Efficient Checking of Actual Causality with SAT Solving - Evaluated Models

Amjad Ibrahim, Simon Rehwald, and Alexander Pretschner
{ibrahim, rehwald, pretschn}@in.tum.de

Department of Informatics, Technical University of Munich, Germany

1 Introduction

In the following, we present and describe the examples which our tool is based on. In summary we prepared 15 different causal models. On the one hand, we took all those models from [3] which consist of binary variables only. Since these examples are rather small and therefore easy to understand, they mainly serve for testing our approaches and showing that they work as expected. On the other hand, we came up with some examples on our own, obtained one from an industrial partner and considered other literature. This leads to the list of causal models shown in Tab. 1. In order to give a feeling for their size, we specified the number of endogenous variables they consist of.

Causal Model	**Source**	**Number of Endogenous Variables**
Rock-Throwing	[3, 5, 6]	5
Forest Fire (conjunctive & disjunctive)	[3, 5, 6]	3
Prisoners	[3, 6]	4
Assassin (first & second variant)	[3]	3
Railroad	[3]	4
Abstract Model 1 & 2	own example	8 & 3
Steal Master Key	industrial partner	36
Steal Master Key with Eight Attackers	industrial partner	91
Leakage in Subsea Production System	[1]	41
Leakage in Subsea Production System with Preemption	based on [1]	41
Binary Tree	own example	15 - 4095
Abstract Model 1 Combined with Binary Tree	own example	4103

Table 1. Evaluated Causal Models

2 Description of the Evaluated Models

2.1 Rock-Throwing

The first model is the Rock-Throwing example explained in [3, 5, 6]. According to the authors, we can assume that Suzy and Billy both throw a rock on a bottle which shatters if one of them hits. Furthermore, we know that Suzy's rock hits the bottle slightly earlier than Billy's and both are accurate throwers. This leads to the endogenous variables ST ("Suzy throws"), BT ("Billy throws"), SH ("Suzy hits"), BH ("Billy hits") and BS ("bottle shatters"). Additionally, since the authors did not explicitly specify the exogenous variables of this example, we introduce the two exogenous variables ST_{exo} and BT_{exo}. In Fig. 1, we can see the corresponding causal graph and obtain the following equations:

- $ST = ST_{exo}$
- $BT = BT_{exo}$
- $SH = ST$,
- $BH = BT \wedge \neg SH$.
- $BS = SH \vee BH$

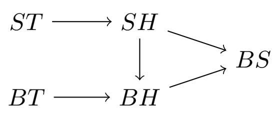


Fig. 1. Rock-throwing example (Source: [5])

2.2 Forest Fire

Another one of Halpern and Pearl's basic examples is a forest fire (FF) that is caused by a lightning (L) or a dropped match (MD, "match dropped") (disjunctive scenario) or only if both occur at the same time (conjunctive scenario). Hence, he actually describes two causal models with this example. The causal graph, which is the same for both variants is depicted in Fig. 2 and the corresponding equations are as follows:

- $L = L_{exo}$
- $MD = MD_{exo}$
- $FF = L \vee MD$ (disjunctive scenario) or $FF = L \wedge MD$ (conjunctive scenario)

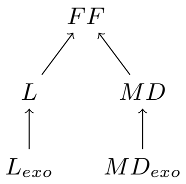


Fig. 2. Causal Graph of Forest Fire Example (Source: [6])

2.3 Prisoners

An additional example found in [6] and [3] is about four prisoners. One of them dies (specified by variable D) if prisoner A loads prisoner B's gun which then shoots or if prisoner C both loads and shoots his gun. The equations in this causal model are straightforward; Fig. 3 shows the causal graph:

- $A = A_{exo}$
- $B = B_{exo}$
- $C = C_{exo}$
- $D = (A \wedge B) \vee C$

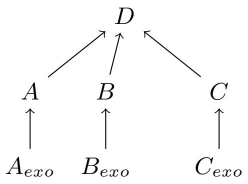


Fig. 3. Causal Graph of Prisoners Example (Source: Own Figure)

