Dynamics of Causal Dependencies in Multi-agent Settings

Maksim Gladyshev, Natasha Alechina, Mehdi Dastani, and Dragan Doder

Utrecht University, Utrecht, Netherlands
{m.gladyshev, n.a.alechina, m.m.dastani, d.doder}@uu.nl

Abstract. In this paper we discuss how causal models can be used for modeling multi-agent interaction in complex organizational settings, where agents' decisions may depend on other agents' decisions as well as the environment. We demonstrate how to reason about the dynamics of such models using concurrent game structures where agents can change the organisational setting and thereby their decision dependencies. In such concurrent game structure, agents can choose to modify their reactions on other agents' decisions and on the environment by intervening on their part of a causal model. We propose a generalized notion of interventions in causal models that allow us to model and reason about the dynamics of agents' dependencies in a multi-agent system. Finally, we discuss how to model uncertainty and reason about agents' responsibility concerning their dependencies and thereby their choices.

Keywords: Causal models · Interventions · Multi-agent systems.

1 Introduction

The complex interactions between agents in multi-agent systems can be described in terms of organizational structures that determine the dependencies between agents' decisions [1, 9, 15, 6]. Such dependencies can be described in a causal manner, allowing us to reason about the cause of agents' decisions and explain what causes a given agent's decision in terms of the organizational structure and the decisions of other agents on which it depends. For example, in an organisational setting such as banking system, the decision of a loan officer to accept or reject a mortgage application may depend on the decision of her manager. It is also clear that in multi-agent systems the agents interact not only with each other, but also with their shared environment, which is also governed by causal relations. In our simple example, accepting a mortgage application may cause a new contract to be added to the administration database, which in turn may cause a notification to be sent to the mortgage applicant. In general, agents' decisions may have a causal effect on each others decisions' and their shared environment, which in turn may have causal affect on the agents' decisions. In order to study such causal interactions between agents and/or the environment, we use causal models developed in the theory of *actual* causality[13].

There exist two different types of causality. The first one is so-called *type causality*, and is critical in machine learning and for prediction purposes. This kind of causality concerns general statements such as 'smoking causes lung cancer', and can be used to predict, e.g., the probability that someone who smokes gets lung cancer. The second kind of inference, is termed *actual causality*, and is essential in tracing and explaining the cause of a specific outcome, which in turn is essential for assigning responsibility for the outcome to a specific component or decision of an AI system. The theory of actual causality was developed in [21, 12, 14, 11, 13].

We assume that the decision-making mechanism of each agent is specified as a part of a causal model, more specifically, as a function that determines the agent's decision based on the current context, the decisions of the agents that she depends on, and the state of the environment. Simply speaking, given an actual context (e.g., a mortgage application is submitted), the decisions of all agents can be determined through the causal model (e.g., the decision of a loan officer is determined by the submitted mortgage application, its decision-making function that specifies an accept/reject decision based on the decisions of her managers, and perhaps the previous mortgage applications of the same applicant stored in the administration database). In this paper, we investigate how agents can change the causal dependency of their decisions, and thereby the structure of their organization. This allows us to reason about causal structures of organisations and their dynamics. So, the proposed causal modelling approach allows us to reason about causal dependencies between agents and possible interventions of agents to modify their dependencies.

From a technical perspective, we employ MAS models to represent and reason about different causal settings. In such causal settings, the agents' behaviour (decisions) is determined by the structure of a causal model and an assignment of exogenous variables called context. At the same time, each agent has a choice to modify her part of the model, which results in an updated causal model. In a new causal setting for updated model and fixed context, the decisions of agents may be different, as well as the state of the environment. We consider the set of all causal settings to be a set of states in a Concurrent Game Structure (CGS). The updates (called interventions) generate the set of possible actions (choices) for the agents. Then the transitions between states of such CGS can be interpreted as strategic abilities of the agents to enforce the corresponding dependency over their decisions and the environment. In this sense, our approach goes along with other works on Concurrent Game Structures semantics for different logics. In particular, CGS semantics for logics of "sees to it that" (STIT) was proposed in [3]. Our work is also close to [16], where the framework for reasoning about agents' knowledge about actual causes is proposed. The main difference with our approach is that [16] uses different formalization, namely situation calculus (SC), while we stick to original Structural Equations Models (SEM) approach and straightforwardly unfold such SEM into CGS. Our approach allows us to employ well-known MAS machinery for reasoning about transformations of causal models interpreted as the choices of multi-agent organizational structures.

The remainder of this paper is structured as follows. In Section 2 we introduce formal definitions related to causal models. In Section 3 we discuss Concurrent Game Structures and demonstrate how to represent possible interventions in a causal model in terms of CGS models. In Section 4 we propose the generalized notion of interventions for causal models that allow us to reason about more complicated behavior of the agents. Finally, in Section 5 we discuss how to model uncertainty in our settings, then we define the notion of strategic responsibility and demonstrate that the proposed generalized interventions can be more suitable for reasoning about agents' responsibility. For simplicity, in this definition we consider only one-step interactions and leave ATL-style machinery for future work.

2   Preliminaries: Causal Models

We start with the general definition of a causal model as used in [21, 14, 13].

Definition 1 (Causal Model). *A signature is a tuple $S = (\mathcal{U}, \mathcal{V}, \mathcal{R})$, where $\mathcal{U}$ is a finite set of exogenous variables, $\mathcal{V}$ is a finite set of endogenous variables, and $\mathcal{R}$ associates with every variable $Y \in \mathcal{U} \cup \mathcal{V}$ a finite nonempty set $\mathcal{R}(Y)$ of possible values for Y, also called range of Y. A causal model over a signature $\mathcal{S}$ is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{F})$, where $\mathcal{F}$ associates with every endogenous variable $X \in \mathcal{V}$ a function $\mathcal{F}_X$ such that $\mathcal{F}_X$ maps $\times_{Z \in (\mathcal{U} \cup \mathcal{V} - \{X\})} \mathcal{R}(Z)$ to $\mathcal{R}(X)$. That is, $\mathcal{F}_X$ describes how the value of the endogenous variable X is determined by the values of all other variables in $\mathcal{U} \cup \mathcal{V}$. The values of exogenous variables $\mathcal{U}$ are determined outside of the model and usually referred to as a context $\vec{u}$.*

To illustrate this definition, consider Example 1, originating in [17] and extensively analysed in the theory of actual causality [13].

Example 1 (Rock-throwing). Suzy and Billy both pick up rocks and throw them at a bottle (encoded as ST=1 and BT=1 respectively). Suzy's rock gets there first, shattering the bottle. We denote the fact that Suzy's rock hits the bottle as SH=1. Similarly, BH=0 denotes the fact that Billy's rock does not hit the bottle. Finally, BS=1 means 'the bottle shatters'. We also know that because both throws are perfectly accurate, Billy's would have shattered the bottle had it not been preempted by Suzy's throw. [1]

---

[1] Although we use this example due to its simplicity and its extensive analysis in the literature, we can also use new interpretation of this example to illustrate the dependencies of agents' decisions in multi-agent organisations. Let Suzy and Billy be two loan officers working in a bank, who decide to accept or reject a mortgage application. Then $ST = 1$ (and $BT = 1$) can indicate that Suzy (and Billy respectively) rejects an application. Then $SH = 1$ (and $BH = 1$) mean that Suzy's (and Billy's) rejection is registered in the administration database. We also assume that Suzy has a priority, so Billy's rejection is registered ($BH = 1$) only if Suzy's is not ($SH = 0$). Then, the mortgage is rejected ($BS = 1$) if $SH = 1$ or $BH = 1$.

