A Efficient Checking of Actual Causality with SAT Solving: Proofs

A.1 Lemma 1

Lemma 2. *In a binary model, if $\vec{X} = \vec{x}$ is a cause of φ, according to HP [9] definition, then every $\vec{x}'$ in the definition of AC2 always satisfies $\forall i \bullet x'_i = \neg x_i$.*

Proof. We use the following notation: $\overrightarrow{X}_{(n)}$ stands for a vector of length n, $X_1, \ldots, X_n$; and $\overrightarrow{X}_{(n)} = \overrightarrow{x}_{(n)}$ stands for $X_1 = x_1, \ldots, X_n = x_n$. Let $\overrightarrow{X}_{(n)} = \overrightarrow{x}_{(n)}$ be a cause for φ in some model M.

1. AC1 yields
$$(M, \overrightarrow{u}) \models (\overrightarrow{X}_{(n)} = \overrightarrow{x}_{(n)}) \wedge (M, \overrightarrow{u}) \models \varphi. \tag{1}$$
2. Assume that the lemma does not hold. Then there is some index k such that $x'_k = x_k$ and AC2 holds. Because we are free to choose the ordering of the variables, let us set $k = n$ wlog. We may then rewrite AC2 as follows:
$$\exists \overrightarrow{W}, \overrightarrow{w}, \overrightarrow{x}'_{(n)} \bullet (M, \overrightarrow{u}) \models (\overrightarrow{W} = \overrightarrow{w}) \implies (M, \overrightarrow{u}) \models \\ [\overrightarrow{X}_{(n-1)} \leftarrow \overrightarrow{x}'_{(n-1)}, X_n \leftarrow x_n, \overrightarrow{W} \leftarrow \overrightarrow{w}]\neg\varphi. \tag{2}$$
3. We will show that equations 1 and 2 give rise to a smaller cause, namely $\overrightarrow{X}_{(n-1)} = \overrightarrow{x}_{(n-1)}$, contradicting the minimality requirement AC3. We need to show that the smaller cause $\overrightarrow{X}_{(n-1)} = \overrightarrow{x}_{(n-1)}$ satisfy AC1 and AC2, as stated by equations 3 and 4 below. This violates the minimality requirement of AC3 for $\overrightarrow{X}_{(n)} = \overrightarrow{x}_{(n)}$.
$$(M, \overrightarrow{u}) \models (\overrightarrow{X}_{(n-1)} = \overrightarrow{x}_{(n-1)}) \wedge (M, \overrightarrow{u}) \models \varphi \tag{3}$$
states AC1 for a candidate "smaller" cause $\overrightarrow{X}_{(n-1)}$. Similarly,
$$\exists \overrightarrow{W}^*, \overrightarrow{w}^*, \overrightarrow{x}'^*_{(n-1)} \bullet (M, \overrightarrow{u}) \models (\overrightarrow{W}^* = \overrightarrow{w}^*) \\ \implies (M, \overrightarrow{u}) \models [\overrightarrow{X}_{(n-1)} \leftarrow \overrightarrow{x}'^*_{(n-1)}, \overrightarrow{W}^* \leftarrow \overrightarrow{w}^*]\neg\varphi \tag{4}$$
formulates AC2 for this candidate smaller cause $\overrightarrow{X}_{(n-1)}$.
4. Let Ψ denote the structural equations that define M. Let Ψ' be Ψ without the equations that define the variables $\overrightarrow{X}_{(n)}$ and $\overrightarrow{W}$; and let Ψ'' be Ψ without the equations that define the variables $\overrightarrow{X}_{(n-1)}$ and $\overrightarrow{W}$. Clearly, $\Psi'' \implies \Psi'$. We can turn equation 1 into a propositional formula, namely
$$E_1 := \big(\Psi \wedge \overrightarrow{X}_{(n-1)} = \overrightarrow{x}_{(n-1)} \wedge X_n = x_n\big) \wedge \varphi. \tag{5}$$

Similarly, equation 3 is reformulated as

$$E_2 := \left(\Psi \wedge \overrightarrow{X}_{(n-1)} = \overrightarrow{x}_{(n-1)}\right) \wedge \varphi. \tag{6}$$

