

# A note on the complexity of the causal ordering problem

Bernardo Gonçalves<sup>a</sup>, Fabio Porto<sup>b</sup>

<sup>a</sup>*IBM Research, São Paulo, Brazil*

<sup>b</sup>*National Laboratory for Scientific Computing (LNCC), Petrópolis, Brazil*

---

## Abstract

In this note we provide a concise report on the complexity of the causal ordering problem, originally introduced by Simon to reason about causal dependencies implicit in systems of mathematical equations. We show that Simon’s classical algorithm to infer causal ordering is NP-Hard—an intractability previously guessed but never proven. We present then a detailed account based on Nayak’s suggested algorithmic solution (the best available), which is dominated by computing transitive closure—bounded in time by  $O(|\mathcal{V}| \cdot |\mathcal{S}|)$ , where  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  is the input system structure composed of a set  $\mathcal{E}$  of equations over a set  $\mathcal{V}$  of variables with number of variable appearances (density)  $|\mathcal{S}|$ . We also comment on the potential of causal ordering for emerging applications in large-scale hypothesis management and analytics.

*Keywords:* Causal ordering, Causal reasoning, Structural equations, Hypothesis management.

---

## 1. Introduction

The causal ordering problem has long been introduced by Simon as a technique to infer the causal dependencies implicit in a deterministic mathematical model [1]. For instance, let  $f_1(x_1)$  and  $f_2(x_1, x_2)$  be two equations defined over variables  $x_1, x_2$ . Then the causal ordering problem is to infer all existing causal dependencies, in this case the only one is  $(x_1, x_2)$ , read ‘ $x_2$  causally depends on  $x_1$ .’ It is obtained by first matching each equation to a variable that appears in it, e.g.,  $f_2 \mapsto x_2$ . Intuitively, this means that  $f_2$  is to be assigned to compute the value of  $x_2$ —using the value of  $x_1$ , which establishes

---

*Email addresses:* [begoncalves@acm.org](mailto:begoncalves@acm.org) (Bernardo Gonçalves), [fporto@lncc.br](mailto:fporto@lncc.br) (Fabio Porto)

a direct causal dependency between these two variables. Indirect dependencies may then arise and can be computed, which is specially useful when the system of equations is very large.

Causal ordering inference can then support users with uncertainty management, say, towards the discovery of what is wrong with a model for enabling efficient and effective modeling intervention. If multiple values of  $x_1$  are admissible for a modeler, then as a user of the causal ordering machinery she has support to track their influence on the values of  $x_2$ . One major application for that is probabilistic database design [2].

Back in the 50th's, Simon was motivated by studies in econometrics and not very concerned with the algorithmic aspects of the Causal Ordering Problem (COP). Yet his vision on COP and its relevance turned out to be influential in the artificial intelligence literature. In a more recent study, Dash and Druzdzel revisit and motivate it in light of modern applications [3]. They show that Simon's original algorithm, henceforth the Causal Ordering Algorithm (COA), is correct in the sense that any valid causal ordering that can be extracted from a self-contained (well-posed) system of equations must be compatible with the one that is output by COA [3]. Their aim has also been (sic.) to validate decades of research that has shown the causal ordering to provide a powerful tool for operating on models. In addition to the result on the correctness of COA, their note also provides a convenient survey of related work that connects to Simon's early vision on causal reasoning.

However, Simon's formulation of COP into COA—originally in [1], and reproduced in [3], turns out to be intractable. There is a need to distinguish Simon's COA from COP itself. The former still seems to be the main entry point to the latter in the specialized literature. In fact, there is a lack of a review on the computational properties of COA—and as we show in this note, it tries to address an NP-Hard problem as one of its steps. The interested reader who needs an efficient algorithmic approach to address COP in a real, large-scale application can only scarcely find some comments spread through Nayak [4, p. 287-91], and then Iwasaki and Simon [5, p. 149] and Pearl [6, p. 226] both pointing to the former. Regarding Simon's COA itself, the classical approach to COP, it is only Nayak who suggests in words that (sic.) '[it] is a worst-case exponential time algorithm' [7, p. 37]. We discuss this ambiguity that exists in the most up-to-date literature shortly in §1.2.

COP is significant also in view of emerging applications in large-scale hypothesis management and analytics [2]. The modeling of physical and socio-economical systems as a set of mathematical equations is a traditional

approach in science and engineering and a very large bulk of models exist which are ever more available in machine-readable format. Simon’s early vision on the automatic extraction of the “causal mechanisms” implicit in (large-scale) models for the sake of informed intervention finds nowadays new applications in the context of open simulation laboratories [8], large-scale model management [9] and online, shared model repositories [10, 11, 12].

In this paper we review the causal ordering problem (§2). Our core contributions are (§3) to originally show that COA aims at addressing an NP-Hard problem, confirming Nayak’s earlier intuition; and then (§4) to organize into a concise yet complete note his hints to solve COP in polynomial time.

### 1.1. Informal Preliminaries

Given a system of mathematical equations involving a set of variables, the *causal ordering problem* consists in detecting the hidden asymmetry between variables. As an intermediate step towards it, one needs to establish a one-to-one mapping between equations and variables [1].

For instance, Einstein’s famous equation  $E = m c^2$  states the equivalence of mass and energy, summarizing (in its scalar version) a theory that can be imposed two different asymmetries for different applications. Say, given a fixed amount of mass  $m = m_0$  (and recalling that  $c$  is a constant), find the particle’s relativistic rest energy  $E$ ; or rather, given the particle’s rest energy, find its mass or potential for nuclear fission. That is, the causal ordering depends on what variables are set as input and which ones are “influenced” by them. Suppose there is uncertainty, say, a user considers two values to set the mass,  $m = m_0$  or  $m = m'_0$ . Then this uncertainty will flow through the causal ordering and affect all variables that are dependent on it (energy  $E$ ).

For really large systems, having structures say in the order of one million equations [13], the causal ordering problem is critical to provide more specific accountability on the accuracy of the system—viz., what specific variables and subsystems account for possibly inaccurate outcomes. This is key for managing and tracking the uncertainty of alternative modeling variations systematically [8, 13].

### 1.2. Related Work

**COA.** Dash and Druzdel [3] provide a high-level description of how Simon’s COA [1] proceeds to discover the causal dependencies implicit in a structure. It returns a ‘partial’ causal mapping: from partitions on the set

of equations to same-cardinality partitions on the set of variables—a ‘total’ causal mapping would instead map every equation to exactly one variable.

They show then that any valid total causal mapping produced over a structure must be consistent with COA’s partial causal mapping. Nonetheless, no observation at all is made regarding COA’s computational properties in [3], leaving in the most up-to-date literature an impression that Simon’s COA is the way to go for COP. In this note we show that Simon’s COA tries to address an NP-Hard problem in one of its steps, and then clearly recommend Nayak’s efficient approach to COP as a fix to COA.

**COP.** Inspired by Serrano and Gossard’s work on constraint modeling and reasoning [14], Nayak describes an approach that is provably efficient to process the causal ordering: extract any valid total causal mapping and then compute the transitive closure of the direct causal dependencies, viz, the causal ordering. Nayak’s is a provably correct approach to COP, as all valid ‘total’ causal mappings must have the same causal ordering.

In this note we arrange those insights into a concise yet detailed recipe that can be followed straightforwardly to solve COP efficiently.

## 2. The Causal Ordering Problem

We start with some preliminaries on notation and basic concepts to eventually state COP formally.

For an equation  $f(x_1, x_2, \dots, x_\ell) = 0$ , we will write  $Vars(f) \triangleq \{x_1, x_2, \dots, x_\ell\}$  to denote the set of variables that appear in it.

**Def. 1.** A **structure** is a pair  $\mathcal{S}(\mathcal{E}, \mathcal{V})$ , where  $\mathcal{E}$  is a set of equations over a set  $\mathcal{V}$  of variables,  $\mathcal{V} \triangleq \bigcup_{f \in \mathcal{E}} Vars(f)$ , such that:

- (a) In any subset  $\mathcal{E}' \subseteq \mathcal{E}$  of  $k > 0$  equations of the structure, at least  $k$  different variables appear, i.e.,  $|\mathcal{E}'| \leq |\mathcal{V}'|$ ;
- (b) In any subset of  $k > 0$  equations in which  $r$  variables appear,  $k \leq r$ , if the values of any  $(r - k)$  variables are chosen arbitrarily, then the values of the remaining  $k$  variables can be determined uniquely—finding these unique values is a matter of solving the equations.

Note in Def. 1 that structures are composed of equations, and variables are only part of them indirectly as part of equations. Accordingly, set inclusion and all set operations such as union, intersection and difference are computed

w.r.t. the equations. That is, if  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  and  $\mathcal{S}'(\mathcal{E}', \mathcal{V}')$  are structures, then we write  $\mathcal{S}' \subset \mathcal{S}$  when  $\mathcal{E}' \subset \mathcal{E}$ . An additional operation for ‘variables elimination’ shall be used. We write  $\mathcal{T} := \mathcal{S} \div \mathcal{S}'$ , to denote a structure  $\mathcal{T}$  resulting from both (i) removing equations  $\mathcal{E}'$  from  $\mathcal{E}$ , and (ii) enforcing elimination of variables  $\mathcal{V}' = \bigcup_{f \in \mathcal{E}'} \text{Vars}(f)$  from  $\mathcal{E} \setminus \mathcal{E}'$ .

