cause of

[4]This shows that for monotone causal models and monotone formulas, we can always take the set $\vec{W}$ in the witness to be empty.

$\neg\varphi$ is essentially the same, again replacing φ by $\neg\varphi$ and switching the role of 0 and 1.

For part (d), checking if $X = 1$ is part of a cause of φ in $(M, \vec{u})$ is clearly in NP: guess a cause $\vec{X} = \vec{1}$ that includes $X = 1$ as a conjunct, and confirm that it is a cause as discussed above.

To show that checking if $X = 1$ is part of a cause of φ in $(M, \vec{u})$ is NP-hard, suppose that we are given a propositional formula φ, with primitive propositions $x_1, \ldots, x_n$. Let φ^r be the result of (i) converting φ to *negation normal form* (so that all the negations are driven in so that they appear only in front of primitive propositions—this conversion can clearly be done in polynomial time, indeed, in linear time if φ is represented by a parse tree) and (ii) replacing all occurrences of $\neg x_i$ by y_i, where y_i is a fresh primitive proposition. Note that φ^r is monotone. (The formula φ^r was first introduced by Goldsmith, Hagen, and Mundhenk [7] for a somewhat different purpose.)

Let $\bar{\varphi}^r$ be the monotone causal formula that results by replacing each occurrence of x_i (resp., y_i) in φ^r by $X_i = 1$ (resp., $Y_i = 1$). Let $\bar{\varphi}^+ = \psi \wedge (\bar{\varphi}^r \vee X = 1)$, where ψ is

$$(X_1 = 1 \vee Y_1 = 1) \wedge \ldots \wedge (X_n = 1 \vee Y_n = 1).$$

Let M be a model where $\mathcal{V} = \{X, X_1, \ldots, X_n, Y_1, \ldots, Y_n\}$ and U is the only exogenous variable. U determines the values of all the variables in $\mathcal{V}$, so again there are no interesting equations. Let u be the context where all these variables are 1. We claim that $X = 1$ is part of a cause of $\bar{\varphi}^+$ in (M, u) iff $\neg\varphi$ is satisfiable. This clearly suffices to prove the NP lower bound (since φ is satisfiable iff $X = 1$ is a cause of $\neg\bar{\varphi}^r$ in (M, u)). To prove the claim, first suppose that $\neg\varphi$ is unsatisfiable, so φ is valid. Let $\vec{Z}$ be a subset of $\{X_1, \ldots, X_n, Y_1, \ldots, Y_n\}$. We claim that if $\vec{Z}$ contains at most one of X_i and Y_i for $i = 1, \ldots n$, then $(M, u) \models [\vec{Z} \leftarrow \vec{0}](\psi \wedge \bar{\varphi}^r)$. The fact that $(M, u) \models [\vec{Z} \leftarrow \vec{0}]\psi$ is immediate. To see that $(M, u) \models [\vec{Z} \leftarrow \vec{0}]\bar{\varphi}^r$, first suppose that $\vec{Z}$ contains exactly one of X_i or Y_i for all $i \in \{1, \ldots, n\}$. Then $\vec{Z}$ determines a truth assignment to $\vec{x}$ in the obvious way, so $(M, u) \models [\vec{Z} \leftarrow \vec{0}]\bar{\varphi}^r$, since φ is valid. Since $\bar{\varphi}^r$ is monotonic, it follows that if $\vec{Z}$ contains at most one of X_i or Y_i for all $i \in \{1, \ldots, n\}$, then we must also have $(M, u) \models [\vec{Z} \leftarrow \vec{0}]\bar{\varphi}^r$. This completes the argument.

