# Constraint-based Causal Discovery: Conflict Resolution with Answer Set Programming
***Paper Supplement***

**Antti Hyttinen** and **Frederick Eberhardt**
California Institute of Technology
Pasadena, CA, USA

**Matti Järvisalo**
HIIT & Department of Computer Science
University of Helsinki, Finland

## A EXTENSION TO MULTIPLE OVERLAPPING DATA SETS

A set of data sets is said to be overlapping if the sets of measured variables share some, but not all the same variables. Tillman et al. (2009), Tillman and Spirtes (2011) and Triantafillou et al. (2010) have explored causal discovery algorithms in this setting under the assumption that the data is passive observational in each data set and that the underlying causal model is acyclic. In particular, the approach in Tillman and Spirtes (2011) uses R.A. Fisher's technique of pooling p-values to integrate test results from multiple tests on the same set of variables. Hyttinen et al. (2013) extend these approaches to include experimental data sets where a subset of the variables has been subject to an intervention. They also allow for feedback cycles in the generating models. However, their algorithm could not handle conflicting constraints that arise from statistical data.

Our present procedure extends naturally to the completely general setting of multiple overlapping experimental or observational data sets, and handles conflicted constraints. In this setting the set of constraints $\mathbf{K}$ that enters into the constraint optimization is expanded to (in the general case) include all possible constraints that can be obtained from each of the individual data sets. Consider a constraint $k \in \mathbf{K}$ obtained from an (experimental) data set $\mathbf{D}$ where the (possibly empty) set of variables $\mathbf{J} \subset \mathbf{V}$ was subject to an intervention. The weight $w(k)$ now enters into the sum of weights for a graph $G$ if the graph $G_{\mathbf{J}} \not\models k$, where $G_{\mathbf{J}}$ is the same as $G$ except that all edges incident on any variable in $\mathbf{J}$ are removed, i.e. $G_{\mathbf{J}}$ is the manipulated version of $G$. The basic idea is that one simply has to keep track of the experimental setting that the constraint was obtained from, and use that as the basis for the minimization. Consequently, the graph that our method returns is optimal (in the sense of the problem statement) across the available data sets.

Variables that are measured in one of the overlapping data sets, but not in another, are treated as marginalized variables in the d-connection graphs for the latter data set. Below, in Appendices C and F, we specify how to add an intervention operation to the ASP-encoding while ensuring that the d-connection properties are appropriately preserved for the d-connection graphs that specify the test results for experimental data sets.

## B PROBABILISTIC INTERPRETATION OF THE LOG-WEIGHTS

Often the log-score can be interpreted in a probabilistic way. Here we investigate the log-scores and their relation to the problem formulation in Section 2 with this aim. Given a set of constraints $\mathbf{K}$, we set out to find $G^*$ such that

$$\begin{aligned} G^* &\in \operatorname*{arg\,min}_{G \in \mathcal{G}} \sum_{k \in \mathbf{K}\,:\, G \not\models k} w(k) \\ w(k) &= \log P(k \mid \mathbf{D}) - \log[1 - P(k \mid \mathbf{D})] \end{aligned}$$

Now, let vector $\mathbf{q}$ be a binary vector indicating whether a constraint is satisfied by a graph $G$ with $q_i = 1$ if the i:th constraint is satisfied and $q_i = 0$ if it is not, i.e. vector $\mathbf{q}$ describes the equivalence class of graphs with respect to the constraints $\mathbf{K}$. Let vector $\mathbf{p}$ be the probability estimates of constraints calculated from the data, i.e. $p_i = P(k_i \mid \mathbf{D})$. The optimization problem can now be re-described as

$$\operatorname*{arg\,min}_{\mathbf{q}} \sum_{i\,:\,q_i = 0} \log p_i - \log(1 - p_i),$$

where we still (implicitly) require that vector $\mathbf{q}$ has to correspond to some graph in the considered model space that satisfies causal Markov and faithfulness. This formulation can now be converted into an equivalent maximization problem

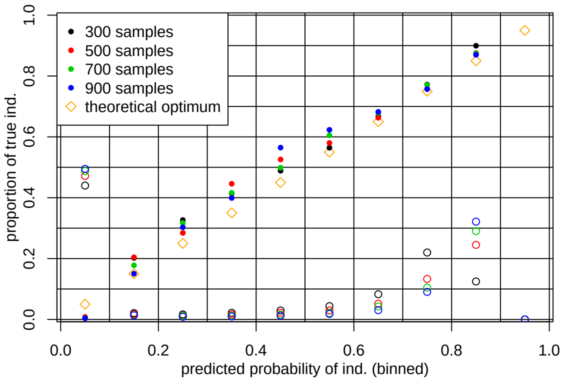

Figure 1: Calibration of probabilities. See text for details.