**Def. 2.** Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a structure. We say that  $\mathcal{S}$  is *self-contained* or **complete** if  $|\mathcal{E}| = |\mathcal{V}|$ .

In short, COP will be concerned with systems of equations that are ‘structural’ (Def. 1) and ‘complete’ (Def. 2), viz., that have as many equations as variables and no subset of equations has fewer variables than equations.<sup>1</sup>

Now Def. 3 introduces a data structure that is an intermediate product towards addressing COP. We shall state COP formally in the sequel.

**Def. 3.** Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a complete structure. Then a **total causal mapping** over  $\mathcal{S}$  is a bijection  $\varphi: \mathcal{E} \rightarrow \mathcal{V}$  such that, for all  $f \in \mathcal{E}$ , if  $\varphi(f) = x$  then  $x \in \text{Vars}(f)$ .

Note that such total causal mapping  $\varphi$  induces a set  $C_\varphi$  of *direct causal dependencies* (see Eq. 1), which shall give us the *causal dependencies* (Def. 4).

$$C_\varphi = \{ (x_a, x_b) \mid \text{there exists } f \in \mathcal{E} \text{ such that } \varphi(f) = x_b \text{ and } x_a \in \text{Vars}(f) \} \quad (1)$$

**Def. 4.** Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a structure with variables  $x_a, x_b \in \mathcal{V}$ , and  $\varphi$  a total causal mapping over  $\mathcal{S}$  inducing set of direct causal dependencies  $C_\varphi$  and indirectly a transitive closure  $C_\varphi^+$ . We say that  $(x_a, x_b)$  is a **direct causal dependency** in  $\mathcal{S}$  if  $(x_a, x_b) \in C_\varphi$ , and that  $(x_a, x_b)$  is a **causal dependency** in  $\mathcal{S}$  if  $(x_a, x_b) \in C_\varphi^+$ .

In other words,  $(x_a, x_b)$  is in  $C_\varphi$  iff  $x_b$  direct and causally depends on  $x_a$ , given the causal asymmetries induced by  $\varphi$ . Then by transitive reasoning on  $C_\varphi$  we shall be able to infer the transitive closure  $C_\varphi^+$ , which is the *causal ordering*. Now we can state COP more formally as Problem 1.

---

<sup>1</sup>Also, for inferring causal ordering the systems of equations given as input is expected to be ‘independent,’ i.e., can only have non-redundant equations.

**Problem 1.** (COP). *Given a complete structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$ , find a total causal mapping  $\varphi$  over  $\mathcal{S}$  and derive a set  $C_\varphi^+$  of causal dependencies induced by it.*

In the sequel we shall see two different algorithmic approaches to COP (Problem 1). First, the classical approach informally described by Simon in the 50th's [1], and reproduced recently in [3]; and then Nayak's one proposed in the 90th's [4]. We shall present the algorithms and analyze their corresponding complexities.

### 3. Simon's Causal Ordering Algorithm and its Complexity

We introduce now additional concepts that are specific to Simon's COA.

**Def. 5.** *Let  $\mathcal{S}$  be a structure. We say that  $\mathcal{S}$  is **minimal** if it is complete and there is no complete substructure  $\mathcal{S}' \subset \mathcal{S}$ .*

**Example 1.** *Consider structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$ , where  $\mathcal{E} = \{f_1(x_1), f_2(x_2), f_3(x_3), f_4(x_1, x_2, x_3, x_4, x_5), f_5(x_1, x_3, x_4, x_5), f_6(x_4, x_6), f_7(x_5, x_7)\}$ . Note that  $\mathcal{S}$  is complete, as  $|\mathcal{E}| = |\mathcal{V}| = 7$ , but not minimal. There are exactly three minimal substructures  $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3 \subset \mathcal{S}$ , whose sets of equations are  $\mathcal{E}_1 = \{f_1(x_1)\}$ ,  $\mathcal{E}_2 = \{f_2(x_2)\}$ ,  $\mathcal{E}_3 = \{f_3(x_3)\}$ .  $\square$*

Now Lemma 1 and Proposition 1 are stated to back up a 'disjointness' assumption that is made by COA upon minimal structures (Def. 5). The proof we present here for Proposition 1 is a conveniently derived alternative to Simon's own proof to his original 'theorem 3.2' [1, p. 59].

**Lemma 1.** *Let  $\mathcal{S}_1(\mathcal{E}_1, \mathcal{V}_1)$  and  $\mathcal{S}_2(\mathcal{E}_2, \mathcal{V}_2)$  be structures. If  $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$  then  $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$  (i.e.,  $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$ ). That is, disjointness of variables is strong enough to warrant disjointness of equations.*

**Proof 1.** *Let  $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ . Now by contradiction assume  $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$ , then there must be at least one shared equation  $f \in \mathcal{E}_1, \mathcal{E}_2$ . Since both  $\mathcal{S}_1, \mathcal{S}_2$  are structures, by Def. 1 we know that  $|\text{Vars}(f)| \geq 1$  and  $\text{Vars}(f) \subseteq \mathcal{V}_1 \cap \mathcal{V}_2$ . Yet  $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ .  $\nabla$ . Therefore  $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$  implies  $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$ .  $\square$*

**Def. 6.** *Let  $\mathcal{S}_1(\mathcal{E}_1, \mathcal{V}_1)$  and  $\mathcal{S}_2(\mathcal{E}_2, \mathcal{V}_2)$  be structures. Then we say that they are **disjoint** if  $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ .*

**Proposition 1.** *Let  $\mathcal{S}$  be a complete structure. If  $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{S}$  are any different minimal substructures of  $\mathcal{S}$ , then they are disjoint.*

**Proof 2.** *We show the statement by case analysis and then contradiction out of Defs. 1–2 and Defs. 5–6. See Appendix A.  $\square$*

Simon’s COA is also based on a data structure introduced in Def. 7.

**Def. 7.** *The **structure matrix**  $A_{\mathcal{S}}$  of a structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$ , with  $f_1, f_2, \dots, f_n \in \mathcal{E}$  and  $x_1, x_2, \dots, x_m \in \mathcal{V}$ , is a  $|\mathcal{E}| \times |\mathcal{V}|$  matrix of 1’s and 0’s in which entry  $a_{ij}$  is non-zero if variable  $x_j$  appears in equation  $f_i$ , and zero otherwise.*

Elementary row operations on the structure matrix may hinder the structure’s causal ordering and then are not valid in general [1]. This also emphasizes that the problem of causal ordering is not about solving the system of equations of a structure, but identifying its hidden asymmetries.

### 3.1. Simon’s Causal Ordering Algorithm

Simon has described his Causal Ordering Algorithm (COA) only informally at a high level of abstraction [1]. It is given a complete structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  and computes a causal mapping  $\varphi$ . The causal ordering itself is to be obtained as a post-processing (transitive closure) out of the causal mapping  $\varphi$  and its induced set  $C_{\varphi}$  of direct causal dependencies. Example 1 (continued) warms up for Simon’s algorithm.

**Example 1.** *(continued). Fig. 1a shows the matrix of the structure  $\mathcal{S}$  given above in this example. By eliminating the variables identified with the minimal substructures  $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3 \subset \mathcal{S}$ , a smaller structure  $\mathcal{T}$  is derived to be input at the next recursive step (see Fig. 1b). This is the main insight of Simon’s to arrive at his recursive causal ordering algorithm, as described next.  $\square$*

Algorithm 1 describes the variant of Simon’s original description that returns a ‘total’ causal mapping (satisfies Def. 3).<sup>2</sup> We refer to its core

---

<sup>2</sup>This slight variation takes place in lines 7–10 of RTCM in Algorithm 1, and is irrelevant to its intractability—which we shall see is due to line 3. Besides, ‘total’ and ‘partial’ causal mappings are interchangeable [3]. In particular, recovering the latter from the former is straightforward: just merge ‘strongly coupled’ variables in a cluster. Intuitively, these are variables whose values can only be determined simultaneously. To be precise, let  $x_1, x_2 \in \mathcal{V}$  be variables in a structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$ . We say  $x_1, x_2$  are *strongly coupled* if  $\mathcal{S}$  is minimal.

