

Introducing Causal Hypergraphs

Michael Baumgartner and Christoph Falk

Abstract

Increased efforts have lately been made to generate visualizations of causal models produced by *configurational comparative methods* such as Qualitative Comparative Analysis (QCA) and Coincidence Analysis (CNA). This paper proposes to visualize these models as causal hypergraphs. Unlike edges of directed acyclic graphs (DAGs), edges of causal hypergraphs can connect more than two nodes and thereby represent causal complexity, which is the main search target of QCA and CNA. The paper defines the notion of a causal hypergraph, introduces the graphical elements of causal hypergraphs, and demonstrates their usefulness. Moreover, it presents the R package **causalHyperGraph**, which automates the drawing of causal hypergraphs.

1 Introduction

The effects of causes can be highly context-sensitive: the use of social media may improve the mental health of some adolescents while impairing that of others (cf. [Valkenburg et al., 2022](#)); a social policy such as the microfinance model of providing small loans to the poor may have favorable effects in one context (e.g. Bangladesh around 1980) but unfavorable ones in another (e.g. Andhra Pradesh around 2010; see [Roy, 2010](#)). Whether a factor X causes one effect or another, or nothing at all, depends not only on the relation between X and its effects but also on the complex configuration of other causes with which X interacts—a phenomenon commonly referred to as *causal complexity*. Standard methods of causal learning, which focus on estimating the net effects of single factors, struggle to account for complexity-induced context-sensitivity of causes. Learning causal complexity from data calls for customized methods.

Configurational comparative methods (CCMs) are designed for this very purpose. CCMs work by grouping multiple causes into conjunctive bundles that become operative only when all of their components are co-instantiated, and by placing these bundles on alternative (disjunctive) causal paths that can bring about corresponding outcomes independently of one

another. The two main members of the CCM family are *Qualitative Comparative Analysis* (QCA; Rihoux & Ragin, 2009; Schneider & Wagemann, 2012) and *Coincidence Analysis* (CNA; Baumgartner & Ambühl, 2020; Whitaker et al., 2020). While QCA has been widely used in the social and political sciences, business administration, and management, CNA has seen a significant uptick in applications in public health in recent years.¹

The models produced by CCMs represent causal complexity by means of complex Boolean functions, which can be challenging to interpret causally for non-specialists. Accordingly, various efforts have been made to cast these models (or parts thereof) into more easily interpretable forms (for an overview, see Robinson, 2019). Cronqvist (2007) proposes to use Carroll diagrams (Carroll, 1958), Fiss (2011) advocates configuration (or “Fiss”) charts for that purpose, and Duşa (2019) recommends the use of Venn diagrams. While all of these visualizations present (parts of) the content of CCM models in a less formal and, thus, more easily accessible manner, none of them entirely closes the interpretative gap between the formal model and the causal structure it represents. Getting from Carroll or Venn diagrams or from configuration charts to a full understanding of the causal structure is still a non-trivial interpretative task.

Most recently, Thiem et al. (2023) followed the lead of Pearl (2009, p. 415) in adapting logic gate diagrams for the purpose of representing causal structures. Not only are logic gates an established tool in digital electronics, computer science, and systems biology, where they are used to express the logical structure of digital circuits or gene regulatory networks, but they can also be given a causal interpretation, thereby narrowing the interpretative gap between CCM models and causation. Their main disadvantage is that they employ various graphical symbols, the mastery of which is only marginally less demanding than that of the Boolean syntax of CCM models. Moreover, although logic gates admit a causal interpretation, that reading is non-standard, and causal structures are customarily visualized in a different graphical syntax.

The standard graphical tools, in most disciplines, for representing causal structures are directed acyclic graphs (DAGs; Greenland et al., 1999; Pearl, 2009; Piccininni et al., 2020; Spirtes et al., 2000). DAGs have a very simple syntax consisting of two elements only: nodes denoting variables of the modeled system and directed edges, which are ordered pairs of variables, standing for direct causal relations. Their causal interpretation is straightforward. However, as edges in DAGs can connect at most two nodes, DAGs cannot effectively

¹Recently, Mkrtchyan et al. (2024) added CORA to the fold of CCMs; but, to our knowledge, no empirical applications of this method have yet been published.

represent the causal complexity that CCMs are designed to uncover.²

In this paper, we introduce an alternative to logic gate diagrams and DAGs, namely *causal hypergraphs*. In contrast to edges in DAGs, edges—or arcs—in hypergraphs can connect more than two nodes. This feature makes them suited for expressing causal complexity. Furthermore, unlike DAGs and combinational logic circuits, hypergraphs can be cyclic, which aligns well with the fact that CCM models (contrary to causal Bayesian networks, for example) may feature causal cycles. Overall, we contend that visualizing CCM models by causal hypergraphs bridges the interpretative gap between these models and causation while striking a balance between simplicity and expressiveness. For that reason, the drawing of causal hypergraphs was automated in the R package `causalHyperGraph`.

The paper is structured as follows. First, we provide some theoretical background about CCMs. Then, we define the notion of a causal hypergraph, introduce the hypergraph notation, and demonstrate its utility. Finally, we illustrate how to draw causal hypergraphs in R and document the corresponding code in an appendix.

2 Background

Causal structures featuring causal complexity—conjunctural causation (Rothman & Greenland, 2005) and equifinality (Gerring, 2005)—tend to pose challenges for standard methods of causal learning utilizing regression techniques or Bayesian networks. One reason is that these structures often violate additivity and involve causes and effects that are not dependent in the data (Culverhouse et al., 2002). Such independence is incompatible with causal faithfulness (Spirtes et al., 2000), a common assumption of standard methods, according to which some form of marginal (i.e., unconditional) dependence between X and Y is necessary for X to cause Y. By contrast, CCMs trace causation as defined by the *(M)INUS theory* of causation, which does not entail that marginal dependence is necessary for causation (cf. Baumgartner & Falk, 2023; Mackie, 1974).³

The MINUS theory defines the relation of causal relevance (i.e., type-level causation) between a factor A taking some value α and a factor B taking a value β in terms of $A=\alpha$ being a *Boolean difference-maker* of $B=\beta$. This means, roughly, that $A=\alpha$ is part of a complex

²To represent complexity, DAGs are typically accompanied by (structural) equations, which makes their causal interpretation considerably more challenging.

