The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)

Temporal Causal Reasoning with (Non-Recursive) Structural Equation Models

Maksim Gladyshev[1], **Natasha Alechina**[2,1], **Mehdi Dastani**[1], **Dragan Doder**[1], **Brian Logan**[3,1]
[1] Utrecht University, The Netherlands
[2] Open Universiteit, The Netherlands
[3] University of Aberdeen, UK
{m.gladyshev, n.a.alechina, m.m.dastani, d.doder, b.s.logan}@uu.nl

Abstract

Structural Equation Models (SEM) are the standard approach to representing causal dependencies between variables in causal models. In this paper we propose a new interpretation of SEMs when reasoning about Actual Causality, in which SEMs are viewed as mechanisms transforming the dynamics of exogenous variables into the dynamics of endogenous variables. This allows us to combine counterfactual causal reasoning with existing temporal logic formalisms, and to introduce a temporal logic, CPLTL, for causal reasoning about such structures. We show that the standard restriction to so-called *recursive* models (with no cycles in the dependency graph) is not necessary in our approach, allowing us to reason about mutually dependent processes and feedback loops. Finally, we introduce new notions of model equivalence for temporal causal models, and show that CPLTL has an efficient model-checking procedure.

Introduction

There has recently been increased interest in causal reasoning from AI researchers and philosophers. The standard framework for reasoning about causal dependencies in both stochastic and deterministic settings is Structural Equation Modelling (SEM) (Pearl 2000; Spirtes, Glymour, and Scheines 2001). Both types of reasoning are crucial for the field of AI: in stochastic domains it is often used in Causal Machine Learning (Peters, Janzing, and Schölkopf 2017) and Causal Discovery, while in deterministic domains it forms the basis for work on Actual Causality (Halpern 2016).

However, reasoning about many real-world phenomena and their dynamics requires reasoning about time and temporal properties. As a result, temporal reasoning in, for example, Linear-time Temporal Logic (LTL) (Pnueli 1977), has become an important technique for reasoning about the dynamics of AI systems. Progress in this field has resulted in many theoretical results in areas such as formal verification and synthesis (Demri, Goranko, and Lange 2016).

While structural equation models are the main tool for analysing cause–effect relations and have been applied in a range of disciplines, e.g., medicine, economics, computer science and industrial engineering, they are not specifically designed for representing the temporal behaviour of a system, which play important role for causality claims in many domains. For example, the effect of some treatment may (causally) depend not only on whether the treatment was given to a patient or not, but also on temporal properties (i.e, dynamics) of this treatment, such as timing, duration and repetition of the treatment. Combining SEM models with temporal reasoning is therefore key in many applications.

Previous work on combining SEM models with temporal reasoning has focused on causal discovery, and several methods for analysing time-series data with SEMs have been developed, e.g., (Assaad, Devijver, and Gaussier 2022; Hyvärinen, Khemakhem, and Monti 2023). In this paper, we propose an approach to temporal reasoning with SEMs for reasoning about actual causality. While a causal model is usually understood as a static representation of causal dependencies transforming values of exogenous variables into the values of endogenous variables, we show it can also be interpreted as a causal mechanism transforming the *dynamics* of exogenous changes into the *dynamics* of endogenous ones. In our framework, we assume that input to a causal model is a (time) series of values assigned to the exogenous variables, which we call the 'temporal context'. We give a procedure that, given a causal model as input, processes a temporal context and transforms it into a (time) series of assignments to endogenous variables. We then show how the framework of actual causality can be combined with the temporal logic LTL to give the logic CPLTL, allowing us to express statements about future and past of the system, e.g., "a fact φ was *always* true", "a fact φ will be true *until* another fact ψ is true", etc.

Our framework has several interesting features. Firstly, interventions (necessary for counterfactual reasoning) become 'time-sensitive': in temporal settings it is necessary to specify not only which intervention happens, but also *when* it happens. Secondly, most existing works on actual causality only deal with *recursive* causal models (models with acyclic dependency graphs). In our approach cycles in the dependency graph have a natural temporal interpretation, so they do not create technical difficulties, but instead provide useful modelling tools. Following Beckers (2021), we introduce new notions of (temporal) equivalence for causal models, which also covers non-recursive cases. Finally, we show that our framework has an efficient model-checking procedure.

Formal Background

In this section we introduce the formal apparatus we use in the rest of the paper: Structural Equation Models (SEM's), also called Causal Models, used for modelling causal dependencies between events, and Linear-time Temporal Logic (LTL) designed for temporal reasoning.

Structural Equation Models

The presentation below essentially follows (Halpern 2016). Let $\mathcal{U}$ and $\mathcal{V}$ be the finite sets of *exogenous* and *endogenous* variables respectively. We say that $\mathcal{S} = (\mathcal{U}, \mathcal{V}, \mathcal{R})$ is a *signature*, where $\mathcal{R} : \mathcal{U} \cup \mathcal{V} \to 2^{\mathbb{R}}$ associates with every variable $Y \in \mathcal{U} \cup \mathcal{V}$ a non-empty *finite* set $\mathcal{R}(Y)$ of possible values, also called *range* of Y.

Definition 1 (Causal model)**.** *Causal Model (or SEM) over $\mathcal{S}$ is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{F})$, where $\mathcal{F}$ associates with every endogenous variable $X \in \mathcal{V}$ a function*

$$\mathcal{F}_X : \prod_{Z \in (\mathcal{U} \cup \mathcal{V})} \mathcal{R}(Z) \to \mathcal{R}(X)$$

which defines the structural equation describing how the value of X depends on the values of $\mathcal{U} \cup \mathcal{V}$.

Informally, in causal models different events are represented by the assignment of different values to abstract variables. Values of *endogenous* variables depend on the values of other variables, while values of *exogenous* variables are determined outside of the model. A complete assignment $(U_1 = u_1, \dots, U_k = u_k)$ of $\mathcal{U}$ is called a *context* and denoted $\vec{u}$. A pair $(\mathcal{M}, \vec{u})$ is called *causal setting*.

Example 1 (Rocks)**.** *Suzy and Billy both pick up rocks and throw them at a bottle (encoded as ST=1 and BT=1 respectively). Both throws are perfectly accurate, so the bottle shatters (BS=1) whenever ST=1 or BT=1.*

Because all the variables are binary here, for simplicity we write ST and $\neg ST$ instead of ST=1 and ST=0 respectively. It is assumed that exogenous variables U_{ST} and U_{BT} determine values of ST and BT: $ST := U_{ST}$, $BT := U_{BT}$. The structural equation for BS is $BS := ST \vee BT$. It is often convenient to represent the structure of the model as a dependency graph. Nodes in the graph represent variables, and directed edges represent (direct) dependencies among variables.

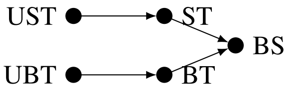


Figure 1: A dependency graph for Example 1.

A model $\mathcal{M}$ is *recursive* if there exists a partial order $\preccurlyeq$ on $\mathcal{V}$, such that unless $X_2 \preccurlyeq X_1$, X_1 is independent of X_2 (Halpern 2016). As a result, a dependency graph of a recursive model is a directed acyclic graph. Recursiveness also guarantees that given $\vec{u}$, the set of structural equations has a unique solution, i.e., a unique assignment of $\mathcal{V}$. For this reason, many papers on actual causality consider recursive causal models only (e.g., (Halpern and Pearl 2005; Beckers and Vennekens 2018)). As we show later, this restriction recursive models is not necessary in our approach.

The last ingredient we need for reasoning about actual and counterfactual courses of events are *interventions*. An intervention $[\vec{X} \leftarrow \vec{x}]\varphi$, meaning "after fixing the values of $\vec{X} \subseteq \mathcal{V}$ to $\vec{x}$, φ holds", results in a new casual model denoted $\mathcal{M}_{\vec{X}\leftarrow\vec{x}}$. $\mathcal{M}_{\vec{X}\leftarrow\vec{x}}$ is the model $\mathcal{M}$ where functions $\mathcal{F}_X$ for any $X \in \vec{X}$ are replaced with a constant function $\mathcal{F}'_X$, which always returns x^*, where $X = x^* \in \vec{X} \leftarrow \vec{x}$ and the remaining functions remain unchanged.