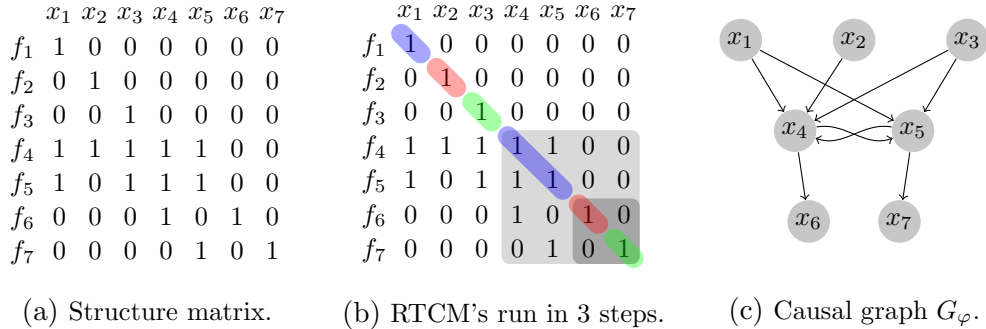


Figure 1: Simon’s RTCM, the core procedure in COA. Fig. 1a: a structure matrix given. Fig. 1b: minimal substructures detected in each recursive step  $k$  are highlighted in shades of gray and have their diagonal elements colored. Fig. 1c: Causal graph  $G_\varphi$  induced by mapping  $\varphi$  over structure  $\mathcal{S}$ . An edge connects a node  $x_i$  towards a node  $x_j$ , with  $x_i, x_j \in \mathcal{V}$ , iff  $x_i$  appears in the equation  $f \in \mathcal{E}$  such that  $\varphi(f) = x_j$ . As the mapping  $\varphi$  is not unique, accordingly the causal graph  $G_\varphi$  is not either—e.g., consider  $\varphi'$  with  $f_4 \mapsto x_5$  and  $f_5 \mapsto x_4$ . The induced graph  $G_{\varphi'}$  would have, e.g., a connection from  $x_2$  to  $x_5$  instead. Yet their graph transitive closure is the same,  $tc(G_\varphi) = tc(G_{\varphi'})$ , as we shall see in §4.

procedure as RTCM (recursive total causal mapping). It comprises the actual source of intractability in Simon’s original description, while lines 3-7 of the COA procedure were not described by himself but only considered as matter of a post-processing. We illustrate RTCM through Example 1 and Fig. 1.

**Example 1.** (continued). Let  $\mathcal{T} = \mathcal{S} \div (\mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3)$  be the structure returned by COA’s first recursive step  $k = 0$  for this example. Then a valid total causal mapping that can be returned at  $k = 1$  (see Fig.1b) is  $COA(\mathcal{T}) = \{\langle f_4, x_4 \rangle, \langle f_5, x_5 \rangle\}$ . Since  $x_4$  and  $x_5$  are strongly coupled, COA maps them arbitrarily (e.g., it could be  $f_4 \mapsto x_5, f_5 \mapsto x_4$  instead). Such total causal mapping  $\varphi$  renders a cycle in the directed causal graph  $G_\varphi$  induced by  $\varphi$  (see Fig.1c), which might not be desirable for some applications.  $\square$

### 3.2. Hardness of Simon’s Recursive COA

First of all, we state a decision problem associated with finding the minimal structures in a given structure (line 3 of Simon’s RTCM procedure in Algorithm 1). For short, we shall refer to this problem as the Complete Substructure Decision Problem (CSDP).



---

**Algorithm 1** Simon’s Causal Ordering Algorithm based on RTCM.

---

```

1: procedure COA( $\mathcal{S}$ : structure over  $\mathcal{E}$  and  $\mathcal{V}$ )
Require:  $\mathcal{S}$  given is complete, i.e.,  $|\mathcal{E}| = |\mathcal{V}|$ 
Ensure: Returns  $C_\varphi^+$ , the causal ordering of  $\mathcal{S}$ 
2:    $\varphi \leftarrow \text{RTCM}(\mathcal{S})$        $\triangleright$  gets total causal mapping  $\varphi$  by Simon’s recursive algorithm
3:    $C_\varphi \leftarrow \emptyset$            $\triangleright$  initializes set of direct causal dependencies
4:   for all  $\langle f, x \rangle \in \varphi$  do
5:     for all  $x_a \in \text{Vars}(f) \setminus \{x\}$  do
6:        $C_\varphi \leftarrow C_\varphi \cup \{(x_a, x)\}$ 
7:   return  $\text{TC}(C_\varphi)$        $\triangleright$  returns the transitive closure of  $C_\varphi$ , as described in §4.2

```

---

```

1: procedure RTCM( $\mathcal{S}$ : structure over  $\mathcal{E}$  and  $\mathcal{V}$ )
Require: Structure  $\mathcal{S}$  given is complete, i.e.,  $|\mathcal{E}| = |\mathcal{V}|$ 
Ensure: Returns total causal mapping  $\varphi : \mathcal{E} \rightarrow \mathcal{V}$ 
2:    $\varphi \leftarrow \emptyset, \mathcal{S}^* \leftarrow \emptyset, D \leftarrow \emptyset$   $\triangleright$  initializes
3:   identify all minimal substructures  $\mathcal{S}' \subseteq \mathcal{S}$ 
4:   for all minimal  $\mathcal{S}' \subseteq \mathcal{S}$  do
5:      $\mathcal{S}^* \leftarrow \mathcal{S}^* \cup \mathcal{S}'$   $\triangleright$  aggregates into  $\mathcal{S}^*$  each minimal substructure scanned
6:     for all  $f \in \mathcal{E}'$ , where  $\mathcal{S}'$  do
7:        $x \leftarrow$  any  $x_a$  such that  $x_a \in \text{Vars}(f)$  and  $x_a \notin D$ 
8:        $\varphi \leftarrow \varphi \cup \langle f, x \rangle$   $\triangleright$  maps to  $f$  some variable  $x \in \text{Vars}(f)$ 
9:        $D \leftarrow D \cup \{x\}$   $\triangleright$  aggregates into  $D$  the variables already ‘matched’
10:   $\mathcal{T} \leftarrow \mathcal{S} \div \mathcal{S}^*$   $\triangleright$  removes  $\mathcal{E}^*$ ; eliminates  $\mathcal{V}^* = \bigcup_{f \in \mathcal{E}^*} \text{Vars}(f)$ , where n.b.,  $\mathcal{V}^* = D$ 
11:  if  $\mathcal{T} \neq \emptyset$  then
12:    return  $\varphi \cup \text{RTCM}(\mathcal{T})$ 
13:  return  $\varphi$ 

```

---

(CSDP). Given a complete structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  with  $|\mathcal{E}| = |\mathcal{V}| = m$  and an integer  $1 \leq \ell < m$ , does  $\mathcal{S}$  have a complete substructure  $\mathcal{S}'(\mathcal{E}', \mathcal{V}')$  with  $|\mathcal{E}'| = |\mathcal{V}'| = \ell$ ?

In this section we carry out an original study on CSDP and show that it is NP-Complete. We consider a basic observation by Nayak [4] apud. [14], that there is a correspondence between Simon’s structures and bipartite graphs. A graph is said *bipartite* if its vertices can be divided into two disjoint sets  $V_1$  and  $V_2$  and every edge connects a vertex in  $V_1$  to one in  $V_2$  [15]. Moreover it is said to be  *$\ell$ -balanced* if  $|V_1| = |V_2| = \ell$ , and is said to be *connected* if  $\text{deg}(w) \geq 1$  for all  $w \in V_1 \cup V_2$ . Def. 8 introduces the mentioned correspondence and provides us some shorthand notation.

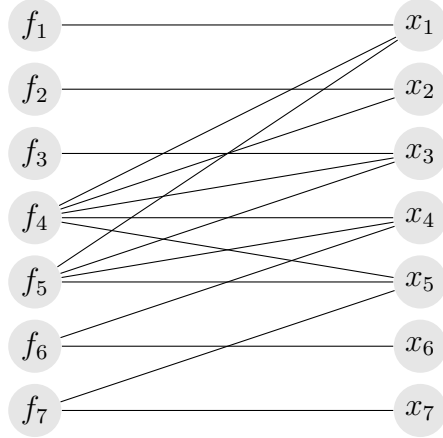


Figure 2: Bipartite graph  $G$  of structure  $\mathcal{S}$  from Example 1.

**Def. 8.** Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a structure, and  $G = (V_1 \cup V_2, E)$  be a bipartite graph where  $V_1 \mapsto \mathcal{E}$  and  $V_2 \mapsto \mathcal{V}$ , and  $E \mapsto \mathcal{S}$  so that an edge  $(f, x) \in E$  if and only if we have  $x \in \text{Vars}(f)$ . We say that  $G$  is the bipartite graph that **corresponds to** structure  $\mathcal{S}$ , and for short write  $G \sim \mathcal{S}$ .

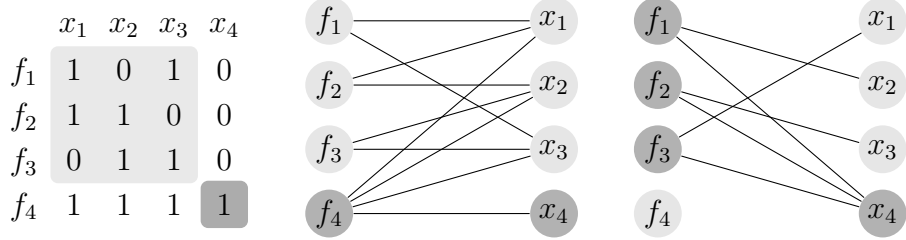
Fig. 2 shows the bipartite graph  $G \sim \mathcal{S}$ , where  $\mathcal{S}$  is the initial structure given in Example 1.

Def. 9 introduces a bipartite graph property of our interest, and then Lemma 2 originally establishes an equivalence of two problems: searching for complete substructures  $\mathcal{S}' \subset \mathcal{S}$  and searching for specific bipartite subgraphs  $G' \subset G$ .

**Def. 9.** Let  $G(V_1 \cup V_2, E)$  be a bipartite graph. We say that  $G$  is **structural** if, for every  $V'_1 \subseteq V_1$ , there is a connected bipartite subgraph  $G'(V'_1 \cup V'_2, E')$  with  $|V'_1| \leq |V'_2|$ . (Note resemblance with Def. 1).