³Originally, “INUS” was an acronym referring to *Insufficient but Non-redundant parts of Unnecessary but Sufficient conditions* (Mackie, 1974, p. 62). Today it serves as a mere name for a theoretical framework, and *MINUS* likewise names, without an expansion, a modification of the original INUS account Beirlaen et al. (2016).

sufficient and necessary condition of $B=\beta$ that does not contain any redundant elements. Factors can either be *crisp-set* (binary), taking two possible values 0 and 1, *fuzzy-set*, taking real values from the unit interval $[0, 1]$, or *multi-value*, taking an arbitrary but finite number of non-negative integers as possible values. As their labels suggest, crisp- and fuzzy-set factors are typically given a set-theoretic interpretation, meaning that their values are understood as membership scores in corresponding sets. Following standard CCM notation, we write “ A ” for membership in the corresponding set and “ a ” for non-membership. That is, italicization carries meaning: “ A ” designates the *factor* while “ A ” and “ a ” stand for factor *values*. By contrast, the notation is not abbreviated for multi-value factors; we write $A=3$ and $A=0$ for A taking the values 3 and 0, respectively.

The MINUS theory borrows much of its formal machinery from Boolean algebra, in particular the operations of *negation*, $\neg A$ (expressing “NOT $A=1$ ”), *conjunction*, $A*B$ (“ $A=1$ AND $B=1$ ”), *disjunction*, $A + B$ (“ $A=1$ OR $B=1$ ”), *implication*, $A \rightarrow B$ (“IF $A=1$, THEN $B=1$ ”), and *equivalence* $A \leftrightarrow B$ (“ $A=1$ IF, AND ONLY IF, $B=1$ ”).⁴ In case of crisp-set and multi-value factors, Boolean operations are given a rendering in classical logic, in case of fuzzy-set factors, they are rendered in fuzzy logic (for details see Baumgartner & Ambühl, 2020, p. 528). The relations of *sufficiency* and *necessity* are defined based on the implication operator. A Boolean expression, for example the conjunction $A*C$, is sufficient for B iff (i.e., if, and only if) $A*C \rightarrow B$, that is, whenever A AND C hold, so does B —or equivalently, it is not the case that A AND C are true and B false. Conversely, an expression, say, $A*C + a*c + D$ is necessary for B iff $B \rightarrow A*C + a*c + D$, meaning that whenever B holds, so does $A*C$ or $a*c$ or D —or equivalently, B does not obtain without any of $A*C$ OR $a*c$ OR D .

Most sufficiency and necessity relations do not reflect causation, but those that are minimal (i.e., redundancy-free) in their component factors and values do (Baumgartner & Falk, 2023; Zhang & Zhang, 2025). $A*C$ is a *minimally sufficient* condition of B iff $A*C \rightarrow B$ holds and no proper part of $A*C$ is sufficient for B , that is, neither $A \rightarrow B$ nor $C \rightarrow B$ hold. $A*C + a*c + D$ is a *minimally necessary* condition of B iff $B \rightarrow A*C + a*c + D$ holds and no proper part is necessary, meaning that no elimination of at least one disjunct from $A*C + a*c + D$ results in a necessary condition of B . Correspondingly, CCMs infer minimally necessary disjunctions of minimally sufficient conjunctions of an investigated outcome from data and output them as *atomic MINUS-formulas*:

$$A*C + a*c + D \leftrightarrow B \quad (1)$$

⁴The symbols “ $*$ ” and “ $+$ ” are used as in Boolean algebra here, which means, in particular, that they do not represent the linear algebraic (arithmetic) operations of multiplication and addition (notational variants of Boolean “ $*$ ” and “ $+$ ” are “ $\wedge$ ” and “ $\vee$ ”). For a standard introduction to Boolean algebra see (Bowran, 1965).

The minimality of MINUS-formulas guarantees that each factor value X_i on the left-hand side of “ $\leftrightarrow$ ” makes a difference to the right-hand side, because without X_i the left-hand side would not satisfy sufficiency or necessity and, thus, would not account for all variation in the factor on the right-hand side. Accordingly, MINUS-formulas have a straightforward causal interpretation: conjunctions represent complex causes (interactions) and disjunctions stand for alternative causes (equifinality). Hence, (1) entails that A and C jointly cause B on one path, a and c on another, and D on a third path. To illustrate, suppose that A , C , and D represent switch positions, *up* and *down*, and B a lamp being on. (1) then asserts that the lamp is caused to be on when A and C are both *up* or both *down*, or D is *up*.

Atomic MINUS-formulas represent causal structures with one outcome only. But outcomes can themselves be causes of other outcomes further downstream. CNA is the only CCM designed to capture this additional complexity dimension. Unlike QCA, CNA does not only output atomic MINUS-formulas but, if warranted by the evidence in the data, also builds *complex MINUS-formulas*, which are redundancy-free conjunctions of atomic MINUS-formulas. Expressions (2) and (3) are examples.

$$(H * K + I \leftrightarrow A) * (A * b + c * D \leftrightarrow E) \quad (2)$$

$$(H=2 * K=1 + I=3 \leftrightarrow A=1) * (A=1 * B=0 + C=0 * D=2 \leftrightarrow E=2) \quad (3)$$

The complex MINUS-formula (3), for instance, expresses a causal chain in which the combination of factor H taking the value 2 and K taking the value 1 is causally relevant for A taking the value 1, which in turn, together with $B=0$, is causally relevant for $E=2$.

Inferring Boolean models like these from real-life data is an intricate task. The number of possible Boolean functions involving even a handful of factors is enormous, and the computational demand on algorithms to navigate in that vast search space grows exponentially with the number of factors in the analysis and as a power of the number of values each factor can take. Moreover, as data are typically fragmented and noisy, protocols for fitting the models to the data and assessing their robustness are needed (Parkkinen & Baumgartner, 2021). The details of these data analytics and model building problems are not relevant for the purposes of this paper. What matters, however, is that the aim of CCMs is to find all causal structures compatible with and equally warranted by the data. That is, they do not just search for (any) *one* model that strikes an optimal balance between data fit, robustness, and complexity but for *all* such models; and there are often multiple. In consequence, contrary to typical regression methods, which return one optimal model only, CCMs regularly output sets of MINUS-formulas that are indistinguishable based on the available evidence, meaning

that the data underdetermine their own causal modeling. In such cases, CCMs cannot specify which of those formulas truthfully represents the causal structure generating the data. All that can be said is that at least one of them is true. It is then up to the analyst to draw on theoretical background knowledge external to the data to select among the candidates. A proper understanding of the different causal implications of the available model candidates is key for this task. But gaining such an understanding directly from MINUS-formulas is challenging, especially for ordinary users of CCMs.