(adding constants does not change the optimal answer):

$$
\begin{aligned}
& \arg\min_{\mathbf{q}} \sum_{i\,:\,q_i=0} \log p_i - \log(1-p_i) \\
\Leftrightarrow\quad & \arg\max_{\mathbf{q}} \sum_{i\,:\,q_i=0} -\log p_i + \log(1-p_i) \\
\Leftrightarrow\quad & \arg\max_{\mathbf{q}} \sum_{i\,:\,q_i=0} [-\log p_i + \log(1-p_i)] + \sum_{i} \log p_i \\
\Leftrightarrow\quad & \arg\max_{\mathbf{q}} \sum_{i\,:\,q_i=1} \log p_i + \sum_{i\,:\,q_i=0} \log(1-p_i) \\
\Leftrightarrow\quad & \arg\max_{\mathbf{q}} \prod_{i} p_i^{q_i}(1-p_i)^{1-q_i}
\end{aligned}
$$

Thus, we are finding the vector $\mathbf{q}$ (representing an equivalence class of graphs) that maximizes the probability distribution $P(\mathbf{q} \mid \mathbf{p})$, with the approximation that the elements $q_i$ are distributed mutually independently, with the corresponding probabilities in the vector $\mathbf{p}$. In fact, some (in)dependence relations imply others, especially under the assumption of Markov and faithfulness. But independently run independence tests are not able to share this information with each other. Instead, this is exactly the task of conflict resolution: One has to find a $\mathbf{q}$ that corresponds to at least on graph in the model space (and respects all the implications between (in)dependences).

Suppose we further assume that the vector $\mathbf{p}$ exhausts all information on $\mathbf{q}$ available in $\mathbf{D}$, so that it forms a 'sufficient statistic', that is (i) $\mathbf{D} \perp\!\!\!\perp \mathbf{q} \mid \mathbf{p}$ and (ii) $\mathbf{p}$ is a deterministic function of $\mathbf{D}$. In that case we have

$$P(\mathbf{q}|\mathbf{D}) \quad = \quad P(\mathbf{q} \mid \mathbf{p}).$$

So in our minimization we are essentially – modulo the assumptions and approximations just specified – finding the equivalence class of graphs that maximizes the posterior probability.

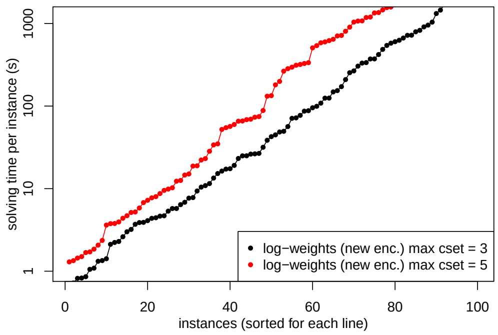

Figure 2: Solving times for the log-weighting scheme.

# C INTERVENTION

Given a d-connection graph $H = (\mathbf{V}, \mathbf{E})_{\mathbf{C}}$, the *intervention* operation $i(H,t)$ for variable $t \in \mathbf{V}$ results in d-connection graph $H' = (\mathbf{V}, \mathbf{E}')_{\mathbf{C}}$, where $\mathbf{E}'$ is related to $\mathbf{E}$ by (i) including in $\mathbf{E}'$ any edges in $\mathbf{E}$ not involving $t$; (ii) including in $\mathbf{E}'$ any edge $x_{a}\!\!-\!t$ if there is an edge $x_{a}\!\!-\!t$ in $\mathbf{E}$; and (iii) not permitting any other edges in $\mathbf{E}'$. Thus, as is standard in the representation of interventions, edges incident on the intervened variable are omitted (they are not copied to $\mathbf{E}'$). Note that the intervention operation cannot be applied to a variable previously conditioned on or marginalized, however these operation can still be applied to a variable after intervention. In the case of conditioning, the purpose of allowing an intervention before conditioning is obvious: One wants to be able to express (in)dependence constraints that condition on an intervened variable. In the case of marginalization the possibility of intervening is necessary in this representation of d-connection graphs in order to reach (in)dependence constraints between non-intervened variables in a data set where some variables have been subject to intervention. Thus, with regard to the "encoding DAG" in Figure 3, there are still multiple options of where the intervention operations could be included, since the order of intervention before marginalization and conditioning only applies to each variable individually. A variety of considerations may make one arrangement of operations in the encoding DAG more useful than another: One may prefer to integrate constraints within one data set first, or one may want to integrate similar constraints across data sets first.

