Dynamics of Causal Dependencies in Multi-agent Settings

Maksim Gladyshev, Natasha Alechina, Mehdi Dastani, Dragan Doder

Department of Information and Computing Sciences
Utrecht University, Netherlands

June 9, 2023

Motivation

Causal models [Pearl, 2000, Halpern and Pearl, 2005, Halpern, 2016] provide us a good tool for reasoning about causal dependencies.

Motivation

Causal models [Pearl, 2000, Halpern and Pearl, 2005, Halpern, 2016] provide us a good tool for reasoning about causal dependencies.
While multi-agent interaction is usually modelled via Action semantics, i.e. Concurrent Game Structures.

Motivation

Causal models [Pearl, 2000, Halpern and Pearl, 2005, Halpern, 2016] provide us a good tool for reasoning about causal dependencies.

While multi-agent interaction is usually modelled via Action semantics, i.e. Concurrent Game Structures.

We'd like to use causal models for modeling multi-agent interaction in complex organizational settings, where agents' decisions may depend on other agents' decisions as well as the environment.

Motivation

Causal models [Pearl, 2000, Halpern and Pearl, 2005, Halpern, 2016] provide us a good tool for reasoning about causal dependencies.

While multi-agent interaction is usually modelled via Action semantics, i.e. Concurrent Game Structures.

We'd like to use causal models for modeling multi-agent interaction in complex organizational settings, where agents' decisions may depend on other agents' decisions as well as the environment.

Preliminaries: Causal Models

Definition (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 $\vec{u}$.

Preliminaries: Causal Models

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.

Preliminaries: Causal Models

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.

In this work, we deal with recursive models only. Intuitively, a causal model $\mathcal{M}$ is *recursive*, if for any context there is a unique solution of the equations in $\mathcal{M}$.

Preliminaries: Causal Models

Example (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.

Preliminaries: Causal Models

Example (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.

Here 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 $Y \in (\mathcal{U} \cup \mathcal{V})$, $\mathcal{R}(Y) = \{0, 1\}$. Structural equations are defined as follows:

- SH=ST;
- BH=(BT$\wedge\neg$SH);
- BS=(SH$\vee$BH).

Preliminaries: Dependency graphs

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.

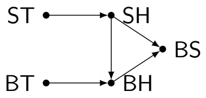


Figure: A dependency graph for the Rock-throwing example.

Preliminaries: Interventions

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 (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.

Definition (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 $\mathcal{L}(\mathrm{C}(\mathcal{S}))$ 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 $\mathcal{L}(\mathrm{C})$ can be evaluated on a causal settings $(\mathcal{M}, \vec{u})$ as follows:

Definition (Semantics)

Given a causal settings $(\mathcal{M}, \vec{u})$, and $\mathcal{L}(\mathrm{C})$ formula φ we define $\models_{HP}$ relation inductively as follows:
$(\mathcal{M}, \vec{u}) \models_{HP} (X = x)$ iff $(X = x)$ in the unique solution of equations in $\mathcal{M}$ for a context $\vec{u}$,
$(\mathcal{M}, \vec{u}) \models_{HP} \neg\varphi$ iff $(\mathcal{M}, \vec{u}) \not\models_{HP} \varphi$,
$(\mathcal{M}, \vec{u}) \models_{HP} (\varphi \wedge \psi)$ iff $(\mathcal{M}, \vec{u}) \models_{HP} \varphi$ and $(\mathcal{M}, \vec{u}) \models_{HP} \psi$,
$(\mathcal{M}, \vec{u}) \models_{HP} [\vec{Y} \leftarrow \vec{y}]\varphi$ iff $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \vec{u}) \models_{HP} \varphi$.

Preliminaries: Action Semantics

Definition (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 $1 \leq i \leq k$, 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.

Preliminaries: Action Semantics

Definition (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 $1 \leq i \leq k$, 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.

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 the environment in causal models.

Translation of Causal Model into CGS

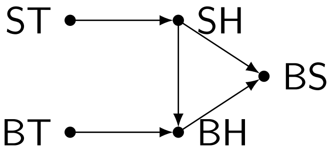
We assume $\mathcal{V} = \mathcal{V}_a \cup \mathcal{V}_e$, where $\mathcal{V}_a$ is the set of agent variables and $\mathcal{V}_e$ is the disjoint set of environment variables. A causal model $\mathcal{M} = (\mathcal{S}, \mathcal{F})$, given a context $\vec{u}$, is translated to a CGS $\Gamma_{\mathcal{M}}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

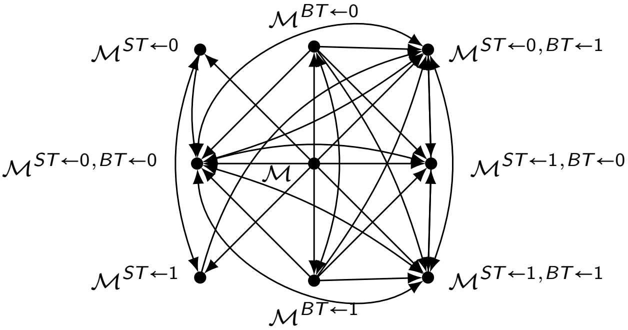
Translation of Causal Model into CGS

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. 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

- $\mathbb{AG} = \mathcal{V}_a$;
- $Q = \{\mathcal{M}^{\vec{X} \leftarrow \vec{x}} \mid \vec{X} \subseteq \mathcal{V}_a \;\&\; \vec{x} \in \times\mathcal{R}(\vec{X})\}$;
- $\Pi = \{Y = y \mid Y \in \mathcal{V} \;\&\; y \in \mathcal{R}(Y)\}$;
- π is defined as $(Y = y) \in \pi(\mathcal{M}')$ iff $(\mathcal{M}', \vec{u}) \models_{HP} (Y = y)$ 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 $\mathcal{M}_1^{Act_{\vec{X}}} = \mathcal{M}_2$.