3 Causal Hypergraphs

The higher the complexity of the MINUS-formulas, the more difficult it is to discern all causal paths expressed by the formulas and the harder it is to fully decode their causal implications. To remedy this problem and facilitate the causal interpretation of MINUS-formulas, we now introduce causal hypergraphs. Causal hypergraphs are related to directed acyclic graphs (DAGs; [Greenland et al., 1999](#); [Pearl, 2009](#); [Spirtes et al., 2000](#)), which are the most widely used tools for visualizing causal structures. But while edges in DAGs connect exactly two nodes, merely indicating the direction of causation, edges in hypergraphs can connect more than two nodes and, thereby, represent more than just the direction of causation. Causal hypergraphs can merge nodes into bundles and then connect these bundles to other nodes. This allows for representing conjunctive and disjunctive groupings of causes and, accordingly, for capturing the causal complexity encoded in MINUS-formulas. Furthermore, while DAGs are assumed not to contain cycles, causal hypergraphs may include cycles to accommodate the fact that MINUS-formulas may express causal feedback structures.

Introducing the notion of a causal hypergraph requires a number of more fundamental terms, which we define by drawing on [Gallo et al. \(1993\)](#) and [Bretto \(2013\)](#). A hypergraph is a pair $(\mathbf{F}, \mathbf{E})$, where $\mathbf{F} = \{F_1, F_2, \dots, F_n\}$ is a set of nodes and $\mathbf{E} = \{E_1, E_2, \dots, E_m\}$ is a set of non-empty subsets of $\mathbf{F}$. Each $E_i \in \mathbf{E}$ is called a *hyperedge*. The *cardinality* of a hyperedge is the number of nodes it contains. A *hyperarc*—or *directed hyperedge* or *arc*, for short—is an ordered pair $\langle \mathbf{F}_1, \mathbf{F}_2 \rangle$ of disjoint subsets of $\mathbf{F}$.⁵ The subset $\mathbf{F}_1$ is called the *tail* of the hyperarc, $\mathbf{F}_2$ is its *head*. A *directed hypergraph* is a hypergraph whose hyperedges are all hyperarcs. A *backward hyperarc* is a hyperarc with a head of cardinality 1, i.e. with a head consisting of exactly one node, and a tail of cardinality ≥ 1 . A *path* connecting two nodes F_i

⁵Note that restricting the sets of nodes connected by a hyperarc to *disjoint* sets excludes self-loops, which provides the basis for a causal interpretation of hyperarcs, as there is no direct self-causation.

and F_j in a hypergraph is a node-hyperedge alternative sequence:

$$F_i = F_1, \mathbf{E}_1, F_2, \mathbf{E}_2, \dots, F_s, \mathbf{E}_s, F_{s+1} = F_j,$$

such that each hyperedge $\mathbf{E}_k$ contains its two neighboring nodes in the sequence, that is, $F_k, F_{k+1} \in \mathbf{E}_k$, for all $k = 1, 2, \dots, s$. A hypergraph is called *connected* iff for any pair of nodes, there is a path that connects these nodes. As a path is defined in terms of hyperedges (irrespective of possible directedness), a directed hypergraph counts as connected if it contains a sequence of arcs between any pair of nodes, such that the elements of the pair can be reached from one another, whether by moving with the arcs or in the opposite direction. Importantly, a connected hypergraph represents one cohering structure, that is, a structure without isolated parts.

Now, we can define the notion of a causal hypergraph or CHG.

Causal Hypergraph (CHG) A causal hypergraph is a connected directed hypergraph $(\mathbf{F}, \mathbf{E})$ whose nodes $\mathbf{F}$ represent factor values and whose hyperedges $\mathbf{E}$ are backward hyperarcs representing the relation of direct causal relevance.

CHGs express cohering structures where an open number of factor values are directly causally relevant for single factor values. To avoid misunderstandings, note the following difference between DAGs and CHGs. While edges in DAGs and backward hyperarcs in CHGs both stand for the relation of direct causal relevance, nodes in DAGs symbolize factors or variables such as A and B, but the nodes in CHGs symbolize factor values such as A or b . Hence, DAGs represent causal relationships between factors or variables, whereas CHGs represent causal relationships between factor values.

We distinguish between two types of CHGs: *set-CHGs* for causal structures involving values of crisp-set or fuzzy-set factors and *mv-CHGs* for causal structures involving values of multi-value factors. Besides nodes and arcs, set-CHGs contain two further graphical elements whose sole purpose is to facilitate their causal interpretation (meaning they are not features of directed hypergraphs): “•” for bundling nodes in a conjunction and “◊” at tails of arcs for negating factor values. Mv-CHGs also symbolize conjunction through “•”, but instead of a negation sign, they carry numeric values at their tail and head ends, indicating which factor values the arc connects. In both set- and mv-CHGs, arcs with the same head value form a disjunction. The operations that these graphical elements can express are summarized in Figure 1.

Figure 2 shows more complex structures built from them. The set-CHG corresponding to the MINUS-formula (2) is shown in Figure 2a and the mv-CHG of (3) is in 2b. Both example

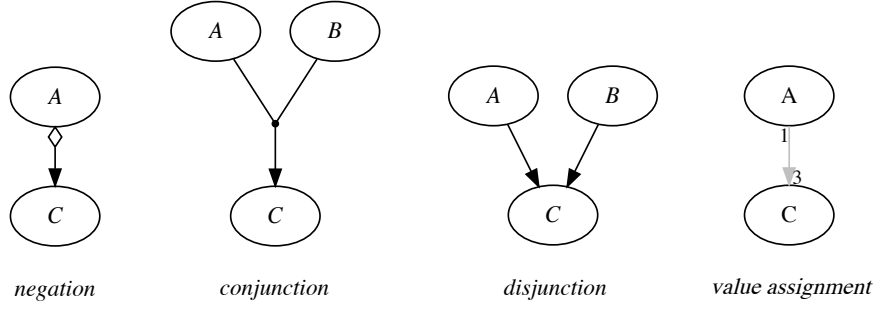


Figure 1: CHG building blocks and the operations they express.

graphs contain values of A and E at heads of arcs, which are the outcomes of the depicted causal structures. Each outcome node is the head of two arcs, indicating that each outcome can be caused along two alternative (disjunctive) causal paths. Both CHGs have one node, A and A=1, respectively, that is a head of two arcs and contained in a tail of another. This signifies that both graphs represent a causal chain with the value 1 of A as middle link.