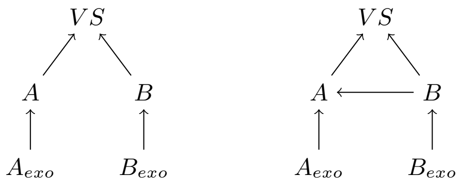
2.4 Assassin

An example very similar to the (disjunctive) forest fire example described previously is about an assassin putting poison into the coffee of its victim. However, the latter's bodyguard has an antidote for the poison which makes the victim survive. In [3], the author describes two variants of this example. In the first one, the assassin puts the poison into the coffee independently from what the bodyguard does. In the second variant, however, the assassin only then puts the poison into the coffee, if the victim's bodyguard uses his antidote. As [3] does not explicitly mention the variables within this example, we use the same ones introduced by [4], who consider this example as well. However, we specify A as "assassin does put in poison", and not "assassin does *not* put in poison", because this makes it easier to model and understand the second variant of this example.

The other variables are the same as in [4]: B "bodyguard puts in antidote" and VS for "victim survives". Adding exogenous variables for A and B, we obtain the following equations (for both variants):

- $B = B_{exo}$
- $A = A_{exo}$ (first variant); $A = A_{exo} \wedge B$ (second variant)
- $VS = \neg A \vee B$

The causal graph for the first variant (Fig. 4a) is structurally equal to the one of the forest fire example (Fig. 2). For the second variant, in which the assassin only then puts the poison into this victim's coffee if the bodyguard does so with his antidote, we additionally have an edge from B to A in the corresponding causal graph (Fig. 4b).



(a) First Variant

(b) Second Variant

Fig. 4. Causal Graphs of Assassin Example (Source: Own Figure)

2.5 Railroad

In this example, [3] describes an engineer that operates a switch which makes an approaching train use the right-hand track if flipped and the left-hand track otherwise. Variable F is 1 if the switch is flipped and 0 if it is not. Two additional variables LB and RB model whether the left- and right-hand track, respectively, is blocked by either being set to 1 (blocked) or 0 (not blocked). The author specifies that the two tracks finally converge. That is, the train arrives at its original destination no matter which of the tracks it took provided the respective track was not blocked. This is captured by variable A, which is 1 if the train arrives and 0 otherwise. The corresponding equations are as follows:

- $F = F_{exo}$
- $LB = LB_{exo}$
- $RB = RB_{exo}$
- $A = \neg((F \wedge RB) \vee (\neg F \wedge LB))$

Figure 5 shows the causal graph. Unfortunately, [3] does not explicitly describe the equations; in particular not for A. Therefore, we assume that it has to be as denoted above: For A being 1 the engineer must flip or not flip the switch such that the train takes a non-blocked track provided that not both tracks are blocked. That is, it must not happen that the engineer flips the switch if the right-hand track is blocked or she does not flip it if the left-hand track is blocked.

Note that [3] describes additional variants of the railroad example. However, we do not consider them here as they are not described with the same degree of detail.

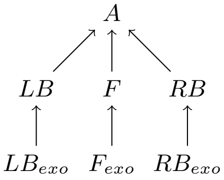


Fig. 5. Causal Graph of Railroad Example (Source: Own Figure)

2.6 Abstract Model 1 & 2

For these two models, we keep the example abstract and just provide variables and corresponding equations as well as the corresponding causal graphs (Figs. 6a & 6a).

Equations of Abstract Model 1:

- $A = A_{exo}$
- $B = B_{exo} \wedge \neg A$
- $C = A \vee B$
- $D = A$
- $E = \neg A$
- $G = \neg C$
- $H = \neg C \wedge \neg G$
- $I = C \vee D \vee E \vee G \vee H$

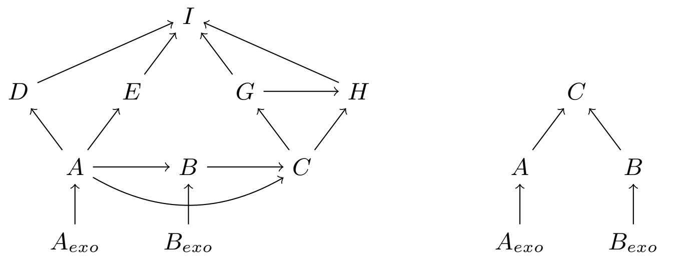
Equations of Abstract Model 2:

