# A Transition System for Causality and Strategic Responsibility

**Sylvia Kerkhove (1154990)**

A thesis for the master Artificial Intelligence, Utrecht University

November 10, 2023

Supervisor: Prof. dr. M.M. (Mehdi) Dastani
Second Examiner: Dr. N.A. (Natasha) Alechina

## Abstract

Causality plays an important role in many day-to-day processes and humans reason about causality all the time. There has however not been much research on causality in (multi-)agent systems. In this work I introduce a way to integrate a structural causal model in multi-agent system models, specifically in a concurrent game structure (CGS). In such a causal CGS, every transition corresponds to an intervention on agent variables of the causal model. The Halpern and Pearl framework of causality is used to determine the effects of a certain value for an agent variable on other variables. This causal CGS allows us to analyse and reason about causal effects of agents' actions on each other and their shared environment in multi-agent settings. In this work I study and analyse the relation between the derived-multi-agent system model, this causal CGS, and the original structural causal model and I analyse the causal CGS to show a relation between strategic responsibility and causality. This work first gives an overview of the literature on causality, labelled transition systems and concurrent game structures, several temporal logics and several notions of responsibility. After that it defines the notion of a causal CGS, and I show how agent strategies in this CGS relate to causality in the original causal model and show a relation between strategic responsibility and causality in the HP framework. This work can in the future be used for (multi-)agent systems where causal relations play an important role. It can be used to plan strategies or to determine causation and responsibility when things go wrong.

# Contents

# 1 Introduction

In his book Actual Causality, Halpern claims that determining causality is crucial in the attribution of responsibility for an outcome [8]. He uses a definition of responsibility directly building upon the definition of causality [5]. Therefore, in that case causality is clearly crucial when attributing responsibility, but this is not true for all definitions of responsibility in multi-agent systems. Other works define responsibility using agent strategies [16, 2]. In this thesis I am investigating the relation between actual causality and strategic responsibility. Before we can do this, we need to look at how both notions compare.

Agent strategies are usually discussed in the context of labelled transition systems (LTS), or more general, concurrent game structures (CGS), which generalise LTS to multi-agent systems [1]. LTS are represented as graphs, where every node corresponds to a state and every edge to a transition. The edges are labelled, usually with actions that bring the transition about [7, 4]. A logic for reasoning about transition systems is linear-time temporal logic (LTL), which is a modal logic that extends propositional logic with a modal operator for 'next', which denotes that something will be true after the current state, and a modal operator for 'until', which denotes that something will be true as long as something else is not true [4]. However, other temporal logics like computation tree logic [4, 11] and alternating-time temporal logic [1, 11] are also used.

A model for actual causality can also be represented as a graph, called a causal network [8]. In such a causal network, the nodes correspond to variables and edges indicate a causal relation between them. Causal models have two different types of variables, exogenous and endogenous variables. The former are variables whose values are determined by causes outside of the model, the latter's values are determined by the variables in the model. The value of an endogenous variable is determined by the other variables of the model, as specified by a structural equation [8]. There is not a single formalism for reasoning about causal models, but some people have suggested branching-time logics [13]. These logics usually have operators for things that will always be true in the future and things that can be true in the future.

The main difference between causal networks and labelled transition systems is that nodes in transition systems represent states of the environment, with the edges representing events that change this state, while in a causal model, the nodes represent variables that can have a certain value in an environment, with the edges representing the causal effect the variables have on one another.

This work will focus on the use of transitions systems and causal models for (multi-)agent systems. One can imagine that actions of one agent in a multi-agent system can have a causal effect on the system as a whole, and other agents in particular. For example, if one agent locks a door, a second agent cannot go through it anymore. The first agent caused the second agent to be unable to go through the door.

When defining some causal variables as agent variables, variables that are directly influenced by an agent, causal model could be used to specify how agents' actions influence the environment. I want to research how a transition system for a (multi-)agent system can be derived from such a causal model. We could imagine that when doing that, we describe the states of the transition system with respect to the variables of the causal model. Actions changing the state could then be interventions that explicitly change one or more variables which leads to other changes of variables according to the causal model, which then leads to a new state consistent with the underlying causal model. I will then look at what such a model can say about causality, strategic responsibility and the relation between the two.

In order to achieve this, in the next section, I will look at Halpern and Pearl's definition of causality [9] and in particular, Halpern's book on causal models and causality [8]. I will also look at several definitions of labelled transition systems, including their original definition by Keller [12] and consider the different temporal logics mentioned above that can be used to reason about these systems. Finally I will discuss several notions of responsibility as discussed by Chockler and Halpern [5], Baier et al. [2] and Yazdanpanah et al. [16]. With this solid theoretical foundation, I will continue with defining a CGS based on a structural causal model in Section 4. In Section 5, I will then show some results on how agent strategies in this CGS relate to causality. Finally, I will shortly relate this to the notion of strategic responsibility in Section 6 and discuss limitations and future work in Section 7.

# 2 Literature Overview

In this section I will first discuss causality, including causal models and a formal definition. Then I will discuss labelled transition systems, including concurrent game structures, their generalisation for multi-agent systems. I will continue with talking about linear temporal logic and finally, I will discuss different notions of responsibility.

## 2.1 Notation

A short overview of the notation I will use throughout this work:

Apart from a few exceptions, sets will be denoted by a capital letter, e.g. $A, B$ or $Q$. Variables will also often be denoted with capital letters. In situations where this would make things unclear, following the practice in [8], I will abuse notation a little and use a vector notation $\vec{X}$ for the set, so that the variables of $\vec{X}$ are $X_1, X_2$, etc. Variable values will be denoted with a small letter, $x, y$. An assignment of values to every variable in a set will be denoted with $X = \mathbf{x}$, where $\mathbf{x} = (x_1, x_2, ..., x_n)$ indicating $(X_1 = x_1, X_2 = x_2, ..., X_n = x_n)$. When using the vector notation for a set, I will also use the vector notation for the assignment, so $\vec{X} = \vec{x}$.

Functions and mappings will generally be denoted with either capital calligraphic text, like $\mathcal{F}$ and $\mathcal{R}$ or with lowercase letters like $f$ and $g$.

## 2.2 Actual Causality

The "actual" part in actual cause, is to distinguish this kind of causality from causality in a more general sense. *Type causality* is considered with more general statements that can be used for prediction, e.g. an expression like: "Doing homework causes a good exam grade." The two events are not directly related, and there are of course people who did do their homework but did not get a good exam grade, but in general doing homework increases the probability of a good exam grade. *Actual causality* on the other hand is interested in more direct relations, an actual cause of an event did actually cause it. For example, if I knock over my glass of water and water spills over my notes, knocking over my glass was an actual cause of my notes becoming unreadable [8][1].

### 2.2.1 Causal Models

Causality has been studied since the ancient Greeks, but formal research into causality started with Hume in 1758, who introduced two definitions of causality [10]. A later notable formal definition we will focus on is the definition by Halpern and Pearl [9]. For this formal definition it is important to create a model that tries to capture all relevant variables in a system and the causal relations between them. We call such a model a *causal model*:

[1] This is a purely hypothetical example and not based on a true story.

**Definition 1** (Causal Model [8])**.** *A* causal model $\mathcal{M}$ *is a pair* $(\mathcal{S}, \mathcal{F})$*, where* $\mathcal{S}$ *is a signature and* $\mathcal{F}$ *defines a set of structural equations, relating the values of the variables.*
*A* signature $\mathcal{S}$ *is a tuple* $(\mathcal{U}, \mathcal{V}, \mathcal{R})$*, where* $\mathcal{U}$ *is a set of exogenous variables,* $\mathcal{V}$*, a set of endogenous variables and* $\mathcal{R}$ *associates with every variable* $X \in \mathcal{U} \cup \mathcal{V}$ *a nonempty set* $\mathcal{R}(X)$ *of possible values for* $X$*.*

*A* causal setting *is a tuple* $(\mathcal{M}, \mathbf{u})$*, where* $\mathcal{M}$ *is a causal model and* $\mathbf{u}$ *a setting for the exogenous variables in* $\mathcal{U}$*.*

The *exogenous variables* are variables whose values depend on factors outside of the model, when we created the model, we chose not to explain how they are caused [8, 14]. The *endogenous variables* on the other hand are fully determined by the variables in the model, or more precise, they are eventually fully determined by the exogenous variables.

To see how that works, we must first look more closely at $\mathcal{F}$, the set of structural equations. $\mathcal{F}$ consists of functions $\mathcal{F}_X$, one for every $X \in \mathcal{V}$. Such a function $\mathcal{F}_X$ then assigns a value to $X$ based on the values of all other variables in the model [14][2]. In practice, not every variable will influence every other variable and hence not every variable will show up in every structural equation.

**Definition 2** (Dependence [8])**.** *Given a causal model* $\mathcal{M}$*, and variables* $X, Y \in \mathcal{U} \cup \mathcal{V}$*,* $Y$ depends *on* $X$ *if there is a setting of all variables in* $\mathcal{U} \cup \mathcal{V}$ *s.t. changing the value of* $X$ *in that setting results in a change of the value of* $Y$*. Formally,* $\exists \mathbf{z} \in (\mathcal{U} \cup \mathcal{V}) \backslash \{X, Y\}$ *and* $\exists x, x' \in \mathcal{R}(X)$ *s.t.* $\mathcal{F}_Y(x, \mathbf{z}) \neq \mathcal{F}_Y(x', \mathbf{z})$*. If* $Y$ *does not depend on* $X$*, we say that* $Y$ *is* independent *of* $X$*.*

In some contexts, the relation described by Definition 2 is called *direct dependence* [8].

Now that we know this, we can see that because only the endogenous variables are described by the structural equations, they must ultimately depend on one or more exogenous variables.

A way to easily show these dependencies is by depicting a causal model as a *causal network*. Such a network is a directed graph with nodes corresponding to the causal variables in $\mathcal{V}$ (and $\mathcal{U}$) with an edge from the node labelled $X$ to the node labelled $Y$ if and only if $\mathcal{F}_Y$ depends on $X$, i.e. if $X$ can influence the value of $Y$ we put an edge from node $X$ to node $Y$ [9] .

Let us now look at an example of such a causal network:

**Example 1.** Lets define a signature $\mathcal{S} = (\mathcal{U}, \mathcal{V}, \mathcal{R})$, with $\mathcal{U} = \{U_A, U_B\}$, $\mathcal{V} = \{A, B, C, D\}$ and $\mathcal{R}(Y) = \{0, 1\}$ for all $Y \in \mathcal{U} \cup \mathcal{V}$. Lets now define a set of structural equations $\mathcal{F}$ for all variables in $\mathcal{V}$:

- $\mathcal{F}_A = U_A$
- $\mathcal{F}_B = \min(U_B, A)$
- $\mathcal{F}_C = \max(A, B)$

[2] Formally this means that $\mathcal{F}_X : (\Pi_{X' \in \mathcal{U} \cup \mathcal{V}} \mathcal{R}(X')) \rightarrow \mathcal{R}(X)$

- $\mathcal{F}_D = (C-1)^2$

We now have a causal model $M = (\mathcal{S}, \mathcal{F})$. The network is given in Figure 1. The network has an edge from $A$ to $B$, which indicates that $B$ depends on $A$. Lets check if that is indeed true according to Definition 2. Let us take $\mathbf{z} \in (\mathcal{U} \cup \mathcal{V}) \backslash \{A, B\}$ such that $\mathbf{z} = (U_A = 1, U_B = 1, C = 1, D = 1)$, let us look at the values of $\mathcal{F}_B(A = 0, \mathbf{z}) = \min(U_B, A) = \min(1, 0) = 0$ and $\mathcal{F}_B(A = 1, \mathbf{z}) = \min(1, 1) = 1$, so $A$ can indeed influence the value of $B$.

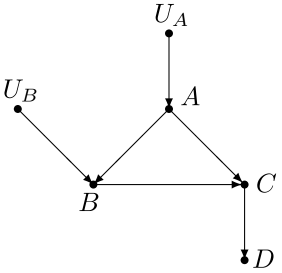


Figure 1: The causal network of the causal model defined in Example 1.

Most of the literature is concerned with models that lead to *acyclic* graphs. In these models there are no cyclic dependencies between variables. That is, if $X$ depends on $Y$ and $Y$ depends on $Z$, then $Z$ can not depend on $X$ [8]. This is equivalent to the notion of a strongly recursive model:

**Definition 3** (Strongly Recursive Models [8])**.** *A model $\mathcal{M}$ is* strongly recursive *if $\exists \preceq$, a partial order on the endogenous variables of $\mathcal{M}$, $\mathcal{V}$, s.t. for any $X, Y \in \mathcal{V}$ unless $X \preceq Y$, $Y$ is not influenced by $X$. $X$* influences *$Y$ if $\exists X_1, ...X_k$ s.t. $X_1 = X$ and $X_k = Y$ and $\forall i \leq k$, $X_{i+1}$ depends on $X_i$.*

**Example 2.** The causal network of Example 1 is an acyclic graph, lets see if the corresponding model is indeed strongly recursive. Define the partial order $\preceq$ such that $\forall X \in \mathcal{V}$, $X \preceq X$, and $A \preceq B$, $A \preceq C$, $B \preceq C$ and $C \preceq D$. Transitivity of partial orders also gives us $A \preceq D$ and $B \preceq D$. For any $(X, Y) \in \preceq$ (so $X \preceq Y$), $X$ does indeed influence $Y$ in this network and for all $(X, Y) \notin \preceq$, $X$ does not influence $Y$. Lets now look at the network in Figure 2. This network is cyclic and so it is supposed to not be strongly recursive. We should hence not be able to define a partial order $\preceq'$ for it, so lets see what happens if we try. $B$ depends on $A$, so we must have $A \preceq' B$. Similarly, we must have $B \preceq' C$, and $C \preceq' A$. However $A$ influences $C$ through $B$ as well, so we must also have $A \preceq' C$, and similarly $B \preceq' A$ and $C \preceq' B$. However, a partial order must be antisymmetric, so for any $X, Y$, if $X \preceq' Y$, $X$ must equal $Y$. This is not the case in this partial order and hence we cannot define a partial order according to the constraints of Example 2 and this network is indeed not strongly recursive.

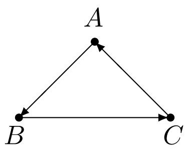


Figure 2: A simple cyclic causal network, the exogenous variables are not displayed.

In a strongly recursive model, a setting $\mathbf{u}$ of the endogenous variables $\mathcal{U}$ fully determines the values of all other (endogenous) variables in the model. After all, we can use the structural equations first to determine the values of the first-level variables, the variables whose values only depend on the exogenous variables [8]. These variables must exist, because the model is acyclic, which means that if there are no endogenous variables that only depend on exogenous variables, they must depend on other endogenous variables and because the model is finite this would inevitably lead to a cycle. After determining the values of these first-level variables, we can determine the values of the second-level variables, whose values depend on the values of the exogenous variables and the first-level variables. Then we can determine the third-level variables, etc. This means that for finite, recursive models, a causal setting $(\mathcal{M}, \mathbf{u})$ fully determines the values of every variable in $\mathcal{M}$. This is illustrated in the following example:

**Example 3.** Consider again the model from Example 1. Note that this model only has one first-level variable: $A$, as $B$ does not just depend on exogenous variables, but also on $A$. This makes $B$ a second-level variable. $C$ is a third-level variable, as it depends on $B$ and that makes $D$ a fourth-level variable.

Now we can compute the values of these variables. Take the setting $\mathbf{u} = (U_A = 1, U_B = 0)$. This leads to the value of $A$ being 1. Given that, we can compute $B = \min(0, 1) = 0$. We can then compute $C = \max(1, 0) = 1$ and then $D = (1 - 1)^2 = 0$. Hence $\mathbf{u}$ fully determines the values of all variables in the model.

### 2.2.2 Formal Definitions of Causality

Now that we have defined the basic terms for causal models, we can look at how causes have been formally defined in the literature. To do that we first need to consider how we will formally depict causes.

Given a signature $\mathcal{S} = (\mathcal{U}, \mathcal{V}, \mathcal{R})$, a formula of the form $X = x$, for $X \in \mathcal{V}$ and $x \in \mathcal{R}(X)$ is called a *primitive event* [9, 8]. These primitive events can be combined with the Boolean connectives $\wedge, \vee$ and $\neg$, to form a *Boolean combinations of primitive events* [9, 8]. A *causal formula* has the form $[Y_1 \leftarrow y_1, ..., Y_k \leftarrow y_k]\varphi$, where $\varphi$ is a Boolean combination of primitive events, $Y_1, ..., Y_k \in \mathcal{V}$ with $\forall i, j, Y_i = Y_j$ if and only if $i = j$, and $y_i \in \mathcal{R}(Y_i), \forall 1 \leq i \leq k$. Such a formula can be shortened to $[Y \leftarrow \mathbf{y}]\varphi$, and when $k = 0$ it is written as just $\varphi$ [9].

Intuitively $[Y \leftarrow \mathbf{y}]\varphi$ says that $\varphi$ holds in the counterfactual world where $Y$ is set to $\mathbf{y}$. In a simpler example, $[Y \leftarrow \mathbf{y}](X = x)$ says that after an intervention that sets all variables of $Y$ to $\mathbf{y}$, it must be the case that $X = x$ [9, 8].

Now, given a causal setting $(\mathcal{M}, \mathbf{u})$, if the causal formula $\psi$ is true in this setting, we write $(\mathcal{M}, \mathbf{u}) \vDash \psi$ [9, 8]. It is clear that in a recursive model, we have $(\mathcal{M}, \mathbf{u}) \vDash X = x$ if and only if the value of $X$ is $x$ after setting the exogenous variables to $\mathbf{u}$ [8]. We can also consider a model after an intervention on the causal model $\mathcal{M}$ that sets the variables in a set $Y$ to $\mathbf{y}$. We denote this model by $M_{Y \leftarrow \mathbf{y}}$, it is called a *submodel* of $\mathcal{M}$ [9]. We can now see that $(\mathcal{M}, \mathbf{u}) \vDash [Y \leftarrow \mathbf{y}]\psi$ and $(M_{Y \leftarrow \mathbf{y}}, \mathbf{u}) \vDash \psi$ are equivalent. In other words, $(\mathcal{M}, \mathbf{u}) \vDash [Y \leftarrow \mathbf{y}]\psi$ if and only if $(M_{Y \leftarrow \mathbf{y}}, \mathbf{u}) \vDash \psi$ [8].