Two further characteristics of CHGs require separate discussion. First, MINUS-formulas may contain different values of the same factor, yet, for reasons of graphical parsimony and transparency, we stipulate that CHGs must *not* contain multiple nodes corresponding to different values of the same factor. That is, each factor in a CHG has a unique node representing its values. Causal relations involving different values of a factor are expressed by attaching negation signs to this unique node in set-CHGs and numeric values in mv-

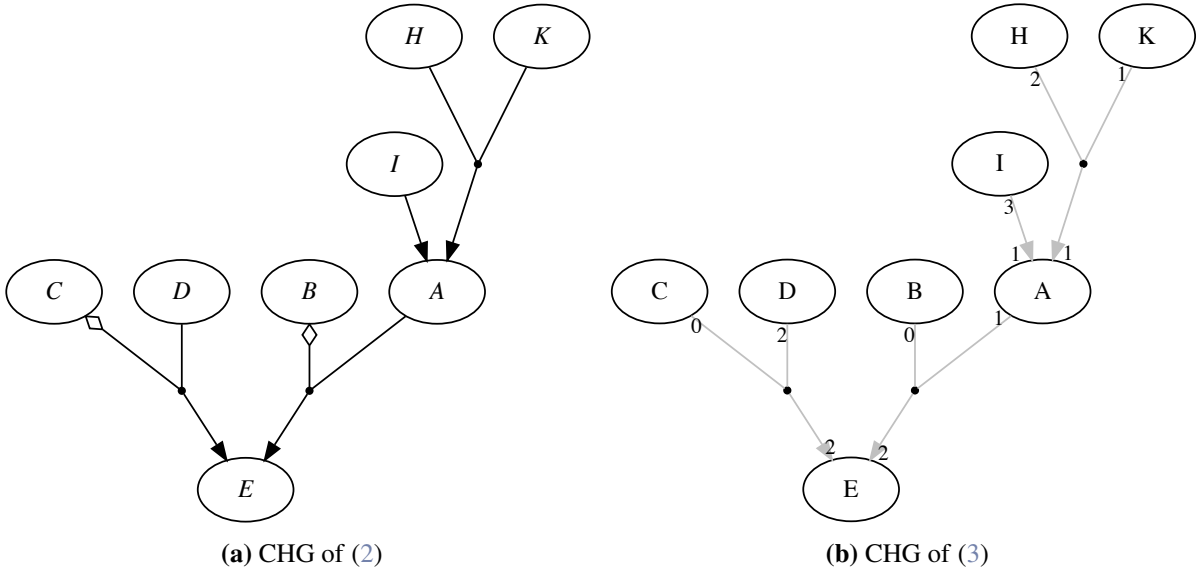


Figure 2: The set-CHG and the mv-CHG corresponding to MINUS-formulas (2) and (3), respectively.

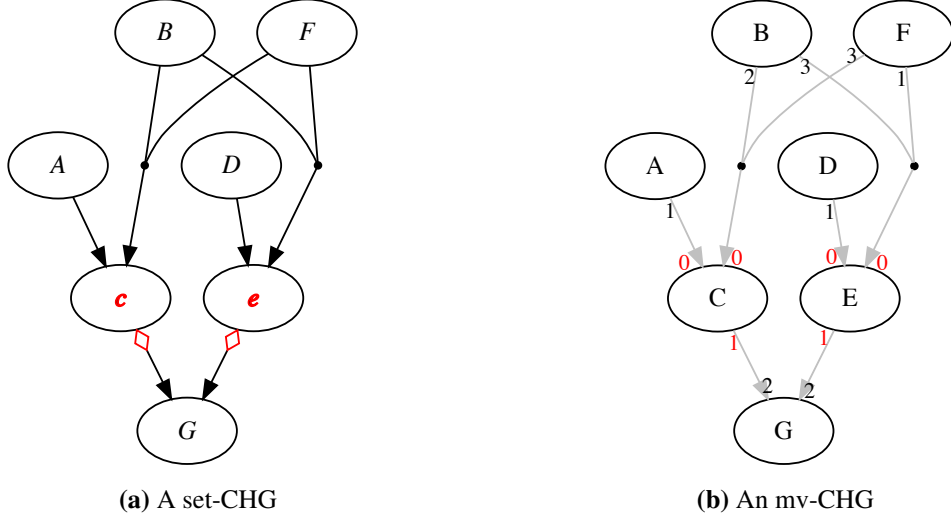


Figure 3: Two examples illustrating the node uniqueness requirement: A set-CHG on the left, (a), and an mv-CHG on the right, (b). Red colorings have mere highlighting purpose.

CHGs. To illustrate, take the MINUS-formulas (4) and (5).

$$(A + B * F \leftrightarrow c) * (D + B * F \leftrightarrow e) * (C + E \leftrightarrow G) \quad (4)$$

$$(A=1 + B=2 * F=3 \leftrightarrow C=0) * (D=1 + B=3 * F=1 \leftrightarrow E=0) * (C=1 + E=1 \leftrightarrow G=2) \quad (5)$$

Formula (4) features different values of the factors C and E as outcomes and causes. The values c and e appear as outcomes in the first two atomic MINUS-formulas of (4), whereas C and E are causally relevant for G in the third. Analogously, $C=0$ and $E=0$ are outcomes in the first two atomic MINUS-formulas of (5), while $C=1$ and $E=1$ appear as causes further downstream. The CHGs corresponding to these two formulas are depicted in Figure 3. The CHG in 3a captures the difference between the factor values figuring as causes and outcomes by the negations at the tails of the arcs connecting c and e to G . As c and e are negations of C and E (i.e. $\neg C$ and $\neg E$), adding further negations results in double negations $\neg\neg C$ and $\neg\neg E$, which cancel out and, thus, reduce to C and E . That is, doing justice to the node uniqueness requirement may call for slight syntactic adaptations of the represented MINUS-formulas. That is not needed to graphically render formula (5). In the CHG of Figure 3b, the arcs comprising $C=0$ and $E=0$ as heads are simply marked with the value 0, while the arcs with $C=1$ and $E=1$ as tails are marked with the value 1.

The second important characteristic of CHGs is that they may contain cycles, just as MINUS-formulas may contain causal cycles. The MINUS theory of causation allows for two types of cyclic structures: *self-reinforcing cycles* and *value-transition loops*. In a self-

reinforcing cycle, a factor value is causally relevant for itself, via a sequence of intermediate links. Such a cycle reinforces itself unless it is interrupted from the outside. Concrete examples are prevalent in many fields. For instance, as a company grows and becomes more profitable, it has more resources to invest in new technologies, expansion, and human capital, which, in turn, leads to further growth, and on to further investments, etc. Or, soil erosion exposes more soil, which causes increased water runoff, producing more erosion, and so on.