- $A = A_{exo}$
- $B = B_{exo}$
- $C = A \,\underline{\vee}$[1] $B = (A \wedge B) \vee (\neg A \wedge \neg B)$

2.7 Steal Master Key

The Steal Master Key example comes from an industrial partner and was originally represented as *attack tree* [7], which is shown in Fig. 8. Basically, it covers the steps an attacker may perform for stealing a master key within a specific system. In particular, we assume that there exist three potential persons U_1, U_2 and U_3 who are able to perform the attack. The corresponding causal graph is depicted in Fig. 7 and the following equations are part of the causal model:

[1] The operator $\underline{\vee}$ is called *XNOR* and denotes that $A \,\underline{\vee}\, B$ is true if both A and B are 1 or 0. Hence, XNOR is equivalent to the logical biconditional $\leftrightarrow$.



(a) Abstract Model 1

(b) Abstract Model 2

Fig. 6. Causal Graphs of Abstract Model 1 & 2 (Source: Own Figure)

- $FS_{U_i} = FS^{exo}_{U_i}$ ("From Script U_i")
- $FN_{U_i} = FN^{exo}_{U_i}$ ("From Network U_i")
- $FF_{U_i} = FN^{exo}_{U_i}$ ("From File U_i")
- $FDB_{U_i} = FN^{exo}_{U_i}$ ("From Database U_i")
- $A_{U_i} = A^{exo}_{U_i}$ ("Access U_i")
- $AD_{U_i} = AD^{exo}_{U_i}$ ("Attach Debugger U_i")
- $GP_{U_i} = FS_{U_i} \lor FN_{U_i}$ ("Get the Passphrase U_i")
- $GK_{U_i} = FF_{U_i} \lor FDB_{U_i}$ ("Get the Key U_i")
- $KMS_{U_i} = AU_i \land AD_{U_i}$ ("From Key Management Service U_i")
- $DK_{U_1} = GP_{U_1} \land GK_{U_1}$ ("Decrypt the Key U_1")
- $DK_{U_2} = GP_{U_2} \land GK_{U_2} \land \neg DK_{U_1}$ ("Decrypt the Key U_2")
- $DK_{U_3} = GP_{U_3} \land GK_{U_3} \land \neg DK_{U_1} \land \neg DK_{U_2}$ ("Decrypt the Key U_3")
- $SD_{U_1} = KMS_{U_1}$ ("Steal Decrypted U_1")
- $SD_{U_2} = KMS_{U_2} \land \neg SD_{U_1}$ ("Steal Decrypted U_2")
- $SD_{U_3} = KMS_{U_3} \land \neg SD_{U_1} \land \neg SD_{U_2}$ ("Steal Decrypted U_3")
- $DK = DK_{U_1} \lor DK_{U_2} \lor DK_{U_3}$ ("Decrypt the Key")
- $SD = SD_{U_1} \lor SD_{U_2} \lor SD_{U_3}$ ("Steal Decrypted")
- $SMK = DK \lor SD$ ("Steal Master Key")

for $i \in \{1, 2, 3\}$

All variables can obtain Boolean values only that denote whether the respective event occurred. As we can see, for stealing the master key (SMK), we need to decrypt it (DK) or steal it in decrypted form (SD). For doing so an attacker U_i needs to either get the passphrase (GP_{U_i}) and the key itself (GK_{U_i}) or get the decrypted key from the key management service (KMS_{U_i}). The passphrase can be obtained from a script (FS_{U_i}) or from the network FN_{U_i}, while the key may be extracted from a file (FF_{U_i}) or a database FDB_{U_i}. For stealing the decrypted key from the key management service, an attacker U_i needs to have access to it (A_{U_i}) and additionally attach a debugger (AD_{U_i}). Notice that we implicitly assume that attackers do not collaborate, i.e. the master key can only be stolen if one attacker alone performs all steps necessary. Additionally, U_1 preempts U_2 and U_3 and U_2 preempts U_3. That is, even if $U2$ or $U3$ were able to get the key and the passphrase or were able to obtain the key from the key management service, we say that their attack is only successful, if U_1 did not decrypt the key (DK_{U_1}) or steal the decrypted key (SD_{U_1}). Analogously for the preemption of U_2 towards U_3. Note that these preemption relationships are not modeled in the original attack tree in Fig. 8.