Because equation 2 holds, we fix some $\overrightarrow{W}, \overrightarrow{w}, \overrightarrow{x}'_{(n)}$ that make it true and rewrite this equation as

$$E_3 := \left(\Psi' \wedge \overrightarrow{X}_{(n-1)} = \overrightarrow{x}'_{(n-1)} \wedge X_n = x_n \wedge \overrightarrow{W} = \overrightarrow{w}\right) \implies \neg\varphi. \tag{7}$$

Finally, in equation 4, we use exactly these values to also fix $\overrightarrow{W}^* = \overrightarrow{W}$, $\overrightarrow{w}^* = \overrightarrow{w}$, and $\overrightarrow{x}'^*_{(n-1)} = \overrightarrow{x}'_{(n-1)}$, and reformulate this equation as

$$E_4 := \left(\Psi'' \wedge \overrightarrow{X}_{(n-1)} = \overrightarrow{x}'_{(n-1)} \wedge \overrightarrow{W} = \overrightarrow{w}\right) \implies \neg\varphi. \tag{8}$$

It is then a matter of equivalence transformations to show that

$$(\Psi'' \implies \Psi') \implies \left((E_1 \wedge E2) \implies (E_3 \wedge E_4)\right) \tag{9}$$

is a tautology, which proves the lemma.

A.2 Theorem 1

Theorem 1. *Formula F constructed within Algorithm.1 is satisfiable iff AC2 holds for a given M, $\vec{u}$, a candidate cause $\vec{X}$, and a combination of events φ.*

Proof. The proof consists of two parts.

Part 1 *SAT(F) $\implies$ AC2, AC2 holds if F is satisfiable*

We show this by contradiction. Assume that F is satisfiable and AC2 does not hold. Based on F's truth assignment, $\vec{v'}$, we cluster the variables into 3 groups.

$$F := \neg\varphi \wedge \bigwedge_{i=1\ldots n} f(U_i = u_i) \wedge \bigwedge_{i=1\ldots \ell} f(X_i = \neg x_i) \wedge \bigwedge_{i=1\ldots m, \nexists j \bullet X_j = V_i} \left(V_i \leftrightarrow F_{V_i} \vee f(V_i = v_i)\right)$$

1. $\vec{X}$: each variable is fixed exactly to the negation of its original value, i.e., $X_i = \neg x_i \forall X_i \in \vec{X}$ (recall $\vec{X} \subseteq \vec{V}$). **2.** $\vec{W}^*$: variables in this group, if they exist, have equal truth and original assignments, i.e., $\langle W_1^*, \ldots, W_s^* \rangle$ s.t. $\forall i \forall j \bullet (i \neq j \Rightarrow W_i^* \neq W_j^*) \wedge (W_i^* = V_j \Leftrightarrow v'_j = v_j)$ **3.** $\vec{Z}$: variables in this group evaluate differently from their original evaluation, i.e., $\langle Z_1, \ldots, Z_k \rangle$ s.t. $\forall i \forall j \bullet (i \neq j \Rightarrow Z_i \neq Z_j) \wedge (Z_i = V_j \Leftrightarrow v'_j \neq v_j) \wedge (\forall i \nexists j \bullet Z_i = X_j)$.

Using $\vec{W}^*, \vec{Z}$, we re-write F as F' which is also satisfiable.

$$F' := \neg\varphi \wedge \bigwedge_{i=1\ldots n} f(U_i = u_i) \wedge \bigwedge_{i=1\ldots \ell} f(X_i = \neg x_i) \wedge \bigwedge_{i=1\ldots s} f(W_i^* = w_i^*) \wedge \bigwedge_{i=1\ldots k} (Z_i \leftrightarrow F_{Z_i})$$

Recall that M is acyclic; therefore there is a unique solution to the equations. Let Ψ be the equations in M without the equations that define the variables $\vec{X}$. Let Ψ_k be Ψ without the equations of some variables in a set $\vec{W}_k$. Since AC2 does not hold, $\forall k \bullet \vec{W}_k \subseteq V \setminus X \Rightarrow (\vec{X} = \vec{\neg x} \wedge \vec{W}_k = \vec{w_k} \wedge \Psi_k \wedge \neg\varphi)$ evaluates to *false*. In case $\vec{W}_k = \vec{W}^*$, the previous unsatisfiable formula is equivalent to the satisfiable F', implying a contradiction.