**Lemma 2.** Let  $\mathcal{S}(\mathcal{V}, \mathcal{E})$  be a complete structure with  $|\mathcal{E}| = |\mathcal{V}| = m$  and  $1 \leq \ell < m$  provide an instance of CSDP. Let also  $G(V_1 \cup V_2, E)$  be a bipartite graph  $G \sim \mathcal{S}$ . Then  $\mathcal{S}$  has a substructure  $\mathcal{S}'$  that gives a yes answer to CSDP if and only if  $G$  has a bipartite subgraph  $G'(V'_1 \cup V'_2, E')$  such that  $G' \sim \mathcal{S}'$  and all of these conditions hold:

- (i) Bipartite subgraph  $G'$  is structural;



(a) COA (2 recursive steps). (b) Bipartite graph  $G$ . (c) Bipartite complement  $G^c$ .

Figure 3: Another example of structure  $\mathcal{S}$  with its correspondent bipartite graph  $G \sim \mathcal{S}$ .

- (ii) For every  $f \in V'_1$ , there is an edge  $(f, x) \in E$  only if  $x \in V'_2$ ;
- (iii) Bipartite subgraph  $G'$  is  $\ell$ -balanced, that is,  $|V'_1| = |V'_2| = \ell$ ;

**Proof 3.** We establish conditions (i-iii) as the bipartite subgraph properties that correspond to a yes answer to CSDP. See Appendix B.  $\square$

We now reach the key property in our argument to show COA's hardness. A *biclique* (or complete bipartite graph) is a bipartite graph  $G = (V_1 \cup V_2, E)$  such that for every two vertices  $u \in V_1, v \in V_2$ , we have  $(u, v) \in E$  [16]. Thus the number of edges in a biclique is  $|E| = |V_1| \cdot |V_2|$ . A biclique with partitions of size  $|V_1| = m$  and  $|V_2| = n$  is denoted  $K_{m,n}$ . For instance, the bipartite graph  $G$  shown in Fig. 2 has a  $K_{2,2}$  biclique, viz.,  $G'(V'_1 \cup V'_2, E')$ , where  $V'_1 = \{f_4, f_5\}$ ,  $V'_2 = \{x_4, x_5\}$  and  $E' = \{(f_4, x_4), (f_4, x_5), (f_5, x_4), (f_5, x_5)\}$ . Let us now consider Example 2.

**Example 2.** We introduce another structure  $\mathcal{S}$ , whose structure matrix is shown in Fig. 3a together with the bipartite graph  $G \sim \mathcal{S}$  in Fig. 3b. Let us consider subgraph  $G'(V'_1 \cup V'_2, E')$  in  $G$  that has  $V'_1 = \{f_1, f_2, f_3\}$  and  $V'_2 = \{x_1, x_2, x_3\}$ . Observe that we have  $G' \sim \mathcal{S}'$ , where  $\mathcal{S}' \subset \mathcal{S}$  is the complete substructure represented by the shaded  $3 \times 3$  matrix in Fig. 3a.

Note also that such bipartite subgraph  $G'$  satisfies the conditions (i-iii) of Lemma 2 and in fact  $\mathcal{S}'$  is a complete substructure in  $\mathcal{S}$ . Clearly,  $G' \sim \mathcal{S}'$  is not a biclique, as it is not the case that  $\deg(w) = 3$  for all  $w \in V'_1 \cup V'_2$ . So there is no obvious connection between identifying complete substructures in a structure and bicliques in a bipartite graph.  $\square$

The key insight to COA’s hardness comes as follows—consider Example 2 and Fig. 3 for illustration. Recall from Lemma 2(ii) that, if we had an edge, say, connecting  $(f_1, x_4) \in E$ , then by Def. 1 the substructure  $\mathcal{S}'(\mathcal{E}', \mathcal{V}')$  with  $\mathcal{E}' = \{f_1, f_2, f_3\}$  would have  $\mathcal{V}' = \bigcup_{f \in \mathcal{E}'} \text{Vars}(f) = \{x_1, x_2, x_3, x_4\}$  instead. That is, it would no more be a complete substructure. In fact, verifying such a negative property (Lemma 2.ii) in structural bipartite graphs translates onto a neat positive property (biclique) in the bipartite complement of bipartite graph  $G$ .

The *bipartite complement* of a bipartite graph  $G(V_1 \cup V_2, E)$  is a bipartite graph  $G^c(V_1 \cup V_2, E^c)$  where an edge  $(u, v) \in E^c$  iff  $(u, v) \notin E$  for every  $u \in V_1$  and  $v \in V_2$ . Given a bipartite graph  $G(V_1 \cup V_2, E)$ , it is easy to see that we can render  $G^c(V_1 \cup V_2, E^c)$  in polynomial time—consider, e.g., the biadjacency matrix of  $G$  (viz., the structure matrix in Fig. 3a), and run a full scan on it to switch the boolean value of each entry in time  $O(|V_1| \cdot |V_2|)$ . Moreover, this operation is clearly invertible, as there is a one-to-one correspondence between  $G$  and  $G^c$ .

Fig. 3c shows the bipartite complement graph  $G^c$  of the bipartite graph  $G$  from Fig. 3b. Note that  $G^c$  has a biclique  $K_{3,1}$  with its vertices shaded in dark grey. To emphasize the point, if we had an edge  $(f_1, x_4) \in E$  (see Fig. 3b), then such a biclique  $K_{3,1}$  would not exist in  $G^c$  (see Fig. 3c). We would have a  $K_{2,1}$  biclique instead with all edges in  $\{f_2, f_3\} \times \{x_4\}$ , but note that  $2 + 1 = 3$  does not sum up to  $|V_1| = |V_2| = m = 4$ .

We can now establish the result we seek. We introduce below the Exact Node Cardinality Decision Problem (ENCD), which is a variant of biclique problem in bipartite graphs that is known to be NP-Complete [17, p. 393]. Theorem 1 establishes its connection with CSDP.

(ENCD). Given a bipartite graph  $G = (V_1 \cup V_2, E)$  and two positive integers  $a, b$ , does  $G$  have a biclique  $K_{a,b}$ ?

**Theorem 1.** *CSDP is NP-Complete.*

**Proof 4.** *We shall construct an instance of ENCD and describe its polynomial-time reduction to an instance of CSDP. We refer to Def. 9 and Lemma 2 and present the argument in detail in Appendix C.  $\square$*

Finally, we formulate an optimization problem associated with CSDP. We refer to it as the Minimal Substructures Problem (MSP). Corollary 1 then finally establishes the hardness of Simon’s COA based on RTCM.

(MSP). Given a complete structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  with  $|\mathcal{E}| = |\mathcal{V}| = m$ , list all its complete substructures  $\mathcal{S}'(\mathcal{E}', \mathcal{V}')$  with  $|\mathcal{E}'| = |\mathcal{V}'| = \ell$  where  $1 \leq \ell < m$  is minimal.

**Corollary 1.** *Let  $\mathcal{S}$  be a complete structure. The extraction of its causal ordering by Simon’s COA( $\mathcal{S}$ ) through its RTCM procedure requires solving MSP, which is NP-Hard.*

**Proof 5.** *Clearly, MSP is the optimization problem that needs to be solved at each recursive step  $k$  of Simon’s RTCM procedure — Algorithm 1, line 3, “find all minimal substructures  $\mathcal{S}' \subseteq \mathcal{S}$ .” But MSP is clearly an optimization problem that subsumes CSDP, which we know from Theorem 1 that is NP-Complete by a reduction from ENCD.*

*In fact, an instance of ENCD’ (as an optimization variant of ENCD) that can be reduced to MSP is as follows: given a bipartite graph  $G(V_1 \cup V_2, E)$  that bears the complement structural property (cf. Theorem 1) and has  $|V_1| = |V_2| = m$ , list all bicliques  $K_{\ell, m-\ell}$  contained in  $G$  where  $1 \leq \ell < m$  is minimal. In worst-case scenario, it requires searching for all bicliques  $K_{\ell, m-\ell}$  for each of the  $m - 1$  possible values of  $\ell$ .*

*ENCD is NP-Complete, therefore ENCD’ is NP-Hard. Accordingly, CSDP is NP-Complete (cf. Theorem 1) therefore MSP is NP-Hard.  $\square$*

COP (Problem 1), nonetheless, can be solved efficiently by means of a different approach due to Nayak [4], which we describe in next section.

#### 4. Nayak’s Efficient Algorithm to COP

The first part of COP requires finding a total causal mapping  $\varphi: \mathcal{E} \rightarrow \mathcal{V}$  over a given complete structure  $\mathcal{S}$ . While Simon’s RTCM goes into an intractable step, inspired by Serrano and Gossard’s work [14] on constraint modeling and reasoning Nayak has found a polynomial-time approach to that task. We cover it next in all of its steps and see their complexity in some detail.

#### 4.1. Total Causal Mappings

Given a structure  $\mathcal{S}$ , there may be more than one total causal mappings over  $\mathcal{S}$  (recall Example 1). So a question that arises is whether the transitive closure  $C_\varphi^+$  is the same for any total causal mapping  $\varphi$  over  $\mathcal{S}$ ; that is, whether the causal ordering of  $\mathcal{S}$  is unique. Proposition 2, from Nayak [4], ensures that is the case.

Before proceeding, we introduce Def. 10 in order to detach the notion of ‘strongly coupled’ variables from ‘minimal structures’ (Def. 5) and connect it to the concept ‘causal dependency’ (Def. 4).