Note that $\vec{X}$ abbreviates $\{X_1, \dots, X_k\}$; $\vec{X} = \vec{x}$ abbreviates $\{X_1 = x_1, \dots, X_k = x_k\}$; $\vec{X} \leftarrow \vec{x}$ abbreviates $\{X_1 \leftarrow x_1, \dots, X_k \leftarrow x_k\}$. Sometimes we slightly abuse this notation and write $X = x \in \vec{X} \leftarrow \vec{x}$ instead of $X \leftarrow x \in \vec{X} \leftarrow \vec{x}$.

In our example in a context $\vec{u} = (U_{ST} = 1, U_{BT} = 1)$, where both Suzy and Billy throw their rocks, formulas $ST = 1$, $BT = 1$, $BS = 1$ are true. At the same time, interventions allow us to formulate statements, such that $[ST \leftarrow 0]BS = 1$, meaning "if Suzy does not throw the rock, the bottle still shatters" or $[ST \leftarrow 0, BT \leftarrow 0]\neg(BS = 1)$, meaning "if neither Suzy nor Billy throw the rock, then the bottle is not shattered". Thus, interventions provide us all the necessary machinery for counterfactual reasoning about SEM's.

Definition 2 (Syntax)**.** *The grammar of the basic causal language is defined as follows:*

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

where $X \in \mathcal{U} \cup \mathcal{V}$, $x \in \mathcal{R}(X)$, $\vec{Y} \subseteq \mathcal{V}$ and $\vec{y} \in \mathcal{R}(\vec{Y})$. Note that $\vec{Y} \leftarrow \vec{y}$ may be empty, so we write φ instead of $[]\varphi$.

The *truth* relation $(\mathcal{M}, \vec{u}) \Vdash \varphi$, meaning that a causal formula φ is true in a causal setting $(\mathcal{M}, \vec{u})$, is defined inductively as follows:
$(\mathcal{M}, \vec{u}) \Vdash (X = x)$ iff $(X = x) \in Sol(\vec{u})$[1];
$(\mathcal{M}, \vec{u}) \Vdash \neg\varphi$ iff $(\mathcal{M}, \vec{u}) \nVdash \varphi$;
$(\mathcal{M}, \vec{u}) \Vdash (\varphi \wedge \psi)$ iff $(\mathcal{M}, \vec{u}) \Vdash \varphi$ and $(\mathcal{M}, \vec{u}) \Vdash \psi$;
$(\mathcal{M}, \vec{u}) \Vdash [\vec{Y} \leftarrow \vec{y}]\varphi$ iff $(\mathcal{M}_{\vec{Y}\leftarrow\vec{y}}, \vec{u}) \Vdash \varphi$.

Linear-time Temporal Logic

Now we introduce some basics of Linear-time Temporal Logic (LTL), for an extensive overview see (Demri, Goranko, and Lange 2016). In this paper we use both future and past LTL operators, so we call it PLTL, and it contains four basic modalities: $\bigcirc\varphi$ meaning "φ will be true in the next moment", $\varphi \mathsf{U} \psi$ meaning "φ will be true until ψ", $\ominus\varphi$ meaning "φ was true in the previous moment" and $\varphi \mathsf{S} \psi$ meaning "φ is true since ψ". The only difference of our approach from the standard PLTL definitions is that we use atomic expressions $(X = x)$ generated by a given signature $\mathcal{S}$ instead of atomic propositions $Prop = \{p, q, \dots\}$.

Definition 3 (PLTL syntax)**.** *Given a signature $\mathcal{S}$,* PLTL *syntax is defined as:*

$$\varphi ::= (X = x) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi \mathsf{U} \varphi \mid \ominus\varphi \mid \varphi \mathsf{S} \varphi,$$

where $X \in \mathcal{V}, x \in \mathcal{R}(X)$.

---

[1] Here, $Sol(\vec{u})$ denotes the unique solution of the equations in $\mathcal{M}$ in context $\vec{u}$ (existing by the recursiveness of $\mathcal{M}$).

We use standard abbreviations for other Boolean connectives, together with derived operators $\mathsf{F}\varphi \equiv \top\mathsf{U}\varphi$ for *eventually*; $\mathsf{G}\varphi \equiv \neg\mathsf{F}\neg\varphi$ for *always* in future; $\mathsf{P}\varphi \equiv \top\mathsf{S}\varphi$ for *sometime* in the past; $\mathsf{H}\varphi \equiv \neg\mathsf{P}\neg\varphi$ for *always* in the past. We refer to the fragments of PLTL without $\{\ominus\varphi, \varphi\mathsf{S}\psi\}$ operators and without $\{\bigcirc\varphi, \varphi\mathsf{U}\psi\}$ operators as to LTL and *pure-past* LTL respectively. We also write $\bigcirc^n$ to abbreviate $\bigcirc$ nested n times. The models of PLTL are infinite sequences of complete assignments to the variables in $\mathcal{V}$.

Definition 4 (Linear model)**.** *For a given signature $\mathcal{S}$, a linear model is an infinite sequence of assignments of all endogenous variables, i.e.,*

$$\sigma : \mathbb{N} \to \prod_{X \in \mathcal{V}} \mathcal{R}(X)$$

Example 2 (Treatment)**.** *Suppose a patient is ill, and a medication exists. But this medication works only if it is given twice, on two consecutive days. The patient is recovered ($R = 1$) at step i if and only if the treatment was given ($T = 1$) at two consecutive previous steps $i-1$ and $i-2$. Once the patient is recovered, they remain so.*

Let us fix $\mathcal{S} = (\mathcal{U}, \mathcal{V}, \mathcal{R})$, with $\mathcal{V} = \{T, R\}$ (T stands for Treatment and R stands for Recovery) and all variables are binary. Consider two linear models over $\mathcal{S}$:
$\sigma_1 = ((T\neg R), (\neg T\neg R), (T\neg R), (\neg T\neg R), (T\neg R), \dots)$
and
$\sigma_2 = ((\neg T\neg R), (T\neg R), (T\neg R), (\neg TR), (\neg TR), \dots)$

The first model σ_1 depicts a situation, when the treatment is given at every even time moment i (including $i = 0$) and the patient is never recovered. In the second model the treatment is given twice in a row, at steps $i = 1$ and $i = 2$, and the patient recovers at $i = 3$. Given a model and a time moment, PLTL logic allows us to express various facts about the past, the present and the future with respect to this time moment.

Definition 5 (PLTL Semantics)**.** *Given a linear model σ, a position $i \in \mathbb{N}$ and a formula $\varphi \in \mathsf{PLTL}$, we define the* truth relation *$\vDash_{\mathsf{PLTL}}$ inductively as follows*
$(\sigma, i) \vDash_{\mathsf{PLTL}} (X = x)$ *iff* $(X = x) \in \sigma(i)$*;*
$(\sigma, i) \vDash_{\mathsf{PLTL}} \neg\varphi$ *iff* $(\sigma, i) \nvDash_{\mathsf{PLTL}} \varphi$*;*
$(\sigma, i) \vDash_{\mathsf{PLTL}} (\varphi \wedge \psi)$ *iff* $(\sigma, i) \vDash_{\mathsf{PLTL}} \varphi$ *and* $(\sigma, i) \vDash_{\mathsf{PLTL}} \psi$*;*
$(\sigma, i) \vDash_{\mathsf{PLTL}} \bigcirc\varphi$ *iff* $(\sigma, i+1) \vDash_{\mathsf{PLTL}} \varphi$*;*
$(\sigma, i) \vDash \varphi\mathsf{U}\psi$ *iff there exists* $j \geq 0$ *such that* $(\sigma, i+j) \vDash_{\mathsf{PLTL}} \psi$ *and for all* $0 \leq k < j : (\sigma, i+k) \vDash_{\mathsf{PLTL}} \varphi$*;*
$(\sigma, i) \vDash_{\mathsf{PLTL}} \ominus\varphi$ *iff* $i \geq 1$ *and* $(\sigma, i-1) \vDash_{\mathsf{PLTL}} \varphi$*;*
$(\sigma, i) \vDash_{\mathsf{PLTL}} \varphi\mathsf{S}\psi$ *iff* $\exists k$ *with* $0 \leq k \leq i$ *such that* $(\sigma, k) \vDash_{\mathsf{PLTL}} \psi$ *and* $\forall j$ *with* $k < j \leq i : (\sigma, j) \vDash_{\mathsf{PLTL}} \varphi$*.*