Now suppose, by way of contradiction, that $X = 1$ is part of a cause of $\bar{\varphi}^+$ in (M, u). Then there exists a subset $\vec{Z}$ of $\{X_1, \ldots, X_n, Y_1, \ldots, Y_n\}$ such that $(M, u) \models [\vec{Z} \leftarrow \vec{0}, X \leftarrow 0]\neg\bar{\varphi}^+$. By the argument above, it cannot be the case $\vec{Z}$ contains at most one of X_i and Y_i for all $i = 1, \ldots, n$, for otherwise, we must have $(M, u) \models [\vec{Z} \leftarrow \vec{0}, X \leftarrow 0](\psi \wedge \bar{\varphi}^r)$, and hence $(M, u) \models [\vec{Z} \leftarrow \vec{0}, X \leftarrow 0]\bar{\varphi}^+$. Thus, it must be the case that $\vec{Z}$ includes both X_i and Y_i for some $i \in \{1, \ldots, n\}$. But then $(M, u) \models [\vec{Z} \leftarrow \vec{0}]\neg\psi$, so $(M, u) \models [\vec{Z} \leftarrow \vec{0}]\neg\bar{\varphi}^+$, which contradicts AC3. Thus, $X = 1$ is not part of a cause of $\bar{\varphi}^r$ in (M, u).

Now suppose that $\neg\varphi$ is satisfiable. Then there is a set $\vec{Z} \subseteq \{X_1, \ldots, X_n, Y_1, \ldots, Y\}$ that includes exactly one of X_i and Y_i, for $i = 1, \ldots, n$, such that $(M, u) \models [\vec{Z} \leftarrow \vec{0}]\neg\bar{\varphi}^r$. Let $\vec{Z}'$ be a minimal subset of $\vec{Z}$ such that $(M, u) \models [\vec{Z}' \leftarrow \vec{0}]\neg\bar{\varphi}^r$. We claim that $\vec{Z}' = 1 \wedge X = 1$ is a cause of $\bar{\varphi}^+$. AC1 trivially holds. Clearly $(M, u) \models [\vec{Z}' \leftarrow \vec{0}, X \leftarrow 0]\neg(\bar{\varphi}^r \vee X = 1)$, so $(M, u) \models [\vec{Z}' \leftarrow \vec{0}, X \leftarrow 0]\neg\bar{\varphi}^+$ and AC2 holds. By choice of $\vec{Z}'$, there is no strict subset $\vec{Z}''$ of $\vec{Z}$ such that $(M, u) \models [\vec{Z}'' \leftarrow \vec{0}]\neg\bar{\varphi}^r$. Since $\vec{Z}'$ contains at most one of X_i or Y_i for $i = 1, \ldots, n$, we have that $(M, u) \models [\vec{Z}' \leftarrow \vec{0}]\psi$. It now easily follows that AC3 holds. Thus, $X = 1$ is part of a cause of $\bar{\varphi}^+$.

Since to get NP-hardness it suffices to consider only CNF formulas, and the result above shows that $X = 1$ is a cause of φ^+ iff $\neg\varphi$ is satisfiable, we can restrict to φ being a DNF formula. 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 $\neg\bar{\varphi}^+$ in (M, u'), where M is as above and u' is the context where all variables in $\mathcal{V}$ get value 0.

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

For the lower bound, using the notation of part (d), if the propositional formula φ 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 $\bar{\varphi}^+$ 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 $\neg\bar{\varphi}^+$, 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 $\vec{Z}^+ = 1$ for φ in (M, u). Since $|\vec{Z}^+| = n + 1$, it follows that the degree of responsibility of $X = 1$ for $\bar{\varphi}$ 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 φ in $(M, \vec{u})$ is in $\mathrm{FP}^{\mathrm{NP}[\log n]}$ (a little more effort in the spirit of [3, Theorem 4.3] shows that it is $\mathrm{FP}^{\mathrm{NP}[\log n]}$-complete), while the problem of computing the degree of blame of $X = x$ for φ relative to an epistemic state $(\mathcal{K}, \Pr)$ is in $\mathrm{FP}^{\mathrm{NP}[n]}$.

We can do even better in conjunctive models.