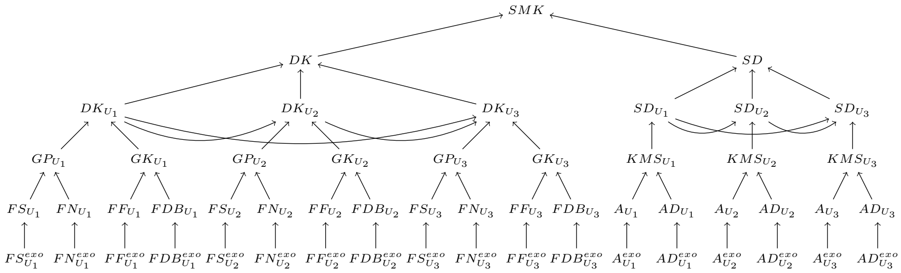


Fig. 7. Causal Graph of Steal Master Key Example (Source: Own Figure)

In addition to the standard model with three attackers, we created another variant of the Steal Master Key model with eight attackers $U_1, ..., U_8$. The general structure and semantics remain the same, but we now have 91 endogenous variables. Regarding the preemption relationships, U_1 now preempts $U_2, ..., U_8$, U_2 preempts $U_3, ..., U_8$ and so on and so forth.

2.8 Leakage in Subsea Production System

For this example, we use the *fault tree* [8] proposed by [1], in which the events that can lead to a leakage in an offshore pipeline system are modeled. We selected this example for several reasons. Firstly, the fault tree contains a relatively large amount of nodes (41 in total), i.e. the causal model is larger than some of the previous ones. Secondly, the fault tree is a real life example with semantics that are not artificially created. Hence, we can interpret causes and effects more intuitively. Thirdly, [1] did not only create the fault tree, but additionally compute its *minimal cut sets*. [8] define the latter as the “smallest combination of component failures which, if they all occur, will cause the top event to occur”. As we can see in Fig. 9, the top event of the current fault tree is the “leakage in an offshore pipeline system”. The minimal cut sets as specified by [1] are shown in Tab. 2. For instance, the two events “overpressure in well” and “failure of control in well” form such a set: If they occur, the top event occurs as well. Although this notion of causality differs from the counterfactual definition of causality used within this thesis, we can use those minimal cut sets as reasonable scenarios for the evaluation of this example. For instance, we expect that “failure of control in well” is a counterfactual cause of the top event under a context such that “overpressure in well” and “failure of control in well” are the