Now we can consider the causality definition of Halpern and Pearl [9].

**Definition 4** (HP Definition [8])**.** $X = \mathbf{x}$ *is an* actual cause *of* $\varphi$ *in the causal setting* $(\mathcal{M}, \mathbf{u})$ *if the following 3 conditions hold (where we take* AC2.(a) *together with either* AC2.(b$^o$) *or* AC2.(b$^u$)*, or only the modified version* AC2.(a$^m$)*):*

AC1. $(\mathcal{M}, \mathbf{u}) \vDash X = \mathbf{x}$ *and* $(\mathcal{M}, \mathbf{u}) \vDash \varphi$*;*

AC2. *(a)* $\exists Z, W \subseteq \mathcal{V}$ *s.t.* $Z \cap W = \emptyset$, $Z \cup W = \mathcal{V}$ *(so a partition of* $\mathcal{V}$*), with* $X \subseteq Z$ *and a setting* $\mathbf{x}'$ *and* $\mathbf{w}$ *of the variables in* $X$ *and* $W$*, respectively s.t.* $(\mathcal{M}, \mathbf{u}) \vDash [X \leftarrow \mathbf{x}', W \leftarrow \mathbf{w}]\neg\varphi$*; and,*

*(b$^o$) If* $\mathbf{z}^*$ *is s.t.* $(\mathcal{M}, \mathbf{u}) \vDash Z = \mathbf{z}^*$*, then* $\forall Z' \subseteq Z \backslash X$*, we have* $(\mathcal{M}, \mathbf{u}) \vDash [X \leftarrow \mathbf{x}, W \leftarrow \mathbf{w}, Z' \leftarrow \mathbf{z}^*]\varphi$*; or,*

*(b$^u$) If* $\mathbf{z}^*$ *is s.t.* $(\mathcal{M}, \mathbf{u}) \vDash Z = \mathbf{z}^*$*, then* $\forall W' \subseteq W$ *and* $\forall Z' \subseteq Z \backslash X$*, we have* $(\mathcal{M}, \mathbf{u}) \vDash [X \leftarrow \mathbf{x}, W' \leftarrow \mathbf{w}, Z' \leftarrow \mathbf{z}^*]\varphi$*; or just,*

*(a$^m$) There is a set* $W$ *of variables in* $\mathcal{V}$ *and a setting* $\mathbf{x}'$ *of variables in* $X$ *s.t. if* $(\mathcal{M}, \mathbf{u}) \vDash W = \mathbf{w}^*$*, then* $(\mathcal{M}, \mathbf{u}) \vDash [X \leftarrow \mathbf{x}', W \leftarrow \mathbf{w}^*]\neg\varphi$*.*

AC3. $X$ *is minimal; there is no strict subset* $X'$ *of* $X$ *s.t.* $X' = \mathbf{x}'$ *satisfies* AC1 *and* AC2*, where* $\mathbf{x}'$ *is the restriction of* $\mathbf{x}$ *to the variables in* $X'$*.*

The original definition considered condition AC2.a and AC2.b$^o$, but there were cases where this did not give a satisfactory result, so they updated the definition to consider conditions AC2.a and AC2.b$^u$. Finally, Halpern came up with a modified definition using only AC2.a$^m$, which he considered to be simpler to work with [8].

When we have a countably finite set of variables written as $A = \{A_1, A_2, ..., A_n\}$, I write $A \leftarrow (a_1, a_2, ..., a_n)$ with $a_1, a_2, ..., a_n \in \mathbb{R}$, to indicate that the $A_1$ gets assigned value $a_1$, $A_2$ gets assigned value $a_2$, etc. The notation $A = (a_1, a_2, ..., a_2)$ indicates that the value of $A_1$ is $a_1$, $A_2$ is $a_2$, etc. This is a bit of an abuse of notation, because here I sometimes treat the set more like a vector, similar notation has however been used in the book by Halpern [8], and I believe this is the simplest way to denote what is happening. For a singleton set $A = \{A_1\}$, I will usually just write $A \leftarrow a_1$ or $A = a_1$. We will see this notation being used in the following example.

**Example 4.** We use again the model from Example 1, we are going to look at whether $A = 1$ is an actual cause of $B = 1$ in the causal setting $(\mathcal{M}, \mathbf{u})$, where $\mathbf{u} = (1, 1)$, according to the original version of the HP definition.

AC1. We have $(\mathcal{M}, \mathbf{u}) \vDash A = 1$ and $(\mathcal{M}, \mathbf{u}) \vDash B = 1$, so this condition is satisfied;

AC2. (a) Take $Z = \{A, B\}$ and $W = \{C, D\}$, set $W \leftarrow (0, 1)$. Now $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 0, W \leftarrow (0, 1)]\neg(B = 1)$ (because $B = 0$). This condition is hence also satisfied;

(b$^o$) In the setting $(\mathcal{M}, \mathbf{u})$, we have $Z = (1, 1) =: \mathbf{z}^*$. There exist two $Z' \subseteq Z\backslash\{A\}$, namely $Z'_1 = \emptyset$ and $Z'_2 = \{B\}$. $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, W \leftarrow (0, 1)](B = 1)$ and $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, W \leftarrow (0, 1), Z'_1 \leftarrow 1](B = 1)$, so this is also satisfied.

This condition is hence also satisfied.

AC3. Since $\{A\}$ is a singleton set, it is minimal.

Hence, $A = 1$ is indeed a cause of $B = 1$ according to the original definition. Let us also check this for the modified and the updated definition:

AC2.(b$^u$) We still have that $\mathbf{z}^* = (1, 1)$, the subsets of $W$ are $\emptyset, \{C\}, \{D\}$ and $\{C, D\}$ itself, with $\mathbf{w} = (0, 1)$ again. The subsets of $Z\backslash\{A\}$ are still $\emptyset$ and $\{B\}$. So let us now check every combination of these.

- $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1](B = 1)$ (for the two empty sets);
- $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, \{B\} \leftarrow 1](B = 1)$;
- $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, \{C\} \leftarrow 0](B = 1)$;
- $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, \{C\} \leftarrow 0, \{B\} \leftarrow 1](B = 1)$;
- $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, \{D\} \leftarrow 1](B = 1)$;
- $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, \{D\} \leftarrow 1, \{B\} \leftarrow 1](B = 1)$;
- $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, \{C, D\} \leftarrow (0, 1)](B = 1)$;
- $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 1, \{C, D\} \leftarrow (0, 1), \{B\} \leftarrow 1](B = 1)$;

AC2.(a$^m$) Take $W = \{C, D\}$ and $A = 0$, then we have $(\mathcal{M}, \mathbf{u}) \vDash W = (1, 0)$ and given that, $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 0, W \leftarrow (1, 0)]\neg(B = 1)$, as $B = 0$ in this setting. Hence this condition holds and it is also true according to the modified definition.

$A = 1$ is thus a cause for $B = 1$ according to all 3 iterations of the HP definition.

The tuple $(W, \mathbf{w}, \mathbf{x}')$ in AC2.a is called a *witness* to the fact that $X = \mathbf{x}$ is a cause of $\varphi$.

**Definition 5** (But-For Cause [8])**.** *We say that $X = x$ is a* but-for cause *of $\varphi$ in $(\mathcal{M}, \mathbf{u})$, if AC1 holds, so both $(\mathcal{M}, \mathbf{u}) \vDash X = x$ and $(\mathcal{M}, \mathbf{u}) \vDash \varphi$, and if $\exists x'$ s.t. $(\mathcal{M}, \mathbf{u}) \vDash [X \leftarrow x']\neg\varphi$.*

Intuitively, a but-for cause of an event $E$ is a cause $A$ such that, but for $A$, $E$ would not have happened [8]. In other words, if $A$ had not happened, $E$ would also not have happened.

**Proposition 1.** [8] *If $X = x$ is a but-for cause of $Y = y$ in $(\mathcal{M}, \mathbf{u})$, then $X = x$ is a cause of $Y = y$ according to all three variants of the HP definition (Definition 4).*

Proposition 1 shows that though the variants of the HP definition have different results in several cases, they luckily do agree on this fairly basic idea of causality.

**Example 5.** Lets check if in the model from Example 1, $A = 1$ is a but-for cause of $B = 1$ in the causal setting $(\mathcal{M}, \mathbf{u} = (1, 1))$. We have already checked in Example 4 that AC1 holds, so we just need to find an $x'$ such that $(\mathcal{M}, \mathbf{u}) \models [A \leftarrow x']\neg(B = 1)$. In fact, such an $x'$ does exist, namely if we set $A = 0$, we get that $B = 0$. Hence $A = 1$ is a but-for cause of $B = 1$ in $(\mathcal{M}, \mathbf{u})$. This also supports Proposition 1, as we have already shown that $A = 1$ is a cause according to all three variants of the HP definition.

Lets now look at how the definitions compare to one another:

**Theorem 1.** [8] *The following claims hold for the different versions of the HP definition:*

*a) If $X = x$ is part of a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the modified HP definition, then $X = x$ is a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the original HP definition.*

*b) If $X = x$ is part of a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the modified HP definition, then $X = x$ is a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the updated HP definition.*

*c) If $X = x$ is part of a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the updated HP definition, then $X = x$ is a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the original HP definition.*

*d) If $X = \mathbf{x}$ is a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the original HP definition, then $|X| = 1$.*

With a *part of a cause*, we mean that if a cause of an event is of the form $X_1 = x_1 \land X_2 = x_2 \land ... \land X_n = x_n$, then each of the $X_i = x_i$ are parts of the cause.

Part d of the theorem looks different from the other 3 parts, but is actually very similar. It can be reformulated to: *If $X = x$ is part of a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the original HP definition, then $X = x$ is a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the original HP definition.* To see that this statement is equivalent, first note that it follows from d). If $X = x$ is a singleton, it only has one part of a cause, which is itself. Now lets see that this statement also implies d). If we assume that this statement is true and $\mathbf{X} = \mathbf{x}$ is a cause of $\varphi$ with $|\mathbf{X}| > 1$ and $X = x$ is a conjunct of $\mathbf{X} = \mathbf{x}$, then according to this statement $X = x$ must also be a cause. However, by AC3, $\mathbf{X} = \mathbf{x}$ is minimal, which it would not be if $|\mathbf{X}| > 1$ and $X = x$ is a cause as well. Hence, $|\mathbf{X}| > 1$ must be false and in fact, $|\mathbf{X}| = 1$ [8].

Another important thing to consider about causality in this sense is that causes do not need to be unique, there can be more than one cause of an event [8].

Causality is not transitive under the HP definition. It is also not true that if $X$ is a cause of $Y$, and $Y$ logically implies $Y'$, that $X$ also is a cause of $Y'$ [8]. We can however prove a result for but-for causes, but before we can do that, we must first state the following definition:

**Definition 6** (Causal Path [8])**.** *A* causal path *in a causal setting* $(\mathcal{M}, \mathbf{u})$ *is a sequence* $(Y_1, ..., Y_n)$ *of variables s.t.* $Y_{j+1}$ *depends on* $Y_i$ *in context* $\mathbf{u}$ *for* $j = 1, ..., k-1$. *In a causal network, a causal path is just a path in the graph. We say that* $Y$ lies *on a causal path in* $(\mathcal{M}, \mathbf{u})$ *from* $X_1$ *to* $X_2$ *if* $Y$ *is a node on a causal path in* $(\mathcal{M}, \mathbf{u})$ *from* $X_1$ *to* $X_2$.

**Proposition 2.** [8] *Suppose that* $X_1 = x_1$ *is a but-for cause of* $X_2 = x_2$ *in the causal setting* $(\mathcal{M}, \mathbf{u})$, $X_2 = x_2$ *is a but-for cause of* $X_3 = x_3$ *in* $(\mathcal{M}, \mathbf{u})$, *and the following conditions hold:*

*a)* $\forall x_2' \in \mathcal{R}(X_2)$, $\exists x_1' \in \mathcal{R}(X_1)$ *s.t.* $(\mathcal{M}, \mathbf{u}) \vDash [X_1 \leftarrow x_1'](X_2 = x_2')$;

*b)* $X_2$ *is on every causal path in* $(\mathcal{M}, \mathbf{u})$ *from* $X_1$ *to* $X_3$,

*Then* $X_1 = x_1$ *is a but-for cause of* $X_3 = x_3$.

Hence, in a special case, but-for causes are transitive. This is illustrated in the following example.

**Example 6.** Consider the following causal model $M = (\mathcal{S}, \mathcal{F})$, with $\mathcal{S} = (\mathcal{U}, \mathcal{V}, \mathcal{R})$, with $\mathcal{U} = \{U_A, U_B, U_C\}$, $\mathcal{V} = \{A, B, C\}$ and $\mathcal{R}(Y) = \{0, 1\}$ for all $Y \in \mathcal{U} \cup \mathcal{V}$. Let the structural equations $\mathcal{F}$ be defined by:

- $\mathcal{F}_A = U_A$
- $\mathcal{F}_B = \min(U_B, A)$
- $\mathcal{F}_C = \min(U_C, B)$

This model is visualised in the network in Figure 3. $A = 1$ is a but-for cause of $B = 1$, which is a but-for cause of $C = 1$ in the causal setting $(\mathcal{M}, \mathbf{u} = (1, 1, 1))$ (notice that these relations are exactly like the relation between $A$ and $B$ in Example 1, it is hence redundant to prove it formally again). $\mathcal{R}(B) = \{0, 1\}$, because $A = 1$ is a but-for cause of $B = 1$, we already know that there is a value of $A$ s.t. setting $A$ to that value gives us $B = 1$. However, we also have that $(\mathcal{M}, \mathbf{u}) \vDash [A \leftarrow 0](B = 0)$, because $B = \min(U_B, A) = \min(1, 0) = 0$ in this case. Therefore, condition a) in Proposition 1 is satisfied. We also have that

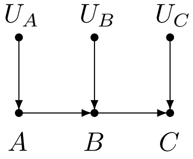


Figure 3: The causal network of the causal model defined in Example 6.

there is only one causal path from $A$ to $C$ and $B$ is on it, therefore according to the proposition, $A$ should also be a but-for cause of $C$. Let us check if that is true. First, AC1 needs to be satisfied, we do indeed have that $(\mathcal{M}, \mathbf{u}) \models A = 1$ and $(\mathcal{M}, \mathbf{u}) \models C = 1$. Now we need to try to find a value of $A$ such that $C$ is not 1 anymore. The only other value we can try is $A = 0$. If $A = 0$, we also get that $B = \min(U_B, A) = 0$, and hence $C = \min(U_C, B) = 0$. Hence, $A = 1$ is indeed a but-for cause of $C = 1$ in this causal setting.

Now that we have discussed the basics of causal models and causality, we can move on to the next topic.

## 2.3 Labelled Fransition Systems

(Labelled) transitions systems were introduced by Robert M. Keller in 1976 as a way to reason about parallel programs [12]. They have been been defined in multiple ways through the years (see [7], [15] and [4]), but we will first consider the following definition:

**Definition 7** (Labelled Transition Systems [12, 4])**.** *A* labelled transition system *(LTS) is a triple $TS = (Q, A, \rightarrow)$ where:*

- *$Q$ is a nonempty, countable set of* states*;*
- *$A$ is the countable set of* labels *(or* action names*) of:*
- *$\rightarrow \subseteq Q \times Q$ is a binary relation on $Q$, called the* set of transitions*.*

We can denote transitions with $q \xrightarrow{a} q'$, where $q, q' \in Q$ and $a \in A$. $q$ is called the transition's source, $q'$ is the target and $a$ is the label of the transition [7]. When a LTS is defined with an initial state, we call it a *rooted labelled transition system*, defined as a pair $(TS, q_0)$, where $TS = (Q, A, \rightarrow)$ is an LTS and $q_0 \in Q$ is the initial state, also called the root [7].

**Example 7.** Lets define an LTS, $TS = (Q, A, \rightarrow)$, where $Q = \{q_1, q_2\}$, $A = \{a_1, a_2\}$ and $\rightarrow = \{(q_1, q_2), (q_2, q_1)\}$. We can denote this LTS in a graph, like in Figure 4.

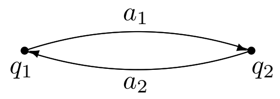


Figure 4: A simple LTS.

A LTS is called *deterministic* if given a transition $\xrightarrow{a}$, for any $q \in Q$ there is at most one $q'$ such that $q \xrightarrow{a} q'$ [12, 15]. It is possible that such a $q'$ does not exist for any state, but if it does exist, we say that $a$ is *enabled* in state $q$ [12].

**Definition 8** (Path [7])**.** *Given an LTS $(Q, A, \to)$, and two states $q, q' \in Q$, a sequence of $n$ transitions ($n$ is allowed to be infinite) $q_1 \xrightarrow{a_1} q'_1$, $q_2 \xrightarrow{a_2} q'_2$, ... , $q_n \xrightarrow{a_n} q'_n$ such that $q = q_1$, $q'_n = q'$ $q'_i = q_i + 1$ for $i = 1, 2, ..., n-1$, is called a* path *of length $n$ from $q$ to $q'$. We say that the path is* acyclic *if $\forall i \neq j$, $q_i \neq q_j$. A rooted LTS $(TS, q_0)$ is acyclic if it contains no cyclic path starting from $q_0$, a regular LTS is acyclic if it contains no cyclic path at all.*

While a path can in general be infinite, it cannot always be infinite in every system. A system may have *deadlock states*, states that have no transition starting from it [7]. It is hence impossible to leave a deadlock state.

We will now look at a few ways to distinguish different LTS:

**Definition 9** (Classes of LTS [7, 15])**.** *A LTS $(Q, A, \to)$ is:*

- Finite-state *if $Q$ and $A$ are finite;*
- Finite *if it is finite-state and acyclic;*
- Boundedly branching *if $\exists k \in \mathbb{N}$ such that $\forall q \in Q$ the set $T_q = \{(q, a, q') | \exists a \in A, \exists q' \in Q$ such that $q \xrightarrow{a} q'\}$ has cardinality at most $k$; the least $k$ satisfying the above condition is called the branching degree of the LTS;*
- Finitely branching *if the set $T_q = \{(q, a, q') | \exists a \in A, \exists q' \in Q$ such that $q \xrightarrow{a} q'\}$ is finite $\forall q \in Q$, otherwise the LTS is* infinitely branching*;*
- Image-finite *if the set $T_{q,a} = \{(q, a, q') | \exists q' \in Q$ such that $q \xrightarrow{a} q'\}$ is finite $\forall q \in Q$ and $\forall a \in A$.*

These classes relate to each other in several ways, as detailed in the following proposition.