So, our endogenous variables $\mathcal{V}$ are $\{ST, BT, SH, BH, BS\}$. Our exogenous variables $\mathcal{U} = \{U_{ST}, U_{BT}\}$ determine the values of ST and BT variables respectively. For all $Y \in (\mathcal{U} \cup \mathcal{V})$, $\mathcal{R}(Y) = \{0, 1\}$. $\mathcal{F}$ in this example can be defined as follows. Let $\vec{z}$ be an assignment of all variables $(\mathcal{U} \cup \mathcal{V}) \backslash \{X\}$ for corresponding $\mathcal{F}_X$.

$$\mathcal{F}_{SH}(\vec{z}) = \begin{cases} 1 & \text{if } (ST = 1) \in \vec{z}, \\ 0 & \text{if } (ST = 0) \in \vec{z}; \end{cases} \mathcal{F}_{BH}(\vec{z}) = \begin{cases} 1 & \text{if } (ST = 0, BT = 1) \in \vec{z}, \\ 0 & \text{otherwise}; \end{cases}$$

$$\mathcal{F}_{BS}(\vec{z}) = \begin{cases} 1 & \text{if } (SH = 1) \in \vec{z} \text{ or } (BH = 1) \in \vec{z}, \\ 0 & \text{otherwise}; \end{cases}$$

Intuitively, $\mathcal{F}_X$ describes some structural equation that specifies how the value of the endogenous variable X is determined by (and depends on) the values of all other variables in $(\mathcal{U} \cup \mathcal{V}) - \{X\}$. For example, in a causal model with three variables X, Y and Z, the function $\mathcal{F}_X(Y, Z) = Y + Z$ defines the structural equation $X = Y + Z$, while $\mathcal{F}_Y(X, Z) = Z$ defines the structural equation $Y = Z$, etc. The later equation demonstrates that Y does not depend on X. For example, given three variables X, Y and Z, the structural equation for X can be defined as X=Y+Z, X=max(Y, Z), X=Y, or any other complex functional specifications. The later equation demonstrates that X does not depend on Z. Additionally, these equations can be written with an 'iff' notation, for example $X = 1$ iff min(Y, Z) = 0, and $X = 0$ iff min(Y, Z) $\neq 0$. For the case of binary variables it is often more convenient to define structural equations using boolean connectives, e.g. $X = \neg(Y \vee X)$. So, by structural equation for any endogenous variable X we understand the way of specifying how the value of X is determined by the values of other variables[2].

Causal models can be represented as a dependency graph. The nodes of such graph represent variables $\mathcal{U} \cup \mathcal{V}$ (we usually omit exogenous variables from the figures), and edges represent the dependencies between the variables. The dependency graph for Example 1 is presented in Figure 1.
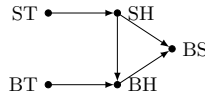
Fig. 1. A dependency graph for the Rock-throwing example.

Now, we need to discuss some restrictions on $\mathcal{F}$ and highlight the difference between recursive and non-recursive models. Following [13], we say that variable Y is independent of X in $(\mathcal{M}, \vec{u})$ if, for all settings $\vec{z}$ of the endogenous variables

---

[2] The detailed overview can be found in [13].

other than X and Y, and all values x and x' of X, $\mathcal{F}_Y(x, \vec{z}, \vec{u}) = \mathcal{F}_Y(x', \vec{z}, \vec{u})$. A model $\mathcal{M}$ is then considered as *recursive* if, for each context $\vec{u}$, there is a partial order $\preceq_{\vec{u}}$ of the endogenous variables such that unless $X \preceq_{\vec{u}} Y$, Y is independent of X in $(\mathcal{M}, \vec{u})$. It guarantees that no cycles can occur in the dependency graph of such model, and then structural equations $\mathcal{F}$ have a unique solution for any $\vec{u}$ [13]. Let $Sol(\vec{u})$ denote a set of all $(X = x)$, where $X \in \mathcal{V}, x \in \mathcal{R}(X)$, such that X has a value x in the unique solution of equations in $\mathcal{M}$ for a context $\vec{u}$.

Causal models allow us to reason not only about an actual context, but also about counterfactual scenarios. These counterfactual scenarios can be described by interventions of the form $[\vec{Y} \leftarrow \vec{y}](Z = z)$, where $\vec{Y} \leftarrow \vec{y}$ abbreviates $(Y_1 \leftarrow y_1, \ldots, Y_k \leftarrow Y_k)$ for $Y_1, \ldots, Y_k \in \mathcal{V}$. We read these formulas as "if $\vec{Y}$ were set to $\vec{y}$, then Z would have a value z". The intervention $\vec{Y} \leftarrow \vec{y}$ in a model $\mathcal{M}$ results in an updated model $\mathcal{M}^{\vec{Y} \leftarrow \vec{y}} = (\mathcal{S}, \mathcal{F}^{\vec{Y} \leftarrow \vec{y}})$.

Definition 2 (Updated Model). *Given a model $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ and intervention $\vec{Y} \leftarrow \vec{y}$, an updated model $\mathcal{M}^{\vec{Y} \leftarrow \vec{y}} = (\mathcal{S}, \mathcal{F}^{\vec{Y} \leftarrow \vec{y}})$ is such that for all $(Y = y) \in \vec{Y} \leftarrow \vec{y}$ and for any assignment $\vec{Z} = \vec{z}$ of all variables other than $Y, \mathcal{F}_Y^{\vec{Y} \leftarrow \vec{y}}(\vec{z}) = y$. So, $\mathcal{F}_Y^{\vec{Y} \leftarrow \vec{y}}$ is a constant function returning y for any input and all $\mathcal{F}_X^{\vec{Y} \leftarrow \vec{y}}$ for $X \notin \vec{Y}$ remain unchanged.*

Next we can define the basic causal language $\mathcal{L}(\mathsf{C})^3$[13].

Definition 3 ($\mathcal{L}(\mathsf{C})$ Syntax). *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 $[Y_1 \leftarrow y_1, \ldots, Y_k \leftarrow y_k]\varphi$, where φ is a Boolean combination of primitive events, $\{Y_1, \ldots, Y_k\} \subseteq \mathcal{V}, y_i \in \mathcal{R}(Y_i)$.*

Language $\mathcal{L}(\mathsf{C}(\mathcal{S}))$ for $\mathcal{S} = (\mathcal{U}, \mathcal{V}, \mathcal{R})$ consists of all Boolean combinations of causal formulas, where the variables in the formulas are taken from $\mathcal{V}$ and the sets of possible values of these variables are determined by $\mathcal{R}$.

Causal formulas from $\mathcal{L}(\mathsf{C})$ can be evaluated on a causal settings $(\mathcal{M}, \vec{u})$ as follows:

Definition 4 (Semantics). *Given a causal settings $(\mathcal{M}, \vec{u})$, and $\mathcal{L}(\mathsf{C})$ formula φ we define $\vDash_{HP}$ relation inductively as follows:*
$(\mathcal{M}, \vec{u}) \vDash_{HP} (X = x)$ iff $(X = x) \in Sol(\vec{u})$,
$(\mathcal{M}, \vec{u}) \vDash_{HP} \neg\varphi$ iff $(\mathcal{M}, \vec{u}) \nvDash_{HP} \varphi$,
$(\mathcal{M}, \vec{u}) \vDash_{HP} (\varphi \wedge \psi)$ iff $(\mathcal{M}, \vec{u}) \vDash_{HP} \varphi$ and $(\mathcal{M}, \vec{u}) \vDash_{HP} \psi$,
$(\mathcal{M}, \vec{u}) \vDash_{HP} [\vec{Y} \leftarrow \vec{y}]\varphi$ iff $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \vec{u}) \vDash_{HP} \varphi$.

As you can see, the nesting of $[\vec{Y} \leftarrow \vec{y}]$ operators is not allowed in $\mathcal{L}(\mathsf{C})$. But if we interpret it as an update operator as Definition 2 suggests, then we can define the result of multiple updates $[\vec{X} \leftarrow \vec{x}][\vec{Y} \leftarrow \vec{y}]$ as a model $(\mathcal{M}^{\vec{X} \leftarrow \vec{x}})^{\vec{Y} \leftarrow \vec{y}}$ updated twice. So, we could reason about the series of model transformations by consecutive interventions $[\vec{X} \leftarrow \vec{x}] \ldots [\vec{Y} \leftarrow \vec{y}]$ (of some agents) on the variables $\mathcal{V}' \subseteq \mathcal{V}$. For example $(\mathcal{M}, \vec{u}) \vDash_{HP} [\vec{X} \leftarrow \vec{x}][\vec{Y} \leftarrow \vec{y}]\varphi$ iff $(\mathcal{M}^{\vec{X} \leftarrow \vec{x}}, \vec{u}) \vDash_{HP} [\vec{Y} \leftarrow \vec{y}]\varphi$ iff $(\mathcal{M}^{\vec{X} \leftarrow \vec{x}})^{\vec{Y} \leftarrow \vec{y}}, \vec{u}) \vDash_{HP} \varphi$.

---

^{3} Please note that for notational convenience we use $\mathcal{L}(\mathsf{C})$ instead of $\mathcal{L}(\mathsf{C}(\mathcal{S}))$.