**Def. 10.** *Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a structure with variables  $x_a, x_b \in \mathcal{V}$ , and  $C_\varphi^+$  be the set of causal dependencies induced by total causal mapping  $\varphi$  over  $\mathcal{S}$ . We say that  $x_a$  and  $x_b$  are **strongly coupled** if we have both  $(x_a, x_b), (x_b, x_a) \in C_\varphi^+$ .*

Recall from Example 1 the strongly coupled variables,  $x_4$  and  $x_5$ . Now we can see it only in terms of causal dependencies.

**Proposition 2.** *Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a structure, and  $\varphi_1: \mathcal{E} \rightarrow \mathcal{V}$  and  $\varphi_2: \mathcal{E} \rightarrow \mathcal{V}$  be any two total causal mappings over  $\mathcal{S}$ . Then  $C_{\varphi_1}^+ = C_{\varphi_2}^+$ .*

**Proof 6.** *The proof is based on an argument from Nayak [4], which we present in a bit more of detail (see Appendix D). Intuitively, it shows that if  $\varphi_1$  and  $\varphi_2$  differ in the variable an equation  $f$  is mapped to, then such variables, viz.,  $\varphi_1(f) = x_a$  and  $\varphi_2(f) = x_b$ , must be causally dependent on each other (strongly coupled).  $\square$*

Another issue is concerned with the precise conditions under which total causal mappings exist (i.e., whether or not all variables in the equations can be causally determined). In fact, by Proposition 3, based on Nayak [4] apud. Hall [16, p. 135-7], we know that the existence condition holds if and only if the given structure is complete. We refer to Even [16] to briefly introduce the additional graph-theoretic concepts that are necessary here:

- A *matching* in a graph is a subset of edges such that no two edges in the matching share a common node.
- A matching is said *maximum* if no edge can be added to the matching (without hindering the matching property).

- Finally, a matching in a graph is said ‘perfect’ if every vertex is an end-point of some edge in the matching — in a bipartite graph, a perfect matching is said to be a *complete* matching.

**Proposition 3.** *Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a structure. Then a total causal mapping  $\varphi: \mathcal{E} \rightarrow \mathcal{V}$  over  $\mathcal{S}$  exists if and only if  $\mathcal{S}$  is complete.*

**Proof 7.** *We observe that a total causal mapping  $\varphi: \mathcal{E} \rightarrow \mathcal{V}$  over  $\mathcal{S}$  corresponds exactly to a complete matching  $M$  in a bipartite graph  $B = (V_1 \cup V_2, E)$ , where  $V_1 \mapsto \mathcal{E}$ ,  $V_2 \mapsto \mathcal{V}$ , and  $E \mapsto \mathcal{S}$ . In fact, by *Even apud. Hall’s theorem* (cf. [16, 135-7]), we know that  $B$  has a complete matching iff (a) for every subset of vertices  $F \subseteq V_1$ , we have  $|F| \leq |E(F)|$ , where  $E(F)$  is the set of all vertices connected to the vertices in  $F$  by edges in  $E$ ; and (b)  $|V_1| = |V_2|$ . By Def. 1 (no subset of equations has fewer variables than equations), and Def. 2 (number of equations is the same as number of variables), it is easy to see that conditions (a) and (b) above hold iff  $\mathcal{S}$  is a complete structure.  $\square$*

The problem of finding a maximum matching is a well-studied algorithmic problem. The Hopcroft-Karp algorithm is a classical solution [18], bounded in polynomial time by  $O(\sqrt{|V_1| + |V_2|} |E|)$ . It solves maximum matching in a bipartite graph efficiently as a problem of maximum flow in a network (cf. [16, p. 135-7], or [19, p. 763]). That is, we can handle the problem of finding a total causal mapping  $\varphi$  over a structure  $\mathcal{S}$  (see Alg. 2) by first translating it to the problem of maximum matching in a bipartite graph in time  $O(|\mathcal{S}|)$ . Then we can just apply the Hopcroft-Karp algorithm to get the matching and finally translate it back to the total causal mapping  $\varphi$ . This procedure has been suggested by Nayak in connection with his Proposition 3 and its respective proof [4].

Fig. 4 shows the complete matching found by the Hopcroft-Karp algorithm for the structure given in Example 1.

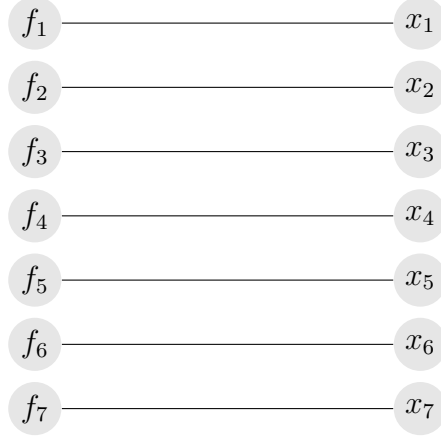


Figure 4: Complete matching  $M$  for structure  $S$  from Example 1.

---

**Algorithm 2** Find a total causal mapping for a given structure.

---

```

1: procedure TCM( $\mathcal{S}$ : structure over  $\mathcal{E}$  and  $\mathcal{V}$ )
Require:  $\mathcal{S}$  given is a complete structure, i.e.,  $|\mathcal{E}| = |\mathcal{V}|$ 
Ensure: Returns a total causal mapping  $\varphi$ 
2:    $B(V_1 \cup V_2, E) \leftarrow \emptyset$ 
3:    $\varphi \leftarrow \emptyset$ 
4:   for all  $\langle f, X \rangle \in \mathcal{S}$  do ▷ translates structure  $\mathcal{S}$  to a bipartite graph  $B$ 
5:      $V_1 \leftarrow V_1 \cup \{f\}$ 
6:     for all  $x \in X$  do
7:        $V_2 \leftarrow V_2 \cup \{x\}$ 
8:        $E \leftarrow E \cup \{(f, x)\}$ 
9:    $M \leftarrow \text{Hopcroft-Karp}(B)$  ▷ solves the maximum matching problem
10:  for all  $(f, x) \in M$  do ▷ translates the matching to a total causal mapping
11:     $\varphi \leftarrow \varphi \cup \{(f, x)\}$ 
12:  return  $\varphi$ 

```

---

Corollary 2 and Remark 1 summarize the results presented in this note.

**Corollary 2.** *Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a complete structure. Then a total causal mapping  $\varphi: \mathcal{E} \rightarrow \mathcal{V}$  over  $\mathcal{S}$  can be found by (Alg. 2) TCM in time that is bounded by  $O(\sqrt{|\mathcal{V}|} \cdot |\mathcal{S}|)$ .*

**Proof 8.** *Let  $B = (V_1 \cup V_2, E)$  be the bipartite graph corresponding to complete structure  $\mathcal{S}$  given to TCM, where  $V_1 \mapsto \mathcal{E}$ ,  $V_2 \mapsto \mathcal{V}$ , and  $E \mapsto \mathcal{S}$ . The*



translation of  $\mathcal{S}$  into  $B$  is done by a scan over it. This scan is of length  $|\mathcal{S}| = |E|$ . Note that number  $|E|$  of edges rendered is precisely the length  $|\mathcal{S}|$  of structure, where the denser the structure, the greater  $|\mathcal{S}|$  is. The re-translation of the matching computed by internal procedure Hopcroft-Karp, in turn, is done at expense of  $|\mathcal{E}| = |\mathcal{V}| \leq |\mathcal{S}|$ . Thus, it is easy to see that TCM is dominated by the maximum matching algorithm Hopcroft-Karp, which is known to be  $O(\sqrt{|V_1| + |V_2|} \cdot |E|)$ , i.e.,  $O(\sqrt{|\mathcal{E}| + |\mathcal{V}|} \cdot |\mathcal{S}|)$ . Since  $\mathcal{S}$  is assumed complete, we have  $|\mathcal{E}| = |\mathcal{V}|$  then  $\sqrt{|\mathcal{V}| + |\mathcal{V}|} = \sqrt{2} \sqrt{|\mathcal{V}|}$ . Therefore, TCM must have running time at most  $O(\sqrt{|\mathcal{V}|} \cdot |\mathcal{S}|)$ .  $\square$

#### 4.2. Computing Transitive Closure

Finally, recall that the set  $C_\varphi$  of direct causal dependencies induced by a total causal mapping  $\varphi$  over a given structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  produces to the so-called ‘causal graph,’ i.e., a directed graph (digraph)  $G(V, E)$  where  $V \mapsto \mathcal{V}$  and  $E \mapsto C_\varphi$ . So, computing set  $C_\varphi^+$  of causal dependencies given  $C_\varphi$  corresponds to computing transitive closure (reachability links) on  $G$ . Note that  $|V| = |\mathcal{V}|$ , and also note that  $|E| = |C_\varphi| = |\mathcal{S}| - |\mathcal{V}| = O(|\mathcal{S}|)$ .

Classical algorithms for such task (e.g., Floyd-Warshall’s) are bounded in time  $O(|\mathcal{V}|^3)$  [19, p. 697]. Another way to do it is by discovering reachability links using either one of the popular graph traversal algorithms, breadth-first search or depth-first search (DFS) [19, p. 603]. Algorithm 3 describes DFS-based transitive closure over digraph  $G(V, E)$ . It runs in time  $O(|V| \cdot |E|)$ , which means  $O(|\mathcal{V}| \cdot |\mathcal{S}|)$  for a complete structure  $\mathcal{S}(\mathcal{E}, \mathcal{V})$ .