**Proposition 3.** [7] *The following results hold for LTS classes:*

1. *A boundedly branching LTS is finitely branching;*
2. *A finitely branching LTS is not in general boundedly branching;*

3. *When a finitely branching LTS has finitely many states, it is boundedly branching;*

4. *A finitely branching LTS that is not boundedly branching cannot be finite-state;*

5. *A finitely branching LTS is image-finite;*

6. *An image-finite LTS is not in general finitely branching.*

*Proof.* I proved all of the above results, they are each shown in turn below:

1. Take a boundedly branching LTS $TS = (Q, A, \rightarrow)$, take $q \in Q$ at random. Because $TS$ is boundedly branching, we have that $\exists k \in \mathbb{N}$ such that $\forall q' \in Q$, $|T_{q'}| \leq k$. So also for our $q$, $|T_q| \leq k < \infty$. As $q$ was taken at random, this must hold for every $q \in Q$, and hence $\forall q \in Q$, $T_q$ is finite. *q.e.d.*

2. I will give an example where an LTS is finitely branching, but not boundedly branching: Define the LTS $TS = (Q, A, \rightarrow)$ with $Q$ being (countably) infinite, with the items being numbered $q_{0,0}, q_{1,0}, q_{1,1}, q_{2,0}, q_{2,1}, q_{2,2}$, etc. Suppose that $\forall i, j \leq i$, $q_{i,j}$ has transitions to $q_{i+1,j'}$ $\forall j' \leq i+1$. So $q_{0,0}$ has 2 transitions, one to $q_{1,0}$ and one to $q_{1,1}$, each of the $q_1$'s has 3 transitions, to each of the $q_2$'s, etc. In general, $\forall i \in \mathbb{Z}_{\geq 0}$, $q_{i,j}$ has $i+2$ transitions for all $0 \leq j \leq i$. So $\forall q \in Q$, $T_q$ is finite. However, there is no $k \in \mathbb{N}$ such that $|T_q| \leq k$ $\forall q$. To see this, you need to see that $\forall k \in \mathbb{N}$, $|T_{q_{k,j}}| = k+2$ and hence there does not exist an upper-bound for $|T_q|$.

3. Now, take a finitely branching LTS $TS = (Q, A, \rightarrow)$ with $|Q| < \infty$. Because $Q$ is finite, we can define $k := \max_{q \in Q}(|T_q|)$. Now, clearly $\forall q \in Q$, $|T_q| \leq k$, in fact, $k$ is the branching degree of $TS$. *q.e.d.*

4. Take the LTS $TS = (Q, A, \rightarrow)$ to be finitely branching, but not boundedly branching. Assume now that it is also finite-state. By definition, $Q$ would be finite and by 3. from Proposition 3 it would have to be boundedly branching. This is a contradiction! Hence $TS$ cannot be finite-state.

5. Take the finitely-branching LTS $TS = (Q, A, \rightarrow)$. It is easy to see that $\forall a \in A, q \in Q$, $T_{q,a} \subseteq T_q$. Since $TS$ is finitely-branching, $T_q$ is finite $\forall q \in Q$. But as $\forall q \in Q, \forall a \in A$, $T_{q,a} \subseteq T_q$, $T_{q,a}$ is also finite $\forall a \in Q, q \in Q$. $TS$ is hence image-finite. *q.e.d.*

6. If the LTS $TS = (Q, A, \rightarrow)$ has for any $q \in Q$, an infinite number of transitions, each with a unique label $a \in A$, but every transition is deterministic. We have that $\forall q \in Q, a \in A$, $|T_{q,a}| \leq 1$ and hence the LTS is image-finite. However, because there are infinitely many transitions, $T_q$ is infinite and hence $TS$ is not finitely branching.

□

There are multiple definitions for equivalence of LTS, each with their own properties [7]. To give an idea of these definitions, we will look at *trace equivalence*, but before we can do that, we must define *traces*.

**Definition 10** (Trace [7, 15])**.** *Let $(Q, A, \rightarrow)$ be an LTS, and let $q \in Q$. A* trace *of $q$ is a string $\sigma \in A^*$, such that $q \xrightarrow{\sigma}^* q'$ for some $q' \in Q$. Here $A^*$ is the set of all strings on $A$, including the empty string $\epsilon$ and $\rightarrow^* \subseteq Q \times A^* \times Q$ is the* reachability relation *defined as the least relation induced by the following:*

$$\frac{}{q \xrightarrow{\epsilon}^* q} \quad \frac{q_1 \xrightarrow{\sigma}^* q_2 \quad q_2 \xrightarrow{a} q_3}{q_1 \xrightarrow{\sigma a}^* q_3}.$$

*The set of traces of $q$ is defined as:*

$$Tr(q) = \{\sigma \in A^* | \exists q' \in Q \text{ such that } q \xrightarrow{\sigma}^* q'\}$$

*Two states $q_1, q_2 \in Q$ are said to be* trace equivalent *if $Tr(q_1) = Tr(q_2)$. Two rooted LTS are trace equivalent if their roots are trace equivalent.*

This basic form of equivalence is not sensitive to deadlock. It may equate two states, where one is a deadlock state and the other is not [7].

In the next section I will discuss several logic systems we can use to reason about transition systems, but before we can do that, I must slightly extend the definition of LTS so that every state is also associated with atomic propositions that hold in that state. Formally:

**Definition 11** (Alternative Definition Labelled Transition System [4, 1])**.** *A* labelled transition system *(LTS) is a tuple $TS = (Q, A, \rightarrow, \Pi, \pi)$ where:*

- *$Q$ is a nonempty, countable set of* states*;*
- *$A$ is the countable set of* labels *(or* actions*) of:*
- *$\rightarrow \subseteq Q \times Q$ is a binary relation on $Q$, called the* set of transitions*;*
- *$\Pi$ is a set of atomic propositions*
- *$\pi : Q \rightarrow 2^{\Pi}$, the labelling function.*

$\pi$ associates with every state $q \in Q$ a (possibly empty) set of atomic propositions that hold in $q$. When we define a LTS in this way they can be seen as a special case of a concurrent game structure [1].

**Example 8.** We can extend the LTS from Example 7 to the following LTS defined according to Definition 11, $TS = (Q, A, \rightarrow, \Pi, \pi)$, where:

- $Q = \{q_1, q_2\}$;
- $A = \{a_1, a_2\}$;
- $\rightarrow = \{(q_1, q_2), (q_2, q_1)\}$;

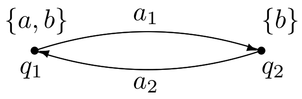


Figure 5: A simple LTS according to Definition 11.

- $\Pi = \{a, b\}$;
- $\pi$ maps $q_1$ to $\{a, b\}$ and $q_2$ to $\{b\}$.

The resulting LTS is shown in Figure 5.

Labelled transition systems are in fact a special case of concurrent game structures, where there is only one player [1]:

**Definition 12** (Concurrent Game Structures [1])**.** *A* concurrent game structure *is a tuple $GS = \langle k, Q, d, \delta, \Pi, \pi \rangle$ with the following components:*

- *A natural number $k \geq 1$ of players. We identify the* players *with the numbers $1, ..., k$.*
- *A finite set $Q$ of states.*
- *For each player $a \in \{1, ..., k\}$ and each state $q \in Q$, a natural number $d_a(q) \geq 1$ of moves available at state $q$ to player $a$. We identify the moves of player $a$ at state $q$ with the numbers $1, ..., d_a(q)$. For each state $q \in Q$, a move vector at $q$ is a tuple $\langle j_1, ..., j_k \rangle$ such that $1 \leq j_a \leq d_a(q)$ for each player $a$. Given a state $q \in Q$, we write $D(q)$ for the set $\{1, ..., d_1(q)\} \times \cdots \times \{1, ..., d_k(q)\}$ of* move vectors*. The function $D$ is called* move function*.*
- *For each state $q \in Q$ and each move vector $\langle j_1, ..., j_k \rangle \in D(q)$, a state $\delta(q, j_1, ..., j_k) \in Q$ that results from state $q$ if every player $a \in \{1, ..., k\}$ chooses move $j_a$ . The function $\delta$ is called* transition function*.*
- *A finite set $\Pi$ of propositions.*
- *For each state $q \in Q$, a set $\pi(q) \subseteq \Pi$ of propositions true at $q$. The function $\pi$ is the labelling function.*

**Example 9.** Let $GS = \langle k, Q, d, \delta, \Pi, \pi \rangle$ be a concurrent game structure, with:

- $k = 2$;
- $Q = \{q_1, q_2\}$;
- $d_{a_1}(q_1) = d_{a_2}(q_1) = 1$, and $d_{a_1}(q_2) = d_{a_2}(q_2) = 2$, so that $D(q_1) = \{\langle 1, 1 \rangle\}$ and $D(q_2) = \{\langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 2, 1 \rangle, \langle 2, 2 \rangle\}$;

- $\delta(q_1, 1, 1) = q_2$, $\delta(q_2, 1, 1) = \delta(q_2, 1, 2) = \delta(q_2, 2, 1) = q_1$, and $\delta(q_2, 2, 2) = q_2$;
- $\Pi = \{a, b\}$;
- $\pi(q_1) = \{a, b\}$ and $\pi(q_2) = \{b\}$.

This is visualised in Figure 6. As we can see, in this concurrent game structure, both players only have one action in $q_1$, taking this action brings them to state $q_2$. In $q_2$ both have two actions, only if both choose action 1, they go back to state $q_1$, in all other cases they stay in state $q_2$.

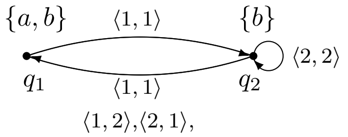


Figure 6: A simple concurrent game structure with two players.

Note that if $k = 1$, a concurrent game structure $S = \langle k, Q, d, \delta, \Pi, \pi\rangle$ simplifies to $d$ only being one function, mapping a state to the number of moves available at that state. The move vector at a state $q$ is just a scalar, a single number. $D$ is the same as $d$ in this case. We also have that $\bigcup_{q \in Q}\{(q, \delta(q, j_1))\}$ equals the set $\rightarrow \subseteq Q \times Q$ of Definition 11.

**Example 10.** Consider the LTS from Example 8, we can describe it as a concurrent game structure $GS = \langle k, Q, d, \delta, \Pi, \pi\rangle$, by setting:

- $k = 1$;
- $Q = \{q_1, q_2\}$;
- $d(q_1) = 1$, $d(q_2) = 1$
- $\delta(q_1, 1) = q_2$, $\delta(q_2, 1) = q_1$;
- $\Pi = \{a, b\}$;
- $\pi(q_1) = \{a, b\}$ and $\pi(q_2) = \{b\}$.

We can describe any LTS as a concurrent game structure in this way. The only difference being that the actions do not have names. This also means that while we could define an LTS like in Figure 7, where both transitions have the same label and could be seen as the same action (imagine a button that when you press it turns the light on or off, where Figure 5 could represent a system with a switch that can be flipped on or off), a concurrent game structure description cannot distinguish between the systems in Figures 5 and 7.

We will now take a look at several logics that can be used in these systems.

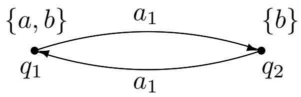


Figure 7: An LTS like in Figure 5, but with only one action label.

## 2.4 Logics for Transition Systems

There are multiple logics one can use to reason about transition systems. I will first discuss linear-temporal logic, after that I will shortly introduce computation-tree logic and alternating-time temporal logic.

### 2.4.1 Linear-Time Temporal Logic

Linear-time temporal logic (LTL for short) is a logic that is used to reason about models that represent time as a straightforward sequence. It can hence be evaluated on paths from LTS. LTL uses operators from propositional logic, and in addition the temporal modalities $\bigcirc$, which can be read as "next", and $\mathcal{U}$, which can be read as "until" [4].

**Definition 13** (LTL Syntax [4, 11])**.** *Given a set* $\Pi$ *of atomic propositions, a formula in linear-time temporal logic is build according to the following grammar:*

$$\varphi ::= \top \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \mathcal{U} \varphi_2,$$

*where* $a \in \Pi$*.*

Besides these basic operators, we can define the other modal operators $\Diamond$ meaning "eventually" and $\Box$ meaning "always", as:

$$\Diamond\varphi := \top\mathcal{U}\varphi \quad \Box\varphi := \neg\Diamond\neg\phi.$$

Generally speaking, LTL-formulas are evaluated on infinite words over $2^\Pi$. A word $\sigma$ is of the form:

$$\sigma = A_0A_1A_2..., \text{ where } A_i \in 2^\Pi, \forall i \in \mathbb{Z}_{\geq 0}.$$

Let $\sigma[j] = A_j$ and let $\sigma[j...]$ denote the fragment $A_jA_{j+1}....$

**Definition 14** (Semantics of LTL [4, 11])**.** *The semantic relation* $\models$ *of linear temporal logic is defined by:*

$\sigma \models p$ *iff* $p \in \sigma[0]$ *(i.e. p is true at the start of the word)*

$\sigma \models \varphi_1 \wedge \varphi_2$ *iff* $\sigma \models \varphi_1$ *and* $\sigma \models \varphi_2$

$\sigma \models \neg\varphi$ *iff* $\sigma \not\models \varphi$

$\sigma \models \bigcirc\varphi$ *iff* $\sigma[1] \models \varphi$

$\sigma \models \varphi_1\mathcal{U}\varphi_2$ *iff* $\exists j \geq 0$ *such that* $\sigma[j...] \models \varphi_2$ *and* $\sigma[i...] \models \varphi_1, \forall 0 \leq i < j.$

The derived operators $\Diamond$ and $\Box$ have the following semantics:

$$\begin{aligned}
&\sigma \vDash \Diamond\varphi && \text{iff } \exists j \in \mathbb{Z}_{\geq 0} \text{ such that } \sigma[j...] \vDash \varphi \\
&\sigma \vDash \Box\varphi && \text{iff } \forall j \in \mathbb{Z}_{\geq 0}, \sigma[j...] \vDash \varphi.
\end{aligned}$$

LTL can be used to reason about labelled transition systems, given an infinite path $\lambda$ in a LTS, we have that $\lambda \vDash \varphi$ if $\varphi$ holds for the infinite word $\sigma = \pi(\lambda[0])\pi(\lambda[1])...$, where the $\lambda[i]$'s are states on the path and $\pi(\lambda[i]) \in 2^{\Pi}$ [4, 11]. We write $\pi(\lambda) = \pi(\lambda[0])\pi(\lambda[1])....$

Lets look at an example of how this works:

**Example 11.** Consider the LTS in Figure 5. This LTS has the infinite path $\lambda = q_1q_2q_1q_2....$ We have that $\lambda \vDash a \wedge b$, because $a$ and $b$ hold in state $q_1$. We also have $\lambda \vDash \bigcirc b$ but $\lambda \nvDash \bigcirc a$, as in the second state of the path, $b$ holds, but $a$ does not. It is also true that $\lambda \vDash a\mathcal{U}b$, after all, if we take the $j$ in the definition to be 1, we have that for all $0 \leq i < 1$, i.e. $i = 0$ that $\lambda[0] \vDash a$ and $\lambda[j] \vDash b$. It is also true that $\lambda \vDash \Box b$, but $\lambda \nvDash \Box a$, because while for any $j \geq 0$, $\lambda[j...] \vDash b$, but for example for $j = 1$, $\lambda[j...] \nvDash a$.

We can also say that an entire LTS satisfies an LTL formula $\varphi$. For this we first need to define *the LT property induced by* $\varphi$:

$$Words(\varphi) := \{\sigma \in (2^{\Pi})^w \mid \sigma \vDash \varphi\} \text{ [4].}$$

Now, given an LTS, $TS = (Q, A, \rightarrow, \Pi, \pi)$, we have that $TS$ satisfies the LTL formula $\varphi$, denoted $TS \vDash \varphi$, if $\{\sigma \in (2^{\Pi})^w \mid \sigma = \pi(\lambda) \text{ for a path } \lambda \text{ in } TS\} \subseteq Words(\varphi)$.

**Example 12.** Consider the LTS of Figure 5 again. For this LTS, the set $\{\sigma \in (2^{\Pi})^w \mid \sigma = \pi(\lambda) \text{ for a path } \lambda \text{ in } TS\} = \{\{a,b\}\{b\}\{a,b\}..., \{b\}\{a,b\}\{b\}...\}$. Hence, this LTS satisfies the formula $b$, as all paths in the LTS satisfy this formula, but not the formula $a$, as the path $q_2q_1q_2...$ does not satisfy this.

### 2.4.2 Computation-Tree Logic

It is impossible for LTL to distinguish between what has to happen and what can possibly happen, it is however sometimes useful to know what might possibly happen on only a single path versus what happens on every possible path [11]. Computation-tree logic (CTL) is an extension of LTL that adds path quantifiers.

**Definition 15** (CTL Syntax [4, 11])**.** *Formulas $\phi$ in CTL are called* state formula *and are build up from atomic propositions and* path formula $\varphi$*. The state formula are build up according to:*

$$\phi ::= \top \mid a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \exists\varphi \mid \forall\varphi.$$

*The path formula are formed according to:*

$$\varphi ::= \bigcirc\phi \mid \phi_1\mathcal{U}\phi_2,$$

*where $\phi, \phi_1$ and $\phi_2$ are state formula. The state formula are interpreted over the states of the model, and the path formula are interpreted over the paths of the model.*

Unlike LTL formulas, which are evaluated on paths in systems, state formulas in CTL are evaluated on a state of the model.