By contrast, value-transition loops are cycles in which a factor value—via intermediate links—is causally relevant for another value of the same factor. A cycle of this latter type comes to a halt after one iteration, unless it is re-initiated from the outside. Typical examples include negative-feedback regulation that maintains an equilibrium. For instance, a high blood glucose level triggers insulin release from the pancreas, which promotes glucose uptake by cells, which in turn decreases the glucose level in the blood. Another example is a staged progression: being a student ($P=1$) is causally relevant for obtaining a degree, which is causally relevant for professional advancement, which in turn is causally relevant for being a manager ($P=2$).

To illustrate formally, consider the MINUS-formulas (6) and (7).

$$(A + B * f \leftrightarrow C) * (C * e + H \leftrightarrow G) * (G + R \leftrightarrow A) \quad (6)$$

$$(A=1 + B=1 * F=0 \leftrightarrow C=2) * (C=2 * E=3 + H=1 \leftrightarrow G=1) * (G=1 + R=2 \leftrightarrow A=2) \quad (7)$$

According to (6), A is causally relevant for C , which is relevant for G , which, in turn, is relevant for A . This cycle reinforces itself unless it is interrupted, say, by setting the factor E to the value 1, which suppresses $C * e$ and (assuming H is absent) prohibits G . By contrast, the cycle in (7) is a value-transition loop, as $A=1$ is causally relevant for $C=2$, which is relevant for $G=1$. However, $G=1$ does not lead back to $A=1$ but to $A=2$. Unless the value of A is reset to 1 (by some cause of $A=1$ not contained in (7)), the cycle stops.

Both types of cycles can be represented by CHGs. To render a self-reinforcing cycle, a CHG must contain a path $\langle F_1, \mathbf{E}_1, F_2, \mathbf{E}_2, \dots, F_s, \mathbf{E}_s, F_{s+1} \rangle$ such that F_1 and F_{s+1} represent the same factor value. The self-reinforcing cycle expressed by (6) is depicted in the CHG of Figure 4a, which comprises the path $\langle A, \langle \{A\}, \{C\} \rangle, C, \langle \{C, e\}, \{G\} \rangle, G, \langle \{G\}, \{A\} \rangle, A \rangle$, among others. To represent a value-transition loop, a CHG must contain a path such that F_1 and F_{s+1} represent different values of the same factor. The value-transition loop in (7) is depicted in Figure 4b, which comprises, among others, the following path from $A=1$ to $A=2$: $\langle A=1, \langle \{A=1\}, \{C=2\} \rangle, C=2, \langle \{C=2, E=3\}, \{G=1\} \rangle, G=1, \langle \{G=1\}, \{A=2\} \rangle, A=2 \rangle$.

In sum, a CHG carries all of the information contained in the Boolean syntax of the

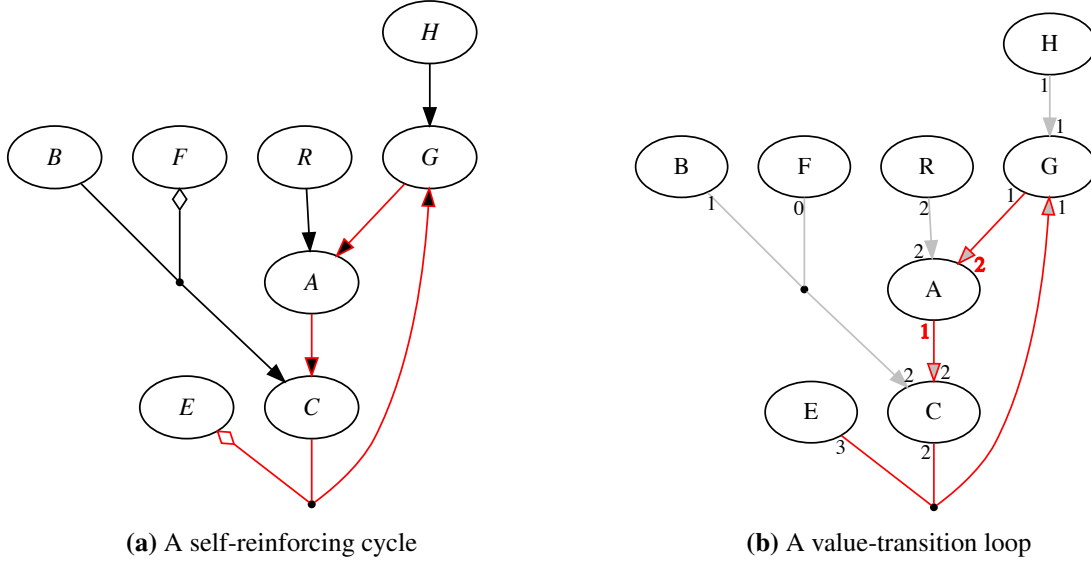


Figure 4: Two types of cycles. Red colorings have mere highlighting purpose.

MINUS-formula it renders: conjunction, disjunction, negation, value assignment, causal chains, and cyclic substructures each receive a graphical expression. A CHG may thus stand for itself, as a self-contained representation of a causal model. In research communities where Boolean notation is not customary, CHGs allow researchers to communicate configurational findings entirely in graphical form.

4 Drawing Causal Hypergraphs in R

The primary software environment in which configurational comparative methods are implemented is R. Functions for QCA are provided in the **QCA** package (Dusa, 2026), and CNA is available in the **cna** package (Ambuehl & Baumgartner, 2025). Accordingly, R is the natural choice for automating the drawing of causal hypergraphs. Based on **DiagrammeR**'s versatile graph functions (Iannone & Roy, 2026), the drawing of causal hypergraphs is implemented in the R package **causalHyperGraph**. This section illustrates the usage of its main functions.

The package has three main components:

1. a standalone function `causalHyperGraph(x)`, or `chg(x)` for short;
2. plotting methods for objects output by various functions of the **cna** package;
3. a function `export_as_svg(x, file)` that exports causal hypergraphs as SVG files for editing with vector graphics software.