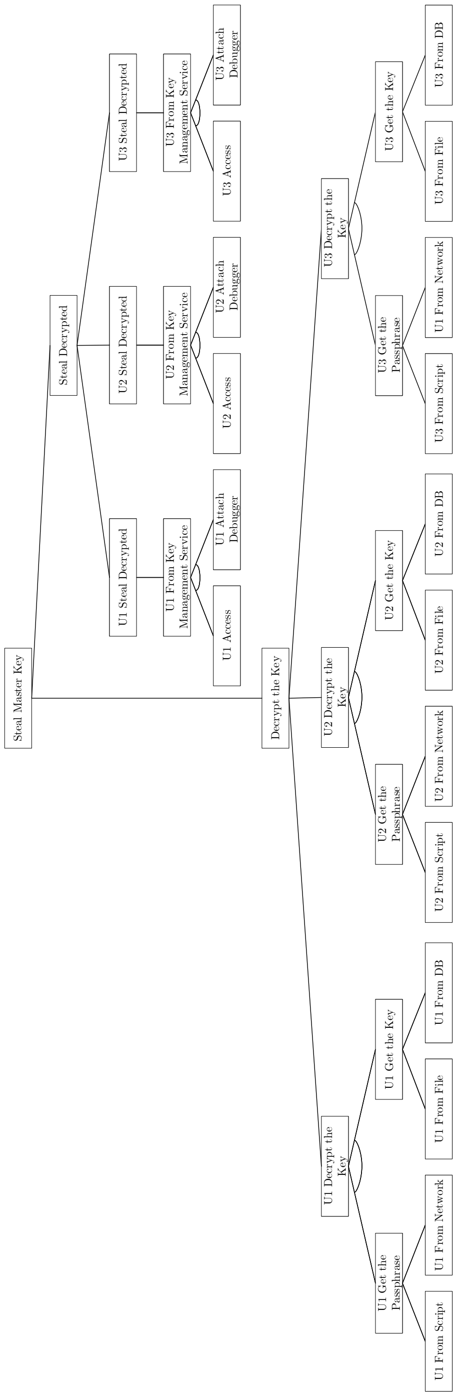


Fig. 8. Steal Master Key Attack Tree (Source: Industrial Partner)

MCS	**Events**	**MCS**	**Events**	**MCS**	**Events**
C_1	X_1, X_2	C_8	X_9, X_{11}	C_{14}	X_{16}, X_{17}
C_2	X_3, X_{11}	C_9	X_{10}, X_{11}	C_{15}	X_{18}, X_{19}
C_3	X_4, X_{11}	C_{10}	X_{12}, X_{17}	C_{16}	X_{20}, X_{21}
C_4	X_5, X_{11}	C_{11}	X_{13}, X_{17}	C_{17}	X_{22}, X_{23}
C_5	X_6, X_{11}	C_{12}	X_{14}, X_{17}	C_{18}	X_{24}, X_{25}
C_6	X_7, X_{11}	C_{13}	X_{15}, X_{17}	C_{19}	X_{26}
C_7	X_8, X_{11}				

Table 2. Minimal Cut Sets of the Fault Tree for Leakage in Subsea Production System (Source: [1])

only basic events, i.e. leaf events, that occur. As they form a minimal cut set, if "failure of control in well" does not occur anymore, the top event should not happen anymore as well. More details on the evaluated scenarios will be given below.

For the sake of readability, we denote each event within the fault tree from now on by X_i where i can be obtained from Fig. 9 (e.g. the top event is abbreviated by X_{41}). Since the corresponding causal graph would look exactly the same as the fault tree from a structure perspective, we skip the former here and define the equations only that we obtain when transforming this example into a causal model:

- for each basic event X_i with $i \in \{1, ..., 26\}$: $X_i = X_i^{exo}$
- $X_{27} = X_3 \vee X_4$
- $X_{28} = X_5 \vee X_6$
- $X_{29} = X_7 \vee X_8$
- $X_{30} = X_9 \vee X_{10}$
- $X_{31} = X_{12} \vee X_{13} \vee X_{14} \vee X_{15} \vee X_{16}$
- $X_{32} = X_{18} \wedge X_{19}$
- $X_{33} = X_{20} \wedge X_{21}$
- $X_{34} = X_{22} \wedge X_{23}$
- $X_{35} = X_{24} \wedge X_{25}$
- $X_{36} = X_{27} \vee X_{28} \vee X_{29} \vee X_{30}$
- $X_{37} = X_{31} \wedge X_{17}$
- $X_{38} = X_1 \wedge X_2$
- $X_{39} = X_{36} \wedge X_{11}$
- $X_{40} = X_{37} \vee X_{32} \vee X_{33} \vee X_{34} \vee X_{35}$
- $X_{41} = X_{38} \vee X_{39} \vee X_{40} \vee X_{26}$

We additionally created causal model which extends the Leakage example described in the above by some preemption relations: We say that "Leakage in pipe (X_{39})" preempts the events "Leakage in gas or oil well (X_{38})", "Leakage in key facilities (X_{40})" and "Third Party Damage (X_{26})". In other words, X_{38}, X_{40} and X_{26} can only occur, if X_{39} does not. Obviously, this example is made up. Nevertheless, we think it is reasonable to argue that if there is a leakage in