The following expressions are true about Example 2:
- $(\sigma_1, 0) \vDash_{\mathsf{PLTL}} \mathsf{G}(\ominus T \to (\neg T \wedge \bigcirc T) \wedge \neg R)$, meaning that at $(\sigma_1, 0)$ "it will always be the case that if the treatment was given yesterday, then it is not given today, but will be given again tomorrow, and the patient will never recover";
- $(\sigma_2, 0) \vDash_{\mathsf{PLTL}} \mathsf{F}((\ominus(T = 1) \wedge \ominus^2(T = 1)) \wedge \mathsf{G}(R = 1))$, meaning that at $(\sigma_2, 0)$ "eventually it will be the case that the treatment is given yesterday and two days ago, and from that moment forward the patient will be recovered".

Definition 6 (Periodic model)**.** *A linear model σ is* ultimately periodic *if $\exists i, l > 0$ such that $\sigma(k) = \sigma(k+l)$ for every $k \geq i$. We call the (possibly empty) finite sequence $\sigma(0), \dots, \sigma(i-1)$ the* prefix *of σ and $\sigma(i), \dots, \sigma(i+l)$ the* loop *of σ and say that σ is of type (i, l).*

An ultimately periodic model can be represented by a finite sequence $\vec{v}_1, \dots, (\vec{v})_{i+l}$ of the assignments of $\mathcal{V}$.

Temporal Interpretation of Causal Models

In order to proceed we need to modify some of the definitions presented already. Firstly, we need to adjust the idea of contexts. Note that normally, the context $\vec{u}$ is understood as an assignment of all exogenous variables (Halpern 2016). But in our setting, we want to consider contexts as a (time) series of such assignments describing how the values of exogenous variables evolve over time.

Definition 7 (Temporal context)**.** *A* temporal *context $\vec{\vec{u}}$ is an infinite sequence of complete assignments of $\mathcal{U}$:*

$$\vec{\vec{u}} : \mathbb{N} \to \prod_{U \in \mathcal{U}} \mathcal{R}(U).$$

We denote a particular time instance of $\vec{\vec{u}}$ as $\vec{\vec{u}}(n)$.

We also need to adjust the definition of interventions. In our framework it is essential to specify not only which interventions take place, but also *when*. We extend the notation to make interventions time sensitive and, instead of $Y \leftarrow y$, we use $Y(n) \leftarrow y$, where $n \in \mathbb{N}$, which means that we intervene on Y with value y at time step n. For multiple interventions, we use the notation $\vec{Y}(\vec{n}) \leftarrow \vec{y} = (Y_1(n_1) \leftarrow y_1, \dots, Y_k(n_m) \leftarrow y_k)$. Note that we allow the same variable Y' to occur in $\vec{Y}(\vec{n}) \leftarrow \vec{y}$ multiple times, meaning that we can intervene on the same variable at multiple time moments.

Given $\mathcal{M}$ describing causal dependencies between the variables, and $\vec{\vec{u}}$ describing how the values of exogenous variables evolve over time, we want to understand how the values of *endogenous* variables evolve over time. We represent this evolution as an (infinite) sequence $\mathcal{C} = (\vec{v}_0, \vec{v}_1, \dots)$ called a *computation*. First we define a *call* to $\mathcal{F}$. Let $\vec{u'}$ and $\vec{v'}$ be complete assignments of $\mathcal{U}$ and $\mathcal{V}$ respectively. We say that a *call* to $\mathcal{F}$ with $(\vec{u'}, \vec{v'})$ takes $(\vec{u'}, \vec{v'})$ and returns $v'' = \prod_{X \in \mathcal{V}} \mathcal{F}_X(\vec{u'}, \vec{v'})$. A *computation* $\mathcal{C}$ starts with a *default* assignment $\vec{v}$ of $\mathcal{V}$ (representing 'initial' configuration of endogenous values)[2] and evolves as a process of iterative calls to $\mathcal{F}$, using values of $\mathcal{U} \cup \mathcal{V}$ from the previous step [3].

Definition 8 (Computation)**.** *Given a tuple $(\mathcal{M}, \vec{\vec{u}}, \vec{v})$, a computation $\mathcal{C}$ over $(\mathcal{M}, \vec{\vec{u}}, \vec{v})$ is a function mapping $\mathbb{N}$ to the complete assignments $\prod_{V \in \mathcal{V}} \mathcal{R}(V)$ of endogenous variables, such that $\mathcal{C}(\mathcal{M}, \vec{\vec{u}}, \vec{v})(0) \coloneqq \vec{v}$ and for all $i > 0$,*

$$\mathcal{C}(\mathcal{M}, \vec{\vec{u}}, \vec{v})(i) \coloneqq \prod_{X \in \mathcal{V}} \mathcal{F}_X\big(\vec{\vec{u}}(i-1), \mathcal{C}(\mathcal{M}, \vec{\vec{u}}, \vec{v})(i-1)\big)$$

---

[2] The idea to use default assignments of $\mathcal{V}$ was also discussed in the context of non-recursive models in (Halpern 2016, Ch.2.7)

[3] The idea of a computation $\mathcal{C}$ is adapted from (Gladyshev et al. 2023).

We use the short notation $\mathcal{C}(i)$ instead of $\mathcal{C}(\mathcal{M},\vec{\vec{u}},\vec{v})(i)$ if $(\mathcal{M},\vec{\vec{u}},\vec{v})$ is clear from the context.

Because any $\mathcal{C}(i)$ is a vector of values whose coordinates are indexed by the elements of $\mathcal{V}$, we write $\mathcal{C}(i)|_X$ to refer to X's value at the i'th step of $\mathcal{C}$. An intervention $int = \vec{Y}(\vec{n}) \leftarrow \vec{y}$ results in an updated computation $\mathcal{C}^{\vec{Y}(\vec{n})\leftarrow\vec{y}}$, defined as follows. Given a default assignment $\vec{v}$, let $\vec{v}^{int}$ be an assignment of $\mathcal{V}$, which agrees with $\vec{v}$ everywhere, except the variables Y, such that $Y(0) \leftarrow y$ occurs in $\vec{Y}(\vec{n}) \leftarrow \vec{y}$. The values of those variables in $\vec{v}^{int}$ are set according to $\vec{Y}(\vec{n}) \leftarrow \vec{y}$.

Definition 9 (Updated Computation)**.** *Given an intervention $int = \vec{Y}(\vec{n}) \leftarrow \vec{y}$ and $(\mathcal{M},\vec{\vec{u}},\vec{v})$, an* updated *computation $\mathcal{C}^{int}$ is defined as $\mathcal{C}^{int}(0) = \vec{v}^{int}$, and $\forall i > 0 \forall X \in \mathcal{V}$:*

$$\mathcal{C}^{int}(i)|_X = \begin{cases} x', \text{ if } X(i) \leftarrow x' \in \vec{Y}(\vec{n}) \leftarrow \vec{y} \\ \mathcal{F}_X(\vec{\vec{u}}(i-1), \mathcal{C}^{int}(i-1)), \text{ otherwise} \end{cases}$$

Simply speaking, an updated computation $\mathcal{C}^{int}$ replaces the values of variables from int on the corresponding steps.