---

#### Algorithm 3 DFS-based transitive closure.

---

```

1: procedure TC(  $G(V, E)$ : digraph)           ▷ where  $G$  is such that  $V \mapsto \mathcal{V}$  and  $E \mapsto C_\varphi$ 
2:    $E^+ \leftarrow \emptyset$ 
3:   for all  $v \in V$  do                           ▷ for all vertices  $v$  in digraph  $G$ 
4:      $D \leftarrow \emptyset$                                ▷ initializes  $D$ 
5:     DFS( $G, v, D$ )                                     ▷ discovers into  $D$  all  $u$ , where  $v$  is reachable from  $u$ 
6:      $D \leftarrow D \setminus \{v\}$                        ▷ enforces an irreflexive transitive closure
7:      $E^+ \leftarrow \bigcup_{u \in D} \{(u, v)\} \cup E^+$ 
8:   return  $G^+(V, E^+)$ 

```

---

```

9: procedure DFS( $G$ : digraph,  $v$ : vertex,  $D$ : global set of discovered vertices)
10:   $D \leftarrow D \cup \{v\}$                                ▷ labels  $v$  as discovered
11:  for all  $u$  where  $(u, v) \in G$  do
12:    if  $u \notin D$  then                                   ▷ vertex  $u$  is not yet labeled as discovered
13:      DFS( $G, u, D$ )
14:  return

```

---

**Remark 1.** Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a complete structure. Then we know (cf. Proposition 3) that a total causal mapping over  $\mathcal{S}$  exists. Let it be defined  $\varphi \triangleq \text{TCM}(\mathcal{S})$ , which can be done in  $O(\sqrt{|\mathcal{V}|} \cdot |\mathcal{S}|)$ . Then the causal ordering implicit in  $\mathcal{S}$  can be correctly extracted (cf. Proposition 2) by computing  $C_\varphi^+$ , the set of causal dependencies induced by  $\varphi$ , in terms of graph transitive closure (TC). The latter is bounded in time by  $O(|\mathcal{V}| \cdot |\mathcal{S}|)$ , that is, the complexity of COP is dominated by TC.

In other words, the complete recipe to solve COP consists in replacing Simon’s RTCM by Nayak’s TCM in COA (Algorithm 1). Transitive closure (TC) in turn is computed as described in Algorithm 3.  $\square$

## 5. Conclusions

Causal ordering inference is a classical problem in the AI literature, and still relevant in light of modern applications [3], e.g., large-scale hypothesis management and analytics [8]. In this note we have shown that Simon’s classical algorithm (COA) tries to address an NP-Hard problem; and then we have given a detailed account on the state-of-the-art algorithms for the causal ordering problem (COP, stated as Problem 1). The key points are:

- By Theorem 1 and Corollary 1, we know (an original hardness result) that Simon’s approach to COP requires solving an NP-Hard problem;
- From the seminal work of Simon [1] (cf. §2) and Nayak [4] (cf. §4, and Propositions 2 and 3), an approach is conveyed to solve COP efficiently;
- By Corollary 2, we know how to process a complete structure into a total causal mapping in time that is bounded by  $O(\sqrt{|\mathcal{V}|} \cdot |\mathcal{S}|)$ . This is a core step to solve COP, which Simon’s COA in turn makes intractable.
- By Remark 1, we know how to extract the *causal ordering* of a complete structure in time  $O(|\mathcal{V}| \cdot |\mathcal{S}|)$ , that is, in sub-quadratic time on the structure density (number of variable appearances). The machinery of causal ordering is then suitable for processing very large structures.

## Acknowledgments

We thank three anonymous reviewers for their careful reading and sharp criticism on a previous version of this manuscript. This work has been

supported by the Brazilian funding agencies CNPq (grants n<sup>o</sup> 141838/2011-6, 309494/2012-5) and FAPERJ (grants INCT-MACC E-26/170.030/2008, ‘Nota 10’ E-26/100.286/2013). We thank IBM for a Ph.D. Fellowship award.

## References

- [1] H. Simon, Causal ordering and identifiability, In Hood & Koopmans (eds.), *Studies in Econometric Methods*, Chapter 3, John Wiley & Sons, 1953.
- [2] B. Goncalves, F. Porto,  $\Upsilon$ -DB: Managing scientific hypotheses as uncertain data, *PVLDB* 7 (11) (2014) 959–62.
- [3] D. Dash, M. J. Druzdzel, A note on the correctness of the causal ordering algorithm, *Artificial Intelligence* 172 (15) (2008) 1800–8.
- [4] P. P. Nayak, Causal approximations, *Artificial Intelligence* 70 (1-2) (1994) 277–334.
- [5] Y. Iwasaki, H. A. Simon, Causality and model abstraction, *Artificial Intelligence* 67 (1) (1994) 143–194.
- [6] J. Pearl, *Causality: Models, Reasoning, and Inference*, Cambridge Univ. Press, 2000.
- [7] P. P. Nayak, *Automated modelling of physical systems*, Springer-Verlag, 1996.
- [8] B. Gonçalves, Managing scientific hypotheses as data with support for predictive analytics, *IEEE Computing in Science & Eng.* 17 (5) (2015) 35–43.
- [9] P. Haas, P. Maglio, P. Selinger, W. Tan, Data is dead... without what-if models, *PVLDB* 4 (12) (2011) 1486–9.
- [10] P. J. Hunter, T. K. Borg, Integration from proteins to organs: the Physioome Project., *Nat. Rev. Mol. Cell. Biol.* 4 (3) (2003) 237–43.
- [11] M. Hines, T. Morse, M. Migliore, N. Carnevale, G. Shepherd, ModelDB: A database to support computational neuroscience, *J. Comput. Neurosci.* 17 (1) (2004) 7–11.

- [12] V. Chelliah, C. Laibe, N. Le Novère, BioModels Database: A repository of mathematical models of biological processes, *Method. Mol. Biol.* (1021) (2013) 189–99.
- [13] B. Gonçalves, Managing large-scale scientific hypotheses as uncertain and probabilistic data, Ph.D. thesis, National Laboratory for Scientific Computing (LNCC), available at CoRR abs/1501.05290, Brazil (2015).
- [14] D. Serrano, D. C. Gossard, Constraint management in conceptual design, in: *Knowledge Based Expert Systems in Engineering: Planning and Design*, Computational Mechanics Publications, 1987, pp. 211–24.
- [15] J. Bondy, U. Murty, *Graph theory with applications*, North-Holland Publishing Co., 1976.
- [16] S. Even, *Graph algorithms*, 2nd Edition, Cambridge Univ. Press, 2011.
- [17] M. Dawande, P. Keskinocak, J. M. Swaminathan, S. Tayur, On bipartite and multipartite clique problems, *J. Algorithms* 41 (2) (2001) 388–403.
- [18] J. E. Hopcroft, R. M. Karp, An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs, *SIAM Journal on Computing* 2 (4) (1973) 225–31.
- [19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, 3rd Edition, The MIT Press, 2009.

## Appendix A. Proof of Proposition 1

*Let  $\mathcal{S}$  be a complete structure. If  $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{S}$  are any different minimal substructures of  $\mathcal{S}$ , then they are disjoint.*

**Proof 9.** *We show the statement by case analysis and then contradiction out of Defs. 1–5. By assumption both  $\mathcal{S}_1, \mathcal{S}_2$  are minimal (hence complete). Let their size be  $|\mathcal{V}_1| = |\mathcal{E}_1| = \ell$  and  $|\mathcal{V}_2| = |\mathcal{E}_2| = m$ . Let also  $\ell \leq m$ . The argument is analogous otherwise but it shall be convenient to keep a placeholder for the size of the smaller substructure (with no loss of generality).*

*By Def. 5 (minimal structures), we know that  $\mathcal{S}_1 \not\subseteq \mathcal{S}_2$  and  $\mathcal{S}_2 \not\subseteq \mathcal{S}_1$ . Now suppose  $\mathcal{S}_1, \mathcal{S}_2$  are not disjoint. Then by Def. 6 there must be at least one shared variable  $x \in \mathcal{V}_1, \mathcal{V}_2$ , and thus we must have  $|\mathcal{V}_1 \cup \mathcal{V}_2| \leq \ell + m - 1$ .*

*We can then proceed through case analysis by inquiring how many equations are shared by  $\mathcal{S}_1, \mathcal{S}_2$ . Since  $\mathcal{S}_1$  is minimal with  $|\mathcal{E}_1| = |\mathcal{V}_1| = \ell$  for*

$1 \leq \ell \leq m$ , the number of equations that are shared with  $\mathcal{S}_2$  could be any  $0 \leq k < \ell$ . (Note that the case of  $k = \ell$  shared equations would lead to the more obvious contradiction that  $\mathcal{S}_1 \subseteq \mathcal{S}_2$ , even though  $\mathcal{S}_2$  is minimal).