**Definition 16** (Semantics of CTL [4])**.** *Let $q \in Q$ be state of the labelled transition system $TS = (Q, A, \rightarrow, \Pi, \pi)$, let $p \in \Pi$ and let $\phi$ and $\psi$ be CTL state formula and let $\varphi$ be a CTL path formula. The semantic relation $\models$ for state formula is defined by:*

$$\begin{array}{ll} q \models p & \textit{iff } p \in \pi(q) \\ q \models \neg\phi & \textit{iff } q \not\models \phi \\ q \models \phi \wedge \psi & \textit{iff } q \models \phi \textit{ and } q \models \psi \\ q \models \exists\varphi & \textit{iff } \sigma \models \varphi \textit{ for some path } \sigma \textit{ starting at } q \\ q \models \forall\varphi & \textit{iff } \sigma \models \varphi \textit{ for all paths } \sigma \textit{ starting at } q. \end{array}$$

*For a path $\sigma$, the semantics for path formulas $\varphi$ are given similarly to the semantics for LTL, by:*

$$\begin{array}{ll} \sigma \models \bigcirc\varphi & \textit{iff } \sigma[1] \models \varphi \\ \sigma \models \varphi_1 \mathcal{U} \varphi_2 & \textit{iff } \exists j \geq 0 \textit{ such that } \sigma[j...] \models \varphi_2 \textit{ and } \sigma[i...] \models \varphi_1, \forall 0 \leq i < j. \end{array}$$

Let us look at an example of how this works in practice:

**Example 13.** Consider again the LTS in Figure 5. The state formula are evaluated over the states of the model. In the simplest form, we can say $q_2 \models b \wedge \neg a$. When we look at the path formulas, we have $q_2q_1q_2... \models \bigcirc(a \wedge b)$, similar to LTL. But now we can introduce the path quantifiers, we have both $q_2 \models \exists \bigcirc (a \wedge b)$ and $q_2 \models \forall \bigcirc (a \wedge b)$, as there is only one path starting from $q_2$ in this system. To illustrate the difference between $\forall$ and $\exists$, we have to modify the system, so lets consider the system in Figure 8. In this system, we still have $q_2 \models \exists \bigcirc (a \wedge b)$, but not $q_2 \models \forall \bigcirc (a \wedge b)$ anymore, because $\bigcirc(a \wedge b)$ is not true on the path one obtains after taking action $a_3$ all the time, $q_2q_2q_2...$, on this path $a$ will never be true.

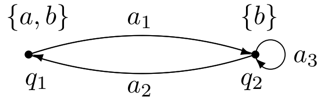


Figure 8: An LTS with two states and three actions.

As the above example shows, CTL can express things LTL cannot. There exists however an even more expressive variant, CTL$^*$, where quantifiers do not have to be followed by one of the temporal operators [11]. However as usual, expressiveness leads to a higher computational cost and makes it more difficult to reason. In this work I will hence focus on CTL.

### 2.4.3 Alternating-Time Temporal Logic

The two logics described above are not sufficiently expressive to describe multi-agent systems, like the concurrent game structures of Definition 12. In order to reason about such systems we use *alternating-time temporal logic* (ATL) [1], which can be seen as an extension of CTL [11].

**Definition 17** (ATL Syntax [1])**.** Alternating-time temporal logic *(ATL) is defined with respect to a set of propositions* $\Pi$ *and a finite set* $\Sigma = \{1, 2, ..., k\}$ *of players. An ATL formula is build up according to:*

$$\varphi ::= \top \mid a \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \langle\langle A \rangle\rangle \bigcirc \varphi \mid \langle\langle A \rangle\rangle \Box \varphi \mid \langle\langle A \rangle\rangle \varphi_1 \mathcal{U} \varphi_2,$$

*where* $A \subseteq \Sigma$ *is a set of players and* $a \in \Pi$.

The temporal operators $\bigcirc, \Box$ and $\mathcal{U}$ have the same meaning as in LTL and CTL, similarly we also write $\langle\langle A \rangle\rangle \Diamond \varphi$ for $\langle\langle A \rangle\rangle \top \mathcal{U} \varphi$ [1]. The operator $\langle\langle \rangle\rangle$ is a path quantifier, like $\forall$ and $\exists$ in CTL [1]. We can write $\langle\langle a_1, a_2, ..., a_k \rangle\rangle$ instead of $\langle\langle \{a_1, a_2, ..., a_k\} \rangle\rangle$ and $\langle\langle \rangle\rangle$ instead of $\langle\langle \emptyset \rangle\rangle$.

ATL formulas are interpreted over the states of a concurrent game structure [1]. Before we can define the semantics of ATL, we must first define the notion of a *strategy* in a concurrent game structure:

**Definition 18** (Strategy in Concurrent Game Structures [1])**.** *Given a concurrent game structure* $S = \langle k, Q, d, \delta, \Pi, \pi \rangle$, *a* strategy *for player* $a \in \Sigma$ *is a function* $f_a$*, that maps any (non-empty) finite sequence* $\lambda$ *of states in* $Q$ *to an action the player can take at the last state of the sequence. I.e. if* $q$ *is the last state of* $\lambda$*, then* $f_a(\lambda) \leq d_a(q)$*. Now, let* $q \in Q$*,* $A \subseteq \Sigma$ *and a set* $F_A = \{f_a \mid a \in A\}$ *of strategies of the players in* $A$*. We define the set of* outcomes *of* $F_A$ *from* $q$ *to be the set* $out(q, F_A)$ *of state sequences the players enforce when following the strategies in* $F_A$*. A sequence* $\lambda = q_0q_1q_2...$ *is in* $out(q, F_A)$ *if* $q_0 = q$ *and* $\forall i \geq 0$*,* $\exists \langle j_1, ..., j_k \rangle \in D(q_i)$ *such that* $j_a = f_a(\lambda[0, i])$ $\forall a \in A$ *and* $\delta(q, j_i, ..., j_k) = q_{i+1}$.

**Example 14.** Consider the concurrent game structure in Example 9. Both players could have the strategy that $f_{a_i}(\lambda) = 1$ if the last state of $\lambda$ is $q_1$ and 2 if the last state is $q_2$. Now, the sequence $q_1q_2q_1q_2...$ is not in $out(q_1, F_A)$, because for $i = 1$, $f_{a_1}(\lambda[0, 1]) = f_{a_2}(\lambda[0, 1]) = 2$. $\delta(q_2, 2, 2) = q_2$ and not $\lambda[2] = q_1$. In fact, the set of outcomes has only one sequence, namely $out(q_1, F_A) = \{q_1q_2q_2q_2...\}$

**Definition 19** (Semantics of ATL [1])**.** *Let* $S = \langle k, Q, d, \delta, \Pi, \pi \rangle$ *be a concurrent transition system, let* $q \in Q$*,* $p \in \Pi$ *and let* $\varphi$ *be an ATL formula. The* *satisfaction relation* $\models$ *of alternating-time temporal logic is defined by:*

$q \models p$ *iff* $p \in \pi(q)$

$q \models \neg\varphi$ *iff* $q \not\models \varphi$

$q \models \varphi_1 \vee \varphi_2$ *iff* $q \models \varphi_1$ *or* $q \models \varphi_2$

$q \models \langle\langle A \rangle\rangle \bigcirc \varphi$ *iff* $\exists F_A$ *such that* $\forall \lambda \in out(q, F_A)$, *we have* $\lambda[1] \models \varphi$

$q \models \langle\langle A \rangle\rangle \Box \varphi$ *iff* $\exists F_A$ *such that* $\forall \lambda \in out(q, F_A)$ *and* $\forall i \geq 0$, *we have* $\lambda[i] \models \varphi$

$q \models \langle\langle A \rangle\rangle \varphi_1 \mathcal{U} \varphi_2$ *iff* $\exists F_A$ *such that* $\forall \lambda \in out(q, F_A), \exists i \geq 0$ *such that* $\lambda[i] \models \varphi_2$
*and* $\forall 0 \leq j \leq i$, *we have* $\lambda[j] \models \varphi_1$

Lets again look an example of how ATL works in practice:

**Example 15.** Consider again the concurrent game structure of Example 9. Clearly, $q_1 \models a$ and $q_2 \models b \vee a$. It becomes more interesting if we consider the path quantifiers. We can for example state $q_2 \models \langle\langle a_1 \rangle\rangle \bigcirc a$, because the strategy where $a_1$ always picks action 1 in any state, has $q_1$ as the next state of every sequence in the outcome set and in $q_1$, $a$ holds. We can however not say $q_2 \models \langle\langle a_1 \rangle\rangle \bigcirc \neg a$, because for any strategy of $a_1$ in state $q_2$, the sequence $q_2 q_1 q_2 q_1 ...$ will be in the outcome set and so $\neg a$ is not true in the next state of all sequences. We can also not state $q_2 \models \langle\langle a_1 \rangle\rangle \Box \neg a$, because no matter what strategy player 1 has, player 2 could always pick action 1, making sure the sequence $q_2 q_1 q_2 q_1 ...$ is in the outcome set, and so $\neg a$ will not hold in all states of the sequence. However, we can state $q_2 \models \langle\langle a_1, a_2 \rangle\rangle \Box \neg a$, if both players always pick action 2 in state $q_2$, $a$ will never be true.

As with CTL and CTL$^*$, there also exists a more expressive variant of ATL, ATL$^*$ [1]. Similar to CTL and CTL$^*$, the difference lies in that quantifiers do not have to be directly followed by a temporal operator in ATL$^*$. So given a set $A \subseteq \Sigma$ of players, $\langle\langle A \rangle\rangle \neg a$ is a valid formula in ATL$^*$, but not in ATL. In this work I will just focus on ATL, as its semantic interpretation is less complex [11].

## 2.5 Responsibility

### 2.5.1 Different Approaches to Responsibility

Many different notions of responsibility have been defined in the literature. [17] gives an overview of different forms of responsibility in a broad sense. Much of the literature is concerned with the notion of strategic responsibility [16, 2], where agents are seen as responsible for a state of affairs if they had the ability to avoid it. This is seen as the 'base form' of responsibility in [17]. Usually, a distinction between forward- and backward-looking responsibility is made. Where the former is considered before an event and the latter afterwards [17, 16, 2].

Closely tied to the definition of responsibility is the notion of blameworthiness, which usually requires agents to have had knowledge of the consequence of their actions [17, 5]. In fact, by some it is seen as a different form of backward-looking responsibility [17]. Accountability and sanctionability are in some cases also seen as other forms of responsibility, where the former has to do with task allocation and the latter with the violation of established norms [17].

In early work, only the cause of an event was seen as (partially) responsible for the occurence of the event [5], but with the notion of strategic responsibility, an agent can be part of a responsible coalition without being a cause of the event.

I now introduce the notion of a computation of a CGS. A *computation* of a CGS $\mathcal{M}$ is an infinite sequence of states $\lambda = q_0, q_1, ...$ such that $q_{i+1}$ is a successor of $q_i$ for all $i > 0$, if a computation starts in $q$, it is called a $q$-computation. A finite sequence of states $q_0, ..., q_n$ is called a *q-history* if $q_n = q$, $n \geq 0$ and $q_{i+1}$ is a successor of $q_i$ for all $0 \leq i \leq n$. A $q$-history starting in $q_i$ with $n$ steps is denoted by $\lambda[q_i, n]$ [16].

In the below, $\bar{S}$ denotes the complement of the set $S$ in $Q$

**Definition 20** (Forward Group Responsibility [16])**.** *Let $\mathcal{M}$ be a CGS, $S$ be a set of states, $q \in S$ a state. We say that a group of agents $\Gamma \subseteq \Sigma$ is* forward responsible *for $S$ in $q$ iff:*

1. *There is a strategy for $\Gamma$, $F_\Gamma$, such that all states on all computations in out$(q, F_\Gamma)$ belong to $\bar{S}$, and*

2. *$\Gamma$ is minimal, that is, there is no $\Gamma' \subsetneq \Gamma$ with the property formulated above.*

**Definition 21** (Backward Group Responsibility [16])**.** *Let $\mathcal{M}$ be a CGS, $S$ be a set of states, $q \in S$ a state, and $\lambda[q_i, k]$ an arbitrary q-history. We say that a group of agents $\Gamma \subseteq \Sigma$ is* backward responsible *for $S$ based on $\lambda[q_i, k]$ iff:*

1. *There is a state $q_j$ in $\lambda[q_i, k]$ such that for some strategy for $\Gamma$, $F_\Gamma$, all states on all computations in out$(q_j, F_\Gamma)$ belong to $\bar{S}$, and*

2. *$\Gamma$ is minimal, that is, there is no $\Gamma' \subsetneq \Gamma$ with the property formulated above.*

Responsible groups of agents are sometimes also called coalitions.

These definitions look very similar and it has actually been proven that a group $\Gamma \subseteq \Sigma$ is backwards responsible for $S$ given some history if and only if it was forward responsible for $S$ in one of the states on the history [16].

**Example 16.** Lets consider responsibility in the rock-throwing example as described in [8]. In this example two agents, Billy and Suzy are throwing rocks at a bottle. Suzy throws faster than Billy, so if they both throw, Suzy's rock is the one to shatter the bottle. This situation could be modelled in a CGS like in Figure 9. In $q_{0,0}$ the coalition containing both Billy and Suzy is forward responsible for the bottle shattering, neither is individually responsible, because

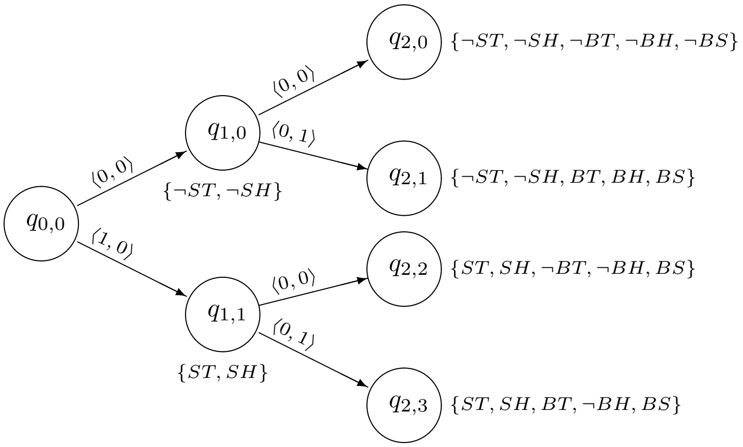


Figure 9: A possible CGS of the rock-throwing example. The variables $ST$ and $BT$ stand for Suzy, respectively Billy, throws. Similarly, $SH$ and $BH$ means Suzy, respectively Billy, hits. $BS$ means that the bottle has shattered. In $q_{0,0}$, Suzy decides to throw or not, in a next state, Billy can decide to throw.

they cannot guarantee that the other will not throw their rock and therefore that the bottle will not be shattered. When we look at backward responsibility it becomes a bit more complicated. Basically, in $q_{2,2}$ and $q_{2,3}$, the coalition is again responsible. Neither can at any point in the history individually assume a strategy that will guarantee that the bottle will not shatter. However, in $q_{2,1}$, Billy is alone backward responsible, as he could have chosen not to throw in $q_{1,0}$.

Yet another approach was introduced in [2]. They consider so-called *causal backward responsibility*, where an agent is held responsible only if changing its actions could have changed the outcome, given that everything else remains fixed, similar to but-for causes. In their paper, they defined responsibility with respect to extensive form games, but it can be naturally translated to the CGS case:

**Definition 22** (Causal Backward Responsibility)**.** *Let $\mathcal{M}$ be a CGS, $S$ be a set of states, $q \in S$ a state, $\lambda[q_i, k]$ an arbitrary q-history, and $F_\Sigma$ be a collective strategy for all agents in $\Sigma$, s.t. $\lambda[q_i, k]$ is contained in $out(q_0, F_\Sigma)$. We say that a group of agents $\Gamma \subseteq \Sigma$ is* causal backward responsible *for $S$ based on $\lambda[q_i, k]$ and $F_\Sigma$ iff:*

1. *There exists a strategy $F_\Gamma$ s.t. for $F'_\Sigma := \{f_a | f_a \in F_\Gamma \text{ if } a \in \Gamma, \text{ else } f_a \in F_\Sigma\}$, all states on all computations in $out(q_0, F'_\Sigma)$ belong to $\bar{S}$, and*

2. *$\Gamma$ is minimal, that is, there is no $\Gamma' \subsetneq \Gamma$ with the property formulated above.*

While this is called causal backward responsibility, this notion is not restricted to only causal CGS, it can be applied in any concurrent game structure.

**Example 17.** When we consider the rock-throwing example again (Figure 9), the coalition $\{Billy, Suzy\}$ is causal backward responsible in the case where they both throw, but only Billy is responsible in the case where only he throws and only Suzy is responsible when Billy does not throw. This is because now it will be guaranteed that the other agent would always have acted the same as it did.

Causal backwards responsibility can be seen a special case of backwards responsibility. The following proposition is stated in a different form in [2] and proven in [3]. As I adapted it to this case, I rewrote the proof.

**Proposition 4.** *Let $\mathcal{M}$ be a CGS, $S$ a set of states, and $q \in S$ a state, $\lambda[q_i, k]$ an arbitrary q-history and $F_\Sigma$ a collective strategy such that this history is contained in $out(q_0, F_\Sigma)$. If the coalition $\Gamma$ is backwards responsible for $S$ based on the history, it contains a coalition $\Gamma' \subseteq \Gamma$ that is causal backward responsible for $S$ based on $\lambda[q_i, k]$.*

*Proof.* Let $\Gamma$ be a backwards responsible coalition, this $\Gamma$ satisfies the first condition of Definition 22. After all, there exists a strategy $F_\Gamma$ such that for a state $q_j$ in $\lambda[q_i, k]$, all states in $out(q_j, F_\Gamma)$ lie in $\bar{S}$, and $q_j$ is a state in $out(q_0, F_\Sigma)$, so when using $F_\Gamma$ and $F_\Sigma$ to create $F'_\Sigma$ like in the definition, it follows that all states on all computations in $out(q_0, F'_\Sigma)$ belong to $\bar{S}$.

If this $\Gamma$ is the minimal set that satisfies this property, it a causal backward responsible. This is however not necessarily true, as a minimal set satisfying the condition of backwards responsibility is not necessarily also a minimal set with respect to the causal backwards responsibility property. However, if $\Gamma$ is not a minimal set it contains a minimal set satisfying the condition which is then a causal responsible coalition, which is what had to be proven. □

### 2.5.2 Distributing Responsibility

In all definitions discussed above, it is possible for a group of agents to be collectively responsible for a state of affairs, but it is sometimes useful to be able to determine how much every individual agent contributed to the result. Chokler and Halpern defined a *degree of responsiblity* for causes of an event [5]. This degree of responsibility depends on how many things have to change until the agent is the sole cause of the event.