3   Concurrent Game Structures

We use Concurrent Game Structures semantics for reasoning about causal models' transformations, through which agents' decision-making dependencies (and thereby organisational structure) may change, and strategic abilities of the agents controlling such transformations. In order to do this, we need to distinguish agents from environment in causal models. As we have seen in Example 1, in causal models variables $\mathcal{V}$ can represent both facts about the agents and the environment. So, in our example, ST and BT can be seen as agents' variables for Suzy and Billy respectively, while SH, BH and BS express some facts about the environment. In these models decisions of agents (understood as the values of agents' variables $\mathcal{V}_a$) determine the values of (some) environmental variables ($\mathcal{V}_e$). But the decisions of these agents can also depend on environmental variables and the decisions of other agents. So, it would be interesting to study what agents can enforce by the right choice of the interventions on their variables. At the same time we do not want to consider how environmental variables could be modified, since we treat the causal dependencies of environmental variables as fixed.

In order to study these series of causal models' transformations, first of all we want to generate a Concurrent Game Structure (CGS) for a given causal model. Concurrent Game Structures are usually defined as follows.

Definition 5 (CGS, pointed). *A concurrent game structure (CGS) is a tuple $\Gamma = (\mathbb{AG}, Q, \Pi, \pi, Act, d, o)$, comprising a nonempty finite set of all agents $\mathbb{AG} = \{1, \ldots, k\}$, a nonempty finite set of states Q, a nonempty finite set of atomic propositions Π and their valuation $\pi : Q \longrightarrow \mathcal{P}(\Pi)$, and a nonempty finite set of (atomic) actions Act. Function $d : \mathbb{AG} \times Q \longrightarrow \mathcal{P}(Act) \backslash \{\varnothing\}$ defines nonempty sets of actions available to agents at each state, and o is a (deterministic) transition function that assigns the outcome state $q' = o(q, (\alpha_1, \ldots, \alpha_k))$ to a state q and a tuple of actions $(\alpha_1, \ldots, \alpha_k)$ with $\alpha_i \in d(i, q)$ and $1 \le i \le k$, that can be executed by $\mathbb{AG}$ in q. A pointed CGS is given by (Γ, q), where Γ is a CGS and q is a state in it.*

Let q' be a successor of q if there exists a complete action profile α, such that $q' = o(q, \alpha)$. Given a CGS Γ, a *play* λ in Γ is an infinite sequence $\lambda = q_0, q_1, \ldots$ of states in Q such that, for all $i \ge 0$, the state q_{i+1} is a successor of the state q_i. For a play λ and positions $i, j \ge 0$, we use $\lambda[i], \lambda[j, i]$ and $\lambda[j, \infty)$ to denote the i'th state of λ, the finite segment $q_j, q_{j+1}, \ldots, q_i$, and the suffix $q_j, q_{j+1}, \ldots$ of λ, respectively. A positional (memoryless) *strategy* for an agent $a \in \mathbb{AG}$ or a-strategy, is a function $str_a : Q \longrightarrow d(a, Q)$. Positional strategy of a coalition G is a tuple str_G of positional strategies, one for each player in G.

We assume $\mathcal{V} = \mathcal{V}_a \cup \mathcal{V}_e$, where $\mathcal{V}_a$ is the set of agent variables and $\mathcal{V}_e$ is the disjoint set of environment variables. Now we demonstrate how to generate a CSG $\Gamma_{\mathcal{M}}$ for a casual model $\mathcal{M}$. A causal model $\mathcal{M} = (\mathcal{S}, \mathcal{F})$, given a context $\vec{u}$, is translated to a CGS $\Gamma_{\mathcal{M}} = (\mathbb{AG}, Q, \Pi, \pi, Act, d, o)$, as follows

- $\mathbb{A}\mathbb{G} = \mathcal{V}_a{}^4$;
- $Q = \{\mathcal{M}^{\vec{X}\leftarrow\vec{x}} \mid \vec{X} \subseteq \mathcal{V}_a \ \& \ \vec{x} \in \times\mathcal{R}(\vec{X})\}$;
- $\Pi = \{Y = y \mid Y \in \mathcal{V} \ \& \ y \in \mathcal{R}(Y)\}$;
- π is defined as $(Y = y) \in \pi(\mathcal{M}')$ iff $(\mathcal{M}', \vec{u}) \vDash_{HP} (Y = y)$ for any $\mathcal{M}' \in Q$;
- $Act = \{X \leftarrow x \mid X \in \mathcal{V}_a \ \& \ x \in \mathcal{R}(X)\} \cup \{\top_X \mid X \in \mathcal{V}_a\}$, where $\top_X$ denotes 'no intervention on X';
- $d : \mathcal{V}_a \times Q \longrightarrow \mathcal{R}(Act)$ is defined as $d(X, \mathcal{M}') \subseteq \{X \leftarrow x \mid x \in \mathcal{R}(X)\}$ for any $X \in \mathcal{V}_a$ and $\mathcal{M}' \in Q$;
- $o : Q \times (Act_{X_1} \times \cdots \times Act_{X_k}) \longrightarrow Q$ for $Act_{X_i} = \{X_i \leftarrow x \mid x \in \mathcal{R}(X_i)\}$ and $\{X_1, \ldots, X_k\} = \mathcal{V}_a$ is such that for any $\mathcal{M}_1, \mathcal{M}_2 \in Q$, $\mathcal{M}_2 \in o(\mathcal{M}_1, Act_{\vec{X}})$ iff $\mathcal{M}_1^{Act_{\vec{X}}} = \mathcal{M}_2$.

So, our states Q are all possible results of $[\vec{X} \leftarrow \vec{x}]$ updates of $\mathcal{M}$ where $\vec{X} \subseteq \mathcal{V}_a$. In other words any $\mathcal{M}' \in Q$ is a result of replacing some $\mathcal{F}_X$'s with constant functions.[5] The set of atomic propositions Π consists of all pairs $(Y = y)$. The valuation function π agrees with $\vDash_{HP}$ relation. Every agent i in any state has a set of available actions $[X_i \leftarrow x]$ for $x \in \mathcal{R}(X_i)$ together with an 'empty' action $\top_{X_i}$ meaning 'do nothing'. So, every agent i may choose to replace her $\mathcal{F}_{X_i}$ with a constant function $\mathcal{F}_{X_i} = x$ for any $x \in \mathcal{R}(X_i)$ or not to change $\mathcal{F}_{X_i}$. The choice $(Act_{X_1} \times \cdots \times Act_{X_k})$ of all agents in any state $q \in Q$ determines its (unique) successor state q' according to o. It guarantees that $\mathcal{M}_2$ is a successor of $\mathcal{M}_1$ by a complete action profile $(\vec{X}_{\mathbb{A}\mathbb{G}} \leftarrow \vec{x})$ in the proposed semantics if and only if $\mathcal{M}_2$ is the result of $\mathcal{M}_1^{\vec{X}_{\mathbb{A}\mathbb{G}}\leftarrow\vec{x}}$ update.

Consider how to obtain a CGS for the causal model from Example 1. Our agents Suzy (s) and Billy (b) control variables ST and BT respectively. So, $\mathcal{V}_a = \{ST, BT\}$. Each agent has 3 options: to replace his/her function $\mathcal{F}_i$ with a constant function returning 1, to replace his/her function $\mathcal{F}_i$ with a constant function returning 0 or not to modify $\mathcal{F}_i$. So, initial state has 9 possible transitions. For example if both agent decide not to change their functions, then $o(\mathcal{M}, (\top_{ST}, \top_{BT})) = \mathcal{M}$, i.e. the agents will stay in the initial state. For other 8 action profiles there is a special state reachable from $\mathcal{M}$ in our CGS. This CGS is illustrated in Figure 2.

Here each state is reachable from the initial one, but interestingly, not any state is reachable from the second one. Other simple properties of this CGS are

- $\mathcal{M}^{BT\leftarrow0}$ is not reachable from $\mathcal{M}^{ST\leftarrow0}$ in Figure 2. Because in $\mathcal{M}^{ST\leftarrow0}$ function $\mathcal{F}_{BT}$ is not modified: it returns 0 if $U_{BT=0}$ and 1 otherwise. While in $\mathcal{M}^{BT\leftarrow0}$, $\mathcal{F}_{BT}^{BT\leftarrow0}$ is a constant function, which cannot be restored to its initial configuration $\mathcal{F}_{BT}$ by any available action for agent b in $\mathcal{M}^{BT\leftarrow0}$.