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

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

In the case of degree of responsibility of $\vec{X} = \vec{0}$ for $\neg\varphi$, observe that for a conjunctive formula φ, there is exactly one cause of $\neg\varphi$ in $(M, \vec{u})$: 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 in a precondition); similarly, Theorem 5.3 applies to team plans that are postcondition minimal. The question remains whether the NP-hardness results in parts (d) and (e) of Theorem 5.2 also apply 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 φ 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 $in_a(t) = 0$ is part of a cause of $\neg pf(Finish)$ in $(M_{\mathcal{P}}, \vec{u})$, where $M_{\mathcal{P}}$ is the causal model determined by a team plan $\mathcal{P}$, 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 $in_a(t) = 0$ 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 $\neg\varphi$ in a trivial monotone causal model, where φ has the form $\psi \wedge (\varphi' \vee X = 1)$, φ' is a formula in DNF whose variables are contained in $\{X_1, \ldots, X_n, Y_1, \ldots, Y_n\}$, and ψ is the formula $(X_1 = 1 \vee Y_1 = 1) \wedge \ldots \wedge (X_n = 1 \vee Y_n = 1)$. Given such a model M and formula φ, we construct a model $M_{\mathcal{P}}$ determined by a team plan $\mathcal{P}$ as follows. Suppose that φ' is the formula $\sigma_1 \vee \ldots \vee \sigma_k$, where σ_j is a conjunction of formulas of the form $X_h = 1$ and $Y_h = 1$. The formula φ is clearly logically equivalent to $\varphi'' = (\sigma_1 \wedge \psi) \vee \ldots \vee (\sigma_k \wedge \psi) \vee (X = 1 \wedge \psi)$. Let ψ' be the formula that results by replacing each disjunct $X_i = 1 \vee Y_i = 1$ in ψ by $W_i = 1$, and let φ^* be the formula that results from replacing each occurrence of ψ in φ'' by ψ'. Clearly, φ^* is monotone.

We construct a team plan $\mathcal{P} = (T, Ag, \prec, \alpha)$ with $T = \{Start, Finish, t_X, t_{X_1}, \ldots, t_{X_n}, t_{Y_1}, \ldots, t_{Y_n}, t_{W_1}, \ldots, t_{W_n}\}$; that is, besides $Start$ and $Finish$, there is a task corresponding to each variable in ψ'. The only nontrivial ordering conditions are $t_{X_i}, t_{Y_i} \prec t_{W_i}$. Take $Ag = \{a_t : t \in T \setminus \{Start, Finish\}\}$ and take α such that each task t in $T \setminus \{Start, Finish\}$ is associated with agent a_t. Finally, we define $prec$ and $post$ so that $clob(t) = \emptyset$ for all actions t, $est(t_{W_i}) = \{\{t_{X_i}\}, \{t_{Y_i}\}\}$, $est(t_{X_i}) = \emptyset$, and $est(t_{Y_1}) = \emptyset$ for $i = 1, \ldots, n$, and $est(Finish) = \{E_{\sigma_1}, \ldots, E_{\sigma_k}, \{t_X, t_{W_1}, \ldots, t_{W_n}\}\}$, where E_{σ_j} consists of the tasks t_{X_i} and t_{Y_j} such that X_i and Y_j appear in σ_j, together with $t_{W_1}, \ldots, t_{W_n}$. This ensures that the equation for $pf(Finish)$ looks like φ^*, except each variable $Z \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_n, W_1, \ldots, W_n\}$ is replaced by $in_{a_{t_Z}}(t_Z)$.