**Definition 23** (Degree of Responsibility [5]). *The* degree of responsibility *of $X = x$ for a causal formula $\varphi$ in $(\mathcal{M}, \mathbf{u})$, denoted $dr((\mathcal{M}, \mathbf{u}), (X = x), \varphi)$, is* 0 *if $x = x$ is not a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$; it is $\frac{1}{k+1}$ if $(X = x)$ is a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$ and there exists a partition $(Z, W)$ and setting $(x', \mathbf{w}')$ for which condition AC2 of the original or updated HP definition for causality holds, s.t.*

1. *$k$ variables in $W$ have different values in $\mathbf{w}'$ than they do in the context $\mathbf{u}$, and*
2. *there is no partition $(Z', W')$ and setting $(x'', \mathbf{w}'')$ satisfying AC2, s.t. only $k' < k$ variables have different values in $\mathbf{w}''$ than they do in $\mathbf{u}$*

The drawbacks of this definition are that it only considers responsibility as causality, while most recent work focuses on strategic responsibility, and that the degree of responsibility does not differ between an agent who is part of one coalition of $k$ agents, or an agent who is part of multiple coalitions of $k$ agents [2]. Intuitively, the latter agent is more powerful and hence more responsible. Because of this, [16] and [2] use the Shapley value to distribute responsibility over a group of agents.

**Definition 24** (Responsibility Value [16])**.** *Let $\mathcal{M}$ be a CGS, $S$ a state of affairs, $q \in S$ a state and $\lambda[q_i, k]$ an arbitrary q-history. We define the responsibility game $\mathcal{G}^S_{q,\lambda[q_i,k]} = (\Sigma, \varrho)$ as a cooperative game where for any coalition $\Gamma \subseteq \Sigma$, the game's characteristic function $\varrho(\Gamma) = 1$ if and only if a coalition $\Gamma' \subseteq \Gamma$ is q-responsible for $S$ given $\lambda[q_i, k]$; otherwise $\varrho(\Gamma) = 0$. The q-responsibility value of agent $a \in \Sigma$ for $S$ given $\lambda[q_i, k]$, denoted $\rho^{a.S}_{q,\lambda[q_i,k]}$, is:*

$$\rho^{a,S}_{q,\lambda[q_i,k]} = \sum_{\Gamma \subseteq \Sigma \setminus \{a\}} \frac{|\Gamma|!(|\Sigma| - |\Gamma| - 1)!}{|\Sigma|!} (\varrho(\Gamma \cup \{a\}) - \varrho(\Gamma)).$$

This definition can be applied to any of the earlier defined forms of responsibility.

**Example 18.** In the rock-throwing example the coalition $\{Billy, Suzy\}$ is backwards responsible in $q_{2,3}$ for the bottle not shattering, as explained in Example 16. This means that Suzy has responsibility value:

$$\begin{aligned}\rho^{Suzy,S}_{q_{2,3},\lambda[q_i,k]} =& \frac{|\emptyset|!(|\{Billy, Suzy\}| - |\emptyset| - 1)!}{|\{Billy, Suzy\}|!}(\varrho(\emptyset \cup \{Suzy\}) - \varrho(\emptyset)) + \\ & \frac{|\{Billy\}|!(|\{Billy, Suzy\}| - |\{Billy\}| - 1)!}{|\{Billy, Suzy\}|!} \\ & (\varrho(\{Billy\} \cup \{Suzy\}) - \varrho(\{Billy\})) \\ =& \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1 = \frac{1}{2}\end{aligned}$$

# 3 Problem Definition

The main goal of this project is to formally model, analyse and reason about causal effects in multi-agent settings. In order to achieve this, I will define several sub-problems.

In Section 2.2 I defined structural causal models that are used to formally analyse and model causality and in Section 2.3 concurrent game structures were defined. These are used to model and reason about multi-agent settings. I will use both these concepts to solve the first sub-problem:

*How can causal models be integrated in multi-agent system models, in order to allow us to analyse and reason about the effects of agents' actions on each other and on their shared environment.*

This sub-problem will be solved by using structural causal models, where the endogenous variables are divided in a set of agent variables, controlled by agents, and a complement set of environment variables. I will use this to create a concurrent game structure that can be used to reason about the agents' actions in the multi-agent setting.

In Section 2.2 and Section 2.4.3 I introduced the HP definition of causality and the notion of strategy in concurrent game structures. This is used for the second sub-problem which can be formulated as follows:

*How can we study and analyse the relation between the derived multi-agent system model and the original structural causal model?*

In particular, I will study the relation between agent strategies in this system and causality in the original model, and investigate the existence of a multi-agent strategy that ensures a certain outcome and whether that says anything about the causal relation between variables of the original structural causal model. I will do this by first formalising what states in the derived multi-agent system model represent and then show certain specific relations between strategies in the derived concurrent game structure and causal relations in the original structural causal model

Finally, in Section 2.5 I discussed the notion of strategic responsibility as used for concurrent game structures. This is used for the final sub-problem:

*Can analysing the derived multi-agent system model show a relation between strategic responsibility and causality?*

Since strategic responsibility relies heavily on the concept of strategy in concurrent game structures, the results of the second sub-problem will be used to relate the concepts of strategic responsibility and causality in the HP sense.

# 4 Basing a Concurrent Game Structure on a Causal Model

Gladyshev et al. (2023) have done work on defining a concurrent game structure from a causal model [6]. In their approach, they partition the endogenous causal variables in a set of agent variables $V_a$, variables directly controlled by an agent, and a set of environment variables, $V_e$, that are not directly controlled by an agent. Therefore, $\mathcal{V} = V_a \cup V_e$ and $V_a \cap V_e = \emptyset$. They then define the set of agents of the CGS to be a bijection to the set of agent variables. The states are the causal models that can be achieved through interventions on the agent variables and those interventions are the possible actions.

This approach is useful for some purposes but I did not find it very useful to compare questions of causality and responsibility, because it takes a 'zoomed out' approach to the causal model. Every state contains a whole causal model, while I am interested in the specific variable values, therefore I came up with a different approach where I tried to make the causal nature of the model also apparent in the CGS. A more similar approach to mine was defined in [3], but they use extensive form games and do not distinguish between agent and environment variables.

## 4.1 Defining a Causal Concurrent Game Structure

Before I can define this model I must first define my notion of rank of a causal variable:

**Definition 25** (Rank of a Causal Variable)**.** *A* ranking function *on a causal model $\mathcal{M}$ is a function $f : \mathcal{V} \to \mathbb{Z}_{>0}$, such that for two causal variables $X$ and $Y$, if $X$ is a descendant of $Y$, then $f(X) > f(Y)$ . The* rank *of a causal variable $X$ is $f(X)$.*

*An* agent ranking function *of a causal model $\mathcal{M}$ corresponding to ranking function $f$ is a function $g : \mathcal{V} \to \{0, ..., n\}$, where $n = |\{f(A) \mid A \in V_a\}|$, such that: For all $A, B \in V_a$, $g(A) > g(B) > 0$ if and only if $f(A) > f(B)$ and $g(A) = g(B)$ if and only if $f(A) = f(B)$. For all $X \in V_e$, $g(X) = g(A) - 1$ if $\exists A \in V_a$ such that $f(X) \leq f(A)$ and $\nexists B \in V_a$ such that $f(X) \leq f(B) < f(A)$. If such an $A$ does not exist, i.e. if $f(X) > f(A)$ for all $A \in V_a$, then $g(X) = n$. The* agent rank *of a variable $A \in V_a$ is $g(A)$.*

The idea behind the agent ranking function $g$ is that it 'compresses' the ranking function to only have as many values as there are agents with distinct ranks. This will later be used to determine in which states of the CGS which variables will be updated. A ranking function gives rise to a unique agent ranking function.

A possible ranking function $f$ on a causal model will be the function that assigns to the endogenous variables the number of their 'level', where first-level variables will be variables that do not depend on other endogenous variables, second-level variable only depend on exogenous variables and first-level variables, etc. This satisfies the condition that descendants need to have a higher rank than their ancestors, because variables of rank $n$ will be $n$th-level variables that depend on exogenous variables and variables of level $n-1$ or less. So all ancestors of these rank $n$ variables will have a lower rank.

Another possible ranking function $f'$ will assign rank 1 to the variable(s) with the longest causal path(s) starting with them, rank 2 to the variable(s) with the second-longest causal path(s) starting with them, etc. This also satisfies the condition, because any ancestor of a variable $X$ will have a longer causal path starting from them, after all, the path from the ancestor to $X$ is added to the longest causal path starting at $X$. This puts them lower in the ranking than $X$.

In the following example I will look at these rankings in the context of a specific causal model and determine the corresponding agent ranking functions.

**Example 19.** Lets look at how these rankings work in the rock-throwing example as described in [8], we will use this as a running example from now on. In this example two agents, Billy and Suzy, are throwing rocks at a bottle. Suzy throws faster than Billy, so if they both throw, Suzy's rock is the one to shatter the bottle. The causal model has endogenous variables Suzy throws, $ST$, Billy throws, $BT$, Suzy's rock hits the bottle, $SH$, Billy's rock hits, $BH$ and the bottle shatters, $BS$. All variables can have value 0 or 1. The structural equations are:

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

The causal network is shown in Figure 10. When we apply $f$ to this example,

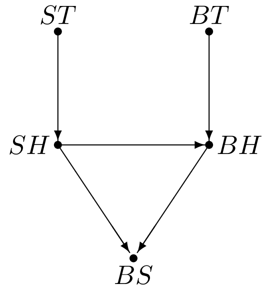


Figure 10: The Causal network for the Rock-Throwing example.

$ST$ and $BT$ get rank 1, $SH$ gets rank 2, $BH$ gets rank 3, and $BS$ gets agent rank 4. The agent variables are $ST$ and $BT$, so the agent ranking function $g$ would give them both rank 1. All other variables would also get rank 1, as there is no agent variable that has a higher or equal rank to $SH, BH$ or $BS$. So all of them get the same agent rank as the highest agent variable, which is 1. For $f'$, the values are: $ST$ gets 1, $BT$ and $SH$ get rank 2, $BH$ gets rank 3 and and $BS$ gets rank 4. Now the agent ranking function $g'$ would give $ST$ agent rank 1 and $BT$ agent rank 2. For the environment variables, $SH$ will get agent rank $g'(BT) - 1 = 1$, as $f'(BT) \geq f'(SH) = 2$. The other two will get agent rank 2, because they both have higher ranks than the highest agent variable.

Now everything is in order to define causal concurrent game structures.

**Definition 26** (Causal CGS)**.** *Given a causal model, $\mathcal{M}$, where the variables can only attain finitely many values, a set of agent variables that is a subset of the endogenous $V_a \subseteq \mathcal{V}$, and where each agent variable depends on exactly one exogenous variable, a context $\mathbf{u}$, and a ranking function $f$ with corresponding agent ranking function $g$. Let $\mathcal{I}$ be a set of initial values for all variables according to the setting $(\mathcal{M}, \mathbf{u})$. A* causal concurrent game structure *is defined as a tuple $GS = \langle k, Q, d, \delta, \Pi, \pi \rangle$ with:*

- *$k = |V_a|$, every agent only controls one agent variable.*
- *A starting state $q_{0,0}$, with the rest of the states being recursively defined as follows: For every combination of values of all variables of agent rank 1 in $V_a$, define a state $q_{1,j}$, $j$ starting at 0 and counting up. Then for every combination of a $q_{1,j}$ state and values of all the variables with agent rank 2, define a state $q_{2,j}$, continue this until the variables with the highest agent rank are considered. In general the number of states $q_{i,j}$ for any $j > 0$ will be:*

$$|\{q_{i,j}| \textit{ for all values of } j\}| = \prod_{\substack{Y \in V_a, \\ g(Y) \leq j}} |R(Y)|.$$

- *$d_{a^X}(q_{i,j}) = \mathcal{R}(X)$ if and only if $a^X$ is an agent controlling variable $X$ and $X$ has agent rank $i+1$. Otherwise $d_{a^X}(q_{i,j}) = \emptyset$. If $d_{a^X}(q_{i,j}) \neq \emptyset$, then an action for agent $a^X$ in state $q_{i,j}$ is denoted $a^X_{i,j} = x$ with $x \in \mathcal{R}(X)$, $X$ being the agent variable that $a$ controls. Else the action is denoted $a^X_{i,j} = 0$.*
- *The state following from the move vector $(a^{X_1}_{i,j} = x_1, ..., a^{X_k}_{i,j} = x_k)$ is defined as $\delta(q_{i,j}, a^{X_1}_{i,j} = x_1, ..., a^{X_k}_{i,j} = x_k) = q_{i+1,j'}$, with:*

  $$j' \in \{j \cdot \#choices, j \cdot \#choices + 1, ..., j \cdot \#choices + (\#choices - 1)\},$$

  *where* $\#choices = \prod_{\substack{Y \in V_a, \\ g(Y) = i+1}} |R(Y)|.$

  *For $i = \max_{Y \in V_a} g(Y)$, the transition $\delta(q_{i,j}, a^{X_1}_{i,j} = 0, ..., a^{X_k}_{i,j} = 0) = q_{i,j}$ for all $j$. Note that this is the only possible move vector for any state with the maximal $i$, because there are no variables of agent rank higher than $i$.*
- *$\Pi = \{X = x|$ for all $X \in V$ and all $x \in \mathcal{R}(X)\}$.*

- *The valuation of each state is defined recursively as:*

$$\begin{aligned} &\pi(q_{0,0}) && = \mathcal{I} \\ &\pi(\delta(q_{i,j}, a_{i,j}^{X_1} = x_1, ..., a_{i,j}^{X_k} = x_k)) && = (\pi(q_{i,j}) \backslash V_{i+1}^{-}) \cup V_{i+1}^{a} \cup V_{i+1}^{e}, \end{aligned}$$

*where*

$$\begin{aligned} V_{i+1}^{-} \quad &= \quad \{X = x \mid g(X) = i+1, (X = x) \in \pi(q_{i,j})\}; \\ V_{i+1}^{a} \quad &= \quad \{X_k = x_k \mid X_k \in V_a \textit{ and } g(X) = i+1\};^{3} \\ V_{i+1}^{e} \quad &= \quad \bigcup_{n=f(A)+1}^{n'} V_{i+1}^{e,n}, \textit{ where } A \in V_a \textit{ with } g(A) = i+1 \textit{ and } n' = f(B), \\ & \qquad \textit{if } \exists B \in V_a \textit{ such that } g(B) = i+2, \textit{ else } n' = \max_{Y \in \mathcal{V}} f(Y) \\ V_{i+1}^{e,n} \quad &= \{X = x \mid X \in V_e \textit{ and } f(X) = n, \\ & \quad x = \mathcal{F}_X((\pi(q_{i,j}) \backslash V_{i+1}^{-}) \cup V_{i+1}^{a} \cup V_{i+1}^{e,f(A)+1} \cup ... \cup V_{i+1}^{e,n-1})\}. \end{aligned}$$

The intuition behind this definition is that agent variables that are earlier on a causal path will earlier get to take an action, to avoid conflicts. The CGS is defined with respect to a context, because in certain models, the agent variables do not fully determine the values of the other variables. The evaluation of the states is defined recursively. After each transition, only a small number of the variable values get updated, the rest stays the same as in the previous state. This is reflected by the fact that we remove the variables that will get updated from the valuation of the previous state and then add the new values. $V_{i+1}^{a}$ contains the values for the agent variables that are currently being changed. $V_{i+1}^{e}$ contains the values of the environment variables that get changed due to the change in agent variables. This set is divided in several smaller sets, because environment variables can also depend on each other, making it necessary to first determine those of the lowest rank, than the slightly higher rank ones, and so on.

One thing to note is that the fact that all states $q_{i,j}$ can only transition to states $q_{i+1,j'}$, will lead to the CGS having a tree structure. It is impossible to return to an earlier state and every node can only branch out.

**Example 20.** We will consider how to build a CGS from the causal model for the Rock-Throwing example. The agent variables are $ST$ and $BT$. We start with the setting $(\mathcal{M}, \mathbf{u})$ with $\mathbf{u} = (U_{ST} = 0, U_{BT} = 0)$ giving $\mathcal{I} = \{ST = 0, BT = 0, SH = 0, BH = 0, BS = 0\}$. Given this, let us determine the causal CGS with $f$ as defined in Example 19.

- $k = |V_a| = |\{ST, BT\}| = 2$
- $Q = \{q_{0,0}, q_{1,0}, q_{1,1}, q_{1,2}, q_{1,3}\}$, because there are 4 combinations of variables of agent rank 1, $(ST = 0, BT = 0), (ST = 0, BT = 1), (ST = 1, BT = 0), (ST = 1, BT = 1)$ and not variables with a higher agent rank.

[3] Recall that the $x_k$ comes from the move vector

- $d_{a^{ST}}(q_{0,0}) = \mathcal{R}(ST) = \{0,1\}$, $d_{a^{BT}}(q_{0,0}) = \mathcal{R}(BT) = \{0,1\}$ and $d_{a^{ST}}(q_{1,j}) = d_{a^{BT}}(q_{1,j}) = \emptyset,\ \forall j \in \{0,1,2,3\}$. Since $ST$ and $BT$ both have agent rank 1, so their agents only get to take an action in $q_{0,0}$.

- $\delta(q_{0,0}, a^{ST}_{0,0} = 0, a^{BT}_{0,0} = 0) = q_{1,0}$,
  $\delta(q_{0,0}, a^{ST}_{0,0} = 0, a^{BT}_{0,0} = 1) = q_{1,1}$,
  $\delta(q_{0,0}, a^{ST}_{0,0} = 1, a^{BT}_{0,0} = 0) = q_{1,2}$,
  $\delta(q_{0,0}, a^{ST}_{0,0} = 1, a^{BT}_{0,0} = 1) = q_{1,3}$,
  $\delta(q_{1,j}, a^{ST}_{1,j} = 0, a^{BT}_{1,j} = 0) = q_{1,j},\ \forall j \in \{0,1,2,3\}$. Since $a^{ST}$ and $a^{BT}$ cannot choose anymore in states $q_{1,j}$, we denote this inaction with 0.

- $\Pi = \{ST = 0, ST = 1, BT = 0, BT = 1, SH = 0, SH = 1, BS = 0, BS = 1\}$.