Recall Example 1 and let $\vec{\vec{u}} = (00, 10, 00, 01, 00\dots)$, so the generated computation for $\vec{\vec{u}}$ and $\vec{v} = 000$ (here we write 000 instead of (ST=0, BT=0, BS=0)) is $\mathcal{C} = (000, 000, 100, 001, 010, 001, \dots)$. Suzy throws (ST=1) at step 2 and Billy (BT=1) at step 4 (we start counting from 0). Then, we can say that the LTL formula $\bigcirc^2(ST = 1) \wedge \bigcirc^3(BS = 1) \wedge \bigcirc^4(BT = 1)$ is true at $\mathcal{C}(0)$. And counterfactually, $\bigcirc(BS = 1)$ is true in $\mathcal{C}^{BT(0)\leftarrow 1(0)}$.[4]

As can be seen, our approach allows us to merge causal time-sensitive interventions with the machinery of PLTL. We call this logic *Causal LTL with Past* (CPLTL).

Definition 10 (Syntax of CPLTL)**.** *The grammar of* CPLTL *is defined as follows:*

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

where $\psi \in$ PLTL (Definition 3) and $\vec{Y}(\vec{n}) \leftarrow \vec{y} = (Y_1(n_1) \leftarrow y_1, \dots, Y_k(n_m) \leftarrow y_k)$, such that $Y_i \in \mathcal{V}$, $y_i \in \mathcal{R}(Y_i)$, $n_i \in \mathbb{N}$ and for any $Y_i(n_i) \leftarrow y_i, Y_j(n_j) \leftarrow y_j \in \vec{Y}(\vec{n}) \leftarrow \vec{y}$, $Y_i = Y_j$ implies $n_i \neq n_j$. $\vec{Y}(\vec{n}) \leftarrow \vec{y}$ may be empty, in this case we write ψ instead of $[]\psi$. We use the same abbreviations for boolean connectives and temporal operators as in PLTL.

In contrast to static causal reasoning, where formulas are evaluated wrt a causal setting $(\mathcal{M}, \vec{u})$, to evaluate CPLTL formulas, we need to know a causal model $\mathcal{M}$, a temporal context $\vec{\vec{u}}$, a default assignment $\vec{v}$, and a time moment t. We call $(\mathcal{M},\vec{\vec{u}},\vec{v})$ a causal scenario. Note that $(\mathcal{M},\vec{\vec{u}},\vec{v})$ together with an intervention $\vec{Y}(\vec{n}) \leftarrow \vec{y}$ produces a computation $\mathcal{C}^{\vec{Y}(\vec{n})\leftarrow\vec{y}}$ according to Definition 9, which is a linear model in the sense of Definition 4, used to define the semantics of the PLTL fragment.

Definition 11 (Semantics of CPLTL)**.** *Given a causal scenario $(\mathcal{M},\vec{\vec{u}},\vec{v})$, $t \in \mathbb{N}$ and $\varphi \in$ CPLTL we define truth relation $(\mathcal{M},\vec{\vec{u}},\vec{v}), t \vDash \varphi$ inductively as follows:*

$(\mathcal{M},\vec{\vec{u}},\vec{v}), t \vDash [\vec{Y}(\vec{n}) \leftarrow \vec{y}]\psi$ iff $(\mathcal{C}^{\vec{Y}(\vec{n})\leftarrow\vec{y}}, t) \vDash_{\mathsf{PLTL}} \psi$, where $\vDash_{\mathsf{PLTL}}$ is introduced in Definition 5;
$(\mathcal{M},\vec{\vec{u}},\vec{v}), t \vDash \neg\varphi$ iff $(\mathcal{M},\vec{\vec{u}},\vec{v}), t \nvDash \varphi$;
$(\mathcal{M},\vec{\vec{u}},\vec{v}), t \vDash \varphi \wedge \chi$ iff $(\mathcal{M},\vec{\vec{u}},\vec{v}), t \vDash \varphi \& (\mathcal{M},\vec{\vec{u}},\vec{v}), t \vDash \chi$.

Non-Recursiveness The temporal approach to SEM's proposed above not only allows to deal with non-recursive models without additional technical adjustments, but also often provides more elegant ways to describe the desired temporal behaviour of the system.

Consider Example 2 again. To model this scenario, we want our model $\mathcal{M}$ to contain $\mathcal{V} = \{T, R\}$ (for Treatment and Recovery), such that (1) $R = 1$ once the treatment is given twice in a row, and (2) once $R = 1$, it remains so. Let variable $U(\mathcal{R}(U) = \{0, 1\})$ represent whether the treatment is given in a given moment. And let $\mathcal{R}(T) = \{0, 1\}, \mathcal{R}(R) = \{0, \frac{1}{2}, 1\}$, where T=1 means the treatment is given. We want to define our structural equations in such a way that R=0 if the treatment is not given on the previous step, R=$\frac{1}{2}$ if the treatment was given once, and R=1 if the treatment was given twice in a row. Additionally, we require that if the patient is recovered, he must remain so.

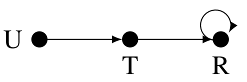


Figure 2: Non-recursive representation of Example 2.

The desired behaviour of the system may be achieved if we define $\mathcal{F}_R$ as follows. $R := 0$ if $(T = 0 \wedge R \neq 1)$; $R := \frac{1}{2}$ if $(T = 1 \wedge R = 0)$; $R := 1$ if $(T = 1 \wedge R = \frac{1}{2}) \vee R = 1$. This model is clearly non-recursive, because $\mathcal{F}_R$ depends of R. However, note that under temporal interpretation every edge in the dependency graph takes (at least) 1 time interval to proceed. It is a feature of computation $\mathcal{C}$, which performs consecutive calls to $\mathcal{F}$, where values of $\mathcal{V}$ at any step i depend on values of $(\mathcal{U} \cup \mathcal{V})$ on the *previous* step. So, $R := 1$ if $((T = 1 \wedge R = \frac{1}{2}) \vee R = 1)$ means that "R=1 is true *now* if on the *previous* step both T=1 and R=$\frac{1}{2}$ were true, or R=1 was true."

Given a temporal context, e.g. $\vec{\vec{u_1}} = (0, 1, 0, 1, 1, 0 \dots)$ and a default assignment $\vec{v} = 00$ (we write 00 instead of (T=0,R=0)), $(\mathcal{M}, \vec{\vec{u_1}}, \vec{v})$ generates a computation $\mathcal{C} = (00, 00, 10, 0\frac{1}{2}, 10, 1\frac{1}{2}, 01, 01, \dots)$, in which $\vec{\vec{u_1}}(1)$=1 triggers (T=1 at t=2), which triggers (R=$\frac{1}{2}$ at t=3). But since $\vec{\vec{u_1}}(2)$=0, T=0 becomes true at t=3, leading to (R=0 at t=4). Later, at step 4, T=1 happens again, triggering (R=$\frac{1}{2}$ at t=5). Since both T=1 and R=$\frac{1}{2}$ are true at t=5, R=1 triggers at t=6. From this moment, R=1 remains true at any t=i, because R=1 holds at i-1. This corresponds to the temporal be-

---

[4] You may notice that in this computation the bottle shatters at step 3 due to Suzy's throw, then BS=0 happens again at step 4 because both (ST=0, BT=0) hold at $\mathcal{C}(3)$, then BS=1 holds again at $\mathcal{C}(5)$ due to Billy's throw. This is an artifact of structural equations being defined in a specific way. We discuss how this can be fixed below.

havior we wanted to achieve in Example 2. Our CPLTL language allows to formulate such statements as $(\mathcal{M}, \vec{\vec{u}}, \vec{v}), 6 \vDash (R=1) \wedge \mathsf{H}\neg(R=1) \wedge [T(0) \leftarrow 1]\ominus^3 \mathsf{G}(R=1)$ meaning that at step 6 it is true that: (1) (R=1); (2) (R=1) has never been true before ($\mathsf{H}\neg(R=1)$); and (3) if the intervention $T \leftarrow 1$ was performed at step 0, R=1 would have been true for 3 time steps already (and would remain so forever). Note also that the same computation could be generated for the trivial context $\vec{\vec{u_2}} = (0, 0, \dots)$ and an intervention $int' = (T(2) \leftarrow 1, T(4) \leftarrow 1, T(5) \leftarrow 1)$, so $\mathcal{C}(\mathcal{M}, \vec{\vec{u_1}}, \vec{v}) = \mathcal{C}^{int'}(\mathcal{M}, \vec{\vec{u_2}}, \vec{v})$.