---

[4] Here we assume for simplicity that each agent in $\mathbb{A}\mathbb{G}$ controls only one variable in $\mathcal{V}_a$, so $|\mathbb{A}\mathbb{G}| = |\mathcal{V}_a|$. But without loss of generality one can assume that $\mathcal{V}_a$ is partitioned into disjoint subsets cotrolled by agents in $\mathbb{A}\mathbb{G}$. In this case $|\mathbb{A}\mathbb{G}| \leq |\mathcal{V}_a|$.

[5] We note that such an intervention (updates) make the agents in $\vec{X}$ independent of other agents as their decision-making functional specifications are now reduced to a constant function. Later in Section 4 we will introduce more general interventions (updates) that can create arbitrary dependencies between agents.
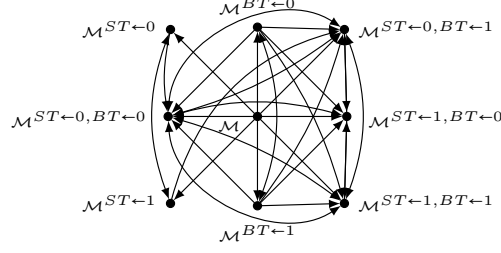
Fig. 2. CGS for the Rock-throwing example. Note that reflexive transitions are omitted from the picture and every transition must be marked with a single or multiple action profiles, which does not fit in the picture.

- There is no requirement that any action profile leads to a different state. Thus, both action profiles $(BT \leftarrow 0, \top_{ST})$ and $(\top_{BT}, \top_{ST})$ in $\mathcal{M}^{BT\leftarrow 0}$ results in a reflexive transition to the same state. But, for example, $(BT \leftarrow 1, \top_{ST})$ will result in the transition to $\mathcal{M}^{BT\leftarrow 1}$ and $(\top_{BT}, ST \leftarrow 0)$ results in the transition to $\mathcal{M}^{BT\leftarrow 0, ST\leftarrow 0}$.
- Different states of such CGS may agree on the valuation on all variables. For example, given a context $\vec{u}$, $(\mathcal{M}, \vec{u})$ and $(\mathcal{M}^{ST\leftarrow 1, BT\leftarrow 1}, \vec{u})$ agree on all $(Y = y)$. But we still treat them as separate states, since these models have different $\mathcal{F}$'s.

Now we can extend $\mathcal{L}(\mathsf{C})$ and allow the nesting of $[\vec{Y} \leftarrow \vec{y}]$ operators.

Definition 6 ($\mathcal{L}(\mathsf{C}_e)$ syntax).

$$\varphi ::= (X = x) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\vec{Y} \leftarrow \vec{y}]\varphi,$$

where X ranges over $\mathcal{V}$, $\vec{Y}$ over $2^{\mathcal{V}}$, x over $\mathcal{R}(X)$ and each y in $\vec{y}$ over $\mathcal{R}(Y)$. We use standard abbreviations for $\top, \bot, \vee$ and $\rightarrow$.

So, now we assume that agents may perform series of updates $[\vec{X} \leftarrow \vec{x}], \ldots, [\vec{Y} \leftarrow \vec{y}]$ in the extended language $\mathcal{L}(\mathsf{C}_e(\mathcal{S}))$. $\mathcal{L}(\mathsf{C}_e(\mathcal{S}))$ formulas can be evaluated by $\vDash_{HP}$ satisfiability relation defined in the same way as in Definition 4.

Proposition 1. *Any $[\vec{X} \leftarrow \vec{x}] \ldots [\vec{Y} \leftarrow \vec{y}]\varphi \in \mathcal{L}(\mathsf{C}_e(\mathcal{S}))$ is equivalent to some $[\vec{X}' \leftarrow \vec{x}', \ldots, \vec{Y}' \leftarrow \vec{y}']\varphi \in \mathcal{L}(\mathsf{C}(\mathcal{S}))$.*

Proof. $[\vec{X} \leftarrow \vec{x}] \ldots [\vec{Y} \leftarrow \vec{y}]$ generates a model $\mathcal{M}^{\vec{X}\leftarrow\vec{x}\cdots^{\vec{Y}\leftarrow\vec{y}}}$ updated multiple times. Our goal is to prove that there exists a model $\mathcal{M}^{\vec{W}\leftarrow\vec{w}}$, such that $\vec{W}$ is a set of variables that occur in $\vec{X}, \ldots, \vec{Y}$ and $\vec{w}$ are the values that occur in $\vec{x}, \ldots, \vec{y}$, such that $\mathcal{M}^{\vec{W}\leftarrow\vec{w}} = \mathcal{M}^{\vec{X}\leftarrow\vec{x}\cdots^{\vec{Y}\leftarrow\vec{y}}}$.

So, let $\vec{W}$ be a set of all variables that occur in $\vec{X}, \ldots, \vec{Y}$. Let $\vec{Z}$ denote a vector $(\vec{X} = \vec{x}, \ldots, \vec{Y} = \vec{y})$. To determine that value of every $W_i \in \vec{W}$ we need to find the right-most $W_i = w_i$ in $\vec{Z}$. So, there is $k \leq |\vec{Z}|$, such that $\vec{Z}[k] = (W_i = w_i)$

(here $\vec{Z}[k]$ denotes the k's element of $\vec{Z}$ of the form $X = x$) and for any $n > k$ and any $w' \in \mathcal{R}(W_i)$ it holds that $\vec{Z}[n] \neq (W_i = w')$. By doing this we enforce that in our model $\mathcal{M}^{\vec{W} \leftarrow \vec{w}}$ all functions $\mathcal{F}_X \in \vec{W}$ are set to constant functions in the exactly same way as they are set in $\mathcal{M}^{\vec{X} \leftarrow \vec{x} \ldots^{\vec{Y} \leftarrow \vec{y}}}$. It guarantees that $(\mathcal{M}, \vec{u}) \vDash_{HP} [\vec{X} \leftarrow \vec{x}] \ldots [\vec{Y} \leftarrow \vec{y}] \varphi$ iff $(\mathcal{M}, \vec{u}) \vDash_{HP} [\vec{W} \leftarrow \vec{w}] \varphi$.

But since it is also clear that every $\varphi \in \mathcal{L}(\mathsf{C})$ is a $\mathcal{L}(\mathsf{C}_e)$ formula, $\mathcal{L}(\mathsf{C})$ and $\mathcal{L}(\mathsf{C}_e)$ are equally expressive. The same result can be seen on CGS's also. For any CGS $\Gamma_{\mathcal{M}}$ obtaned from $\mathcal{M}$, it holds that if some state $q' \in \Gamma_{\mathcal{M}}$ is reachable from initial state q_0, then it is reachable in 1 step.

4   Arbitrary Updates

In this section we demonstrate how the proposed framework can be generalized to allow creating arbitrary dependencies between agents. This is done by allowing interventions that change the functional specifications $\mathcal{F}_X$ to an arbitrary $\mathcal{F}'_X$ for any agent X. It is clear that interventions $[\vec{X} \leftarrow \vec{x}]$ are not the only possible operations modifying $\mathcal{F}$. In other words, there are more ways to update $\mathcal{F}$ instead of replacing some $\mathcal{F}_X$'s with a constant functions. For example, we can allow agents to modify the value of $\mathcal{F}_X(\vec{z})$ on a specific input $\vec{z}$. We denote it as $X(\vec{z}) \leftarrow x$, where $X \in \mathcal{V}, x \in \mathcal{R}(X)$ and $\vec{z}$ is the assignment of all variables in $\mathcal{V}$ except X.

To illustrate it, assume that in the Rock-throwing example we allow Suzy to make an additional action (act^*): to update $\mathcal{F}_{ST}$ in such a way that $\mathcal{F}_{ST}^{act^*}(\vec{z}) = 1$ on all inputs $\vec{z}$ containing $(U_{ST} = 1)$. Now we can generate a new CGS Γ' which contains more possible transitions. The updated CGS is preseinted in Figure 3.
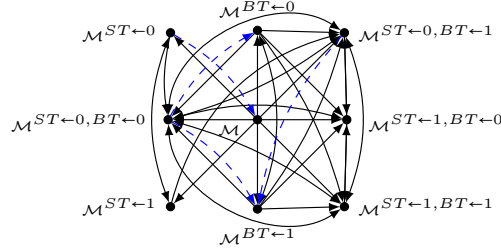

Fig. 3. Extended CGS Γ' for the Rock-throwing example. Dashed blue arrows indicate new transitions.