Part 2 *AC2 $\implies$ SAT(F); F is satisfiable if AC2 holds*

Assume that AC2 holds and F is unsatisfiable. Then $\exists \vec{W}, \vec{w}, \vec{x}' \bullet (M, \vec{u}) \models (\vec{W} = \vec{w}) \implies (M, \vec{u}) \models \left[\vec{X} \leftarrow \vec{x}', \vec{W} \leftarrow \vec{w}\right]\neg\varphi$. By definition [9], $(M, \vec{u}) \models [Y_1 \leftarrow y_1..Y_k \leftarrow y_k]\varphi$ is equivalent to $(M_{Y_1 \leftarrow y_1..Y_k \leftarrow y_k}, \vec{u}) \models \varphi$, i.e., we replace specific equations in M to obtain a new model $M' = M_{Y_1 \leftarrow y_1,\ldots,Y_k \leftarrow y_k}$. So,we replace the equations of the variables in $\vec{X}, \vec{W}$ in M to obtain a new model, M', such that $(M', \vec{u}) \models \neg\varphi$. Equations of $\vec{X}, \vec{W}$ variables are now of the form $V_i = v_i$, i.e., each variable is equal to a constant value. Note that M' is only different from M in the equations of $\vec{X}, \vec{W}$. Hence, M' is acyclic and has a unique solution for a given $\vec{U} = \vec{u}$. We construct a formula, F' (shown below), that is a conjunction of the variables in sets X', W', U in M'. Because of their equations, each variable is represented by a constant, i.e., a positive or a negative literal. Based on the nature of this formula, it is satisfiable with exactly the same truth assignment as the unique solution of M'.

$$F' := \bigwedge_{i=1\ldots n} f(U_i = u_i) \wedge \bigwedge_{i=1\ldots s} f(W_i' = w_i') \wedge \bigwedge_{i=1\ldots \ell} f(X_i' = x_i')$$

Now, we add the remaining variables, i.e., $\forall i \bullet V_i \notin (\vec{X} \cup \vec{W})$, as formulas using the $\leftrightarrow$ operator. The overall formula F'', is satisfiable because we have an assignment that makes each equivalence relation true.

$$F'' := F' \wedge \bigwedge_{i=1\ldots m, \nexists j \bullet X_j = V_i, W_j = V_i} (V_i \leftrightarrow F_{V_i})$$

We have $(M', \vec{u}) \models \neg\varphi$, which says that the model evaluates $\neg\varphi$ to true with its unique solution (same assignment of F''). We add another clause to F'' which evaluates to true and keeps the formula satisfiable. That is, $F''' := F'' \wedge \neg\varphi$. Last, we only have to show the relation between (F from Algorithm.1 and F'''). We can rewrite F (shown at the beginning of the proof) such that we remove all disjuncts of the form $(V_i \leftrightarrow F_{V_i})$ for the variables in $\vec{W}$. Similarly, we remove all disjuncts of the form $f(V_i = v_i)$ for all the variables that are not in $\vec{W}$. According to our assumption, F is still unsatisfiable, since we removed disjunctions from the clauses. Then, we reach a contradiction since F is equivalent to F''' which is satisfiable for the same clauses.