Consider the causal model $M_{\mathcal{P}}$. We claim that $X = 0$ is part of a cause of $\neg\varphi^*$ in (M, u), where u sets all endogenous variables to 0, iff $in_{a_{t_X}}(t_X) = 0$ is a part of a cause of $\neg pf(Finish)$ in $(M_{\mathcal{P}}, \vec{u}_{\mathcal{P}})$, where $\vec{u}_{\mathcal{P}}$ is such that $in_{a_t}(t) = 0$ for all tasks $t \in T \setminus \{Start, Finish\}$. Suppose that $X = 0$ is part of a cause of $\neg\varphi^*$ in $(M, \vec{u})$. Then there exists some $\vec{V} \subseteq \{X_1, \ldots, X_n, Y_1, \ldots, Y_n\}$ such that $\vec{V} = \vec{0} \wedge X = 0$ is a cause of $\neg\varphi^*$. The corresponding conjunction $(\wedge_{V \in \vec{V}} in_{a_{t_V}}(t_V) = 0 \wedge in_{a_{t_X}}(t_X) = 0)$ is a cause of $\neg pf(Finish)$ in $(M_{\mathcal{P}}, \vec{u}_{\mathcal{P}})$, so $in_{a_{t_X}}(t_X) = 0$ is part of a cause of $\neg pf(Finish)$.

Conversely, suppose that $in_{a_{t_X}}(t_X) = 0$ is part of a cause of $\neg pf(Finish)$ in $(M_{\mathcal{P}}, \vec{u}_{\mathcal{P}})$. Thus, there exists a set $\vec{V}$ such that $\vec{V} = \vec{0} \wedge in_{a_{t_X}}(t_X) = 0$ is a cause of $\neg pf(Finish)$ in $(M_{\mathcal{P}}, \vec{u}_{\mathcal{P}})$. Note that $in_{a_{t_{W_i}}}(t_{W_i}) \notin \vec{V}$ for $i = 1, \ldots, n$. For it is easy to see that $(M_{\mathcal{P}}, \vec{u}_{\mathcal{P}}) \models [in_{a_{t_{W_i}}}(t_{W_i}) \leftarrow 1] pf(Finish)$, so AC3 would be violated if $in_{a_{t_{W_i}}}(t_{W_i}) \in \vec{V}$. The same holds true if $en(t_{W_i}) \in \vec{V}$ or if $pf(t_{W_i}) \in \vec{V}$. Next note that if $en(t_Z) \in \vec{V}$ then it can be replaced by $in_{a_{t_Z}}(t_Z)$, for $Z \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_n\}$, and similarly for $pf(t_Z)$. That is, if $\vec{V}'$ is the set obtained after doing this replacement, then $\vec{V} \wedge X = 0$ is a cause of $\neg pf(Finish)$ iff $\vec{V}' \wedge X = 0$ is a cause of $\neg pf(Finish)$. The upshot of this discussion is that, without loss of generality, we can take $\vec{V}$ to be a subset of $\{in_{a_{t_Z}}(t_Z) : Z \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_n\}\}$. It now easily follows that if $\vec{V}^*$ is the corresponding subset of $\{X_1, \ldots, X_n, Y_1, \ldots, Y_n\}$, then $\vec{V}^* = \vec{0} \wedge X = 0$ is a cause of $\neg\varphi^*$ in (M, u). This completes the proof for part of a cause.

The argument for the degree of responsibility is similar to Theorem 5.2(e). 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 $in_{a_t}(t)$), as in Theorem 5.2(e). ∎

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

[1] H. Aldewereld, V. Dignum, and W. Vasconcelos. We ought to; they do; blame the management! - A conceptualisation of group norms. In *Coordination, Organizations, Institutions, and Norms in Agent Systems (COIN 2013)*, volume 8386 of *LNCS*, pages 195–210, 2013.

[2] G. Aleksandrowicz, H. Chockler, J. Y. Halpern, and A. Ivrii. The computational complexity of structure-based causality. In *Proc. Twenty-Eighth National Conference on Artificial Intelligence (AAAI '14)*, pages 974–980, 2014.

[3] H. Chockler and J. Y. Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 20:93–115, 2004.

[4] F. de Jonge, N. Roos, and C. Witteveen. Primary and secondary diagnosis of multi-agent plan execution. *Autonomous Agents and Multi-Agent Systems*, 18(2):267–294, 2009.