We see that after intervention $ST \leftarrow 0$ Suzy can always 'return' $\mathcal{F}_{ST}$ to the initial behavior by act^*. So, the blue transitions are the new options. Now, from $\mathcal{M}^{ST \leftarrow 0}$ Billy and Suzy can return to $\mathcal{M}$ if their action profile is $(act^*, \top_{BT})$. Note also that no new states were generated in the extended example. Because

additional action act_1^* for Suzy cannot produce new configuration of $\mathcal{F}_{ST}$ which is different from $\mathcal{F}_{ST}, \mathcal{F}_{ST}^{ST \leftarrow 1}$ or $\mathcal{F}_{ST}^{ST \leftarrow 0}$. But assume that we add another possible action act_2^* which can be expressed as $ST(\mathcal{U}_{ST} = 1) \leftarrow 0$, meaning that $\mathcal{F}_{ST}^{act_2^*}(\vec{z}) = 0$ if $(\mathcal{U}_{ST} = 1) \in \vec{z}$. How should we generate a CGS Γ' now? In this case there will be a possible Suzy's strategy to make $ST \leftarrow 1$ intervention first and then act_2^*. Then, whatever Billy does, $ST = 1$ if $U_{ST} = 0$ and $ST = 0$ if $U_{ST} = 1$. But such model cannot be reached by any strategy if only possible actions for Suzy are interventions $ST \leftarrow 1$ and $ST \leftarrow 0$. To better illustrate the problem, consider another example.

Example 2. Suppose that there are two agents a_1 and a_2 who can give an order to the third agent a_3. There are three alternative decisions a_1 and a_2 may choose: order '1', order '-1' and not to give an order '0'. The only environmental variable P determines the priority of a_1's or a_2's order. Finally, a_3 must choose one of three possible actions: 1, -1 or 0 (to 'wait').

More formally, our variables are $\mathcal{V}_a = \{a_1, a_2, a_3\}, \mathcal{V}_e = \{P\}$. Their ranges are $\mathcal{R}(a_1) = \mathcal{R}(a_2) = \mathcal{R}(a_3) = \{-1, 0, 1\}, \mathcal{R}(P) = \{1, 2\}$. The values of a_1, a_2 and P depend on the context $\vec{u}$, while a_3 depends on all of them. The values for a_3 are determined as follows $\mathcal{F}_{a_3}(\vec{z}) = 1$ if $((P = 1) \in \vec{z}$ and $(a_1 = 1) \in \vec{z})$ or $((P = 2) \in \vec{z}$ and $(a_2 = 1) \in \vec{z})$, $\mathcal{F}_{a_3}(\vec{z}) = 0$ if $((P = 1) \in \vec{z}$ and $(a_1 = 0) \in \vec{z})$ or $((P = 2)$ and $(a_2 = 0))$, $\mathcal{F}_{a_3}(\vec{z}) = -1$ if $((P = 1) \in \vec{z}$ and $(a_1 = -1) \in \vec{z})$ or $((P = 2) \in \vec{z}$ and $(a_2 = -1) \in \vec{z})$. So, agent a_3 checks who has a priority and follows the order.
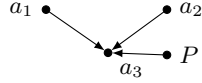
Fig. 4. Dependency graph for Example 2.

Assume that in our context $\vec{u}$, a_1's order has a priority over a_2's according to $\mathcal{F}_P$, so a_3 follows the a_1's order. Decisions of a_1 and a_2 are determined by the context, but each of them can enforce a desirable order by intervention on their variables. So, each of the agents can modify her response to the environment by updating $\mathcal{F}_{a_i}$ (in our case by making it a constant function). Agent a_3 depends on all other variables a_1, a_2 and P. But standard interventions $[X \leftarrow x]$ does not allow a_3 to adjust its behavior while staying dependent on a_1's or a_2's orders. For example, assume that a_3 no longer trusts a_1 and decides to ignore him completely and always follow the a_2's order. This situation is clearly not expressible by standard interventions. But if we extend possible actions of a_3 with any combination of $a_3(\vec{z}) \leftarrow x$, where $x \in \mathcal{R}(a_3)$ and $\vec{z}$ is the assignment of all variables expect a_3, then we can encode much more complex behavior. In particular, let trust_{a_2} be an action encoded as

$$\bigcup_{\vec{z}, s.t.(a_2=1) \in \vec{z}} (a_3(\vec{z}) \leftarrow 1) \cup \bigcup_{\vec{z}', s.t.(a_2=0) \in \vec{z}'} (a_3(\vec{z}') \leftarrow 0) \cup \bigcup_{\vec{z}'', s.t.(a_2=-1) \in \vec{z}''} (a_3(\vec{z}'') \leftarrow -1)$$

This action allows a_3 to modify $\mathcal{F}_{a_3}$ and obtain $\mathcal{F}_{a_3}^{\mathsf{trust}_{a_2}}$. According to this function, agent a_3 always follows the order of a_2 and ignores a_1. We can also imagine that order 1 is very risky for a_3 and in case this agent receives this order from the prioritized agent, he wants to check if second agent also gives this order independently of P's value. This behavior can also be encoded with basic general interventions of the form $X(\vec{z}) \leftarrow x$. Let a_3's action doublecheck be defined as follows

$$\bigcup_{\vec{z}',s.t.(P=1,a_1=1,a_2\neq 1)\in \vec{z}'} (a_3(\vec{z}') \leftarrow 0) \cup \bigcup_{\vec{z}'',s.t.(P=2,a_2=1,a_1\neq 1)\in \vec{z}''} (a_3(\vec{z}'') \leftarrow 0)$$

In this settings, if a_3 receives an order 1 from the prioritized agent, but the order of second agent is not 1, then a_3 decides to wait ($a_3 = 0$) according to $\mathcal{F}_{a_3}^{\mathsf{doublecheck}}$. So, this action will result in one of the updated models $\{\mathcal{M}'_1, \ldots, \mathcal{M}'_l\}$, depending on the actions of other agents. But we know that for any such model $\mathcal{M}'_i$ it holds that $(\mathcal{M}'_i, \vec{u}) \vDash ((a_1 \neq 1) \vee (a_2 \neq 1)) \rightarrow (a_3 \neq 1)$.

We can formalize these generalized interventions as follows.

Definition 7 (Generally updated model). *For any $X \in \mathcal{V}_a$, any assignment $\vec{z}$ of all variables other than X and any $x \in \mathcal{R}(X)$, let $X(\vec{z}) \leftarrow x$ be a generalized intervention that results in the update $\mathcal{F}_X^{X(\vec{z})\leftarrow x}$ of function $\mathcal{F}_X$, such that*

$$\mathcal{F}_X^{X(\vec{z})\leftarrow x}(z') = \begin{cases} x & \text{if } \vec{z}' = \vec{z}, \\ \mathcal{F}_X(\vec{z}') & \text{otherwise}; \end{cases}$$

Let $\vec{X}(\vec{z}) \leftarrow \vec{x}$ denote $X_1(\vec{z}) \leftarrow x_1, \ldots, X_k(\vec{z}') \leftarrow x_k$, where same variable from $\mathcal{V}_a$ can occur multiple times in $X_1, \ldots, X_k$. For any general intervention $\vec{X}(\vec{z}) \leftarrow \vec{x}$, an updated model is a pair $\mathcal{M}^{\vec{X}(\vec{z})\leftarrow \vec{x}} = (\mathcal{S}, \mathcal{F}^{\vec{X}(\vec{z})\leftarrow \vec{x}})$.

The intervention $[X \leftarrow x]$ can be encoded as a set of generalized interventions: $X \leftarrow x \equiv \cup_{\vec{z}} X(\vec{z}) \leftarrow x$. Since $X \leftarrow x$ replaces the value of $\mathcal{F}_X$ for each input $\vec{z}$.