CGS for the Rock-throwing Example





Problems with Interventions

Note that such an intervention (updates) $\vec{X} \leftarrow \vec{x}$ make the agents in $\vec{X}$ independent of other agents as their decision-making functional specifications are now reduced to a constant function.

Problems with Interventions

Note that such an intervention (updates) $\vec{X} \leftarrow \vec{x}$ make the agents in $\vec{X}$ independent of other agents as their decision-making functional specifications are now reduced to a constant function.

It is also 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.

Arbitrary updates

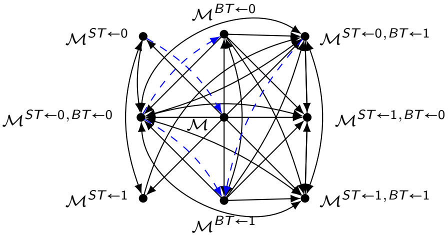
Definition (Generally updated model)

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 $\mathcal{M}^{\vec{X}(\vec{z})\leftarrow\vec{x}} = (\mathcal{S}, \mathcal{F}^{\vec{X}(\vec{z})\leftarrow\vec{x}})$.

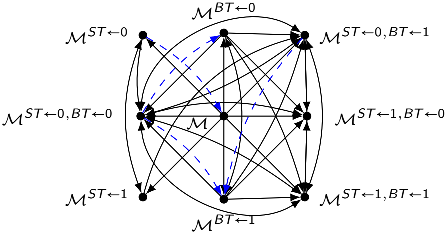
Arbitrary updates

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.



Arbitrary updates

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.

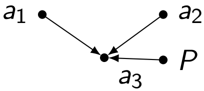


Now, Suzy can return to the initial state after $[ST \leftarrow 0]$ intervention.

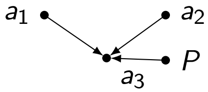
Arbitrary updates

Example

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.

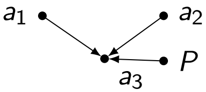


Arbitrary updates



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.

Arbitrary updates



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

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

Arbitrary updates

For extended set of operations on models we can generate a new 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 $(\mathcal{M}', \vec{u}) \models' (Y = y)$ for any $\mathcal{M}' \in Q$;
- $Act^* = \{X(\vec{z}) \leftarrow x \mid X \in \mathcal{V}_a \ \& \ \vec{z} \in \times_{Z\in(\mathcal{U}\cup\mathcal{V})\setminus\{X\}}\mathcal{R}(Z) \ \& \ x \in \mathcal{R}(X)\} \cup \{\top_X \mid X \in \mathcal{V}_a\}$, where $\top_X$ denotes 'no intervention on X';
- $d^*(X, \mathcal{M}') \subseteq \{X(\vec{z}) \leftarrow x \mid x \in \mathcal{R}(X), \vec{z} \in \times_{Z\in(\mathcal{U}\cup\mathcal{V})\setminus\{X\}}\mathcal{R}(Z)\}$ 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 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 $X \leftarrow x$.

Future Work

The presented translation into CGS allows us to deploy a well-studied machinery of ATL- or SL- style logics for reasoning about agents' choices of organisational structure (and their decision making policy).

Future Work

The presented translation into CGS allows us to deploy a well-studied machinery of ATL- or SL- style logics for reasoning about agents' choices of organisational structure (and their decision making policy).

It also allows us to reason about strategic responsibility or blameworthiness [Yazdanpanah et al., 2019], [Friedenberg and Halpern, 2019], [Alechina et al., 2017] with respect to the choices of organisational structures.

Future Work

The presented translation into CGS allows us to deploy a well-studied machinery of ATL- or SL- style logics for reasoning about agents' choices of organisational structure (and their decision making policy).

It also allows us to reason about strategic responsibility or blameworthiness [Yazdanpanah et al., 2019], [Friedenberg and Halpern, 2019], [Alechina et al., 2017] with respect to the choices of organisational structures.

Various 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.

References I

Alechina, N., Halpern, J. Y., and Logan, B. (2017).
Causality, responsibility and blame in team plans.
In Das, S., Durfee, E., Larson, K., and Winikoff, M., editors, *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems AAMAS '17*.

Friedenberg, M. and Halpern, J. Y. (2019).
Blameworthiness in multi-agent settings.
Proceedings of the AAAI Conference on Artificial Intelligence, 33:525–532.

Halpern, J. Y. (2016).
Actual Causality.
The MIT Press.

References II

Halpern, J. Y. and Pearl, J. (2005).
Causes and explanations: A structural-model approach. part i: Causes.
The British Journal for the Philosophy of Science, 56(4):843–887.

Pearl, J. (2000).
Causality: Models, Reasoning, and Inference.
Cambridge University Press.

Yazdanpanah, V., Dastani, M., Jamroga, W., Alechina, N., and Logan, B. (2019).
Strategic responsibility under imperfect information.
In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 592–600, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

The End