A.3 Theorem 2

Before presenting the proof, we define the term *non-minimal cause.*

Definition 3. ***Non-minimal Cause*** *is a candidate cause $\vec{X_{(n)}} = \vec{x}_{(n)}$ that satisfies AC2 yet contains at least one element X_n that satisfies one of the following conditions*

- ***NMC1.*** *AC2 holds for the smaller cause $\vec{X}_{(n-1)}$ regardless whether $X_n \in \vec{W}$ or not, i.e., $(M, \vec{u}) \models \left[\vec{X}_{(n-1)} \leftarrow \vec{x}'_{(n-1)}, \vec{W} \leftarrow \vec{w}\right]\neg\varphi$ and $(M, \vec{u}) \models \left[\vec{X}_{(n-1)} \leftarrow \vec{x}'_{(n-1)}, X_n \leftarrow x_n, \vec{W} \leftarrow \vec{w}\right]\neg\varphi$ both hold.*

– ***NMC2.*** *AC2 holds for the smaller cause $\vec{X}_{(n-1)}$ only if $X_n \notin \vec{W}$, i.e.,* $(M, \vec{u}) \models \left[\vec{X}_{(n-1)} \leftarrow \vec{x}'_{(n-1)}, \vec{W} \leftarrow \vec{w}\right]\neg\varphi$

Informally, **NMC1** deals with irrelevant variables, i.e., those that do not affect the cause relation to the effect. **NMC2** targets the case of relevant variables that are affected by the cause but not necessary for it to be a cause.

Theorem 2. *Algorithm.2 returns false iff $\vec{X}$ is a non-minimal cause.*

Proof. **Part 1** *If the cause is a non-minimal cause, then Algorithm 2 returns false.*
For the algorithm to return *false*, G must be satisfiable first, and the cardinality check is passed. So we prove this part by showing that if any of the conditions in Def.3 hold then G is satisfiable and the check is passed and the algorithm returns *false.*

1. Recall $G := \neg\varphi \wedge \bigwedge_{i=1\ldots\ell} f(U_i = u_i) \wedge \bigwedge_{i=1\ldots m, \nexists j \bullet X_j = V_i} \big(V_i \leftrightarrow F_{V_i} \vee f(V_i = v_i)\big) \wedge \bigwedge_{i=1\ldots\ell}(X_i \vee \neg X_i)$. Let us rewrite the formula to abstract the first part as, $G := G_{base} \wedge \bigwedge_{i=1\ldots n}(X_i \vee \neg X_i)$.
2. Note how $\vec{X_{(n)}}$ is added to G as $(X_1 \vee \neg X_1) \wedge (X_2 \vee \neg X_2) \ldots (X_n \vee \neg X_n)$. Re-write this big conjunction to have its equivalent disjunctive normal from (DNF) i.e., $(\neg X_1 \wedge \neg X_2 \cdots \wedge \neg X_n) \vee (\neg X_1 \wedge \neg X_2 \cdots \wedge X_n) \cdots \vee (X_1 \wedge X_2 \cdots \wedge X_n)$. Assume wlog that all the original values of $\vec{X_{(n)}}$ (which lead to φ holding true) were *true*, hence to check them in AC2 we present them as negative literals like $\neg X_i$. Looking at the DNF, we have 2^n clauses that list all the possible cases of negating or fixing the elements in $\vec{X}$. Then, we partition G according to the clauses, i.e, $G := G_1 \vee G_2 \ldots G_{2^n}$, where $G_1 := G_{base} \wedge (\neg X_1 \wedge \neg X_2 \cdots \wedge \neg X_n)$. In the case of the clause where all variables $\vec{X_{(n)}}$ are negated, the corresponding G, i.e., G_1, is exactly formula F from Algorithm.1.
3. Each G_i, other than G_1, fixes some group of elements to their original evaluation (X_i) and negates some, possibly none (G_{2^n}), other elements $(\neg X_i)$. Clearly, G_i is an F formula (from Algorithm.1) for all the negated variables, in a clause, as $\vec{X}$ but with x special fixed variables that are added to $\vec{W}$. Hence, a G_i is a check of AC2 for a specific subset of the causes given that the other part (fixed) of the cause is in $\vec{W}$. This is the case of **NMC1** in Def.3, i.e, AC2 still holds after transferring some elements $\vec{X}^*$ from $\vec{X}$ to $\vec{W}$. $\vec{X}^*$ is guaranteed to be expressed in one of the 2^n clauses, and hence, by a specific G_k. According to Theorem.1, such a G_k is satisfiable since AC2 holds. Then, for a non-minimal cause based on **NMC1**, G is satisfiable since G_k is satisfiable. For this case, we only have to show that it passes the cardinality check in the algorithm. It is clear that $\forall i \in \vec{X}^*\ v'_i = v_i$ since they will be in $\vec{W}$. This makes the condition in the algorithm evaluates to true and hence, the algorithm returns false.
4. Similarly, for the second case **NMC2**, i.e., the non-minimal part should not be in $\vec{W}$. AC2 holds for the non-minimal cause, i.e, F and G_1 are satisfiable