Now we can extend our generalized syntax $\mathcal{L}(\mathsf{C}_g)$ with a new operator:

Definition 8 ($\mathcal{L}(\mathsf{C}_g)$ syntax).

$$\varphi ::= (X = x) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\vec{Y}(\vec{z}) \leftarrow \vec{y}]\varphi,$$

Note that since any variable Y may occur multiple times in $[\vec{Y}(\vec{z}) \leftarrow \vec{y}]$, every agent $i \in \mathbb{AG}$ can modify $\mathcal{F}_i$ in an arbitrary way in $\mathcal{L}(\mathsf{C}_g)$. The satisfiability relation $\vDash_g$ is identical to Definition 4 in all items other than $[\vec{Y}(\vec{z}) \leftarrow \vec{y}]\varphi$, for which it is defined as
$(\mathcal{M}, \vec{u}) \vDash_g [\vec{Y}(\vec{z}) \leftarrow \vec{y}]\varphi$ iff $(\mathcal{M}^{\vec{Y}(\vec{z})\leftarrow \vec{y}}, \vec{u}) \vDash \varphi$.

Now we can generate a CGS for the extended set of operations on models. Note that the set $\{X(\vec{z}) \leftarrow x \mid X \in \mathcal{V}_a \ \& \ \vec{z} \in \times_{Z\in(\mathcal{U}\cup\mathcal{V})\setminus\{X\}} \mathcal{R}(Z) \ \& \ x \in \mathcal{R}(X)\}$ will generate a larger set of actions Act^* for Γ^*. The set of states Q^* in Γ^* will also contain more elements, because now we have more choices to construct updated causal model $\mathcal{M}^{\vec{X}(\vec{z})\leftarrow \vec{x}}$ for any $\vec{X}(\vec{z}) \leftarrow \vec{x}$. In fact, we need to be sure that we will generate every $\mathcal{M}^{\vec{X}(\vec{z})\leftarrow \vec{x}\cdots^{\vec{Y}(\vec{z}')\leftarrow \vec{y}}}$. This is possible because there are

only finitely many such models: there only finitely many possible functions $\mathcal{F}_X$ mapping $\times_{Z\in(\mathcal{U}\cup\mathcal{V}-\{X\})}\mathcal{R}(Z)$ to $\mathcal{R}(X)$. So, we want the set of states $Q^* \in \Gamma^*$ to contain a model $\mathcal{M}'$ for any possible updated functions $\mathcal{F}'_{X_1},\ldots,\mathcal{F}'_{X_k}$ for $\mathcal{V}_a = \{X_1,\ldots,X_k\}$. But as we show in Proposition 2, the set of all $\mathcal{M}^{\vec{X}(\vec{z})\leftarrow\vec{x}}$'s is equal to the set of all $\mathcal{M}^{\vec{X}(\vec{z})\leftarrow\vec{x}\cdots^{\vec{Y}(\vec{z'})\leftarrow\vec{y}}}$'s. Next, Π^*, π^* and d^* are defined as before. We say that $o(\mathcal{M}',\vec{X}(\vec{z})\leftarrow\vec{x}) = \mathcal{M}''$ iff $\mathcal{M}'' = (\mathcal{M}')^{\vec{X}(\vec{z})\leftarrow\vec{x}}$. Thus, given a causal model $\mathcal{M} = (\mathcal{S},\mathcal{F})$ and a context $\vec{u}$ we can generate a general CGS $\Gamma^*_\mathcal{M}$ as follows

- $\mathbb{AG} = \mathcal{V}_a$;
- $Q^* = \{\mathcal{M}^{\vec{X}(\vec{z})\leftarrow\vec{x}} \mid \vec{X} \subseteq \mathcal{V}_a \ \& \ \vec{x} \in \times\mathcal{R}(\vec{X}) \ \& \ \vec{z} \in \times_{Y\in\mathcal{U}\cup\mathcal{V}}\mathcal{R}(Y)\}$;
- $\Pi^* = \{Y = y \mid Y \in \mathcal{V} \ \& \ y \in \mathcal{R}(Y)\}$;
- π^* is defined as $(Y = y) \in \pi(\mathcal{M}')$ iff $(\mathcal{M}',\vec{u}) \vDash' (Y = y)$ for any $\mathcal{M}' \in Q$;
- $Act^* = \{X(\vec{z}) \leftarrow x \mid X \in \mathcal{V}_a \ \& \ \vec{z} \in \times_{Z\in(\mathcal{U}\cup\mathcal{V})\backslash\{X\}}\mathcal{R}(Z) \ \& \ x \in \mathcal{R}(X)\} \cup \{\top_X \mid X \in \mathcal{V}_a\}$, where $\top_X$ denotes 'no intervention on X';
- $d^*(X,\mathcal{M}') \subseteq \{X(\vec{z}) \leftarrow x \mid x \in \mathcal{R}(X), \vec{z} \in \times_{Z\in(\mathcal{U}\cup\mathcal{V})\backslash\{X\}}\mathcal{R}(Z)\}$ for any $X \in \mathcal{V}_a$ and $\mathcal{M}' \in Q$;
- $o^*(\mathcal{M}',\vec{X}(\vec{z}) \leftarrow \vec{x}) = \mathcal{M}''$ iff $\mathcal{M}'' = \mathcal{M}'^{\vec{X}(\vec{z})\leftarrow\vec{x}}$ for any $\mathcal{M}',\mathcal{M}'' \in Q^*$;

This general CGS differs from our previous construction, because the set of general interventions $X(\vec{z}) \leftarrow x$ generates a different set of actions Act^* and a set of possible states Q^* comparing to standard interventions $X \leftarrow x$. Now we can establish the result similar to Proposition 1.

Proposition 2. *For any $\mathcal{L}(\mathsf{C}_g)$ formula of the form $[\vec{X}(\vec{z}) \leftarrow \vec{x}]\ldots[\vec{Y}(\vec{z'}) \leftarrow \vec{y}]\varphi$ there exists an equivalent formula of the form $[X'(\vec{z_i}) \leftarrow x',\ldots,Y'(\vec{z_j}) \leftarrow y']\varphi$.*

Proof. Let $\vec{Z}$ be a vector $(\vec{X}(\vec{z}) \leftarrow \vec{x},\ldots,\vec{Y}(\vec{z'}) \leftarrow \vec{y})$. So, each element of $\vec{Z}$ is a basic intervention of the form $Y(\vec{z}) \leftarrow y$. We denote k's element of $\vec{Z}$ as $\vec{Z}[k]$. Let W be a set of all pairs $(Y,\vec{z})$ for which $Y(\vec{z}) \leftarrow y$ occurs in $\vec{Z}$. So, there is $k \leq |\vec{Z}|$, such that $\vec{Z}[k] = (Y(\vec{z}) \leftarrow y)$ and for any $n > k$ and any $y' \neq y$ it holds that $\vec{Z}[n] \neq (Y(\vec{z}) \leftarrow y')$. Let $\vec{w}$ be vector of such values for all elements of $\vec{W}$. Then, the resulting models $\mathcal{M}^{(\vec{X}(\vec{z})\leftarrow\vec{x}\cdots^{\vec{Y}(\vec{z'})\leftarrow\vec{y}}}$ and $\mathcal{M}^{\vec{W}\leftarrow\vec{w}}$ are equivalent. So, it holds that $(\mathcal{M}^{(\vec{X}(\vec{z})\leftarrow\vec{x}\cdots^{\vec{Y}(\vec{z'})\leftarrow\vec{y}}},\vec{u}) \vDash_g \varphi$ iff $(\mathcal{M}^{\vec{W}\leftarrow\vec{w}},\vec{u}) \vDash_g \varphi$.

This proposition in particularly implies, that for any two states $q,q' \in Q^*$, if q' is reachable from q by some series of updates $[\vec{X}(\vec{z}) \leftarrow \vec{x}]\ldots[\vec{Y}(\vec{z}) \leftarrow \vec{y}]$, then q' is reachable from q' in one step by some update $[X'(\vec{z}) \leftarrow x,\ldots,Y'(\vec{z}) \leftarrow y']$.