[5] T. De Lima, L. M. M. Royakkers, and F. Dignum. Modeling the problem of many hands in organisations. In *19th European Conference on Artificial Intelligence (ECAI 2010)*, pages 79–84, 2010.

[6] I. Georgievski and M. Aiello. HTN planning: Overview, comparison, and beyond. *Artificial Intelligence*, 222:124–156, 2015.

[7] J. Goldsmith, M. Hagen, and M. Mundhenk. Complexity of DNF minimization and isomorphism testing for monotone formulas. *Information and Computation*, 206(6):760–775, 2008.

[8] N. Griffiths, M. Luck, and M. d'Inverno. Annotating cooperative plans with trusted agents. In R. Falcone, K. S. Barber, L. Korba, and M. P. Singh, editors, *Trust, Reputation, and Security: Theories and Practice*, volume 2631 of *Lecture Notes in Computer Science*, pages 87–107. Springer, 2003.

[9] D. Grossi, F. Dignum, L. M. M. Royakkers, and J.-J. C. Meyer. Collective obligations and agents: Who gets the blame? In *Deontic Logic in Computer Science (DEON 2004)*, volume 3065 of *LNCS*, pages 129–145, 2004.

[10] D. Grossi, L. M. M. Royakkers, and F. Dignum. Organizational structure and responsibility. *Artificial Intelligence Law*, 15(3):223–249, 2007.

[11] J. Y. Halpern. A modification of the Halpern-Pearl definition of causality. In *Proc. 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 3022–3033, 2015.

[12] J. Y. Halpern. *Actual Causality*. MIT Press, Cambridge, MA, 2016.

[13] J. Y. Halpern and J. Pearl. Causes and explanations: A structural-model approach. Part I: Causes. In *Proc. Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI 2001)*, pages 194–202, 2001.

[14] J. Y. Halpern and J. Pearl. Causes and explanations: A structural-model approach. Part I: Causes. *British Journal for Philosophy of Science*, 56(4):843–887, 2005.

[15] T. D. Lima, L. M. M. Royakkers, and F. Dignum. A logic for reasoning about responsibility. *Logic Journal of the IGPL*, 18(1):99–117, 2010.

[16] E. Lorini and F. Schwarzentruber. A logic for reasoning about counterfactual emotions. *Artificial Intelligence*, 175(3-4):814–847, 2011.

[17] A. Meliou, W. Gatterbauer, K. F. Moore, and D. Suciu. The complexity of causality and responsibility for query answers and non-answers. *Proc. VLDB Endowment*, 4(1):33–45, 2010.

[18] R. Micalizio and P. Torasso. Cooperative monitoring to diagnose multiagent plans. *Journal of Artificial Intelligence Research*, 51(1):1–70, Sept. 2014.

[19] R. Micalizio, P. Torasso, and G. Torta. On-line monitoring and diagnosis of multi-agent systems: A model based approach. In R. L. de Mántaras and L. Saitta, editors, *Proceedings of the 16th Eureopean Conference on Artificial Intelligence (ECAI 2004)*, pages 848–852. IOS Press, 2004.

[20] C. H. Papadimitriou. The complexity of unique solutions. *Journal of ACM*, 31:492–500, 1984.

[21] C. H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *J. Comput. Syst. Sci.*, 28(2):244–259, 1982.

[22] M. P. Singh, A. K. Chopra, and N. Desai. Commitment-based service-oriented architecture. *IEEE Computer*, 42(11):72–79, 2009.

[23] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Comput. Sci.*, 3:1–22, 1977.

[24] D. S. Weld. An introduction to least commitment planning. *AI Magazine*, 15(4):27–61, 1994.

[25] C. Witteveen, N. Roos, R. van der Krogt, and M. de Weerdt. Diagnosis of single and multi-agent plans. In *4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 805–812, 2005.