# D PROOF OF THEOREM 1

**Proof:** Assume there is a path $p$ of type $x_{a}\cdots{}_{b}y$ that is d-connecting given $\mathbf{C}'' \supseteq \mathbf{C}'$ in $H$. Since $p$ is d-connecting, every collider on $p$ is in $\mathbf{C}''$ and every non-collider is not in $\mathbf{C}''$. In the conditioning operation all edges not adjacent to $w$ are preserved. So the only parts of $p$ that may be broken in $H'$ are due to edges connected to the newly conditioned variable $w$. Since $w \in \mathbf{C}''$ and the path $p$ is d-connecting given $\mathbf{C}''$, it follows that $w$ must be a collider on $p$, which means it can only appear in components of the type $s{\scriptstyle a}{\rightarrow}w[{\leftrightarrow}w]^{*}{\leftarrow}{\scriptstyle b}t$ on the path $p$, where $s \neq w, t \neq w$ and the part in the brackets may appear zero or more times. Let $p'$ be the path between $x$ and $y$ where all such components are replaced by $s{\scriptstyle a}{-}{\scriptstyle b}t$. This replacement edge is in $H'$ due to rule (ii) in the definition of the conditioning operation. Furthermore, since there is no other way to break a path, $p'$ as a whole is present in $H'$, and $p'$ is still d-connecting given $\mathbf{C}''$. The path $p'$ will have the same end points as path $p$, and thus a path of type $x{\scriptstyle a}{\cdots}{\scriptstyle b}y$ is present in $H'$ that is d-connecting given $\mathbf{C}''$.

Assume there is a path $p'$ of type $x{\scriptstyle a}{\cdots}{\scriptstyle b}y$ that is d-connecting given $\mathbf{C}'' \supseteq \mathbf{C}'$ in $H'$. As all other edges in $H'$ are just copies of edges in $H$, for the path $p'$ not to exist in $H$ there must be edges $s{\scriptstyle a}{-}{\scriptstyle b}t$ on the path $p'$ that are added by rule (ii) in the definition of the conditioning operation. But then for each such edge there has to be a triple $s{\scriptstyle a}{\rightarrow}w{\leftarrow}{\scriptstyle b}t$ in $H$, as otherwise rule (ii) would not have added $s{\scriptstyle a}{-}{\scriptstyle b}t$ to $H'$. Such a triple cannot be blocked since $w \in \mathbf{C}''$. Replacing all edges not present in $H$ by such triples similarly results in a path $p$ in $H$ that is d-connecting given $\mathbf{C}''$ and still of the type $x{\scriptstyle a}{\cdots}{\scriptstyle b}y$. □

The proof for the marginalization and intervention operations follow the exact same idea.

# E FURTHER SIMULATIONS

Figure 1 shows the probability calibration plot for the estimates of the probability of independence determined by the model comparison described in Section 4.3. The estimates are based on 300 causally insufficient and possibly cyclic models under passive observation. The predictions were divided into 10 bins with equal width, denoted by the black lines in the plot. We then calculated how many times the true result was in fact an independence. The filled circles mark these true proportions against the predicted probability for different sample sizes. The probabilities seem to be roughly calibrated for the different sample sizes when the prior is set to $\alpha = 0.5$ (and the equivalent sample size prior for the local-score is set to 20). The unfilled circles show the proportions of predictions in each bin. Note that for these sample sizes, the test does not yet predict a probability over $0.9$ for an independence. This seems natural since it is very hard to be sure of an independence, it might just be a very weak dependence. Otherwise the high and low probabilities are predicted more often when the sample size grows, while the probabilities in the middle bins are predicted somewhat more often for lower sample sizes.

Figure 2 further explores the scalability of the method using log-weights for 7-variable graphs. The running times are in log-scale, and again the instances are sorted according to their solving times. Limiting the maximimum conditioning set size to 3, which already allows most inferences to be made, cuts down the number of constraints and allows for faster solving.

# F FULL ASP ENCODING

Finally, we give the full ASP-encoding with the *intervention*, *conditioning* and *marginalization* operations, presented in Figure 3. In the full encoding, an additional set $\mathbf{J}$ is introduced that represents the intervention set. In analogy with $cond(\mathbf{V}, \mathbf{C}, \mathbf{J}, z)$ and $marg(\mathbf{V}, \mathbf{C}, \mathbf{J}, z)$, the input predicate $intervene(\mathbf{V}, \mathbf{C}, \mathbf{J}, z)$ enables intervening on a variable $z$ in a d-connection graph that has exactly the variables $\mathbf{V}$ and conditioning set $\mathbf{C}$.

## References

Hyttinen, A., Hoyer, P. O., Eberhardt, F., and Järvisalo, M. (2013). Discovering cyclic causal models with latent variables: A general SAT-based procedure. In *Proceedings of UAI*, pages 301–310. AUAI Press.