There are of course different ways to introduce additional restrictions on the set of available actions d^*. And there may be different motivation for these restrictions. Firstly, it seems reasonable to require that if some variable $X \in \mathcal{V}_a$ is independent of some other variable $Y \in \mathcal{V}$ in the initial model, then it must remain so for any updated model. Note that the contrary does not hold: if X depends on Y it may become independent of it even after standard intervention

$X \leftarrow x$ since $\mathcal{F}_X^{X \leftarrow x}$ becomes a constant function. But this restriction does not look universal: it is easy to imagine that in some situations agent may decide to take into account some information he previously ignored. Secondly, it seems important to allow agents to rewrite the changes in their $\mathcal{F}_i$'s back. Formally, assume that in some state q, the i's function is defined as $\mathcal{F}_i^1$ and i has a strategy str_i, such that for any $\lambda \in plays(q, str_i)$ it holds that in all states $q' = \lambda[1]$ i's function is defined as $\mathcal{F}_i^2$. Then, agent must have a strategy str_i^*, such that for any $\lambda \in plays(q', str_i^*)$ it holds that the $\mathcal{F}_i$ in $\lambda[1]$ is defined as $\mathcal{F}_i^1$. So, i can return $\mathcal{F}_i$ to its initial configuration after any change. This restriction sounds reasonable for multi-agent systems, yet it does not generally hold if the possible actions for agents are standard interventions $[\vec{X}_{\mathbb{AG}} \leftarrow \vec{x}]$ described in Section 3. Finally, some actions (updates) can turn a recursive model into a non-recursive one. So, the choice of adequate restrictions remains an important issue.

Even though we introduced $\mathcal{L}(\mathsf{C}_e)$ and $\mathcal{L}(\mathsf{C}_g)$ to reason about sequences of updates performed by agents as their strategies, essentially we worked with one-shot games, because everything was reachable in one step in the corresponding CGS as shown in Propositions 1 and 2. But this may not be the case depending on the additional restrictions on the set of available actions (interventions) $d \in \Gamma$. But these restrictions go beyond the scope of this paper.

5 Uncertainty and Responsibility

Reasoning about strategic abilities often includes reasoning about agents' uncertainty [8]. For example, an agent may be unaware of other agents' choices or about some fact of the world.

Note that previously we generated a CGS for a fixed context $\vec{u}$. But now we assume that the actual context may be unknown to the agent. So, we want to model uncertainty over the pairs $(\mathcal{M}, \vec{u})$, which is a standard assumption for modelling uncertainty in causal models [5, 10]. Basically, we want to generate a state for any possible pair $(\mathcal{M}^*, \vec{u}')$. Formally, given a causal model $\mathcal{M}$ we want to generate a CGS $\Gamma_{(\mathcal{M}, \vec{u})}$ for every context $\vec{u}$. Then, let Γ be a disjoint union of all $\Gamma_{(\mathcal{M}, \vec{u})}$. In other words, $Q^\Gamma = \{(\mathcal{M}^{\vec{X}(\vec{z}) \leftarrow \vec{x}}, \vec{u}) \mid \vec{X} \subseteq \mathcal{V}_a \ \& \ \vec{x} \in \times \mathcal{R}(\vec{X}) \ \& \ \vec{z} \in \times_{Y \in \mathcal{U} \cup \mathcal{V}} \mathcal{R}(Y) \ \& \ \vec{u} \in \times_{U \in \mathcal{U}} \mathcal{R}(U)\}$.

We say that an *Epistemic* CGS (ECGS) $\Gamma = (\mathbb{AG}, Q, \{\sim_i\}_{i \in \mathbb{AG}}, \Pi, \pi, Act, d, o)$ is a CGS extended with an epistemic relations $\sim_i \subseteq Q \times Q$ for each $i \in \mathbb{AG}$, such that all $\sim_i$'s are equivalence relations. To obtain ECGS Γ^*, we need to extend Γ with these epistemic relations $\sim_i$. We assume that they are already given.

To illustrate the role of knowledge and uncertainty, return to Example 2 again. We want to model a situation when a_3 observes only his own actions and does not know what actions other agents make. Assume also that a_3 does not know the context $\vec{u}$, i.e. the assignment of exogenous variables $\mathcal{U}$ which determine the values of a_1, a_2, P. Figure 5 represents this epistemic state for a_3.

Here a_3 can choose three available actions: not to modify $\mathcal{F}_{a_3}$ (denoted as $\top$), to follow a_2's decision (trust_{a_2}) or to double-check order '1' ($\mathsf{doublecheck}$). If a_3 decides not to modify $\mathcal{F}_{a_3}$, then he knows that he is in one of the states
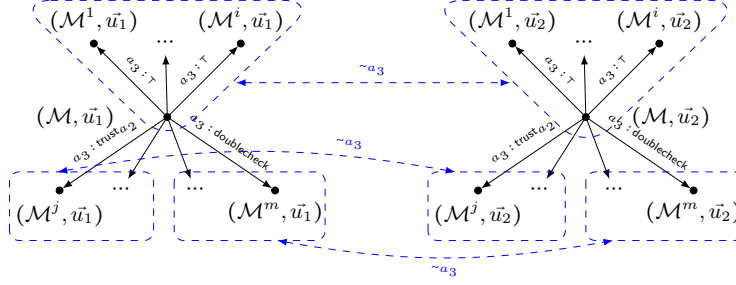
Fig. 5. Epistemic scenario with a_3's uncertainty for Example 2. Note that only two contexts $\vec{u_1}, \vec{u_2}$ are included in the picture, but in general case there can be any possible context $\vec{u'}$. The labels on the transitions demonstrates a_3's action, while other agents' decisions are omitted.

form $(\mathcal{M}, \vec{u_1}), (\mathcal{M}^1, \vec{u_1}), \ldots (\mathcal{M}^i, \vec{u_1})$ or $(\mathcal{M}, \vec{u_2}), (\mathcal{M}^1, \vec{u_2}), \ldots (\mathcal{M}^i, \vec{u_2})$. So, in this epistemic state the agent does not know what the actual context is as well as what the decisions of a_1 and a_2 are, i.e. how they react to the context $\vec{u}$. But a_3 still knows that $(P = 1 \wedge a_1 = 0) \rightarrow (a_3 = 0)$, $(P = 2 \wedge a_2 = 1) \rightarrow (a_3 = 1)$, etc. In other words, even though a_3 does not know what configuration of the environment is (will be) and how other agents (will) react on it, he still knows his own response to any possible situation (because the choice is up to him).

Syntactically, we can extend any of the previously mentioned languages with knowledge operators K_i, where the formula $K_i\varphi$ reads as 'agent i knows φ'. The standard semantics of this operator is defined as

$$(\Gamma, q) \vDash K_i\varphi \text{ iff } \forall q' \in Q, q \sim_i q' \Rightarrow (\Gamma, q') \vDash \varphi$$

Being able to model agents' strategic abilities and uncertainty, we can define such notions as strategic responsibility (or blameworthiness) in the proposed framework.

5.1   Expressing Strategic Responsibility

There are a number of approaches dealing with notions of responsibility and blameworthiness proposed in a literature on causal models [5, 13, 10, 2] as well as for CGS semantics [4, 19, 22]. The various definitions differ in details, but the main idea is that the group of agents G is responsible for some outcome φ if G could prevent φ independently of their epistemic state. For blameworthiness it is usually required that G had a knowledge *how to* (and hence *could*) prevent φ. Though this distinction is useful in many settings, in this section we discuss the notion of strategic responsibility, which takes into account both strategic ability and epistemic state.

Another important criteria for the definition of responsibility is a minimality condition. We want to claim that the group G is responsible for φ only if G is the minimal coalition that could prevent φ. Without this condition, responsibility

would always be distributed to super-groups, so the grand coalition $\mathbb{AG}$ would be responsible for φ whenever a sub-group $G \subset \mathbb{AG}$ is. Note that there can be multiple minimal coalitions responsible for the same φ.

Finally, some approaches (e.g. [5, 13, 10, 2]) deal with a notion of a *degree* of responsibility (or blameworthiness). In these settings, if the group G is responsible (blameworthy) for some outcome φ, then this responsibility (blameworthiness) can be shared and distributed over individual members of G. In this paper we do not discuss the degree of responsibility and assume that the group responsibility is not distributed to the individual members of the group. But, of course, additional procedure for such distribution of responsibility can be defined as an extension. So, in our framework it is the case that if a group G is responsible for φ, then for all $i \in G$ it holds that i is not responsible for φ. If it does not hold and some $i \in G$ is responsible for φ, then G does not satisfy the minimality condition, which contradicts our initial assumption. This property may look counterintuitive, but it guarantees that agents are not considered responsible for φ until they have no strategic power to prevent it (given their uncertainty).

