From Checking to Inference: Actual Causality Computations as Optimization Problems- 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 37 different causal models. On the one hand, we took all those models from [6] 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	[6, 9]	5
Forest Fire (conjunctive & disjunctive)	[6, 9]	3
Prisoners	[6, 9]	4
Assassin (first & second variant)	[6]	3
Railroad	[6]	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	[3]	41
Leakage in Subsea Production System with Preemption	based on [3]	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 [**?**, 6, 9]. 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:

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

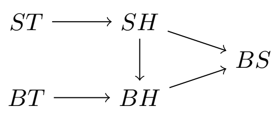


Fig. 1. Rock-throwing example (Source: [**?**])

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)

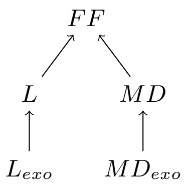


Fig. 2. Causal Graph of Forest Fire Example (Source: [9])

2.3 Prisoners

An additional example found in [9] and [6] 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:

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

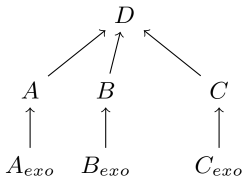


Fig. 3. Causal Graph of Prisoners Example (Source: Own Figure)

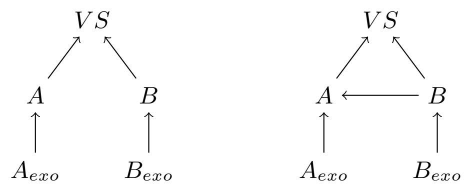
2.4 Assassin

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 [6], 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 [6] does not explicitly mention the variables within this example, we use the same ones introduced by [8], 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 [8]: 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, [6] 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, [6] 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 [6] 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.

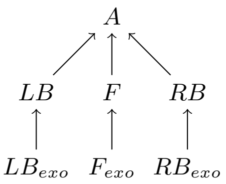


Fig. 5. Causal Graph of Railroad Example (Source: Own Figure)

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:

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

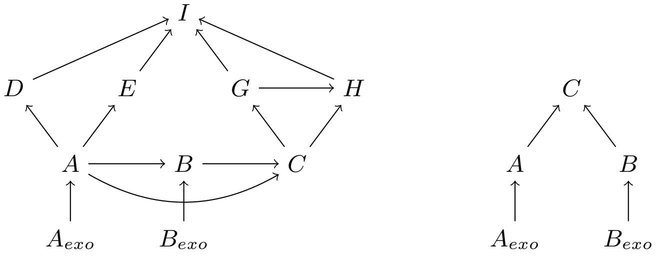
Equations of Abstract Model 2:

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

2.7 Leakage in Subsea Production System

For this example, we use the *fault tree* [19] proposed by [3], 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

[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)

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, [3] did not only create the fault tree, but additionally compute its *minimal cut sets*. [19] 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. 7, the top event of the current fault tree is the “leakage in an offshore pipeline system”. The minimal cut sets as specified by [3] 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 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.

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: [3])

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. 7 (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 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}:

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

2.8 Ueberlingen mid-air Collision

On the night of the first of July 2002, two aircraft collided near the German town of Ueberlingen.[2] The collision happened between a Tupolev Tu154M passenger