- $\pi(q_{0,0}) = \mathcal{I}$,
  $\pi(q_{1,0}) = \pi(\delta(q_{0,0}, a^{ST}_{0,0} = 0, a^{BT}_{0,0} = 0)) = (\pi(q_{0,0}) \backslash V_1^-) \cup V_1^a \cup V_1^e$, where
  $V_1^- = \{ST = 0, BT = 0, SH = 0, BH = 0, BS = 0\}$, as all variables have agent rank 1,
  $V_1^a = \{ST = 0, BT = 0\}$, as $a^{ST}_{0,0} = 0$ and $a^{BT}_{0,0} = 0$.
  $V_1^e = \bigcup_{n=2}^{n'} V_1^{e,n}$, where $n' = \max_{Y \in \mathcal{V}} f(Y) = 4$, now
  - $V_1^{e,2} = \{SH = \mathcal{F}_{SH}(ST = 0, BT = 0) = 0\}$
  - $V_1^{e,3} = \{BH = \mathcal{F}_{BH}(ST = 0, BT = 0, SH = 0) = 0\}$
  - $V_1^{e,4} = \{BS = \mathcal{F}_{BS}(ST = 0, BT = 0, SH = 0, BH = 0) = 0\}$, so
  $\pi(q_{1,0}) = (\{ST = 0, BT = 0, SH = 0, BH = 0, BS = 0\} \backslash \{ST = 0, BT = 0, SH = 0, BH = 0, BS = 0\}) \cup \{ST = 0, BT = 0\} \cup (\{SH = 0\} \cup \{BH = 0\} \cup \{BS = 0\} = \{ST = 0, BT = 0, SH = 0, BH = 0, BS = 0\}$.
  Now, for $\pi(q_{1,1}) = \pi(\delta(q_{0,0}, a^{ST}_{0,0} = 0, a^{BT}_{0,0} = 1))$, we have that $V_1^-$ is the same as above, as the rank is still the same. The difference starts with $V_1^a$, we now have that $V_1^a = \{ST = 0, BT = 1\}$, as $a^{ST}_{0,0} = 0$ and $a^{BT}_{0,0} = 1$. $V_1^e$ is still the union of $V_1^{e,2}, V_1^{e,3}$ and $V_1^{e,4}$, but these sets are now: - $V_1^{e,2} = \{SH = \mathcal{F}_{SH}(ST = 0, BT = 1) = 0\}$
  - $V_1^{e,3} = \{BH = \mathcal{F}_{BH}(ST = 0, BT = 1, SH = 0) = 1\}$
  - $V_1^{e,4} = \{BS = \mathcal{F}_{BS}(ST = 0, BT = 1, SH = 0, BH = 1) = 1\}$, so
  $\pi(q_{1,1}) = (\{ST = 0, BT = 0, SH = 0, BH = 0, BS = 0\} \backslash \{ST = 0, BT = 0, SH = 0, BH = 0, BS = 0\}) \cup \{ST = 0, BT = 1\} \cup (\{SH = 0\} \cup \{BH = 1\} \cup \{BS = 1\} = \{ST = 0, BT = 1, SH = 0, BH = 1, BS = 1\}$.
  The valuation for the other two states is done similar, so that $\pi(q_{1,2}) = \{ST = 1, BT = 0, SH = 1, BH = 0, BS = 1\}$ and $\pi(q_{1,3}) = \{ST = 1, BT = 1, SH = 1, BH = 1, BS = 1\}$.

The corresponding graph is given in Figure 11.

Let us now consider the causal CGS for $f'$ as defined in Example 19. This means we will now start with just Suzy being able to make a decision.

From now on if a variable like $ST$ can attain only two values I will write $ST$ and $\neg ST$ instead of $ST = 1$ and $ST = 0$ respectively.

- $k$ and $\Pi$ are the same as in the above example.

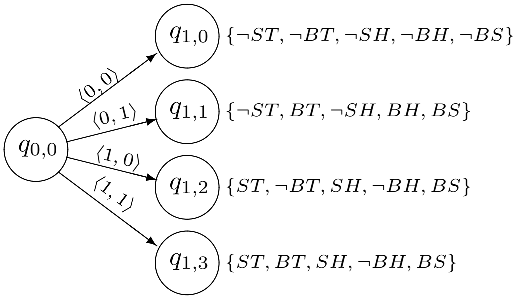

$q_{0,0}$
$\langle 0,0\rangle$
$\langle 0,1\rangle$
$\langle 1,0\rangle$
$\langle 1,1\rangle$
$q_{1,0}$ $\{\neg ST, \neg BT, \neg SH, \neg BH, \neg BS\}$
$q_{1,1}$ $\{\neg ST, BT, \neg SH, BH, BS\}$
$q_{1,2}$ $\{ST, \neg BT, SH, \neg BH, BS\}$
$q_{1,3}$ $\{ST, BT, SH, \neg BH, BS\}$

Figure 11: The causal CGS of the rock-throwing example, using ranking function $f$. As a convention I will not show the initial values in the starting state, nor any values that are not changed in a state, except for the leaf states. I also do not show the transitions to the same state in the leaf states.

- $Q = \{q_{0,0}, q_{1,0}, q_{1,1}, q_{2,0}, q_{2,1}, q_{2,2}, q_{2,3}\}$, as $|\{q_{1,j}|\forall j\}| = \prod_{Y \in \{ST\}} |\mathcal{R}(Y)| = |\{0,1\}| = 2$ and $|\{q_{2,j}|\forall j\}| = \prod_{Y \in \{ST,BT\}} |\mathcal{R}(Y)| = |\{0,1\}| \cdot |\{0,1\}| = 4$. These are all states, because the highest agent rank value of $g'$ is 2.
- The available actions for each agent in each state are:

$$\begin{array}{lll} d_{a^{ST}}(q_{0,0}) = \{0,1\}, & d_{a^{BT}}(q_{0,0}) = \emptyset, & \\ d_{a^{ST}}(q_{1,j}) = \emptyset, & d_{a^{BT}}(q_{1,j}) = \{0,1\}, & \forall j \in \{0,1\} \\ d_{a^{ST}}(q_{2,j}) = \emptyset, & d_{a^{BT}}(q_{2,j}) = \emptyset, & \forall j \in \{0,1,2,3\}. \end{array}$$

- The transitions are:
  $\delta(q_{0,0}, a^{ST}_{0,0} = 0, a^{BT}_{0,0} = 0) = q_{1,0}$,
  $\delta(q_{0,0}, a^{ST}_{0,0} = 1, a^{BT}_{0,0} = 0) = q_{1,1}$,
  $\delta(q_{1,0}, a^{ST}_{1,0} = 0, a^{BT}_{1,0} = 0) = q_{2,0}$,
  $\delta(q_{1,0}, a^{ST}_{1,0} = 0, a^{BT}_{1,0} = 1) = q_{2,1}$,
  $\delta(q_{1,1}, a^{ST}_{1,1} = 0, a^{BT}_{1,1} = 0) = q_{2,2}$,
  $\delta(q_{1,1}, a^{ST}_{1,1} = 0, a^{BT}_{1,1} = 1) = q_{2,3}$,
  $\delta(q_{2,j}, a^{ST}_{2,j} = 0, a^{BT}_{2,j} = 0) = q_{1,j}$, $\forall j \in \{0,1,2,3\}$.
- For the valuations we have again that $\pi(q_{0,0}) = \mathcal{I}$, as by definition. To determine $\pi(q_{1,0})$, we determine $V_1^- = \{\neg ST, \neg SH\}$, as both $g'(ST)$ and $g'(SH)$ equal 1. $V_1^a = \{\neg ST\}$, as $a^{ST}_{0,0} = 0$. $V_1^e = V_1^{e,2}$, as $n' = 2$, since $g'(BT) = 2$ and $f'(BT) = 2$ as well. Now, the only environment variable of rank 2 is $SH$, and so $V_1^e = \{\neg SH\}$, as $\mathcal{F}_{SH}((\mathcal{I} \backslash \{\neg ST, \neg SH\}) \cup \{\neg ST\}) = 0$. Putting this together gives $\pi(q_{1,0}) = \{\neg ST, \neg BT, \neg SH, \neg BH, \neg BS\}$. The valuation of $\pi(q_{1,1})$ is done very similar. $V_1^-$ is the same as above. $V_1^a$ is now $\{ST\}$, as $a^{ST}_{0,0} = 1$. $V_1^e$ again equals $V_1^{e,2}$, which is now $\{SH\}$. So

$\pi(q_{1,1}) = (\mathcal{I}\backslash\{\neg ST, \neg SH\})\cup\{ST\}\cup\{SH\} = \{ST, \neg BT, SH, \neg BH, \neg BS\}$.
Lets now look at $\pi(q_{2,1}) = \pi(\delta(q_{0,0}, a^{ST}_{0,0} = 0, a^{BT}_{0,0} = 1))$. $V_2^- = \{\neg BT, \neg BH, \neg BS\}$, since all these variables have agent rank 2. $V_2^a = \{BT\}$, as $BT$ is the only agent variable of agent rank 2 and $a^{BT}_{0,0} = 1$. $V_2^e = V_2^{e,3} \cup V_2^{e,4}$. There is no agent variable with a higher agent rank than 2 and the maximum value of $f'$ is 4. We have that $V_2^{e,3} = \{BH\}$, since $\mathcal{F}_{\mathcal{BH}}(\{\neg ST, \neg SH, BT\}) = 1$, and $V_2^{e,4} = \{BS\}$, since $\mathcal{F}_{\mathcal{BS}}(\{\neg ST, \neg SH, BT, BH\}) = 1$.
So, $\pi(q_{2,1}) = (\{\neg ST, \neg BT, \neg SH, \neg BH, \neg BS\}\backslash\{\neg BT, \neg BH, \neg BS\})\cup\{BT\}\cup \{BH\} \cup \{BS\} = \{\neg ST, BT, \neg SH, BH, BS\}$. The valuation of $q_{2,0}, q_{2,2}$ and $q_{2,3}$ is done similar. So $\pi(q_{2,0}) = \{\neg ST, \neg BT, \neg SH, \neg BH, \neg BS\}$, $\pi(q_{2,2}) = \{ST, \neg BT, SH, \neg BH, BS\}$ and $\pi(q_{2,3}) = \{ST, BT, SH, BH, BS\}$.

The graph corresponding to this is given in Figure 12.

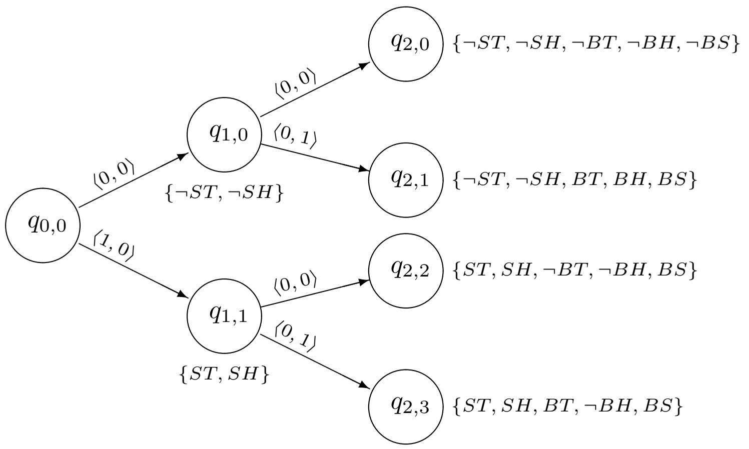


Figure 12: The causal CGS of the rock-throwing example using ranking function $f'$. Again I do not show the initial values in the starting state, nor any values that are not changed in a state. I also do not show the transitions to the same state in the leaf states.

So, in the second case, the agents that (indirectly) influence most variables get to take an action first, while in the first case, the agents that are influenced by the smallest number of variables get to go first.

Some other examples taken from [8]:

**Example 21.** Consider the fighter planes example from [8]. In this example Billy and Suzy pilot fighter planes, with the mission to destroy a target. Billy's task is to shoot any enemies that might show up to shoot Suzy, Suzy's task is to bomb the target. The Causal model is described by the causal variables $BGU$

(Billy goes up), $ESU$ (enemy shows up), $BPT$ (Billy pulls trigger), $EE$ (enemy eludes Billy), $ESS$ (enemy shoots Suzy), $SBT$ (Suzy bombs target) and $TD$ (target destroyed), which can all be true or false, with structural equations:

- $BPT = BGU \wedge ESU$
- $EE = ESU \wedge \neg BPT$
- $ESS = EE$
- $SBT = \neg ESS$
- $TD = SBT$.

The causal network is given in Figure 13.

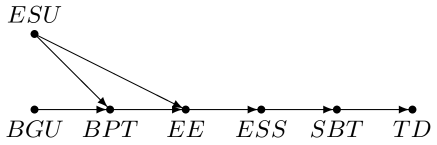


Figure 13: The causal network for the fighter planes example.

In this example we have that $V_a = \{BGU, ESU, SBT\}$, as those are the first actions of each agent, and we could assume the enemy will also try to shoot Suzy if they show up and that Billy will actually shoot the enemy if he gets the chance. Let $\mathcal{I} = \{\neg BGU, \neg ESU, \neg BPT, \neg E, \neg ESS, \neg SBT, \neg TD\}$.

Now, note that both ranking functions we defined earlier give the same ranking. The rank of $BGU$ and $ESU$ is 1, the rank of $BPT$ is 2, the rank of $EE$ is 3, for $ESS$ it is 4, for $SBT$ it is 5 and the rank of $TB$ is 6. The corresponding agent rank of $BGU, ESU, BPT, EE$ and $ESU$ is 1 and for $SBT$ and $TD$ it is 2. Because of this, we only get one concurrent game structure as given in Figure 14.

I will not fully write out the whole specification of the causal CGS, but I will show how the evaluation of $q_{2,3}$ is determined:

$\pi(q_{2,3}) = \pi(\delta(q_{1,1}, a^{BGU}_{1,1} = 0, a^{ESU}_{1,1} = 0, a^{SBT}_{1,1} = 1)) = (\{\neg BGU, ESU, \neg BPT, EE, ESS, \neg SBT, \neg TD\} \backslash V_2^-) \cup V_2^a \cup V_2^e$. In this case, $V_2^- = \{\neg SBT, \neg TD\}$, as those are the two variables of agent rank 2. $V_2^a = \{SBT\}$, because $a^{SBT}_{1,1} = 1$. $V_2^e = V_2^{e,6} = \{\neg TD\}$, as $0 = \mathcal{F}_{TD}(\{\neg BGU, ESU, \neg BPT, EE, ESS, \neg SBT\})$. Therefore, $\pi(q_{2,3}) = (\{\neg BGU, ESU, \neg BPT, EE, ESS, \neg SBT, \neg TD\} \backslash \{\neg SBT, \neg TD\}) \cup \{\neg SBT\} \cup \{\neg TD\} = \{\neg BGU, ESU, \neg BPT, EE, ESS, \neg SBT, \neg TD\}$. Which is what is pictured in Figure 14.

The following example shows why we need to define the CGS in general given a context.

**Example 22.** Lets now consider the railroad switch example from [8]. In this example an agent is capable of flipping a switch that determines on which track

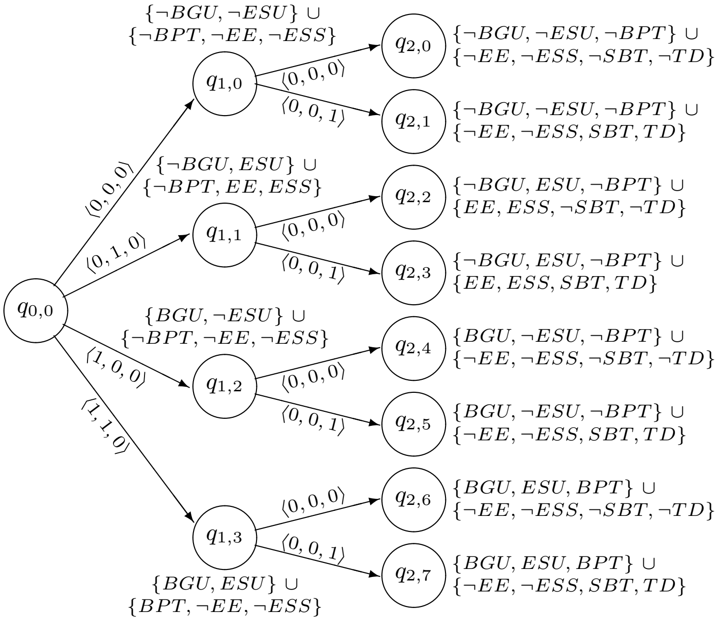


Figure 14: The causal CGS for the fighter planes example. The full evaluations of the starting state and states $q_{1,j}$ are again not pictured.

a train will continue. The train will standard go down the left track, but will follow the right track if the switch is flipped. Though both tracks arrive at the same destination, one of them can be blocked. This is modelled with four variables, $LB$, left track blocked, $RB$, right track blocked, $F$, for switch flipped and $A$, train arrives at destination. The only relevant structural equation is:

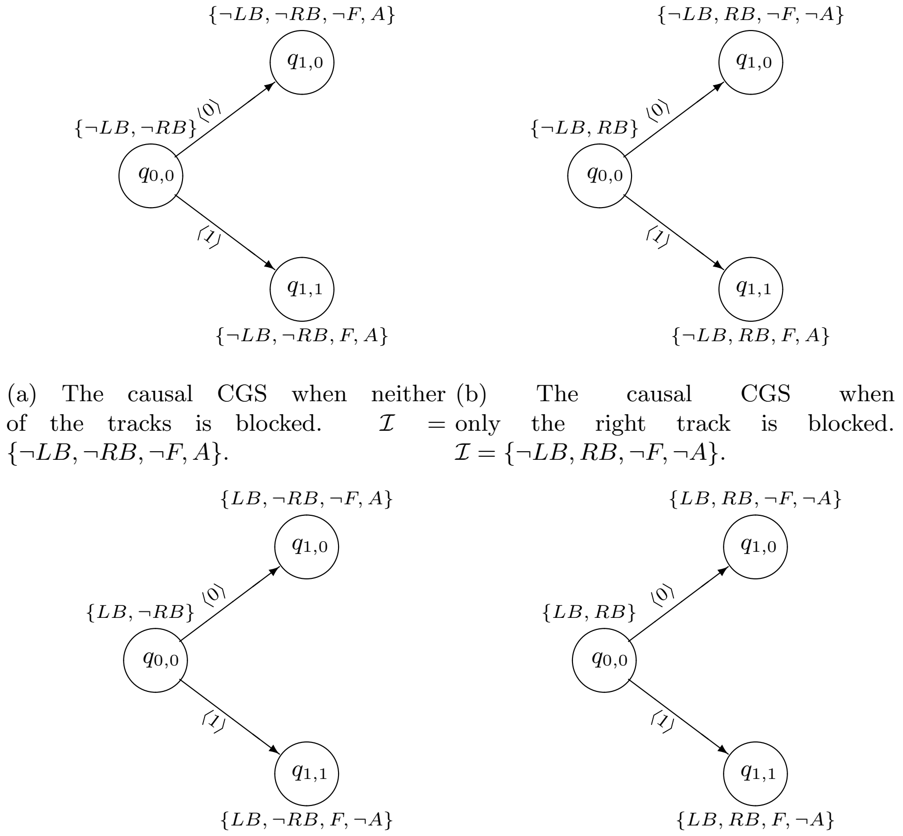
$$A = (F \land \neg LB) \lor (\neg F \land \neg RB).$$

$F$ is the only agent variable, but to determine the effect of $F$, it is essential to take the context into account. The concurrent game structures for all four possible contexts are given in Figure 15.