Before we provide a formal definition, we need to introduce the notion of a uniform strategy. Formally, a strategy str_a for agent $a \in \mathbb{AG}$ is called *uniform* if for any states $q, q' \in Q$, such that $q \sim_a q'$, it holds that $str_a(q) = str_a(q')$. A coalition strategy str_G is uniform if it is uniform for every $a \in G$. As we said before any $q \in Q$ is reachable from q_0 in one step. So, it is sufficient to check the strategic ability of agents in the initial state q_0. Let φ be a Boolean combination of basic formulas of the form $Y = y$ for $Y \in \mathcal{V}, y \in \mathcal{R}(Y)$.

Definition 9 (Strategic Responsibility). *A group G is strategically responsible for φ in (Γ, q) if the following three conditions hold:*
1. $\Gamma, q \vDash \varphi$;
2. There is a uniform strategy str_G for G, such that for all q', s.t. $q_0 \sim_G q'$ and for all $\lambda \in plays(q', str_G)$ it holds that $\Gamma, \lambda[1] \vDash \neg\varphi$;
3. No proper subset of G satisfies (2),

Using this definition we can better illustrate the role of general interventions proposed in Section 4.

Example 3. Imagine a simple model with two agents i and j. Let j depend on i's decision, so $\mathcal{F}_j(\vec{z}) = x$ iff $(i = x) \in \vec{z}$. Variable O (outcome) depends on both i and j as follows: $O = 1$ if $i \neq j$ and $O = 0$ otherwise. All variables are binary: $\mathcal{R}(i) = \mathcal{R}(j) = \mathcal{R}(O) = \{0, 1\}$. Assume also that j is uncertain about the actual context as well as about i's actions.

Clearly, agent i cannot prevent $O = 0$ in this settings, so, i is not responsible for $O = 0$. But for agent j the situation is more complicated. If the set of available actions for j is defined by the interventions $[\vec{X} \leftarrow \vec{x}]$ from $\mathcal{L}(\mathsf{C})$ language, then j has an option to guess i's decision and make an intervention $j \leftarrow x$, where $x = \mathcal{F}_i(\vec{u})$. But until we assume that j is unaware of i's decision and/or the context $\vec{u}$, then Definition 9 does not identify j being responsible for $O = 0$. According to our definition the group $\{i, j\}$ is the minimal coalition that can prevent $O = 0$
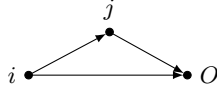
Fig. 6. Dependency graph for Example 3.

given the uncertainty (by choosing either $(i \leftarrow 1, j \leftarrow 0)$ or $(i \leftarrow 0, j \leftarrow 1)$). But if we allow the generalized interventions $[\vec{X}(\vec{z}) \leftarrow \vec{x}]$ from $\mathcal{L}(\mathsf{C}_g)$ to form the set of available actions Act, then $\{i, j\}$ is no longer the minimal coalition that can enforce $O = 1$. Now agent j has available action

$$\mathsf{not}_i := \bigcup_{(i=0) \in \vec{z}} (j(\vec{z}) \leftarrow 1) \cup \bigcup_{(i=1) \in \vec{z}} (j(\vec{z}) \leftarrow 0)$$

Now j can enforce the fact that his decision is opposed to that of i in any context. Thus, action not_i for j can prevent $O = 0$ in his epistemic state and hence j is strategically responsible according to Definition 9. So, the distinction between standard $\vec{X} \leftarrow \vec{x}$ and proposed $\vec{X}(\vec{z}) \leftarrow \vec{x}$ interventions is important for reasoning about responsibility.

6 Discussion

In this paper we demonstrate how causal models can be used for modeling multi-agent interaction in organizational structures, where decisions of agents may depend on other agents as well as the environment. Such causal models provide us a tool for specification of the behaviour of the agents and the changes of the environment. Moreover, these models contain additional counterfactual information. So, they describe the behaviour of agents and the environment not only for the actual context, but also for any counterfactual scenario. Then we demonstrate how to reason about updates (interventions) of such models in terms of concurrent game structures. In such CGS, agents can choose to modify their reaction on the environment and other agents' decisions by updating their part of a causal model. Then we discuss how the notion of intervention on a causal model can be generalized for reasoning about more complex behavior. Finally, we demonstrate how strategic responsibility can be defined in our settings. We believe that the proposed framework can be useful for reasoning about multi-agent systems.

However, there are still many open questions left for future work. Firstly, as we mentioned before, different restrictions of the set of available actions for agents require closer study. The choice of these restrictions affects the strategic power of the agents and thus determines what these agents can achieve, which may obviously affect responsibility statements. Secondly, we represent the transformations of a causal model in terms of standardly defined CGS, which allows us to deploy a well-known machinery developed in the field of multi-agent systems for reasoning about such structures. The obvious examples of such machinery are widely used logics dealing with strategic power, such as Coalition logic CL [20], Alternating-time temporal logic ATL [7] and Strategy logic SL [18].

References

1. Ahmady, G.A., Mehrpour, M., Nikooravesh, A.: Organizational structure. Procedia - Social and Behavioral Sciences **230**, 455–462 (2016)
2. Alechina, N., Halpern, J.Y., Logan, B.: Causality, responsibility and blame in team plans. In: Das, S., Durfee, E., Larson, K., Winikoff, M. (eds.) Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems AAMAS '17 (2017)
3. Boudou, J., Lorini, E.: Concurrent game structures for temporal STIT logic. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. p. 381–389. AAMAS '18, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2018)
4. Bulling, N., Dastani, M.: Coalitional responsibility in strategic settings. In: Leite, J., Son, T.C., Torroni, P., van der Torre, L., Woltran, S. (eds.) Computational Logic in Multi-Agent Systems. pp. 172–189. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
5. Chockler, H., Halpern, J.Y.: Responsibility and blame: A structural-model approach. Journal of Artificial Intelligence Research **22**, 93–115 (2004)
6. Dastani, M., van der Torre, L.W.N., Yorke-Smith, N.: Commitments and interaction norms in organisations. Autonomous Agents and Multi Agent Systems **31**(2), 207–249 (2017)
7. Demri, S., Goranko, V., Lange, M.: Temporal Logics in Computer Science: Finite-State Systems. Cambridge University Press, Cambridge (2016)
8. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.: Reasoning about Knowledge. MIT Press, Cambridge, Massachusetts (1995)
9. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: An organizational view of multi-agent systems. In: Giorgini, P., Müller, J.P., Odell, J. (eds.) Agent-Oriented Software Engineering IV. pp. 214–230. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
10. Friedenberg, M., Halpern, J.Y.: Blameworthiness in multi-agent settings. Proceedings of the AAAI Conference on Artificial Intelligence **33**, 525–532 (2019)
11. Halpern, J.Y.: A modification of the Halpern-Pearl definition of causality. In: Proc. 24th International Joint Conference on Artificial Intelligence (IJCAI 2015). pp. 3022–3033 (2015)
12. Halpern, J.Y.: Axiomatizing causal reasoning. Journal of Artificial Intelligence Research **12**, 317–337 (5 2000)
13. Halpern, J.Y.: Actual Causality. The MIT Press (2016)
14. Halpern, J.Y., Pearl, J.: Causes and explanations: A structural-model approach. part i: Causes. The British Journal for the Philosophy of Science **56**(4), 843–887 (2005)
15. Hübner, J.F., Boissier, O., Kitio, R., Ricci, A.: Instrumenting multi-agent organisations with organisational artifacts and agents. Autonomous Agents and Multi Agent Systems **20**, 369–400 (2010)
16. Khan, S.M., Lespérance, Y.: Knowing why — on the dynamics of knowledge about actual causes in the situation calculus. In: Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems. p. 701–709. AAMAS '21, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2021)
17. Lewis, D.: Causation as influence. Journal of Philosophy **XCVII**, 182–197 (2000)

18. Mogavero, F., Murano, A., Perelli, G., Vardi, M.Y.: Reasoning about strategies: On the model-checking problem. ACM Trans. Comput. Logic **15**(4) (nov 2014)
19. Naumov, P., Tao, J.: An epistemic logic of blameworthiness. Artificial Intelligence **283**, 103269 (2020)
20. Pauly, M.: A Modal Logic for Coalitional Power in Games. Journal of Logic and Computation **12**(1), 149–166 (02 2002)
21. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press (2000)
22. Yazdanpanah, V., Dastani, M., Jamroga, W., Alechina, N., Logan, B.: Strategic responsibility under imperfect information. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. p. 592–600. AAMAS '19, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2019)