Let us start with the case  $\mathcal{E}_1 \cap \mathcal{E}_2 = \emptyset$  to illustrate the rationale in its simplest form. In this case, no equations are shared yet at least one variable is. Then we have  $|\mathcal{E}_1 \cup \mathcal{E}_2| = \ell + m$ , but  $|\mathcal{V}_1 \cup \mathcal{V}_2| \leq \ell + m - 1$ . Since we have both  $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{S}$ , in fact we have their sets of equations  $\mathcal{E}_1, \mathcal{E}_2 \subset \mathcal{E}$  as well and then  $\mathcal{E}_1 \cup \mathcal{E}_2 \subseteq \mathcal{E}$ . Now, by Def. 1 (valid structure), we know that in any subset of  $k > 0$  equations of  $\mathcal{S}$ , at least  $k$  different variables must appear. But rather we have  $|\mathcal{E}_1 \cup \mathcal{E}_2| = \ell + m$  and yet  $|\mathcal{V}_1 \cup \mathcal{V}_2| \leq \ell + m - 1$ . That is, we reach a contradiction to Def. 1, viz.,  $|\mathcal{E}_1 \cup \mathcal{E}_2| > |\mathcal{V}_1 \cup \mathcal{V}_2|$ .  $\zeta$ .

The next case is when one equation is shared ( $|\mathcal{E}_1 \cap \mathcal{E}_2| = 1$ ). Note that, if we had  $|\mathcal{E}_1| = |\mathcal{V}_1| = \ell = 1$  in particular then the only equation  $f \in \mathcal{E}_1$  would have  $|\text{Vars}(f)| = 1$  and be shared with  $\mathcal{E}_2$ , making  $\mathcal{S}_1 \subseteq \mathcal{S}_2$  even though  $\mathcal{S}_2$  is assumed minimal.  $\zeta$ . We rather know that  $|\mathcal{E}_1| = \ell \geq 2$ . Also, note that we must have  $|\text{Vars}(f)| \geq 2$  for all  $f \in \mathcal{E}_1$ , otherwise there would be some  $g \in \mathcal{E}_1$  with  $|\text{Vars}(g)| = 1$  even though  $|\mathcal{E}_1| \geq 2$ . That is, we would have a minimal substructure within  $\mathcal{S}_1$ , although it is minimal.

So, since one equation is shared and for all  $f \in \mathcal{E}_1$  we have  $|\text{Vars}(f)| \geq 2$ , then at least two variables must be shared. Observe then that  $|\mathcal{E}_1 \cup \mathcal{E}_2| = \ell + m - 1$  (since exactly one equation is shared) and  $|\mathcal{V}_1 \cup \mathcal{V}_2| \leq \ell + m - 2$  (at least two variables are shared). Again, we see the same contradiction in face of Def. 1, viz.,  $|\mathcal{E}_1 \cup \mathcal{E}_2| > |\mathcal{V}_1 \cup \mathcal{V}_2|$ .  $\zeta$ .

Now we complete the case analysis by making the argument abstract for any number of shared equations,  $0 \leq k < \ell$  (an inductive step, n.b., is not required because  $k \in \mathbb{N}$  is bounded). Note that, for any such number  $0 \leq k < \ell$ , we must have at least  $k + 1$  shared variables, otherwise the shared substructure having  $k$  equations, formed out of  $\mathcal{E}_1 \cap \mathcal{E}_2$ , would be minimal as well even though  $\mathcal{E}_1 \cap \mathcal{E}_2 \subseteq \mathcal{E}_1, \mathcal{E}_2$  (that is, rendering both  $\mathcal{S}_1, \mathcal{S}_2$  non-minimal.  $\zeta$ ). However, once more we see that this contradicts Def. 1.  $\zeta$ .  $\square$

## Appendix B. Proof of Lemma 2

Let  $\mathcal{S}(\mathcal{V}, \mathcal{E})$  be a complete structure with  $|\mathcal{E}| = |\mathcal{V}| = m$  and  $1 \leq \ell < m$  provide an instance of CSDP. Let also  $G(V_1 \cup V_2, E)$  be a bipartite graph  $G \sim \mathcal{S}$ . Then  $\mathcal{S}$  has a substructure  $\mathcal{S}'$  that gives a yes answer to CSDP if and only if  $G$  has a bipartite subgraph  $G'(V'_1 \cup V'_2, E')$  such that  $G' \sim \mathcal{S}'$  and all of these conditions hold:

- (i) Bipartite subgraph  $G'$  is structural;
- (ii) For every  $f \in V_1'$ , there is an edge  $(f, x) \in E$  only if  $x \in V_2'$ ;
- (iii) Bipartite subgraph  $G'$  is  $\ell$ -balanced, that is,  $|V_1'| = |V_2'| = \ell$ ;

**Proof 10.** First, we consider the ‘if’ statement—that is, all conditions (i-iii) together are sufficient. Let  $G' \subset G$  be a bipartite subgraph  $G'(V_1' \cup V_2', E')$  that satisfies all conditions (i-iii), and  $\mathcal{S}'(\mathcal{E}', \mathcal{V}')$  be a substructure of  $\mathcal{S}$  with  $G' \sim \mathcal{S}'$ . We shall see that such  $\mathcal{S}'$  does give a yes answer to CSDP, that is, it is a complete substructure with  $|\mathcal{E}'| = |\mathcal{V}'| = \ell$ .

From condition (i) we know that  $G'$  is structural (Def. 9). That is, for every  $V_1'' \subseteq V_1'$ , there is a connected bipartite subgraph  $G''(V_1'' \cup V_2'', E'')$  with  $|V_1''| \leq |V_2''|$ . Since  $V_1' \mapsto \mathcal{E}'$ ,  $V_2' \mapsto \mathcal{V}'$  and  $E' \mapsto \mathcal{S}'$ , such property bears obvious resemblance with Def. 1 (structure). That is, the ‘connected’ bipartite subgraph aspect implies that, for any subset of  $|\mathcal{E}''|$  equations in  $\mathcal{E}'$ , at least  $|\mathcal{V}''| \geq |\mathcal{E}''|$  variables appear and each such variable  $x \in \mathcal{V}''$  appears in some  $f \in \mathcal{E}''$ , otherwise  $x \in V_2''$  would be disconnected in  $G''(V_1'' \cup V_2'', E'')$ .

Condition (ii) ensures in addition that  $\bigcup_{f \in \mathcal{E}'}, \text{Vars}(f) = \mathcal{V}'$ . That is, the variables in  $\mathcal{V}'$  are exhaustive w.r.t.  $\mathcal{E}'$ . Thus, structure  $\mathcal{S}'$  satisfies Def. 1. Finally, condition (iii) ensures that  $\mathcal{S}'$  is complete with  $|\mathcal{E}'| = |\mathcal{V}'| = \ell$ .

We consider now the ‘only if’ statement—i.e., every condition (i-iii) is necessary. We assume a bipartite graph  $G' \sim \mathcal{S}'$  and show that lacking any one such condition implies that  $\mathcal{S}'$  cannot be complete or cannot be a structure at all. First, it is easy to see that when condition (iii) does not hold for  $G'$  then a structure  $\mathcal{S}'$  with  $G' \sim \mathcal{S}'$  cannot be complete.

Now suppose condition (ii) does not for  $G'$ . That is, there is some  $f \in V_1'$  that has incidence with some  $x \in V_2 \setminus V_2'$ . Thus we have  $V_1' \mapsto \mathcal{E}'$  and  $V_2' \mapsto \mathcal{V}'$  but  $\bigcup_{f \in \mathcal{E}'}, \text{Vars}(f) \neq \mathcal{V}'$ . Therefore either  $\mathcal{S}'$  does not satisfy Def. 1 or we cannot actually have  $G' \sim \mathcal{S}'$ .  $\zeta$ .

Finally, consider that  $G'$  is not structural (Def. 9). That is, there is some  $V_1'' \subseteq V_1'$  such that no connected bipartite subgraph  $G''(V_1'' \cup V_2'', E'')$  exists in  $G'$  with  $|V_1''| \leq |V_2''|$ . Considering  $G' \sim \mathcal{S}'$ , that would imply for  $\mathcal{S}'(\mathcal{E}', \mathcal{V}')$  either an equation  $f \in \mathcal{E}'$  with no variables (a disconnected vertex  $x \in V_1'$ ), or a redundant subset of equations—number of equations is larger than number of variables appearing in it. Either conditions violate Def. 1, so  $\mathcal{S}'$  cannot be a structure even though  $G' \sim \mathcal{S}'$ .  $\zeta$ .  $\square$

## Appendix C. Proof of Theorem 1

*CSDP is NP-Complete.*

**Proof 11.** *We shall construct an instance of ENCD and describe its polynomial-time reduction to an instance of CSDP by using Lemma 2.*

**Constructing an instance of ENCD.** *Let  $G(V_1 \cup V_2, E)$  be a bipartite graph such that, for every  $V'_1 \subseteq V_1$ , there is a bipartite subgraph  $G'(V'_1 \cup V'_2, E')$  with  $|V'_1| \leq |V'_2|$  and  $\deg(f) < |V'_2|$  for all  $f \in V'_1$ . Note that this is the complement property of the structural bipartite graph property (see Def. 9). It is meant to ensure that the bipartite complement graph  $G^c(V_1 \cup V_2, E^c)$  of  $G$  is structural—satisfies Def. 9. That is, when we produce  $G^c$ , we know that it can possibly correspond to a structure  $\mathcal{S}$  such that  $G^c \sim \mathcal{S}$ . Let also  $G$  have  $|V_1| = |V_2| = m$  in order to ensure that such structure  $\mathcal{S}$  will be complete as well—recall that  $\mathcal{S}$  given in CSDP is assumed complete indeed.*