In many cases non-recursive causal models are the only way to represent mutually dependent variables and feedback loops processes, which are necessary to model many interesting phenomena. However, most of the literature on actual causality is restricted to recursive models because (in static settings) non-recursiveness may create serious technical difficulties, leading to non-uniqueness of the solution of structural equations with no clear way to choose the 'correct' one (Halpern 2016, Ch. 2.7). We argue that non-recursive models do not create any technical difficulty under the temporal interpretation of SEM's. But also sometimes provide fruitful modelling tools, as we will later see.

Modelling Assumptions Here we list our modelling assumptions. First of all, in our settings the time is discrete. This is a standard assumption for LTL-style temporal logics. We also assume that the temporal context $\vec{\vec{u}}$ represents a time series of exogenous changes given to us as an input. In this time series equal intervals between indexes correspond to equal time intervals. And similarly, equal time intervals correspond to equally spaced indices of the computation $\mathcal{C}$. Given $\vec{\vec{u}}$ time series, we want our model to return the correct time series of $\mathcal{V}$ values that (temporally) correspond to the behaviour of the phenomena of our interest. So, our framework requires causal models to contain correct temporal information, which affects the way we design them.

To illustrate this, let us revisit Example 1. Assume we know that Suzy's throws are consistently faster that Billy's. Let us say it takes n time steps (e.g. seconds) for Suzy's rock to reach the bottle, and k for Billy's, where $n < k$. So, whenever Suzy decides to throw the rock at time t_S and Billy at t_B, the Suzy's rock will reach the bottle (if it is still there) at time $t_S + n$ and Billy's at time $t_B + k$. We want our model to predict when the bottle will be shattered, given a temporal context $\vec{\vec{u}}$. So, our model must contain the information about 'delays' between the Suzy's (or Billy's) throw and the bottle shattering. One way to achieve this, is to add 'chains' of hidden (i.e. dummy) variables in the model (Figure 3 (a)).

Firstly, we assume that if BS=1 at some step, then it should remain so, because once the bottle is shattered it obviously remains so. This creates a reflexive arrow in the dependency graph. If they throw simultaneously at step i, then the bottle shatters at step $(i+n)$, because $n < k$. But now we can model situations when they decide to throw a rock at different time. So, if Suzy decides to throw at t_S and Billy at t_B, then the bottle will be shattered at $t^* := \min((t_S+n),(t_B+k))$, i.e. for $\vec{\vec{u}} = (00, 00, \dots)$,

$$(\mathcal{M}, \vec{\vec{u}}, \vec{v}), 0 \vDash [ST(t_S) \leftarrow 1, BT(t_B) \leftarrow 1]\bigcirc^{t^*}(BS=1).$$

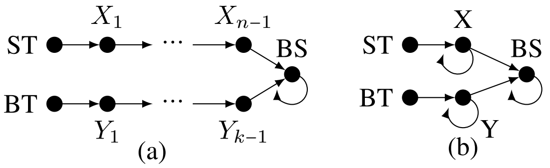
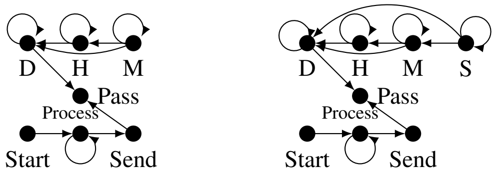


Figure 3: (a) 'Long' and (b)'Chain-free' models.



(a) Time scale = 1 min. (b) Time scale = 1 sec.

Figure 4: Dependency graphs for Example 3.

Such a representation is not compact, and non-recursive models provide us a better way to represent this example. Instead of adding a long chain of dependencies to capture time intervals between events and their effects, we can add a single variable to abbreviate each chain (Figure 3 (b)). Note, however, that it is not enough to specify the range of these new variables, X and Y, as $\mathcal{R}(X) = \{0, \dots, n-1\}$ and $\mathcal{R}(Y) = \{0, \dots, k-1\}$, where each value 'emulates' position of the rock on the original chain. This is because nothing prevents multiple variables in the corresponding chains $X_1 \dots X_{n-1}$ and $Y_1 \dots Y_{k-1}$ to have value 1 at the same time moment. This situation can be interpreted as multiple rocks thrown at different moments. To properly encode the temporal behaviour of the original model using a 'chain-free' model, the range of X and Y must contain all binary strings of the length $n-1$ and $k-1$ respectively. In other words, the new values must represent not only in which position the rock is at a given time moment, but also how many rocks there are. The equations then can be defined straightforwardly, and we omit the formal description due to lack of space. This construction provides a more compact graphical representation of the model, by reducing long chains of dependencies. It is easy to verify that as long as we are interested only in the variables $\{ST, BT, BS\}$, these two models behave identically wrt any context and time moment. We discuss the notion of temporal equivalence in detail in the next section.

Our models must adequately represent temporal behaviour of a system. This, in turn, requires a clear temporal semantics of each 'tick' of the model. To illustrate the problem, we present our final example.

Example 3 (Deadline). *Assume the agent has a deadline to perform some task. Let variables* D, H, M *with* $\mathcal{R}(D) = \{Mon, \dots, Sun\}), \mathcal{R}(H) = \{0, \dots, 23\}$ *and* $\mathcal{R}(M) = \{0, \dots, 59\}$ *represent days, hours and minutes respectively. At any moment the agent may decide to Start the* *task (Start=1). It takes 8 hours for the agent to process the task, so the range of Process variable contains* $8 * 60 = 480$ *values, representing minutes. Once the task is completed, it is sent to the server (Send=1). If this happens not later than Friday (D≠Sat∧D≠Sun), then the task is passed (Pass = 1); otherwise Pass = 0.*

The model shown in Figure 4 (a) obviously allows us to reason about the agent's decisions, and whether the deadline is met. However the interesting point is that there are clocks embedded in the model, and is obvious from the model itself which time interval we denote as 1 computational step. We can also easily change our time scale by adding a variable S for seconds and modifying the structural equation in obvious way, see Figure 4 (b). Note, however, that the two models are not equivalent (in the sense of the previous example), because now it takes 60 times more steps for any of the variable D, H, M to change value. At the same time, it allows us to model various sub-processes with higher accuracy. So, in our framework it is crucial to understand how 1 computational step of a model is interpreted in terms of real-world time, i.e., to understand which clocks are supposed to tick along with a model.

Temporal Equivalence for Causal Models

When dealing with structural equation modelling, we usually have many alternative causal models that describe the same underlying process. These models may have different sets of variables and describe causal dependencies in different ways. Moreover, at some point we may expand the set of variables in a model or reconsider some dependencies due to new discoveries. The only requirement ensuring that different models talk about the same process is that the models share some set of common variables. In such settings, it is crucial to have an adequate notion of equivalence between models to guarantee that different models correctly represent causal (and temporal) properties of some process with respect to the variables of interest (Beckers 2021). Since previous work has focused on static interpretation of SEMs (and so usually applicable only to recursive models), in this section we discuss how model equivalence can be treated in our framework.

Following (Beckers 2021), we assume that, two models $\mathcal{M}_1$ and $\mathcal{M}_2$ share the same exogenous variables ($\mathcal{U}_1 = \mathcal{U}_2$) and a set of *observable* variables $\mathcal{O} \subseteq (\mathcal{V}_1 \cap \mathcal{V}_2)$. So, we can only observe the values of and perform interventions on $\mathcal{O}$.

Definition 12 (Equivalent Computations)**.** *Consider two computations $\mathcal{C}_1$ and $\mathcal{C}_2$ sharing some set of variables $\mathcal{O}$. We call $\mathcal{C}_1$ and $\mathcal{C}_2$* temporally equivalent *wrt $\mathcal{O}$ if* $\forall X \in \mathcal{O}, \forall i > 0 : (X = x) \in \mathcal{C}_1(i)$ *iff* $(X = x) \in \mathcal{C}_2(i)$.