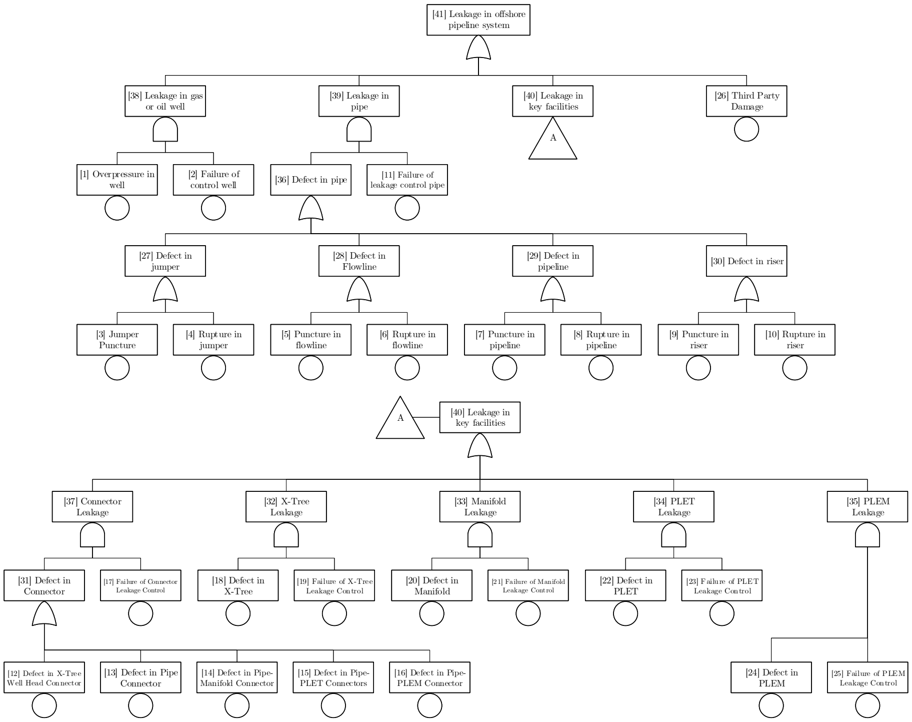

[41] Leakage in offshore pipeline system
[38] Leakage in gas or oil well
[39] Leakage in pipe
[40] Leakage in key facilities
A
[26] Third Party Damage
[1] Overpressure in well
[2] Failure of control well
[36] Defect in pipe
[11] Failure of leakage control pipe
[27] Defect in jumper
[28] Defect in Flowline
[29] Defect in pipeline
[30] Defect in riser
[3] Jumper Puncture
[4] Rupture in jumper
[5] Puncture in flowline
[6] Rupture in flowline
[7] Puncture in pipeline
[8] Rupture in pipeline
[9] Puncture in riser
[10] Rupture in riser
A
[40] Leakage in key facilities
[37] Connector Leakage
[32] X-Tree Leakage
[33] Manifold Leakage
[34] PLET Leakage
[35] PLEM Leakage
[31] Defect in Connector
[17] Failure of Connector Leakage Control
[18] Defect in X-Tree
[19] Failure of X-Tree Leakage Control
[20] Defect in Manifold
[21] Failure of Manifold Leakage Control
[22] Defect in PLET
[23] Failure of PLET Leakage Control
[12] Defect in X-Tree Well Head Connector
[13] Defect in Pipe Connector
[14] Defect in Pipe-Manifold Connector
[15] Defect in Pipe-PLET Connectors
[16] Defect in Pipe-PLEM Connector
[24] Defect in PLEM
[25] Failure of PLEM Leakage Control

Fig. 9. Fault Tree for Leakage in Subsea Production System (Source: [1])

the pipe of a pipeline system, then it is possibly counter-intuitive, if other events which might independently lead to the top event, are considered as (parts of the) cause as well. This is similar to the Rock-Throwing example where we say we want to be able to call Suzy's throw alone a cause for the bottle shattering even if Billy would have actually shattered the bottle when Suzy had not. Finally, we obtain the following new equations for the events preempted by X_{39}:

- $X_{26} = X_{26}^{exo} \wedge \neg X_{39}$
- $X_{38} = X_1 \wedge X_2 \wedge \neg X_{39}$
- $X_{40} = (X_{37} \vee X_{32} \vee X_{33} \vee X_{34} \vee X_{35}) \wedge \neg X_{39}$

2.9 Binary Tree

This example has the structure of a full binary tree and we specifically created it for measuring the efficiency of our approach. Such a model and various versions of it that exhibit a different height can be easily generated by a computer. For the sake of simplicity, we assume that the equation of each non-leave variable is defined as the disjunction of its two children. That is, the equation of n_1 in Fig. 10 would be given by $n_1 = n_3 \vee n_4$. Analogously for n_{root} and n_2. All other variables, i.e. the leaves, are defined by an exogenous variable. Since the number

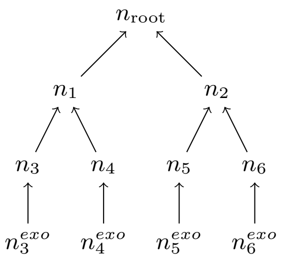


Fig. 10. Causal Graph of one Variant of the Binary Tree Example (Source: Own Figure)

of nodes in a full binary tree is $n = 2^h - 1$ with height $h \in \mathbb{N}$[2] [2], the generation of causal models with a very high number of nodes is simple, which makes it even more interesting for benchmarking.

2.10 Abstract Model 1 Combined with Binary Tree

The problem with a pure Binary Tree as causal model is that the semantics of the latter do not include preemption. Therefore, we combine two of our previous causal models, the Abstract Model 1 and a Binary Tree with $h = 12$. The causal graph in Fig. 11 illustrates how this combination works. Basically, we replace the equation of A of the Abstract Model 1, i.e. $A = A_{exo}$, with $A = n_{\text{root}}$. That is, we connect A with the root node of the Binary Tree model; all other semantics of these causal models remain unchanged.

2.11 Details of the Scenarios Presented in the Paper

Having introduced the models, we present the details of the scenarios presented in the results section of the paper. Table 3 shows the same scenarios but with details about $\vec{X}$ and φ showing in the third and the fourth of the column.

Table 3. Discussed scenarios as part of the analysis

	ID	$\vec{X} = \vec{x}$	φ	AC1	AC2	AC3	$\lvert\vec{W}\rvert$	BRUTE	SAT	SAT$^{\text{OPT}}$	SAT$^{\text{MIN}}$	SAT$^{\text{COM}}$
SMK_3	3	$FS_{U_1} = 1 \wedge FN_{U_1} = 1 \wedge \ A_{U_1} = 1$	$SMK = 1$	✓	✓	✓	4	N/A	0.9269ms 1.054MB	0.9761ms 1.139MB	1.080ms 1.218MB	0.7335ms 0.90132MB
	24	$FS_{U_3} = 1 \wedge FN_{U_3} = 1 \wedge \ A_{U_3} = 1$		✓	✓	✓	0	N/A	0.759ms 0.9122MB	0.7606ms 0.9745MB	0.7832ms 0.915MB	0.7867 ms 0.914MB
	29	$A_{U_3} = 1 \wedge AD_{U_3} = 1$	$SD = 1$	✓	✓	✗	0	0.1360ms 0.226MB	0.7517ms 0.9232MB	0.7194ms 0.93073MB	0.7298ms 0.9244MB	0.7675ms 0.9232MB
SMK_8	11	$FS_{U_1} = 1 \wedge FN_{U_1} = 1 \wedge \ A_{U_1} = 1$	$DK_{U_1} = 1 \vee SD_{U_1} = 1$	✓	✓	✓	0	N/A	11.77ms 8.011MB	2.410ms 2.741MB	11.33ms 7.988MB	12.07ms 8.017MB
LSP	3	$X_1 = 1 \wedge X_2 = 1$	$X_{41} = 1$	✓	✓	✗	0	0.1552ms 0.2519MB	0.8881ms 1.086MB	0.7893ms 1.052MB	0.8954ms 1.094MB	0.9440ms 1.093MB
BT_{11}	35	$n_{4091} = 1 \wedge n_{4092} = 1$ $\wedge\, n_{4093} = 1 \wedge n_{4094} = 1$	$n_{\text{root}} = 1$	✓	✗	✓	0	N/A	6949ms 2030MB	6717ms 2031MB	6618ms 2029MB	2940ms 1054MB
ABT	1	$n_{4093} = 1$	$I = 1$	✓	✓	✓	4	N/A	3071ms 1054.5MB	3043ms 1054.87MB	15545ms 4018.7MB	3131ms 1054.9MB
	4	$n_{4093} = 1 \wedge n_{4094} = 1$		✓	✓	✓	4	N/A	6379ms 2034.6MB	6940ms 2032.84MB	19043ms 4997MB	3073ms 1057MB
	5	$n_{4092} = 0 \wedge n_{4093} = 1 \wedge n_{4094} = 1$		✓	✓	✗	5	N/A	7906ms 2035.9MB	7933ms 2032.6MB	19271ms 4998MB	2926 1058MB