## 4.2 Properties of Causal Concurrent Game Structures

I already mentioned that a causal CGS has a tree structure, because of this, I will call states $q_{i,j}$, with $i = \max_{X \in \mathcal{V}} g(X)$, with $g$ the used agent ranking function, *leaf-states*.

I will call actions in states where an agent does not control a variable, i.e. $a^X_{i,j} = 0$, with $g(X) = i + 1$, trivial actions. It is also useful to define an *action path* for a state $q_{i,j}$, that contains all the non-trivial actions that led to the state. In other words, only the actions that agents took in a state where



(a) The causal CGS when neither of the tracks is blocked. $\mathcal{I} = \{\neg LB, \neg RB, \neg F, A\}$.

(b) The causal CGS when only the right track is blocked. $\mathcal{I} = \{\neg LB, RB, \neg F, \neg A\}$.

(c) The causal CGS when only the left track is blocked. $\mathcal{I} = \{LB, \neg RB, \neg F, A\}$.

(d) The causal CGS when both tracks are blocked. $\mathcal{I} = \{LB, RB, \neg F, \neg A\}$.

Figure 15: All four possible causal CGS for the railroad switch example.

they could actually take an action. I will denote this set of actions as $\alpha[q_{i,j}]$. Formally, for $0 \leq n \leq k$, an action $(a^{X_n}_{i',j'} = x) \in \alpha[q_{i,j}]$ if and only if $q_{i',j'} \in \lambda[q_{i,j}, i]$ (the history of $q_{i,j}$) and there exists a move vector containing this action: $(a^{X_1}_{i',j'} = x_1, ..., a^{X_n}_{i',j'} = x, ..., a^{X_k}_{i',j'} = x_k)$, such that $\delta(q_{i',j'}, a^{X_1}_{i',j'} = x_1, ..., a^{X_n}_{i',j'} = x, ..., a^{X_k}_{i',j'} = x_k) \in \lambda[q_{i,j}, i]$. In other words, an action is on the action path for a state $q_{i,j}$, if the state in which the action is taken lies on the history of $q_{i,j}$, and the successor of this state on the history can be reached when taking this action.

The statement in the lemma below is a direct consequence of the way the valuation of states are determined in a causal CGS. It states that a variable can only be assigned a new value once on a computation in a causal CGS. After a change of value, the variable will keep that value in all following states of the computation.

**Lemma 1.** *Given a causal CGS based on the causal model $\mathcal{M}$ with ranking function $f$ and corresponding agent ranking function $g$. For any causal variable $X \in \mathcal{V}$ of $\mathcal{M}$, with $g(X) = i$, it holds that $(X = x) \in \pi(q_{i,j})$ for some state $q_{i,j}$, if and only if $(X = x) \in \pi(q_{i',j'})$ for all states $q_{i',j'}$ that are descendants of $q_{i,j}$.*

*Proof.* In state $q_{i,j}$, only propositions with variables of agent rank $i$ are changed, all other variables will have the same value as in the previous state by definition. In particular propositions with variables of a lower agent rank remain unchanged. This means that if $(X = x) \in \pi(q_{i,j})$, then the value of $X$ will not change in any descendant states $q_{i',j'}$ of $q_{i,j}$, after all, $i' > i$, and all variables with agent rank less than $i'$ remain unchanged, so in particular those variables with agent rank $i$ and hence $X = x \in \pi(q_{i',j'})$ for all states $q_{i',j'}$ that are descendant of $q_{i,j}$.

If $(X = x) \in \pi(q_{i',j'})$ for all states $q_{i',j'}$ that are descendants of $q_{i,j}$, then in none of those states, the value of $X$ was changed, as only variables of agent rank $i'$ are changed in states $q_{i',j'}$ and $X$ has agent rank $i$. That means that the $X$ must have the same value in $q_{i,j}$, so $(X = x) \in \pi(q_{i,j})$. □

**Definition 27** (Correspondence)**.** *We say that a state $q_{i,j}$ of a causal CGS* corresponds *to a causal setting $(\mathcal{M}, \mathbf{u})$ if for all causal variables $X$ of $\mathcal{M}$, $(X = x) \in \pi(q_{i,j})$ if and only if $(\mathcal{M}, \mathbf{u}) \models X = x$[4].*

I will also sometimes say that a causal setting $(\mathcal{M}, \mathbf{u})$ corresponds to a state $q_{i,j}$ of a causal CGS and mean the same thing.

Before we can state our main result, we first need to proof the following lemma.

**Lemma 2.** *Given a causal CGS based on a causal model $\mathcal{M} = (\mathcal{S}, \mathcal{F})$ with $\mathcal{S} = (\mathcal{U}, \mathcal{V}, \mathcal{R})$, context $\mathbf{u}$ and ranking function $f$, with corresponding agent ranking function $g$. Let $X \in \mathcal{V}$ be any causal variable of $\mathcal{M}$, with $g(X) \leq i$, then for all states $q_{i,j}$ of the causal CGS, $(X = x) \in \pi(q_{i,j})$ if and only if $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models X = x$, where $\vec{Y} \leftarrow \vec{y} = \{Y \leftarrow y \mid Y \in V_a \text{ and } (a^Y_{i,j} = y) \in \alpha[q_{i,j}]\}$, with $\alpha[q_{i,j}]$ the action path for $q_{i,j}$.*

*Proof.* We have $X \in \mathcal{V}$ with $g(X) \leq i$. If $X \in V_a$, let $(X = x) \in \pi(q_{i,j})$, as $X \in V_a$, it holds that the action $a^X_{i,j} = x$ was taken on the transition to $q_{i,j}$, so $(a^X_{i,j} = x) \in \alpha[q_{i,j}]$. That means that $(X \leftarrow x) \in \vec{Y} \leftarrow \vec{y}$ and that hence $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models X = x$ by the definition of an intervention. That proves one direction for agent variables.

Now suppose $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models X = x$, $X$ has to be in $\vec{Y}$, since $\vec{Y}$ contains all agent variables of agent rank less or equal than $i$, since all agents with agent variables with agent rank $i$ get to take an action in the states $q_{i-1,j'}$. As $\vec{Y}$ contains $X$ and $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models X = x$, it holds that $(X \leftarrow x) \in \vec{Y} \leftarrow \vec{y}$. This

[4] So the causal variable $X$ has value $x$ in the causal setting $(\mathcal{M}, \mathbf{u})$.

means that $\alpha[q_{i,j}]$ contains $a^X_{i-1,j'} = x$, which implies that $(X = x) \in V_i^a$ for this state, and hence $(X = x) \in \pi(q_{i,j})$. That proofs that the statement holds for agent variables.

Now suppose $X \in V_e$ instead. I am going to prove the statement by induction on the rank of $X$, $f(X)$:

Base Step Let $f(X) = 1$, then $g(X) = 0$ by definition of agent rank. First suppose that $(X = x) \in \pi(q_{i,j})$. Since $X$ has agent rank 0, by Lemma 1 it holds that $(X = x) \in \pi(q_{0,j'})$ for some state $q_{0,j'}$, but there is only one such state, $q_{0,0}$. So, given that $(X = x) \in \pi(q_{0,0})$, we have that $(X = x) \in \mathcal{I}$. We have that $(\mathcal{M}, \mathbf{u}) \vDash \mathcal{I}$, and hence $(\mathcal{M}, \mathbf{u}) \vDash X = x$. Since $X$ has rank 1, it does not depend on any other variables and its value will not change given interventions on other variables. Therefore $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \vDash X = x$ as well.

Now, for the other direction, suppose that $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \vDash X = x$, again, since $X$ has rank 1, it must have the same value in $(\mathcal{M}, \mathbf{u})$. Since $\mathcal{I}$ has a value for every endogenous variable of $\mathcal{M}$ and $(\mathcal{M}, \mathbf{u}) \vDash \mathcal{I}$, we have that $(X = x) \in \mathcal{I} = \pi(q_{0,0})$. By Lemma 1 it follows that $(X = x) \in \pi(q_{i,j})$ as well.

Induction Hypothesis Suppose that there is a $n$ such that for $f(X) \leq n$ with $g(X) \leq i$, it holds that for all states $q_{i,j}$, that $(X = x) \in \pi(q_{i,j})$ if and only if $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \vDash X = x$.

Induction Step I need to show that it holds for an $X'$, with $f(X') = n + 1$. There are two cases, either the agent rank of $X'$ is the same as for $X$ from the induction hypothesis (IH) or $g(X') = g(X) + 1$. I will first consider the first case, without loss of generality, we can say that the agent rank of $X'$ is $i$, the prove stays the same for a lower rank. First suppose that $(X' = x') \in \pi(q_{i,j})$, that means that $(X' = x') \in V_i^e$ and specifically in $V_i^{e,n+1}$, so $x' = \mathcal{F}_X((\pi(q_{i,j}) \backslash V_i^-) \cup V_i^a \cup V_i^{e,f(A)+1} \cup ... \cup V_1^{e,n})$. By the IH, we have that the values for all variables of rank less or equal to $n$ in $\pi(q_{i,j})$ follow from $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u})$, so $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \vDash V_i^a \cup V_i^{e,f(A)+1} \cup ... \cup V_1^{e,n}$ (for $V_i^a$ this follows from the first part of the proof), this also holds for those variables in $(\pi(q_{i,j}) \backslash V_i^-)$ of rank less or equal $n$. Since $X'$ by definition of rank only depends on variables of rank less or equal to $n$, it must also hold that $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \vDash X' = x'$.

Now for the other direction, if $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \vDash X' = x'$, I need to show that $(X' = x') \in V_i^{e,n+1}$. By the IH, we have that all variables that $X$ depends on in $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u})$ are in $\pi(q_{i,j})$, since their rank must be smaller than than the rank of $X'$. Lets denote this set of variables that $X'$ depends on with $\vec{Z}$. $\vec{Z} \subset (\pi(q_{i,j}) \backslash V_i^-) \cup V_i^a \cup V_i^{e,f(A)+1} \cup ... \cup V_1^{e,n}$, so $x' = F_{X'}((\pi(q_{i,j}) \backslash V_i^-) \cup V_i^a \cup V_i^{e,f(A)+1} \cup ... \cup V_1^{e,n})$ and hence

$(X' = x') \in \pi(q_{i,j})$.

Now to prove the second case. We can again assume without loss of generality that $g(X') = i$ and now $g(X) = i - 1$. If $(X' = x') \in \pi(q_{i,j})$, then $(X' = x') \in V_i^{e,n+1}$, but now $n$ is also the rank of any agent variable $A$ such that $g(A) = i$, so $x' = \mathcal{F}_{X'}(\pi(q_{i-1,j'} \backslash V_i^- \cup V_i^a)$, where $q_{i-1,j'}$ is the parent state of $q_{i,j}$. The first half of this proof has already shown that $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models V_i^a$ and the IH shows that the variables of $\pi(q_{i-1,j'} \backslash V_i^-$ with rank less or equal to $n$ also follow from $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u})$ and since $X'$ only depends on those variables with rank less or equal to $n$, we have that it must also hold that $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models X' = x'$.

Finally, suppose that $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models X' = x'$, we must show that $(X' = x') \in V_i^{e,n+1}$, in other words, that $x' = \mathcal{F}_{X'}((\pi(q_{i-1,j'}) \backslash V_i^-) \cup V_i^a)$. All variables that $X'$ depends on in $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u})$ have rank $n$ or less, hence for the environment variables $Z^e$ that $X'$ depends on, the IH implies that if for any of those $Z^e$, it holds that if $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models Z^e = z$, then $(Z^e = z) \in \pi(q_{i-1,j'})$. Since all those variables $Z^e$ have rank $n$ or less, $g(Z^e) < i$ and hence $(Z^e = z) \in \pi(q_{i-1,j'}) \backslash V_i^-$ as well. For the agent variables $Z^a$ that $X'$ depends on, it is shown above that if $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models Z^a = z'$, then if $g(Z^a) < i$, $(Z^a = z') \in \pi(q_{i-1,j'}) \backslash V_i^-$ and if $g(Z^a) = i$, then $(Z^a = z') \in V_i^a$. Hence all the variable-value combinations that $X' = x'$ depends on are in $(\pi(q_{i-1,j'}) \backslash V_i^-) \cup V_i^a$ and hence it must hold that $x' = \mathcal{F}_{X'}((\pi(q_{i-1,j'}) \backslash V_i^-) \cup V_i^a)$.

□

The following proposition actually follows directly from Lemma 2, however it is the most important result as we will use it in the next section when we define the notion of causality in causal CGS.

**Proposition 5.** *Given a causal CGS based on a causal model $\mathcal{M}$, with context $\mathbf{u}$, for every leaf-state $q_{i,j}$ of this causal CGS, $q_{i,j}$ corresponds to the causal setting $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u})$, where $\vec{Y} \leftarrow \vec{y} = \{A \leftarrow a \mid A \in V_a \text{ and } (a_{i,j}^A = a) \in \alpha[q_{i,j}]\}$, with $\alpha[q_{i,j}]$ is the action path for $q_{i,j}$.*

*Proof.* Recall that for a leaf-state $q_{i,j}$, $i = \max_{X \in \mathcal{V}} g(X)$. Therefore, as every variable in the causal model has agent rank $i$ or less, we can use Lemma 2 on all variables. Hence, for all endogenous variables $X$ of the causal model, it holds that $(X = x) \in \pi(q_{i,j})$ if and only if $(\mathcal{M}^{\vec{Y} \leftarrow \vec{y}}, \mathbf{u}) \models X = x$. This is the definition of correspondence and hence the statement is proven. □

# 5 Causality in Concurrent Game Structures

Now that I have defined causal concurrent game structures and shown what their states represent, it is time to look at what else they can be used for. In this section I will show three relations between causal concurrent game structures and the modified HP definition, but before I can do that I must shortly introduce a few concepts.

I will occasionally say that an agent $a^X$ causes $\varphi$ in a causal setting $(\mathcal{M}, \mathbf{u})$ if there is a value $x$ of agent variable $X$ such that $X = x$ causes $\varphi$ in this setting. The set of all agents in a model will be denoted by $\Sigma$. Specifically, for a causal CGS, $\Sigma = \{a^X \mid X \in V_a\}$. This set will also be called the *grand coalition* at times.

Now I can state the first claim. The following proposition states that if causal formula $\varphi$ is caused by a set of agents in a causal setting, then there is at least one leaf-state in the causal CGS corresponding to this causal setting, that also contains $\varphi$.

**Proposition 6.** *Given a set of agents $\Gamma = \{a^X \mid X \in \vec{X}\}$ and a setting $\vec{x}$ for the variables in $\vec{X}$. If $\vec{X} = \vec{x}$ is a cause of a causal formula $\varphi$, according to the modified HP definition, in causal setting $(\mathcal{M}, \mathbf{u})$, then, in the causal CGS based on this causal setting and a corresponding ranking function $f$ and agent ranking function $g$, the grand coalition has a strategy $F_\Sigma$ that guarantees $\varphi$ in the leaf-state resulting from this strategy.*

*Proof.* Let $F_\Sigma$ be the strategy where every agent in the system takes the same action as in $(\mathcal{M}, \mathbf{u})$, i.e. $F_\Sigma = \{f_{a^Y} \mid Y \in V_a,$ and $f_{a^Y}(q_{i,j}) = y,$ with $y$ such that $(\mathcal{M}, \mathbf{u}) \vDash Y = y$ if $i = g(Y)$, else $f_{a^Y}(q_{i,j}) = 0\}$. Because $F_\Sigma$ defines an action for every agent in the causal CGS, following this strategy will result in a unique computation $\lambda$. In other words, the set $out(q_{0,0}, F_\Sigma)$ is a singleton set only containing $\lambda$. By Proposition 5 the leaf state in this computation will correspond to causal setting $(\mathcal{M}^{V_a \leftarrow \vec{y}}, \mathbf{u})$, where $V_a$ is the set of agent variables and $\vec{y}$ is the set of values these agent variables were set to according to the strategy. As no values were actually changed in this causal setting when compared to the original causal setting $(\mathcal{M}, \mathbf{u})$, $\varphi$ holds in this setting, because as $\vec{X} = \vec{x}$ causes $\varphi$ in $(\mathcal{M}, \mathbf{u})$, we have by the first condition for causality that $(\mathcal{M}, \mathbf{u}) \vDash \varphi$. The leaf-state must hence guarnatee $\varphi$. This proofs the proposition. □

The above statement is fairly trivial, as all what had to be done was making sure the agents did not change the values of their variables along the path to the leaf-state.

The following two propositions are more interesting, they make a statement the other way around, they claim that if there is a set of agents in the causal CGS that have a strategy that can guarantee $\varphi$, then there is a set of agents that causes $\varphi$. The propositions are fairly similar to each other, but the second is more specific than the first one.

**Proposition 7.** *Given a causal setting $(\mathcal{M}, \mathbf{u})$ with $(\mathcal{M}, \mathbf{u}) \vDash \varphi$ and $(\mathcal{M}, \mathbf{u}) \vDash A = a$ for some $A \in V_a$. Given the causal CGS GS based on $(\mathcal{M}, \mathbf{u})$, if agent*

*$a^A$ has a strategy to bring about $\neg\varphi$ in all leaf-states resulting from this strategy, then $A = a$ causes $\varphi$ in $(\mathcal{M}, \mathbf{u})$, according to the modified HP definition.*

*Proof.* To show that $A = a$ causes $\varphi$ in $(\mathcal{M}, \mathbf{u})$, we need to show all three conditions of the modified HP definition:

AC1 This holds by the conditions in the proposition. $(\mathcal{M}, \mathbf{u}) \models \varphi \wedge (A = a)$.