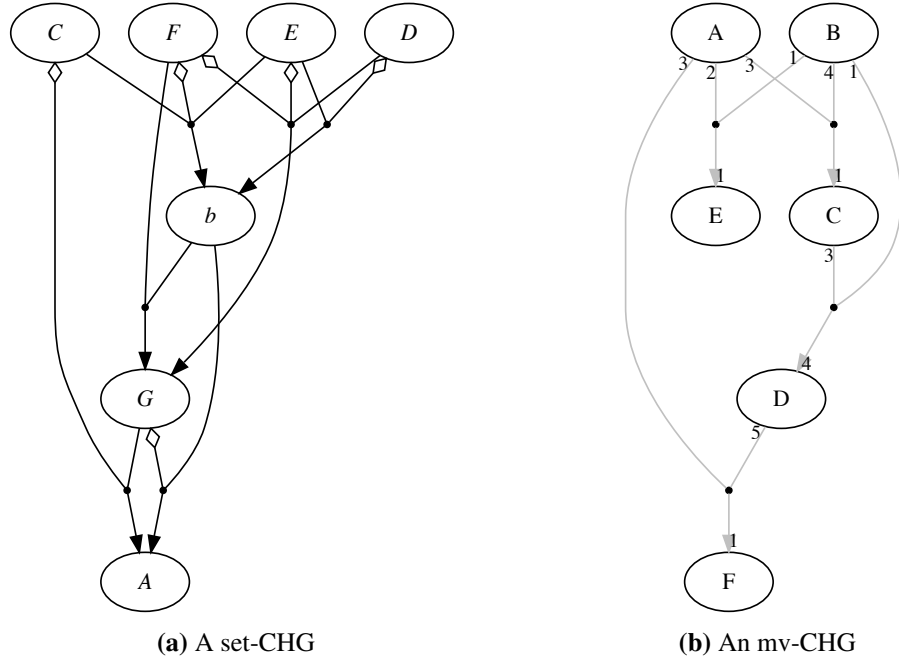


Figure 5: The hypergraphs rendering `csFormula` and `mvFormula`.

The `causalHyperGraph(x)` function takes a character vector `x` as input that contains atomic or complex MINUS-formulas in “upper/lower case” notation or “factor=value” notation. It returns the causal hypergraphs corresponding to the elements of `x` as `htmlwidgets`. The following code chunk draws the two graphs in Figure 2:

```
> library(causalHyperGraph)
> x <- c("(H*K + I <-> A)*(A*b + c*D <-> E)",
        "(H=2*K=1 + I=3 <-> A=1)*(A=1*B=0 + C=0*D=2 <-> E=2)")
> causalHyperGraph(x)
```

Any MINUS-formula with Boolean or multi-value factors can be plotted with `chg()`. The following code plots a crisp-set formula, `csFormula`, and a multi-value formula, `mvFormula`, as a set-CHG and an mv-CHG, respectively. The reader is encouraged to modify these formulas to draw other causal structures. The resulting graphs are in Figure 5.

```
> csFormula <- "(d*E + E*C*f <-> b)*(f*e*D + b*F <-> G)*(c*G + b*g <-> A)"
> mvFormula <- "(B=4*A=3 <-> C=1)*(B=1*C=3 <-> D=4)*(A=3*D=5 <-> F=1)*
        (A=2*B=1 <-> E=1)"
> chg(c(csFormula, mvFormula))
```

When comparing the formal expressions of `csFormula` and `mvFormula` to the hypergraphs in Figure 5a and 5b, the gain in transparency and intuitive accessibility afforded by

the graphs is palpable. It is immediately visible from the graphs that both formulas represent causal chains with two intermediate links; it is evident which factors are the starting (root) nodes of the chains, the intermediate links, and the end nodes. Distilling all that information from the Boolean syntax of `csFormula` and `mvFormula` requires a trained eye and careful inspection. That is, causal hypergraphs contribute substantively to closing the interpretative gap between MINUS-formulas and causation.

The **causalHyperGraph** package is tailored for use in combination with the **cna** package, which takes data as input and outputs all equally fitting MINUS-formulas, termed *atomic* and *complex solution formulas* (*asf* and *csf*) in **cna**. Correspondingly, **causalHyperGraph** provides plotting methods for output objects of core functions such as `cna()`, `csf()` and `condition()`. To illustrate, we analyze the data `d.jobsecurity`, which is taken from Emmenegger’s (2011) study on the determinants of high job security regulations in Western democracies and which is contained in the **cna** package. It is a fuzzy-set dataset with factor JSR (job security regulations with 1 = ”high” and 0 = ”not high”) as outcome and 6 exogenous factors measuring S (statism), C (non-market coordination), L (labor movement strength), R (Catholicism), P (religious party strength), and V (institutional veto points) in Western democracies. Here is the head of the data:

```
> library(cna)
> head(d.jobsecurity)
```

	S	C	L	R	P	V	JSR
AU	0.00	0.00	0.57	0.2	0.0	1.00	0.14
AT	0.67	1.00	0.57	1.0	0.8	0.67	0.71
BE	1.00	0.67	0.43	1.0	1.0	0.67	0.57
CA	0.00	0.00	0.14	0.2	0.0	1.00	0.14
DK	0.00	0.67	0.86	0.4	0.4	0.00	0.29
FI	0.67	1.00	0.71	0.4	0.4	0.00	0.43

The first line in the following code snippet analyzes `d.jobsecurity` with CNA’s fit thresholds of consistency and coverage both set to 0.85 while identifying *JSR* as outcome. The subsequent line then plots the solution object `ana`.

```
> ana <- cna(d.jobsecurity, outcome = "JSR", con = .85, cov = .85)
> plot(ana)
```

This analysis yields 18 models (*asf*) that fit the data equally well but vary in complexity. When increased model complexity does not significantly improve fit, as seen in this analysis, it is preferable to choose the least complex models to avoid overfitting. Figure 6 displays

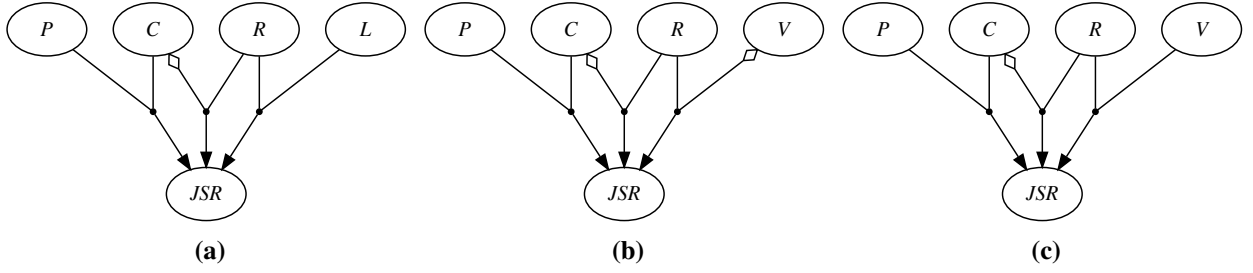


Figure 6: Three of the least complex models from the CNA solution object `ana`.

the output of `plot(ana)` corresponding to three of the least complex models. These examples demonstrate the effectiveness of CHGs in highlighting the differences and similarities among various model candidates. The graphs in Figure 6 show that the model candidates for this analysis have two conjunctive causes on two paths in common, while they differ slightly in the third path. This level of causal transparency facilitates bringing to bear theoretical background knowledge about the underlying data-generating structure. While the Boolean form of the models in the solution object `ana` does not explicitly convey its causal interpretation, the hypergraph form clearly does.