*Now let  $G$  and an integer  $1 \leq \ell < m$  provide an instance of ENCD for integers  $a = \ell$  and  $b = m - \ell$ . That is, our problem is to decide whether  $G$  has a biclique  $K_{\ell, m-\ell}$ . Imposing both of the above properties on  $G$ , n.b., incurs in no loss of generality w.r.t. ENCD as it does not open a pruning opportunity in searching for a biclique  $K_{\ell, m-\ell}$  in powerset  $\mathcal{P}(V_1 \times V_2)$ . Such a biclique  $K_{\ell, m-\ell}$ , if existing in  $G$ , can be denoted  $C(V'_1 \cup V_2^*, K)$ , where  $|V'_1| = \ell$  and  $|V_2^*| = m - \ell$ , and  $K$  is a complete set of edges between  $V'_1$  and  $V_2^*$ . Note also that  $V'_1 \subset V_1$  and  $V_2^* \subset V_2$ .*

**Production of an instance of CSDP from the ENCD one.** *Let  $G^c(V_1 \cup V_2, E^c)$  be the bipartite complement graph of  $G$ , where an edge  $(f, x) \in E^c$  if and only if  $(f, x) \notin E$  for  $f \in V_1$  and  $x \in V_2$ . Clearly, bipartite graph  $G^c$  can be produced in polynomial time from  $G$ —as mentioned in §3.2, consider the ‘structure matrix’ (biadjacency matrix) of  $G$  and run a full scan on it to switch the boolean value of each entry in time  $O(|V_1| \cdot |V_2|)$  and then get  $G^c$ .*

**Decision problem correspondence.** *Now we show that a biclique  $K_{\ell, m-\ell}$  in  $G$ , if existing, corresponds to a bipartite subgraph  $G^{c'}(V'_1 \cup V'_2, E^{c'})$  in  $G^c$  that satisfies the conditions (i-iii) of Lemma 2. That is, we show that a yes answer to ENCD implies a yes answer to CSDP.*

*In fact, as  $G^c$  is the bipartite complement graph of  $G$ , then the biclique  $C(V'_1 \cup V_2^*, K)$  in  $G$  becomes a bipartite subgraph  $C^c(V'_1 \cup V_2^*, \emptyset)$  in  $G^c$ . Now let  $G^{c'}(V'_1 \cup V'_2, E^{c'})$  be such that  $V'_2 = V_2 \setminus V_2^*$ . We observe that:*

- (i) *The presence of biclique  $C(V'_1 \cup V_2^*, K)$  in  $G$  indicates that  $V_2^*$  could not have contributed to satisfy the complement structural property for*

$V'_1$ , only  $V'_2 = V_2 \setminus V_2^*$  could. But such property turns into the structural property in  $G^c$ , thus  $G^{c'}(V'_1 \cup V'_2, E^{c'})$  must be structural indeed. That is, condition (i) of Lemma 2 is ensured.

(ii) By the fact that we have  $C^c(V'_1 \cup V_2^*, \emptyset)$  in  $G^c$  we know that, for all  $f \in V'_1$ , there can only be an edge  $(f, x) \in E^c$  if  $x \in V'_2$  indeed. That is, condition (ii) of Lemma 2 is ensured.

(iii) The presence of biclique  $C(V'_1 \cup V_2^*, K)$  of form  $K_{\ell, m-\ell}$  in  $G$  implies that  $V'_1$  has size  $|V'_1| = \ell$ . Besides,  $V'_2$  will have size  $|V'_2| = |V_2| - |V_2^*| = m - (m - \ell) = \ell$ . That is, we must have  $|V'_1| = |V'_2| = \ell$  and then condition (iii) of Lemma 2 is ensured as well.

We have then established that the existence of a biclique  $C \subset G$  of form  $K_{\ell, m-\ell}$  implies the existence of a bipartite subgraph  $G^{c'} \subset G^c$ , where  $G^{c'}$  satisfies the conditions (i-iii) of Lemma 2. That is, we get a yes answer to CSDP if we find one to ENCD. It remains to show the ‘only if’ part of the correspondence.

In fact, suppose no biclique  $C(V'_1 \cup V_2^*, K)$  of form  $K_{\ell, m-\ell}$  exists in  $G(V_1 \cup V_2, E)$ . Clearly, it means that for any  $V'_1 \subset V_1$  where  $|V'_1| = \ell$ , there is at least one  $f \in V'_1$  such that an edge  $(f, x)$  with  $x \in V_2^*$  is missing from  $E$ . Accordingly, in  $G^c(V_1 \cup V_2, E^c)$ , we cannot have  $G^{c'} \subset G^c$  with condition (ii) of Lemma 2 satisfied.

ENCD is NP-Complete. Thus CSDP must be NP-Complete as well.  $\square$

## Appendix D. Proof of Proposition 2

Let  $\mathcal{S}(\mathcal{E}, \mathcal{V})$  be a structure, and  $\varphi_1: \mathcal{E} \rightarrow \mathcal{V}$  and  $\varphi_2: \mathcal{E} \rightarrow \mathcal{V}$  be any two total causal mappings over  $\mathcal{S}$ . Then  $C_1^+ = C_2^+$ .

**Proof 12.** The proof is based on an argument from Nayak [4], which we reproduce here in a bit more of detail. Intuitively, it shows that if  $\varphi_1$  and  $\varphi_2$  differ in the variable an equation  $f$  is mapped to, then such variables, viz.,  $\varphi_1(f)$  and  $\varphi_2(f)$ , must be causally dependent on each other (strongly coupled).

To show  $C_1^+ = C_2^+$  reduces to show both  $C_1^+ \subseteq C_2^+$  and  $C_2^+ \subseteq C_1^+$ . We show the first containment, and the second is understood as following by symmetry. Closure operators are extensive,  $X \subseteq cl(X)$ , and idempotent,



$cl(cl(X)) = cl(X)$ . That is, if we have  $C_1 \subseteq C_2^+$ , then we shall have  $C_1^+ \subseteq (C_2^+)^+$  and, by idempotence,  $C_1^+ \subseteq C_2^+$ .

Then it suffices to show that  $C_1 \subseteq C_2^+$ , i.e., for any  $(x', x) \in C_1$ , we must show that  $(x', x) \in C_2^+$  as well. Observe by Def. 3 that both  $\varphi_1$  and  $\varphi_2$  are bijections, then, invertible functions. If  $\varphi_1^{-1}(x) = \varphi_2^{-1}(x)$ , then we have  $(x', x) \in C_2$  and thus, trivially,  $(x', x) \in C_2^+$ . Else,  $\varphi_1$  and  $\varphi_2$  disagree in which equations they map onto  $x$ . But we show next, in any case, that we shall have  $(x', x) \in C_2^+$ .

Take all equations  $g \in \mathcal{E}' \subseteq \mathcal{E}$  such that  $\varphi_1(g) \neq \varphi_2(g)$ , and let  $n \leq |\mathcal{E}'|$  be the number of such 'disagreed' equations. Now, let  $f \in \mathcal{E}'$  be such that its mapped variable is  $x = \varphi_1(f)$ . Construct a sequence of length  $2n$  such that,  $s_0 = \varphi_1(f) = x$  and, for  $1 \leq i \leq 2n$ , element  $s_i$  is defined  $s_i = \varphi_2(\varphi_1^{-1}(s_{i-1}))$ . That is, we are defining the sequence such that, for each equation  $g \in \mathcal{E}'$ , its disagreed mappings  $\varphi_1(g) = x_a$  and  $\varphi_2(g) = x_b$  are such that  $\varphi_1(g)$  is immediately followed by  $\varphi_2(g)$ . As  $x_a, x_b \in \text{Vars}(g)$ , we have  $(x_a, x_b) \in C_2$  and, symmetrically,  $(x_b, x_a) \in C_1$ . The sequence is of form  $s = \langle \underbrace{x, x_f, \dots}_{f}, \underbrace{x_a, x_b, \dots}_{g}, \underbrace{x_{2n-1}, x_{2n}}_{h} \rangle$ .

Since  $x$  must be in the codomain of  $\varphi_2$ , we must have a repetition of  $x$  at some point  $2 \leq k \leq 2n$  in the sequence index, with  $s_k = x$  and  $s_{k-1} = x''$  such that  $(x'', x) \in C_2$ . If  $x'' = x'$ , then  $(x', x) \in C_2$  and obviously  $(x', x) \in C_2^+$ . Else, note that  $x_f$  must also be in the codomain of  $\varphi_1$ , while  $x''$  in the codomain of  $\varphi_2$ . Let  $\ell$  be the point in the sequence,  $3 \leq \ell \leq 2n-1$ , at which  $s_\ell = x_f = x_a$  and  $s_{\ell+1} = x_b$  for some  $x_b$  such that  $(x_f, x_b) \in C_2$ . It is easy to see that, either we have  $x_b = x''$  or  $x_b \neq x''$  but  $(x_b, x'') \in C_2^+$ . Thus, by transitivity on such a causal chain, we must have  $(x_f, x'') \in C_2^+$  and eventually  $(x_f, x) \in C_2^+$ . Finally, since  $x' \in \text{Vars}(f)$  and  $\varphi_2(f) = x_f$ , we have  $(x', x_f) \in C_2$  and, by transitivity,  $(x', x) \in C_2^+$ .  $\square$