and then G is also satisfiable. Since the non minimal parts in this case are not in $\vec{W}$ or $\vec{X}$, then they follow their equations in the model, and hence $\exists i \bullet v'_i = [\vec{V} \mapsto \vec{v}']F_{X_i}$, which results in false returned by the algorithm.

Part 2 *If Algorithm2 returns false, then the cause is a non-minimal cause*

To prove this part, we show that if $\vec{X}$ is a minimal cause, the algorithm does not return false. The algorithm returns false if G is satisfiable ($\vec{X}$ or a subset of it fulfill AC2), and the cause passes the cardinality check. A minimal cause will have a satisfiable G. For the cardinality check, by Lemma.1, a cause should have all its elements negated. Hence the first conjunct in line 5 of Algorithm.2 will be *true* for each element. If the second conjunct in the same line ($v'_i \neq [\vec{V} \mapsto \vec{v}']F_{X_i}$) evaluates to *false* for any element then, this is not a minimal cause. Hence, for a minimal cause the two conjuncts will evaluate to true for all the elements in $\vec{X}$, and then a false is never returned for such a case.

A.4 Theorem 3

Theorem 3. *Formula G' is satisfiable iff AC3 is violated.*

Proof. The proof follows from the fact that to check minimality, it is sufficient to check $\vec{X}$'s subsets of cardinality k where $k = |\vec{X}| - 1$, i.e., the subsets of $\vec{X}$ with one element less. We denote these subsets by $\mathcal{P}_k(X)$. A set $\vec{X}$ of size l has l subsets of size $l - 1$.

1. Recall that $G := \neg\varphi \wedge \bigwedge_{i=1...\ell} f(U_i = u_i) \wedge \bigwedge_{i=1...m, \nexists j \bullet X_j = V_i}(V_i \leftrightarrow F_{V_i} \vee f(V_i = v_i)) \wedge \bigwedge_{i=1...\ell}(X_i \vee \neg X_i)$.
2. $G' := G \wedge \bigvee_{i=1...\ell} \big((X_i \leftrightarrow F_{X_i}) \vee (f(X_i = x_i)\big)$. Let us take the *base case*: $l = 2$, then $G' := G \wedge \big((X_1 \leftrightarrow F_{X_1}) \vee (f(X_1 = x_1)\big) \vee \big((X_2 \leftrightarrow F_{X_2}) \vee (f(X_2 = x_2)\big)$. If we distribute the conjunction over disjunction, then $G' := G \wedge ((X_1 \leftrightarrow F_{X_1}) \vee (f(X_1 = x_1)) \vee G \wedge ((X_2 \leftrightarrow F_{X_2}) \vee (f(X_2 = x_2))$. Call $G \wedge ((X_i \leftrightarrow F_{X_i}) \vee (f(X_i = x_i))$, G_i^*. Then $G' = G_1^* \vee G_2^* ... \vee G_l^*$. For the base case $G' = G_1^* \vee G_2^*$
3. A G_i^* represents the case where one cause variable X_i is removed from the cause set by adding this clause $\big((X_i \leftrightarrow F_{X_i}) \vee (f(X_i = x_i)\big)$ which then makes G_i^* only satisfiable if X_i was not negated, i.e., not part of the cause. G_i^* then can be seen as an AC2 check of a smaller cause (in relation with formula F from Theorem 1). Since G' is a disjunction of all $G_i^* \in \mathcal{P}_k(X)$, G' is an AC2 check of all subsets of $\vec{X}$ with size k. G' is only satisfiable if one or more G_i^* clauses are satisfiable. This satisfiability of a G_i^* makes the corresponding X_i an irrelevant cause and hence AC3 is violated.
4. By induction, a cause $\vec{X}$ of size n written (by distributing the conjunction over disjunction) as $G' = G_1^* \vee G_2^* ... G_n^*$ is only satisfiable if AC2 holds for a subset-cause of size $n - 1$, and consequently AC3 is violated.