Tillman, R. E., Danks, D., and Glymour, C. (2009). Integrating locally learned causal structures with overlapping variables. In *Proceedings of NIPS 2008*, pages 1665–1672.

Tillman, R. E. and Spirtes, P. (2011). Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *Proceedings of AISTATS*. JMLR.

Triantafillou, S., Tsamardinos, I., and Tollis, I. G. (2010). Learning causal structure from overlapping variable sets. In *Proceedings of AISTATS*, pages 860–867. JMLR.

Conditioning on variable $z \in \mathbf{V}, \forall x, y \in \mathbf{V} \setminus z$:

$$\begin{aligned}
\mathit{th}(x, y, \mathbf{V} \setminus z, \mathbf{C} \cup z, \mathbf{J}) &\text{ :- } \mathit{th}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{cond}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{th}(x, y, \mathbf{V} \setminus z, \mathbf{C} \cup z, \mathbf{J}) &\text{ :- } \mathit{th}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{hh}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{cond}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z).
\end{aligned}$$

$$\begin{aligned}
\mathit{hh}(x, y, \mathbf{V} \setminus z, \mathbf{C} \cup z, \mathbf{J}) &\text{ :- } \mathit{hh}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{cond}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{hh}(x, y, \mathbf{V} \setminus z, \mathbf{C} \cup z, \mathbf{J}) &\text{ :- } \mathit{hh}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{hh}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{cond}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z).
\end{aligned}$$

$$\begin{aligned}
\mathit{tt}(x, y, \mathbf{V} \setminus z, \mathbf{C} \cup z, \mathbf{J}) &\text{ :- } \mathit{tt}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{cond}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{tt}(x, y, \mathbf{V} \setminus z, \mathbf{C} \cup z, \mathbf{J}) &\text{ :- } \mathit{th}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{th}(y, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{cond}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z).
\end{aligned}$$

Marginalizing a variable $z \in \mathbf{V}, \forall x, y \in \mathbf{V} \setminus z$:

$$\begin{aligned}
\mathit{th}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{th}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{th}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{tt}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{th}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{th}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{th}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{th}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{th}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{tt}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{hh}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{th}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{th}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{tt}(z, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{hh}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z).
\end{aligned}$$

$$\begin{aligned}
\mathit{hh}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{hh}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{hh}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{th}(z, x, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{th}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{hh}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{hh}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{th}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{hh}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{th}(z, x, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{hh}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{hh}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{hh}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{tt}(z, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{hh}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z).
\end{aligned}$$

$$\begin{aligned}
\mathit{tt}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{tt}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{tt}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{tt}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{tt}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{tt}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{th}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{tt}(z, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{tt}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{tt}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{th}(y, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{tt}(x, y, \mathbf{V} \setminus z, \mathbf{C}, \mathbf{J}) &\text{ :- } \mathit{th}(x, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{tt}(z, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{th}(y, z, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{marg}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z).
\end{aligned}$$

Intervening on a variable $z \in \mathbf{V}, \forall x, y \in \mathbf{V}$:

$$\begin{aligned}
\mathit{th}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J} \cup z) &\text{ :- } \mathit{th}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), z \neq y, \\
&\qquad \mathit{intervene}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{hh}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J} \cup z) &\text{ :- } \mathit{hh}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), x \neq z, y \neq z, \\
&\qquad \mathit{intervene}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z). \\
\mathit{tt}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J} \cup z) &\text{ :- } \mathit{tt}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{intervene}(\mathbf{V}, \mathbf{C}, \mathbf{J}, z).
\end{aligned}$$

Inferring failures to satisfy (in)dependencies $\forall x \forall y > x, \forall \mathbf{C}, \mathbf{V} = \{x, y\}$:

$$\begin{aligned}
\mathit{fail}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W) &\text{ :- } \mathit{tt}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{indep}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W). \\
\mathit{fail}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W) &\text{ :- } \mathit{th}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{indep}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W). \\
\mathit{fail}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W) &\text{ :- } \mathit{th}(y, x, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{indep}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W). \\
\mathit{fail}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W) &\text{ :- } \mathit{hh}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \mathit{indep}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W). \\
\mathit{fail}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W) &\text{ :- } \text{not } \mathit{th}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \text{not } \mathit{th}(y, x, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \text{not } \mathit{hh}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \text{not } \mathit{tt}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}), \\
&\qquad \mathit{dep}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W).
\end{aligned}$$

Weak constraints $\forall x \forall y > x, \forall \mathbf{C}, \mathbf{J}, \mathbf{V} = \{x, y\}$:

$$:\sim \mathit{fail}(x, y, \mathbf{V}, \mathbf{C}, \mathbf{J}, W). \, [W]$$

Figure 3: The ASP encoding, including the encoding of the intervention operation.