[2] A tree consisting of one node only has height 1. In Fig. 10, the binary tree out of which the causal graph was created has height 3.

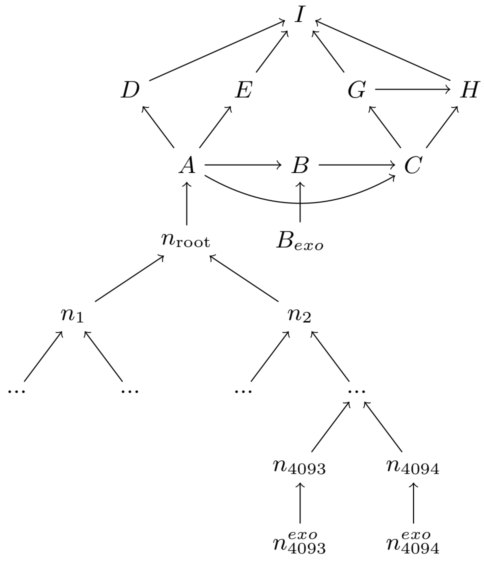


Fig. 11. Causal Graph of Abstract Causal Model 1 Combined with Binary Tree (Source: Own Figure)

References

1. Cheliyan, A.S., Bhattacharyya, S.K.: Fuzzy fault tree analysis of oil and gas leakage in subsea production systems. Journal of Ocean Engineering and Science **3**(1), 38 – 48 (2018). https://doi.org/https://doi.org/10.1016/j.joes.2017.11.005, `http://www.sciencedirect.com/science/article/pii/S2468013317300591`
2. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Second Edition. The MIT Press and McGraw-Hill Book Company (2001)
3. 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 2015, Buenos Aires, Argentina, July 25-31, 2015. pp. 3022–3033 (2015), `http://ijcai.org/Abstract/15/427`
4. Halpern, J.Y., Hitchcock, C.: Actual causation and the art of modeling. In: Causality, Probability, and Heuristics: A Tribute to Judea Pearl, pp. 383–406. London: College Publications (2010)
5. Halpern, J.Y., Pearl, J.: Causes and explanations: A structural-model approach - part I: causes. In: UAI '01: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, University of Washington, Seattle, Washington, USA, August 2-5, 2001. pp. 194–202 (2001), `https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=100&proceeding_id=17`
6. Halpern, J.Y., Pearl, J.: Causes and explanations: A structural-model approach. part i: Causes. The British Journal for the Philosophy of Science **56**(4), 843–887 (2005). https://doi.org/10.1093/bjps/axi147, `http://dx.doi.org/10.1093/bjps/axi147`
7. Schneier, B.: Attack trees. Dr. Dobb's journal **24**(12), 21–29 (1999)
8. Vesely, W., Goldberg, F., Roberts, N., Haasl, D.: Fault tree handbook (1981)