AC2 If $A$ has a strategy $f_{a^A}$ in the causal CGS $GS$ to bring about $\neg\varphi$, define the strategy $F_\Sigma = \{f_a\} \cup \{f_{a^X} \mid X \in V_a \backslash \{A\}$, and for all states$q_{i',j'}$ van $GS$, $f_{a^X}(q_{i',j'}) = x$, where $x$ is such that $(\mathcal{M}, \mathbf{u}) \models X = x$, if $i' = g(X)$, else $f_{a^X}(q_{i',j'}) = 0\}$. Now, the valuation of the leaf-state $q_{i,j}$ resulting from this strategy $F_\Sigma$ when applied from the initial state $q_{0,0}$, must also contain $\neg\varphi$. However, by Proposition 5, this leaf-state $q_{i,j}$ corresponds to causal setting $(\mathcal{M}^{V_a \leftarrow \vec{y}}, \mathbf{u})$, where $V_a$ is the set of agent variables and $\vec{y}$ is the set of values these agent variables were set to according to the strategy. Define $\vec{W} = V_a \backslash \{A\}$, let $\mathbf{w}^*$ be the set of values such that $(\mathcal{M}, \mathbf{u}) \models \vec{W} = \mathbf{w}^*$. Because $F_\Sigma$ is defined to keep the values of all agent variables that are not $A$ the same as in the original setting, these values $\mathbf{w}^*$ are also the values $\vec{W}$ will have in the valuation of $q_{i,j}$. Let $a'$ be such that $A = a' \in \pi(q_{i,j})$. Now, $(\mathcal{M}^{A \leftarrow a', \vec{W} \leftarrow \mathbf{w}^*}, \mathbf{u})$ is the same causal setting as $(\mathcal{M}^{V_a \leftarrow \vec{y}}, \mathbf{u})$. Since the latter corresponds to $q_{i,j}$ and $q_{i,j}$ is a leaf-state resulting from strategy $f_{a^A}$, $\neg\varphi \in \pi(q_{i,j})$, but as $(\mathcal{M}^{A \leftarrow a', \vec{W} \leftarrow \mathbf{w}^*}, \mathbf{u})$ corresponds to $q_{i,j}$, it holds that $(\mathcal{M}, \mathbf{u}) \models [A \leftarrow a', \vec{W} \leftarrow \mathbf{w}^*]\neg\varphi$, which proves the second condition.

AC3 Since the set $\{A\}$ is a singleton set, it is minimal by definition, proving the third condition.

As all three conditions are proven, we have that $A = a$ does indeed cause $\varphi$ in $(\mathcal{M}, \mathbf{u})$ according to the modified HP definition. □

As I said before, the following proposition is quite similar to the previous, the difference is that the above proposition is about a specific agent, while the following proposition is states that when the grand coalition has a strategy to guarantee a certain outcome, given that the negation of this outcome holds in the initial state, then there is a subset of agents that causes this negation.

**Proposition 8.** *Given a causal setting $(\mathcal{M}, \mathbf{u})$, with $(\mathcal{M}, \mathbf{u}) \models \varphi$ and a causal CGS based on this causal setting. If the grand coalition has a strategy $F_\Sigma$ to bring about $\neg\varphi$ in a leaf-state of the causal CGS, then there is a set of agents $\Gamma = \{a^X \mid X \in \vec{X}\}$ and a setting $\vec{x}$ for the variables in $\vec{X}$ such that $\vec{X} = \vec{x}$ causes $\varphi$ in $(\mathcal{M}, \mathbf{u})$, according to the modified HP definition.*

*Proof.* Let $q_{i,j}$ be the leaf-state resulting from the strategy $F_\Sigma$. Let $\Gamma \subset \Sigma$ be the set of agents such that if $X \in V_a$, $(\mathcal{M}, \mathbf{u}) \models X = x$ and $(X = x) \notin \pi(q_{i,j})$, then $a^X \in \Gamma$. Let $\vec{X}$ be those variables $X$ such that $X \in \vec{X}$ if $a^X \in \Gamma$ and let $\mathbf{x}$ be the values of those variables in $(\mathcal{M}, \mathbf{u})$ (such that $(\mathcal{M}, \mathbf{u}) \models \vec{X} = \mathbf{x}$). I will now prove that $\vec{X} = \mathbf{x}$ causes $\varphi$ in $(\mathcal{M}, \mathbf{u})$.

AC1 $(\mathcal{M}, \mathbf{u}) \models \varphi \wedge (\vec{X} = \mathbf{x})$, by the condition in the proposition and the construction of $\vec{X}$ and $\mathbf{x}$ above.

AC2 Let $\mathbf{x}'$ be the values of the actions that the agents in $\Gamma$ took on $\alpha[q_{i,j}]$ (i.e. $\mathbf{x}' = \{x \mid (a^X = x) \in \alpha[q_{i,j}], a^X \in \Gamma\}$). Let $\vec{W} = V_a \backslash \vec{X}$, these are exactly those variables whose values do not change when $F_\Sigma$ is followed. Let $\mathbf{w}^*$ be such that $(\mathcal{M}, \mathbf{u}) \models \vec{W} = \mathbf{w}^*$. Now, $(\mathcal{M}^{\vec{X} \leftarrow \mathbf{x}', \vec{W} \leftarrow \mathbf{w}^*}, \mathbf{u})$ corresponds to $q_{i,j}$, as $\vec{X} \cup \vec{W} = V_a$ and the values these sets are set too are exactly those values resulting from the strategy $F_\Sigma$. It holds that $\neg\varphi \in \pi(q_{i,j})$ by the condition of the proposition. Therefore by the definition of correspondence, it must hold that $(\mathcal{M}^{\vec{X} \leftarrow \mathbf{x}', \vec{W} \leftarrow \mathbf{w}^*}, \mathbf{u}) \models \neg\varphi$, which implies $(\mathcal{M}, \mathbf{u}) \models [\vec{X} \leftarrow \mathbf{x}', \vec{W} \leftarrow \mathbf{w}^*]\neg\varphi$. Which proves this condition.

AC3 If $\vec{X}$ is minimal, then I am done. If $\vec{X}$ is not minimal, then it must contain a minimal proper subset satisfying AC1 and AC2, but then that is also controlled by a set of agents $\Gamma' \subset \Sigma$, making $\Gamma'$ a cause of $\varphi$. This also proves the proposition.

$\square$

Note that in both proofs above, the set $\vec{W}$ does not have to be the only set $\vec{W}$ that satisfies the conditions of the modified HP definition. There might very well be smaller sets that accomplish the same thing, but for the construction of the proofs this set worked fine.

The propositions in this section show several relations between causality and the notion of strategy. The show several ways to relate the HP notion of causality to a notion specific to concurrent game structures. This can be used in the next section and in future work to relate the HP notion of causality to strategic responsibility.

# 6 Responsibility in Concurrent Game Structures

The responsibility notions as defined in Section 2.5 cannot be effectively used in the context of causal concurrent game structures (causal CGS), when applied just like they stand right now. We recall that all three discussed notions of responsibility call a coalition responsible for an outcome, only if all states in all computations, resulting from a strategy for this coalition starting at a certain state, to belong to a set $\bar{S}$ of states that do not have this outcome. In a causal CGS this does not really make sense, as in general, not all variables are allowed to change their values in every state. This means that in some cases the result of an action will only be seen in a later state. Therefore it seems contra-productive to require all states on a computation to belong to this set $\bar{S}$. Moreover, the tree-structure of a causal CGS lends itself to only evaluating the result of an action at the leaf-states, I therefore suggest to slightly modify the notions of responsibility for this case.

**Definition 28** (Forward Group Responsibility)**.** *Let $\mathcal{M}$ be a CGS, $S$ be a set of states, $q \in S$ a state. We say that a group of agents $\Gamma \subseteq \Sigma$ is* forward responsible *for $S$ in $q$ iff:*

1. *There is a strategy for $\Gamma$, $F_\Gamma$, such that all leaf-states of all computations in out$(q, F_\Gamma)$ belong to $\bar{S}$, and*

2. *$\Gamma$ is minimal, that is, there is no $\Gamma' \subsetneq \Gamma$ with the property formulated above.*

**Definition 29** (Backward Group Responsibility )**.** *Let $\mathcal{M}$ be a CGS, $S$ be a set of states, $q \in S$ a state, and $\lambda[q_i, k]$ an arbitrary q-history. We say that a group of agents $\Gamma \subseteq \Sigma$ is* backward responsible *for $S$ based on $\lambda[q_i, k]$ iff:*

1. *There is a state $q_j$ in $\lambda[q_i, k]$ such that for some strategy for $\Gamma$, $F_\Gamma$, all leaf-states of all computations in out$(q_j, F_\Gamma)$ belong to $\bar{S}$, and*

2. *$\Gamma$ is minimal, that is, there is no $\Gamma' \subsetneq \Gamma$ with the property formulated above.*

**Definition 30** (Causal Backward Responsibility)**.** *Let $\mathcal{M}$ be a CGS, $S$ be a set of states, $q \in S$ a state, $\lambda[q_i, k]$ an arbitrary q-history, and $F_\Sigma$ be a collective strategy for all agents in $\Sigma$, s.t. $\lambda[q_i, k]$ is contained in out$(q_0, F_\Sigma)$. We say that a group of agents $\Gamma \subseteq \Sigma$ is* causal backward responsible *for $S$ based on $\lambda[q_i, k]$ and $F_\Sigma$ iff:*

1. *There exists a strategy $F_\Gamma$ s.t. for $F'_\Sigma := \{f_a | f_a \in F_\Gamma \text{ if } a \in \Gamma, \text{ else } f_a \in F_\Sigma\}$, all leaf-states of all computations in out$(q_0, F'_\Sigma)$ belong to $\bar{S}$, and*

2. *$\Gamma$ is minimal, that is, there is no $\Gamma' \subsetneq \Gamma$ with the property formulated above.*

Note that the only difference with the earlier definitions is that I now only require the leaf-states to be in $\bar{S}$.

Lets see how this definition works in the causal CGS for the rock-throwing example.

**Example 23.** Recall that we had defined two different causal CGS for the rock-throwing example, the one of Figure 11 and the one in Figure 12. I will first consider the first causal CGS.

Let $S = \{q_{i,j} \mid \neg BS \in \pi(q_{i,j})\}$. We have that $q_{0,0} \in S$. Now, both Suzy and Billy are forward responsible for $S$ in $q_{0,0}$. After all, if either of them follows the strategy to throw, the bottle will shatter, putting the resulting state in $\bar{S}$. Similarly, they are backward and causal backward responsible for $S$ in $q_{1,0}$. Now, let $S' = \{q_{i,j} \mid BS \in \pi(q_{i,j})\}$. Only the coalition of both Billy and Suzy is backward responsible for $S'$ in any of the states in $S'$. This is also true for causal backward responsibility in $q_{1,3}$, however in $q_{1,1}$, only Billy is causal backward responsible and in $q_{1,2}$, only Suzy is causal backward responsible.

In the other CGS, we have mostly the same story, only in $q_{2,1}$, only Billy is backward responsible for $S'$.

Using these definitions, we can make a connection between responsibility and causality.

**Proposition 9.** *Given a causal CGS based on causal setting $(\mathcal{M}, \mathbf{u})$ with $(\mathcal{M}, \mathbf{u}) \vDash \varphi$, if an agent $a^A$ of this CGS is forward responsible for $S = \{q_{i,j} \mid \pi(q_{i,j}) \vDash \varphi\}$ in $q_{0,0}$, then $a^A$ is a cause of $\varphi$ in $(\mathcal{M}, \mathbf{u})$.*

*Proof.* Let $(\mathcal{M}, \mathbf{u}) \vDash A = a$. The agent $a^A$ is forward responsible for $S$ in $q_{0,0}$, this means it has a strategy $F_A$ such that all leaf states in $out(q_{0,0}, F_A)$ belong to $\bar{S}$. This set $\bar{S}$ is exactly the set of those states where $\neg\varphi$ holds. Hence $a^A$ has a strategy to bring about $\neg\varphi$ in all the leaf states resulting from this strategy and hence the condition from Proposition 7 holds, which means that $A = a$ causes $\varphi$. Hence we can say that $a^A$ causes $\varphi$. □

So far, I cannot yet generalise the above proposition to a set of agents, because I have not yet found a relation between a general set of agents having a strategy to reach a certain an outcome and being a cause of this outcome.

# 7 Conclusion and Discussion

The main goal of this project was to find a way to reason about causal effects in multi-agent settings. To do this I had formulated three sub-problems, the first was *How can we define a way to derive a system used for reasoning about multi-agent settings, from a causal model that describes the effects of agents' action on the environment?* To handle this problem I introduced a way to construct a causal CGS on the basis of a structural causal model. This construction works by first dividing the endogenous variables in a set of agent variables, directly controlled by agents, and a set of environment variables, which are all the other variables. Then the variables get ranked according to a function that respects their causal order. This ranking will be used to determine when agents will get to take actions in the causal CGS. I defined the causal CGS as starting from a starting state, where agents with the lowest rank get to take actions, leading to new states, in which all the agents with the next lowest rank get to take actions, and so on until all agents have taken an action. This leads to a tree-structure for the CGS. The starting state of a causal CGS will correspond to a causal setting while every agent action will be reminiscent of an intervention on this causal setting. Not all variables get evaluated in every state, in principle, all variables will only be evaluated once on a path through the causal CGS and in every state, only the agent variables of the agents that just took an action and all environment variables that depend on those variables and not on higher ranked agent variables will be evaluated. Because of this only the leaf states will fully correspond to interventions on the original causal setting.

The second sub-problem was *to study the relation between the derived system and the original structural causal model.* I showed that agent strategies in this CGS can be related to causality in the original causal setting through three separate propositions.

The final sub-problem asked whether *this derived model can help with showing a relation between strategic responsibility and causality?* I defined a modified version of strategic forward responsibility, that only looks at the outcome of a strategy in the leaf-states of the causal CGS instead of at all states resulting from the strategy. I then showed a relation between this and the HP definition of causality, but there could be more research done into this topic in the future.

A limitation of this approach is that since the agent actions are seen as interventions, not all causal relations are carried over in the causal CGS. After all, an intervention on a variable removes its causal connection to its ancestors [14]. An intervention on a variable fixes its value, this means that the system is changed. After an intervention on a variable $X$ that sets its value to $x$, the value of an ancestor has no influence on the value of $X$ anymore, the value of $X$ will always be $x$. That is why for some causal models, the actions in the last states fully determine the values of all the variables. That also means that a causal CGS in general is unable to say something about all the causes in a causal model.

Another limitation is that the causal CGS is created with respect to a specific causal setting. The result only applies to a single context. This means that if the context is uncertain, multiple causal CGS have to be made to evaluate all possible outcomes. However, it is possible that this problem can be solved by using a version of an epistemic CGS. This can be researched in the future.

In the future the relationship between causality and responsibility in this framework could be researched more. A possible direction for this research is how being a part of a cause might impact strategic responsibility. So far, I have only showed a relationship between a single agent being a cause and the coalition consisting of only this agent being forward responsible. It might be possible to show a more general result when an agent is not a cause on its own. Another possible direction is looking at it from the opposite direction, can we say anything about whether an agent is a cause when we know it is responsible for an outcome? It might also be interesting to see how the type of causality might relate to responsibility. With type of causality I mean whether it is a but-for cause, or not, and if not, what does the witness look like? It is possible that the witness influences which group of agents can be held responsible. Finally, one can also look at distributed responsibility and how causality influences to what degree an agent can be held responsible.

Another possible direction for future research is to look at how the causal CGS itself can be adapted and refined. Right now, agents' actions are completely independent from other things that are happening in the system. It could be possible to let agents be influenced by earlier actions or variable values. In the causal CGS as it stands now, agent actions are also seen as some kind of intervention on the causal model. It might be interesting to see if there are other options for modelling the agent actions. If this is possible, it could also be interesting to compare the relationship that such a model has with causality and responsibility, with the causal CGS as defined in this work.

This research could be used in multi-agent systems with a clear causal structure. Examples of this are traffic control environments, think of planes that cannot land when another is departing, trains that cannot travel over the same track at the same time and traffic lights that cannot all give the green light at the same time. Other applications could be in the analysis of multi-player games, after all, players could cause other players to make a certain move, or even energy management systems, where supply and demand of electricity influence each other in a myriad of ways. In these situations this research could be used to help making decisions, or after something has gone wrong to help attributing responsibility for this.

# References

[1] Rajeev Alur, Thomas A Henzinger, and Orna Kupferman. Alternating-time temporal logic. *Journal of the ACM (JACM)*, 49(5):672–713, 2002.

[2] Christel Baier, Florian Funke, and Rupak Majumdar. A game-theoretic account of responsibility allocation. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 1773–1779. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Main Track.

[3] Christel Baier, Florian Funke, and Rupak Majumdar. A game-theoretic account of responsibility allocation. *arXiv preprint arXiv:2105.09129*, 2021.

[4] Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT press, 2008.

[5] Hana Chockler and Joseph Y Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

[6] Maksim Gladyshev, Natasha Alechina, Mehdi Dastani, and Dragan Doder. Dynamics of causal dependencies in multi-agent settings. 2023.

[7] Roberto Gorrieri. *Process algebras for Petri nets: the alphabetization of distributed systems.* Springer, 2017.

[8] Joseph Y Halpern. *Actual causality.* MiT Press, 2016.

[9] Joseph Y Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part i: Causes. *The British journal for the philosophy of science*, 2005.

[10] D Hume. An enquiry concerning human understanding (1748) reprinted by open court press. *LaSalle, IL*, 1958.

[11] Wojciech Jamroga. *Logical methods for specification and verification of multi-agent systems.* Institute of Computer Science, Polish Academy of Sciences, 2015.

[12] Robert M Keller. Formal verification of parallel programs. *Communications of the ACM*, 19(7):371–384, 1976.

[13] Samantha Kleinberg and Bud Mishra. The temporal logic of causal structures. *arXiv preprint arXiv:1205.2634*, 2012.

[14] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer.* John Wiley & Sons, 2016.

[15] RJ van Glabbeek. The linear time - branching time spectrum. *CWI Amsterdam Report*, 1988.

[16] Vahid Yazdanpanah, Mehdi Dastani, Natasha Alechina, Brian Logan, and Wojciech Jamroga. Strategic responsibility under imperfect information. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2019*, pages 592–600. IFAAMAS, 2019.

[17] Vahid Yazdanpanah, Enrico H Gerding, Sebastian Stein, Corina Cirstea, MC Schraefel, Timothy J Norman, and Nicholas R Jennings. Different forms of responsibility in multiagent systems: Sociotechnical characteristics and requirements. *IEEE Internet Computing*, 25(6):15–22, 2021.