One viable approach to resolving model ambiguities of this sort is to draw on available knowledge about individual cases in the data. The analyst may know that particular causal processes cannot be operative in certain cases. Hence, if a candidate model entails that causal processes are operative in cases where background knowledge determines they cannot be, these models can be discarded (see [Baumgartner and Ambuehl \(2025\)](#)). The function `condition(x, data)` takes a character vector `x` of models (*asf* or *csf*) as first input and `data` as second, and it evaluates the models on the level of individual cases in the `data`. This reveals which paths of a model are operative in which cases. To further assist in model selection and ambiguity resolution, the **causalHyperGraph** package also provides a plotting method for objects output by `condition()`. Case knowledge can only select among model candidates if the causal implications of these candidates are clear, and making the causal implications of MINUS-formulas explicit is what causal hypergraphs do.

```
> condition(csf(ana)$condition, d.jobsecurity) |> plot()
```

Finally, to facilitate the manual editing of automatically generated hypergraphs the function `export_as_svg(x, file)` exports CHGs as SVG files. Here, `x` is a vector of CHGs to be exported, and `file` defines the path and file names to be used. For example, the following command exports all 18 models in the solution object `ana` from the above analysis of `d.jobsecurity` as scalable vector graphics to the working directory, naming them

“ana.svg”, “ana1.svg”, “ana2.svg”, etc.

```
> export_as_svg(plot(ana), "ana")
```

Scalable vector graphics can be manually edited in software such as Inkscape, Adobe Illustrator, Apache Batik, or Boxy. Moreover, SVG files can easily be converted to PDF or other formats.

5 Summary

Configurational comparative methods (CCMs) such as CNA and QCA are used in public health, the social and political sciences, business administration, and management to learn causal complexity and context-sensitivity from data. The causal models produced by CCMs have the form of complex Boolean functions. Although the underlying MINUS theory of causation offers clear rules for causally interpreting such functions, it is challenging—even for experts—to extract all causal implications from them. This paper introduced causal hypergraphs (CHGs) to bridge the gap between the Boolean form of CCM models (i.e. MINUS-formulas) and their causal interpretation. CHGs add just the right amount of expressiveness to the familiar DAG notation that is required to represent the causal complexity encoded in CCM models. A CHG carries all of the information contained in the Boolean syntax of the MINUS-formula it renders, which makes it a self-contained model representation. As a result, researchers can choose to communicate their findings entirely in graphical form. Moreover, the paper illustrated the drawing of CHGs using the R package **causalHyperGraph**.

By making models expressing causal complexity more intuitive and accessible, CHGs help researchers to better understand their results and to communicate them beyond the CCM community, which shows what configurational methods are capable of and may thereby contribute to their broader adoption across the sciences. This, in turn, promises to increase awareness of causal complexity as a fundamental characteristic of the empirical world, ultimately yielding a more nuanced and accurate understanding of the systems we study.

References

Ambuehl, M., & Baumgartner, M. (2025). cna: Causal modeling with Coincidence Analysis [Computer software manual]. <https://CRAN.R-project.org/package=cna> (R package version 4.0.3)

- Baumgartner, M., & Ambuehl, M. (2025). **cna**: An R package for configurational causal inference and modeling. *Vignette included in R package cna, version 4.0.3*. <https://CRAN.R-Project.org/package=cna>
- Baumgartner, M., & Ambühl, M. (2020). Causal modeling with multi-value and fuzzy-set Coincidence Analysis. *Political Science Research and Methods*, 8, 526-542. (doi: 10.1017/psrm.2018.45)
- Baumgartner, M., & Falk, C. (2023). Boolean difference-making: A modern regularity theory of causation. *The British Journal for the Philosophy of Science*, 74(1), 171-197. <http://dx.doi.org/10.1093/bjps/axz047>
- Beirlaen, M., Leuridan, B., & Van De Putte, F. (2016). A logic for the discovery of deterministic causal regularities. *Synthese*, 195(1), 367-399. doi: 10.1007/s11229-016-1222-x
- Bowran, A. P. (1965). *A Boolean algebra. Abstract and concrete*. London: Macmillan.
- Bretto, A. (2013). *Hypergraph theory: An introduction*. Springer International Publishing.
- Carroll, L. (1958). *Symbolic logic and the game of logic*. New York: Dover Publications.
- Cronqvist, L. (2007). *TOSMANA: Tool for small-n analysis* (Tech. Rep.). (Available at: <https://www.tosmana.net>)
- Culverhouse, R., Suarez, B. K., Lin, J., & Reich, T. (2002). A perspective on epistasis: Limits of models displaying no main effect. *The American Journal of Human Genetics*, 70(2), 461-471. <http://dx.doi.org/10.1086/338759>
- Duša, A. (2019). *QCA with R*. Springer International Publishing. <http://dx.doi.org/10.1007/978-3-319-75668-4>
- Dusa, A. (2026). *QCA: Qualitative Comparative Analysis* [Computer software manual]. <https://CRAN.R-project.org/package=QCA> (R package version 3.25)
- Emmenegger, P. (2011). Job security regulations in western democracies: A fuzzy set analysis. *European Journal of Political Research*, 50(3), 336-364. <http://dx.doi.org/10.1111/j.1475-6765.2010.01933.x>
- Fiss, P. C. (2011). Building better causal theories: A fuzzy set approach to typologies in organization research. *Academy of Management Journal*, 54(2), 393-420. doi: 10.5465/amj.2011.60263120
- Gallo, G., Longo, G., Pallottino, S., & Nguyen, S. (1993). Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2), 177-201. <https://www.sciencedirect.com/science/article/pii/0166218X9390045P>
- Gerring, J. (2005). Causation: A unified framework for the social sciences. *Journal of Theoretical Politics*, 17(2), 163-198. <http://dx.doi.org/10.1177/0951629805050859>