References

1. Aleksandrowicz, G., Chockler, H., Halpern, J.Y., Ivrii, A.: The computational complexity of structure-based causality. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
2. Beer, A., Heidinger, S., Kühne, U., Leitner-Fischer, F., Leue, S.: Symbolic causality checking using bounded model checking. In: Model Checking Software - 22nd International Symposium, SPIN (2015)
3. Beer, I., Ben-David, S., Chockler, H., Orni, A., Trefler, R.J.: Explaining counterexamples using causality. Formal Methods in System Design **40**(1), 20–40 (2012)
4. Bertossi, L.: Characterizing and computing causes for query answers in databases from database repairs and repair programs. In: International Symposium on Foundations of Information and Knowledge Systems. pp. 55–76. Springer (2018)
5. Chockler, H., Halpern, J.Y.: Responsibility and blame: A structural-model approach. J. Artif. Intell. Res. **22**, 93–115 (2004)
6. Chockler, H., Halpern, J.Y., Kupferman, O.: What causes a system to satisfy a specification? ACM Transactions on Computational Logic (TOCL) **9**(3), 20 (2008)
7. Eén, N., Sörensson, N.: An extensible sat-solver. In: Theory and Applications of Satisfiability Testing, 6th International Conference. pp. 502–518 (2003)
8. Feigenbaum, J., Jaggard, A.D., Wright, R.N.: Towards a formal model of accountability. In: 2011 New Security Paradigms Workshop, NSPW '11. pp. 45–56 (2011)
9. Halpern, J.Y.: A modification of the halpern-pearl definition of causality. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI. pp. 3022–3033 (2015)
10. Halpern, J.Y.: Actual causality. The MIT Press, Cambridge, Massachussetts (2016)
11. Halpern, J.Y., Kleiman-Weiner, M.: Towards formal definitions of blameworthiness, intention, and moral responsibility. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18) (2018)
12. Hopkins, M.: Strategies for determining causes of events. In: AAAI/IAAI (2002)
13. Hume, D.: An Enquiry Concerning Human Understanding (1748)
14. Ibrahim, A., Rehwald, S., Pretschner, A.: Efficient checking of actual causality via sat solving - paper proofs, https://github.com/amjadKhalifah/HP2SAT1.0/blob/master/doc/proofs.pdf
15. Kacianka, S., Kelbert, F., Pretschner, A.: Towards a unified model of accountability infrastructures. In: Proceedings of CREST@ETAPS 2016,. pp. 40–54 (2016)
16. Künnemann, R., Esiyok, I., Backes, M.: Automated verification of accountability in security protocols. CoRR **abs/1805.10891** (2018)
17. Leitner-Fischer, F., Leue, S.: Causality checking for complex system models. In: Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI 2013, Rome, Italy, January 20-22, 2013. Proceedings (2013)
18. Lewis, D.: Causation. Journal of Philosophy **70**(17), 556–567 (1973)
19. Meliou, A., Gatterbauer, W., Halpern, J.Y., Koch, C., Moore, K.F., Suciu, D.: Causality in databases. IEEE Data Eng. Bull. **33**(3), 59–67 (2010)
20. Moore, M.S.: Causation and responsibility : an essay in law, morals, and metaphysics. Oxford Univ. Press (2009)
21. Newsham, Z., Ganesh, V., Fischmeister, S., Audemard, G., Simon, L.: Impact of community structure on sat solver performance. In: International Conference on Theory and Applications of Satisfiability Testing. pp. 252–268. Springer (2014)
22. Salimi, B., Bertossi, L.: From causes for database queries to repairs and model-based diagnosis and back (2014)