In other words, at any step equivalent computations agree on the values of all variables in $\mathcal{O}$.

Definition 13 (Model Equivalence)**.** *Two models $\mathcal{M}_1$ and $\mathcal{M}_2$ are temporally equivalent wrt $\mathcal{O}$ if for any intervention $\vec{Y}(\vec{n}) \leftarrow \vec{y}$, where $\vec{Y} \subseteq \mathcal{O}$ and for any $(\vec{\vec{u}}, \vec{v}_1)$ there exists $\vec{v}_2$, such that computations for $(\mathcal{M}_1^{\vec{Y}(\vec{n})\leftarrow\vec{y}}, \vec{\vec{u}}, \vec{v}_1)$ and $(\mathcal{M}_2^{\vec{Y}(\vec{n})\leftarrow\vec{y}}, \vec{\vec{u}}, \vec{v}_2)$ are equivalent, and vice versa.*

Temporal equivalence of causal models guarantees that no matter what intervention $\vec{Y}(\vec{n}) \leftarrow \vec{y}$ we use, there is no difference in $\mathcal{C}_1^{\vec{Y}(\vec{n})\leftarrow\vec{y}}$ and $\mathcal{C}_2^{\vec{Y}(\vec{n})\leftarrow\vec{y}}$ in how the endogenous changes in $\mathcal{O}$ proceed with exogenous changes $\vec{\vec{u}}$.

Observation 1. *Models $\mathcal{M}_a$ and $\mathcal{M}_b$ in Figure 3 are temporally equivalent for $\mathcal{O} = \{ST, BT, BS\}$.*

It is easy to check, that whatever the context $\vec{\vec{u}}$, default assignment $\vec{v}_1$ and an intervention $\vec{Y}(\vec{n}) \leftarrow \vec{y}$ for $\vec{Y} \subseteq \mathcal{O}$ are, there exists a default assignment $\vec{v}_2$, such that $(\mathcal{M}_a^{\vec{Y}(\vec{n})\leftarrow\vec{y}}, \vec{\vec{u}}, \vec{v}_1)$ and $(\mathcal{M}_b^{\vec{Y}(\vec{n})\leftarrow\vec{y}}, \vec{\vec{u}}, \vec{v}_2)$ generate $\mathcal{O}$-equivalent computations. However, this notion of equivalence is too strong to capture the similarities in Example 3.

Observation 2. *Models $\mathcal{M}_a$ and $\mathcal{M}_b$ from Example 3 are **not** temporally equivalent wrt $\mathcal{O} = \{Start, Pass\}$.*

We therefore need a more general notion of equivalence. Note that the models in Example 3 describe identical processes, but on a different time scale. To capture this aspect, we introduce the notion of *rescalable* equivalence.

Definition 14 (Rescalably Equivalent Computations)**.** *$\mathcal{C}_2$ is* rescalably *equivalent to $\mathcal{C}_1$ wrt $\mathcal{O}$ (with a coefficient $k \in \mathbb{N}$) if $\forall X \in \mathcal{O}, \forall i > 0 : (X = x) \in \mathcal{C}_1(i)$ iff $(X = x) \in \mathcal{C}_2(i \cdot k)$.*

Informally, k demonstrates how many ticks of $\mathcal{M}_2$ are needed to emulate one tick of $\mathcal{M}_1$. Given an intervention $\vec{Y}(\vec{n}) \leftarrow \vec{y}$ and a coefficient k, $\vec{Y}(\vec{n}^k) \leftarrow \vec{y}$ denotes an intervention, in which all indexes from $\vec{n}$ are multiplied by k.

Definition 15 (Rescalably Equivalent Models)**.** *A model $\mathcal{M}_2$ is* rescalably *equivalent (with a coefficient k) to $\mathcal{M}_1$ wrt $\mathcal{O}$ if for any intervention $\vec{Y}(\vec{n}) \leftarrow \vec{y}$ ($\vec{Y} \subseteq \mathcal{O}$) and for any $(\vec{\vec{u}}, \vec{v}_1)$ there exists $\vec{v}_2$, such the computation for $(\mathcal{M}_2^{\vec{Y}(\vec{n}^k)\leftarrow\vec{y}}, \vec{\vec{u}}, \vec{v}_2)$ is rescalably equivalent (with a coefficient k) to the computation for $(\mathcal{M}_1^{\vec{Y}(\vec{n})\leftarrow\vec{y}}, \vec{\vec{u}}, \vec{v}_1)$.*

Observation 3. *Model $\mathcal{M}_b$ in Example 3 is rescalably equivalent to $\mathcal{M}_a$ wrt $\mathcal{O} = \{Start, Pass\}$ with $k = 60$.*

We disagree with Beckers (2021) that it is impossible to come up with a useful notion of equivalence that takes into account 'numerical' properties. We believe that we have proposed such a notion, and it is useful because in temporal settings it is essential to consider not only *what* happened, but also how much time it took to happen.

Model-Checking

In this section, we study the model-checking complexity of CPLTL.

Definition 16 (Model-checking)**.** *The* CPLTL *model-checking problem is, given $(\mathcal{M}, \vec{\vec{u}}, \vec{v}, t)$ and a* CPLTL *formula φ, to decide whether $(\mathcal{M}, \vec{\vec{u}}, \vec{v}), t \models \varphi$.*

Note that the input to the problem is not necessarily finite or finitely presentable, since only $\mathcal{M}, \vec{v}$ and t are finite objects. To ensure that $\vec{\vec{u}}$ is finitely presentable, analogously to Definition 6, we require that $\vec{\vec{u}}$ is *ultimately periodic* of type (n, m), i.e. there exist $n, m > 0$ such that $\vec{\vec{u}}(k) = \vec{\vec{u}}(k + m)$

Algorithm 1: Computing $\mathcal{C}^{\vec{Y}(\vec{n})\leftarrow\vec{y}}$ of $(M, \overrightarrow{u}, \vec{v}, \vec{Y}(\vec{n}) \leftarrow \vec{y})$, type of $\overrightarrow{u}$ is (n, m)

```
1:  procedure PERIODIC-COMP($M, \overrightarrow{u}, \vec{v}, \vec{Y}(\vec{n}) \leftarrow \vec{y}$)
2:      $n^{int} \leftarrow \max(n_1, \dots, n_k \in \vec{n})$
3:      $n^* \leftarrow max(n, n^{int})$
4:      $C(0) \leftarrow \{X = x' \mid X(0) \leftarrow x' \in int\}$
5:      $C(0) \leftarrow C(0) \cup \{X = x'' \mid X = x'' \in \vec{v} \wedge$
                                  $\nexists x(X(0) \leftarrow x \in int)\}$
6:      for $i \in [1, n^*]$ do
7:          if $i < n^{int}$ then
8:              $C(i) \leftarrow \{X = x' \mid X(i) \leftarrow x' \in int\}$
9:              $C(i) \leftarrow C(i) \cup \{X = \mathcal{F}_X(\overrightarrow{u}(i-1), C(i-1)) \mid$
                                  $\nexists x(X(i) \leftarrow x \in int)\}$
10:         $C(i) \leftarrow \{X = \mathcal{F}_X(\overrightarrow{u}(i-1), C(i-1) \mid X \in \mathcal{V}\}$
11:     repeat
12:         $i \leftarrow i + 1$
13:         $C(i) \leftarrow \{X = \mathcal{F}_X(\overrightarrow{u}(i-1), C(i-1) \mid X \in \mathcal{V}\}$
14:     until $C(i) = C(j) \wedge (i \mod m) = (j \mod m)$
            for some $n^* \leq j < i$
15:     $y \leftarrow i - j$
16:     $x \leftarrow i - y$
17:     $\mathcal{C}^{int} \leftarrow C$
18: end procedure
```

for every $k \geq n$. We say that $\overrightarrow{u}[0, n-1]$ is the prefix and $\overrightarrow{u}[n, n+m]$ is the loop of $\overrightarrow{u}$.