- Greenland, S., Pear, J., & Robins, J. M. (1999). Causal diagrams for epidemiologic research. *Epidemiology*, 10(1), 37-48.
- Iannone, R., & Roy, O. (2026). Diagrammer: Graph/network visualization [Computer software manual]. <https://CRAN.R-project.org/package=DiagrammeR> (R package version 1.0.12)
- Mackie, J. L. (1974). *The cement of the universe. A study of causation*. Oxford: Clarendon Press.
- Mkrtchyan, L., Thiem, A., & Sebechlebská, Z. (2024). CORA: An open-source software tool for combinational regularity analysis. *Social Science Computer Review*, 43(4), 850-871. <http://dx.doi.org/10.1177/08944393241275640> doi: 10.1177/08944393241275640
- Parkkinen, V.-P., & Baumgartner, M. (2021). Robustness and model selection in configurational causal modeling. *Sociological Methods & Research*, 52(1), 176-208. <http://dx.doi.org/10.1177/0049124120986200>
- Pearl, J. (2009). *Causality* (Second ed.). Cambridge: Cambridge Univ. Press.
- Piccininni, M., Konigorski, S., Rohmann, J. L., & Kurth, T. (2020). Directed acyclic graphs and causal thinking in clinical risk prediction modeling. *BMC Medical Research Methodology*, 20(1). <http://dx.doi.org/10.1186/s12874-020-01058-z>
- Rihoux, B., & Ragin, C. C. (Eds.). (2009). *Configurational comparative methods. Qualitative Comparative Analysis (QCA) and related techniques*. Thousand Oaks: Sage.
- Rothman, K. J., & Greenland, S. (2005). Causation and causal inference in epidemiology. *American Journal of Public Health*, 95(S1), S144-S150. <http://dx.doi.org/10.2105/AJPH.2004.059204>
- Roy, A. (2010). *Poverty capital. Microfinance and the making of development*. New York: Routledge.
- Rubinson, C. (2019). Presenting Qualitative Comparative Analysis: Notation, tabular layout, and visualization. *Methodological Innovations*, 12(2), 205979911986211. <http://dx.doi.org/10.1177/2059799119862110>
- Schneider, C. Q., & Wagemann, C. (2012). *Set-theoretic methods for the social sciences: A guide to Qualitative Comparative Analysis*. Cambridge University Press.
- Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, prediction, and search* (2nd ed.). Cambridge: MIT Press.
- Thiem, A., Sebechlebská, Z., & Mkrtchyan, L. (2023). Logic diagrams: A visual tool with untapped potential. *Nature Reviews Methods Primers*, 3(1). <http://dx.doi.org/10.1038/s43586-023-00210-5>

- Valkenburg, P. M., Meier, A., & Beyens, I. (2022). Social media use and its impact on adolescent mental health: An umbrella review of the evidence. *Current Opinion in Psychology*, 44, 58-68. (<https://doi.org/10.1016/j.copsyc.2021.08.017>) doi: 10.1016/j.copsyc.2021.08.017
- Whitaker, R. G., Sperber, N., Baumgartner, M., Thiem, A., Cragun, D., Damschroder, L., ... Birken, S. (2020). Coincidence Analysis: A new method for causal inference in implementation science. *Implementation Science*, 15, 1–10.
- Zhang, J., & Zhang, K. (2025). A new minimality condition for Boolean accounts of causal regularities. *Erkenntnis*, 90(1), 67–86. (<https://doi.org/10.1007/s10670-023-00685-4>)

A R code for drawing causal hypergraphs

The commands below reproduce the figures of this paper and illustrate some plotting methods not shown in the main text. They presuppose only the two packages **cna** and **causalHyperGraph**, loaded in the first code chunk. The commands were run under **cna** 4.0.3 and **causalHyperGraph** 0.1.0. The commands for Figures 2, 5, and 6 are also shown in the main text; those for Figures 1, 3, and 4 appear only here.

```
> library(cna)
> library(causalHyperGraph)
```

Figure 1, the graphical building blocks (negation, conjunction, disjunction, and value assignment):

```
> chg(c("a <-> C", "A*B <-> C", "A + B <-> C", "A=1 <-> C=3"))
```

Figure 2, formula (2) as a set-CHG and formula (3) as an mv-CHG:

```
> x <- c("(H*K + I <-> A)*(A*b + c*D <-> E)",
        "(H=2*K=1 + I=3 <-> A=1)*(A=1*B=0 + C=0*D=2 <-> E=2)")
> chg(x)
```

Figure 3, formulas (4) and (5), including the negations that the node uniqueness requirement makes necessary:

```
> uniqueNodes <- c("(A + B*F <-> c)*(D + B*F <-> e)*(C + E <-> G)",
                  "(A=1 + B=2*F=3 <-> C=0)*(D=1 + B=3*F=1 <-> E=0)*",
                  "(C=1 + E=1 <-> G=2)")
> chg(uniqueNodes)
```

Figure 4, formulas (6) and (7), a self-reinforcing cycle and a value-transition loop:

```
> cycles <- c("(A + B*f <-> C)*(C*e + H <-> G)*(G + R <-> A)",
              "(A=1 + B=1*f=0 <-> C=2)*(C=2*e=3 + H=1 <-> G=1)*
              (G=1 + R=2 <-> A=2)")
> chg(cycles)
```

Figure 5, the two formulas whose hypergraphs the figure shows:

```
> csFormula <- "(d*e + e*c*f <-> b)*(f*e*d + b*f <-> G)*(c*g + b*g <-> A)"
> mvFormula <- "(B=4*A=3 <-> C=1)*(B=1*C=3 <-> D=4)*(A=3*D=5 <-> F=1)*
              (A=2*B=1 <-> E=1)"
> chg(c(csFormula, mvFormula))
```

Figure 6, the analysis of the job security data and the plot of its solution:

```
> head(d.jobsecurity)
> ana <- cna(d.jobsecurity, outcome = "JSR", con = .85, cov = .85)
> plot(ana)
```

SVG export of all 18 graphs of the solution, named "ana.svg", "ana1.svg", and so on:

```
> export_as_svg(plot(ana), "ana")
```

Further plotting methods for different solution objects output by **cna**:

```
> plot(asf(ana)) # atomic solution formulas
> plot(csf(ana)) # complex solution formulas
> plot(ana, what = "asf") # same, directly on the cna object
> plot(ana, what = "msc", outcome = "JSR") # minimally sufficient conditions
> condition(csf(ana)$condition, d.jobsecurity) |> plot()
```

The functions used above, with their arguments:

```
> causalHyperGraph(x, show_formula = FALSE, formula_font = "Courier",
                  formula_spaces = NULL, n = 10, ask = length(x) > 1)
> export_as_svg(x, file, sep = "", verbose = TRUE)
```

The drawing of graphs is customized via optional arguments of `causalHyperGraph()`: `show_formula = TRUE` prints the formula below the graph in the font `formula_font`, with a space around the characters contained in `formula_spaces`. If `x` comprises several graphs, `n` limits their number and `ask` decides whether rendering pauses for <Return> between graphs. Since `export_as_svg()` names its output files via `make.unique()`, exporting multiple graphs under one name yields “ana.svg”, “ana1.svg”, and so on.