Definition 17 (Model Size)**.** *The size of a periodic temporal context $\overrightarrow{u}$ of type (n, m) is*

$$|\overrightarrow{u}| = \sum_{0 \leq j \leq (n+m)} |\overrightarrow{u}(j)|.$$

The size of a model $\mathcal{M}$, denoted $|\mathcal{M}|$, is

$$|\mathcal{M}| := |\mathcal{U}| + |\mathcal{V}| + |\mathcal{R}| + |F|,$$

where $|\mathcal{R}| = \Sigma_{X \in (\mathcal{U} \cup \mathcal{V})} |\mathcal{R}(X)|$ and $|\mathcal{F}|$ is the cardinality of the set of tuples in the extensional definition of $\mathcal{F}$ (which is $|\mathcal{V}| \cdot \prod_{X \in (\mathcal{U} \cup \mathcal{V})} |\mathcal{R}(X)|$)). $|\vec{v}| = |\mathcal{V}|$. We assume that all numbers are written in unary, in particular $|t| = t$. Then, $|(\mathcal{M}, \overrightarrow{u}, \vec{v}, t)| = |\mathcal{M}| + |\overrightarrow{u}| + |\vec{v}| + |t|$.

The size of φ, denoted $|\varphi|$, is the number of symbols in φ, assuming that numbers are written in unary.[5]

Informally, our approach is as follows. First, we show that computation $\mathcal{C}^{int}$ for $(\mathcal{M}, \overrightarrow{u}, \vec{v})$ and an intervention int from φ is ultimately periodic of type (x, y) for some $x, y \in \mathbb{N}$. Then, we use a PLTL *path* model-checker (Markey 2002) to verify $(\mathcal{C}^{int}, t) \vDash_{\mathsf{PLTL}} \psi$ for each intervention subformula of φ of the form $[int]\psi$ and compute the value of their boolean combination.

Algorithm 1 first computes the prefix of the computation by applying all the interventions (lines 7–9) and, if necessary, continues the computation until the end of the prefix of $\overrightarrow{u}$ (line 10). We then start searching for a cycle (lines 11-14). Note that, to find a cycle, it is not sufficient to find i, j such that $\mathcal{C}(i) = \mathcal{C}(j)$ and $\overrightarrow{u}(i) = \overrightarrow{u}(j)$, as $\overrightarrow{u}(i)$ can be the same at different points in the loop of $\overrightarrow{u}$. Instead, we need to consider positions $p_0, \dots, p_{m-1}$ in the loop of $\overrightarrow{u}$ and find i, j such that $\mathcal{C}(i) = \mathcal{C}(j)$ and $\overrightarrow{u}$ *is at the same position in its loop* at i and at j.

Observe that the loop at lines (11–14) terminates after at most $m \cdot |\times_{X \in \mathcal{V}} \mathcal{R}(X)|$, where $\times_{X \in \mathcal{V}} \mathcal{R}(X)$ is the set of all possible assignments of the variables in $\mathcal{V}$. This is also the upper bound on y (the length of loop of $\mathcal{C}^{\vec{Y}(\vec{n})\leftarrow\vec{y}}$).

Given a periodic computation $\mathcal{C}$, a PLTL formula φ and a natural number i, the *path* model-checking problem is to decide whether $(\mathcal{C}, i) \vDash_{\mathsf{PLTL}} \varphi$. The future modalities $\bigcirc\psi$ and $\chi \mathsf{U} \psi$ can be solved straightforwardly by a standard labelling algorithm for LTL (e.g., (Demri, Goranko, and Lange 2016)) in time $\mathcal{O}(|\mathcal{C}| \cdot |\varphi|)$. However, this approach does not extend to the full PLTL. We therefore use the technique presented in (Markey 2002). Let $h_P(\varphi)$ denote the *past*-temporal height of φ, which is the maximum number of nested *past* modalities in φ.

Lemma 1 (Markey (2002))**.** *For any periodic $\mathcal{C}$ of type (x, y), $\varphi \in \mathsf{PLTL}$ and $k \geq x + h_P(\varphi) \cdot y$: $(\mathcal{C}, k) \vDash_{\mathsf{PLTL}} \varphi$ iff $(\mathcal{C}, k + y) \vDash_{\mathsf{PLTL}} \varphi$.*

This lemma states that after some initial segment of $\mathcal{C}$ past modalities in φ cannot distinguish how many times the loop has been repeated. In other words, in order to verify if $(\mathcal{C}, i) \vDash \varphi$ for any i, it is sufficient to check only the first $x + (h_P(\varphi) + 1) \cdot y$ elements of the computation $\mathcal{C}$, and if $i > x + (h_P(\varphi) + 1) \cdot y$, then it is sufficient to find $k = (i - (x + (h_P(\varphi) + 1) \cdot y) \mod y$ and check $(\mathcal{C}, (k + x + h_P(\varphi) \cdot y)) \vDash \varphi$.

Corollary 1 (Path model-checking)**.** *Given an ultimately periodic $\mathcal{C}$ of type (x, y), $\varphi \in \mathsf{PLTL}$ and $i \in \mathbb{N}$, path model-checking $(\mathcal{C}, i) \vDash_{\mathsf{PLTL}} \varphi$ can be done in time $\mathcal{O}((x + (h_P(\varphi) + 1) \cdot y) \cdot |\varphi|)$.*

Original proofs of Lemma 1 and Corollary 1 can be found in (Markey 2002). The only difference with our approach is that we use atomic statements $(X = x)$ instead of atomic propositions. Now we are ready to establish the main result.

Theorem 1. *CPLTL model-checking is in* P.

Proof. Let our input be $((\mathcal{M}, \overrightarrow{u}, \vec{v}, t), \varphi)$ where $\overrightarrow{u}$ has the prefix length n and the loop length m. Note that any CPLTL formula is a boolean combination of intervention subformulas of the form $[\vec{Y}(\vec{n}) \leftarrow \vec{y}]\psi$, where $\psi \in \mathsf{PLTL}$. For every intervention subformula $\chi' = [\vec{Y'}(\vec{n'}) \leftarrow \vec{y}]\psi'$ of φ we generate a computation $\mathcal{C}^{\vec{Y'}(\vec{n'})\leftarrow\vec{y'}}$ using Algorithm 1 and verify if the PLTL formula ψ' holds at $(\mathcal{C}^{\vec{Y'}(\vec{n'})\leftarrow\vec{y'}}, t)$. By Definition 11, $(\mathcal{C}^{\vec{Y'}(\vec{n'})\leftarrow\vec{y'}}, t) \vDash_{\mathsf{PLTL}} \psi'$ iff $(\mathcal{M}, \overrightarrow{u}, \vec{v}), t \vDash \chi'$. Finally, it remains to substitute the truth values of all subformulas in φ. The overall procedure requires $\mathcal{O}(|Sub(\varphi)|)$ calls to Algorithm 1, and each call takes $\mathcal{O}((\max(|\overrightarrow{u}|, |\varphi|) + |\overrightarrow{u}|) \cdot |\mathcal{M}|)$ steps to generate $\mathcal{C}'$. Checking each $(\mathcal{C}', t) \vDash_{\mathsf{PLTL}} \psi'$ can be done in $\mathcal{O}((n + (h_P(\varphi) + 1) \cdot m) \cdot |\varphi|)$ by Corollary 1, where $\mathcal{O}(|h_P(\varphi)|) = \mathcal{O}(|Sub(\varphi)|)$. The final substitution takes $\mathcal{O}(|Sub(\varphi)|)$ more steps. Thus, CPLTL model-checking problem is solvable in polynomial time. ☐

---

[5] Unary encoding gives more intuitive results, e.g., traversing a path up to position n is linear rather than exponential in $|n|$.

Related Work

The computational complexity of verifying actual causation in SEMs have been studied by many researchers, for example, Halpern (2015), Gladyshev et al. (2023) and de Lima and Lorini (2024).

Although Beckers and Vennekens (2018) argued that accounting for temporal information is crucial for actual causation judgments, to the best of our knowledge the framework proposed in this paper is the first attempt to integrate LTL-style temporal reasoning into actual causality settings.

Previous work on combining SEM models with temporal reasoning has focused on causal discovery, and several methods for analysing time-series data with SEMs have been developed, e.g., (Assaad, Devijver, and Gaussier 2022; Hyvärinen, Khemakhem, and Monti 2023). These methods typically use Full Time Causal Graphs to represent and reason about time-series using SEMs. Full Time Causal Graphs use time-indexed variables (Peters, Janzing, and Schölkopf 2017, Ch. 10) and thus potentially require specifying infinitely many structural equations. However, if all the causal relations remain constant over time, a Full Time Causal Graph can be abbreviated with a (finite) Window Causal Graph or a (finite) Summary Causal Graph (Assaad, Devijver, and Gaussier 2022). Though the dependency graphs of our models resemble both Window and Summary graphs in the way they interpret cyclic dependencies, there are important differences. Firstly, conventional models of time-series allow contemporaneous dependencies, i.e., a (time-indexed) variable X_t may depend on the value of another variable Y_t at the same time step t. This modelling choice is usually driven by the observational limitations in the field of causal discovery, e.g., the sampling frequency of the time series may not be able to separate causes and effects. In the field of actual causality, in contrast, we assume that a given causal model provides a complete representation of reality, and thus we assume that the unit interval is sufficiently small to separate causes and subsequent effects. So, contemporaneous relations do not occur in our framework. Secondly, both Full Time and Window graphs allow arbitrary 'lagged' dependencies, i.e., a variable X at time t may (directly) depend on another variable Y at time $t-n$ for arbitrary n. In our models we interpret all the direct dependencies as 1-step lagged. We conjecture that for any arbitrary lagged model there exists an equivalent (Definition 13) 1-step lagged model.

In contrast to causal discovery, where cyclic causal models are sometimes used, e.g. (Bongers et al. 2021), current research in actual causality is mostly focused on acyclic (recursive) models. To the best of our knowledge, there has been relatively little work which considers non-recursive models, e.g., Halpern (2000) studies axiomatizations for different classes of SEM's, in particularly non-recursive ones, and Halpern (2016, Ch. 2.7) discusses the definition of an *actual cause* in such models. Finally, Halpern and Peters (2022) introduce so-called *Generalized* SEMs (GSEM) and axiomatize different classes of GSEMs, including non-recursive ones. Note that in GSEMs structural equations $\mathcal{F}$ are replaced with $\mathbf{F}$ mapping contexts and interventions to sets of outcomes. We argue that, by staying closer to original formalism, our temporal framework accommodates non-recursive models without significant adjustments.

Defining the equivalence of causal models is another well-recognized problem both in causal discovery and actual causality. We have already mentioned the work of Beckers (2021), but the problem was recognized already by Verma and Pearl (1990). A similar notion of causal consistency has also been studied in (Rubenstein et al. 2017).

Similarly to equivalence, causal consistency, intuitively guarantees that two models agree in their predictions of the effects of interventions. Rubenstein et al. (2017) introduced the notion of exact transformations between SEMs preserving causal consistency, and Willig et al. (2023) proposed another type of transformations, called consolidating mechanisms, to transform large-scale SEMs into smaller, computationally efficient ones. Both of these approaches to consistency-preserving transformations are applicable to time-series models. We believe studying our models as transformations of existing time-series models based on (Rubenstein et al. 2017; Willig et al. 2023) is an interesting direction for future work.

Discussion

We have proposed a novel conceptual interpretation of existing formalisms in the field of actual causality. While causal models have been developed as a useful tool for representing static dependencies between the variables, we demonstrate that this formalism (with minor modifications) can be treated as a mechanism able to transform a time series of exogenous values into a time series of endogenous ones. Though our approach does not allow us to extract temporal information from existing static models which were not designed with this purpose, it provides new insights and techniques for (temporal) structural equation modelling.

We make a number of technical contributions. We introduced the core concept of a computation, which treats structural equations as 'time-lagged' and allows us to 'unwind' a causal scenario $(\mathcal{M}, \vec{\vec{u}}, \vec{v})$ into a time series $\mathcal{C}$ of the assignments to $\mathcal{V}$. In addition, we proposed the logic CPLTL, which combines the temporal logic LTL with causal time-sensitive interventions. Finally, we introduced new notions of temporal equivalence for causal models and showed that the model-checking problem for CPLTL is in P.

We believe there are a number of interesting directions for future work. First, in this paper we discussed only finite models, however the extension of our framework to models with infinitely many variables (and potentially infinite range) seems to be straightforward. Secondly, the problems of defining a (bi)simulation relation between temporal causal models (which would imply their temporal equivalence) as well as defining weaker notions of temporal equivalence are left open. Finally, another promising direction for future work is adaptation of our framework to probabilistic settings, when we have a probability distribution on temporal contexts or non-deterministic structural equations as recently proposed by Beckers (2024).

References

Assaad, C. K.; Devijver, E.; and Gaussier, E. 2022. Survey and Evaluation of Causal Discovery Methods for Time Series. *J. Artif. Int. Res.*, 73.

Beckers, S. 2021. Equivalent Causal Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35: 6202–6209.

Beckers, S. 2024. Nondeterministic Causal Models. arXiv:2405.14001.

Beckers, S.; and Vennekens, J. 2018. A principled approach to defining actual causation. *Synthese*, 195(2): 835–862.

Bongers, S.; Forré, P.; Peters, J.; and Mooij, J. M. 2021. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 49(5): 2885 – 2915.

de Lima, T.; and Lorini, E. 2024. Model Checking Causality. In *Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)*, 3324–3332.

Demri, S.; Goranko, V.; and Lange, M. 2016. *Temporal Logics in Computer Science: Finite-State Systems*. Cambridge: Cambridge University Press.

Gladyshev, M.; Alechina, N.; Dastani, M.; Doder, D.; and Logan, B. 2023. Dynamic Causality. In *Proceedings of the 26th European Conference on Artificial Intelligence*, 867–874.

Halpern, J. Y. 2000. Axiomatizing Causal Reasoning. *Journal of Artificial Intelligence Research*, 12: 317–337.

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

Halpern, J. Y. 2016. *Actual Causality*. The MIT Press. ISBN 9780262336611.

Halpern, J. Y.; and Pearl, J. 2005. Causes and Explanations: A Structural-Model Approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 56(4): 843–887.

Halpern, J. Y.; and Peters, S. 2022. Reasoning about Causal Models with Infinitely Many Variables. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5): 5668–5675.

Hyvärinen, A.; Khemakhem, I.; and Monti, R. 2023. Identifiability of latent-variable and structural-equation models: from linear to nonlinear. arXiv:2302.02672.

Markey, N. 2002. Past is for Free: on the Complexity of Verifying Linear Temporal Properties with Past. *Electronic Notes in Theoretical Computer Science*, 68(2): 87–104. EXPRESS'02, 9th International Workshop on Expressiveness in Concurrency (Satellite Workshop of CONCUR 2002).

Pearl, J. 2000. *Causality: Models, Reasoning, and Inference.* New York: Cambridge University Press. ISBN 0-521-77362-8.

Peters, J.; Janzing, D.; and Schölkopf, B. 2017. *Elements of Causal Inference. Foundations and Learning Algorithms*. Cambridge, Massachusetts: The MIT Press.

Pnueli, A. 1977. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science*, 46–57.

Rubenstein, P. K.; Weichwald, S.; Bongers, S.; Mooij, J. M.; Janzing, D.; Grosse-Wentrup, M.; and Schölkopf, B. 2017. Causal consistency of structural equation models. *arXiv preprint arXiv:1707.00819*.

Spirtes, P.; Glymour, C.; and Scheines, R. 2001. *Causation, Prediction, and Search*. The MIT Press. ISBN 9780262284158.

Verma, T.; and Pearl, J. 1990. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, 255–270. USA: Elsevier Science Inc. ISBN 0444892648.

Willig, M.; Zečević, M.; Dhami, D. S.; and Kersting, K. 2023. Do Not Marginalize Mechanisms, Rather Consolidate! In *Thirty-seventh Conference on Neural Information Processing Systems*.