[2] https://en.wikipedia.org/wiki/2002_Uberlingen_mid-air_collision

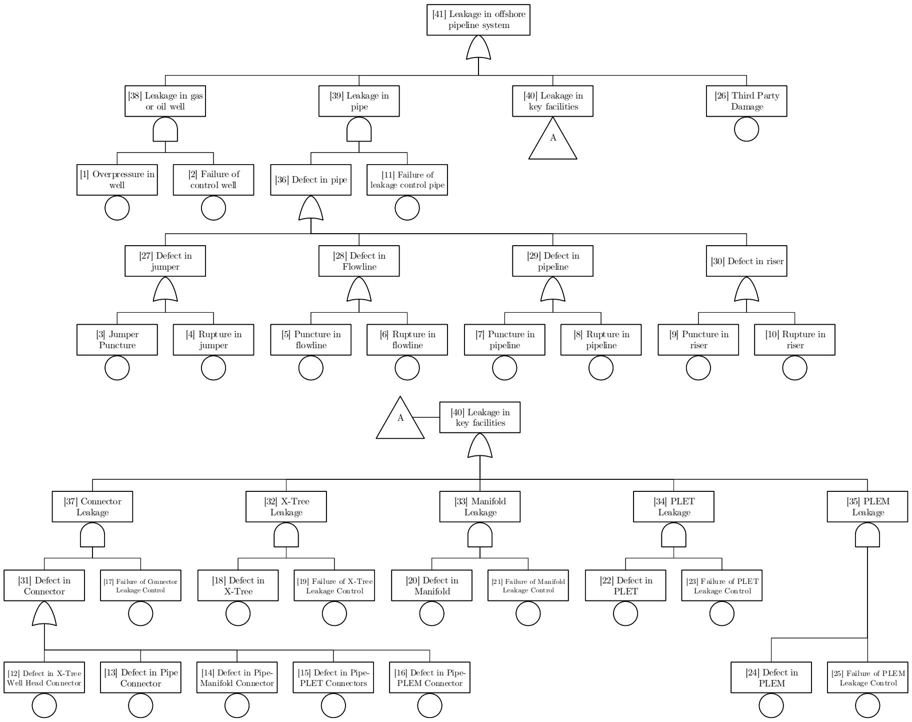

[41] Leakage in offshore pipeline system
[38] Leakage in gas or oil well
[39] Leakage in pipe
[40] Leakage in key facilities
A
[26] Third Party Damage
[1] Overpressure in well
[2] Failure of control well
[36] Defect in pipe
[11] Failure of leakage control pipe
[27] Defect in jumper
[28] Defect in Flowline
[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. 7. Fault Tree for Leakage in Subsea Production System (Source: [3])

jet (Bashkirian Airlines Flight 2937 from Moscow to Barcelona), and a Boeing 757-200 cargo jet (DHL Flight 611 from Bergamo to Brussels). A total of 71 passengers and crew members were killed in the accident [1]. The context of the collision comprised many confusing factors. We summarize some of them; a complete list (sources: [1, 16], and [17] (German)) of translated factors is shown in Table 3. The area of the accident is under the control of Zurich Air Traffic Controller (ATC), who is in charge of keeping the routes clear. An ATC is equipped with a radar and a system: short term collision avoidance (STCA), that alerts visually and aurally about 2-3 mins before a collision [14]. Also, both aircraft were equipped with traffic alert and collision avoidance systems (TCAS). TCAS warns the crew in aircraft in under 50 seconds before a collision, with a complementing resolution advisory commands (RA) to either climb or descends [14,16]. According to the regulations, the ATC is responsible for keeping aircraft separated; TCAS is an additional system only.

The accident day, a maintenance operation at Zurich ATC office deactivated STCA, ran the radar in degraded mode, and caused the telephone system (used to communicate with nearby airports and other ATCs) to run in fallback mode. Given the low density of flights at night, these limitations were approved. The ATC at Karlsruhe (Germany) was also monitoring the path of the collision with a fully operational radar. Due to a problem with the telephone system, ATC at Karlsruhe could not communicate with Zurich [14]. Additionally, ATC at Zurich spent around 5 minutes guiding an unexpected late flight to airport Friedrichshafen (Germany). The ATC was working alone that night because the operating company tolerated the case of taking long breaks at night [1].

Still, a collision can be avoided with all these systems installed. The ATC noticed the potential collision on the radar and instructed the Tu154M to descend flight level at 21:34:49; the Tu154M descends (21:34:56) exactly when TCAS generates an RA to the Tu154M to climb and to the B757 to descend. Then, the B757 descends (21:34:58) also, and the Tu154M crew discusses the contradictory commands (ATC to descend and TCAS to climb). The confusion is complicated by the fact that the Russian and the European procedures are not standardized in such a situation. Nineteen seconds before the collision, the ATC repeats descent advisory to the Tu154M, and wrongly advises the crew of traffic at "2 o'clock" while the DHL is at 10 o'clock. It is not clear whether the wrong location would have changed anything or not. The aircraft collided at 21:35:32. The original story has more factors that we omitted for simplicity.

With all these human, technical, and organizational factors, it is hard to draw conclusions. The official investigation by the German Federal Bureau of Aircraft Accident Investigation (BFU) was issued 2 years after the accident [1]. It concluded that both, the ATC (late intervention) and the TU154M crew (followed the ATC instruction contrary to the TCAS RA), made a series of mistakes that are considered as immediate causes, but the primary systematic cause is the negligence by the Air Traffic Control company of Switzerland. In 2005, researchers [3] conducted a Why-Because-Analysis of the accident and presented a model that contained 95 factors [18] which used in our work.

The model The Why-Because-Analysis (WBA) process [12–14] is a systematic procedure to organize facts related to an accident. The process results in a graphical understanding, called Why-Because-Graph WBG, of all the related facts and their causal relations. We can use WBG with some adaptions as a causal model. The complete causal model can be inspected in Fig.8. In the following, we use e_i to refer to the event with ID=(i) in the table. We consider each node in the WBG to be an endogenous variable. For each leaf in the graph, we create an exogenous variable that sets its value. The WBG embeds the context because it is created based on facts, thus the exogenous variables are always true [12]. To come up with the equations that describe each variable, we manually inspected each one to decide on its equation. In disjunctive equation (events that disjunctively depended on other events), preemption relations are crucial to infer actual causality [11]. These relations are especially important in contexts where events coincide. Some preemption relations express temporal order of events, but others may reflect a discrepancy of the causal importance among events. For example, in the rock-throwing example, the fact that Suzy threw the rock slightly earlier is modeled using a preemption relation. We use the same concept to edit the WBG of to add preemption relations among the

[3] `https://rvs-bi.de/research/WBA/`

events leading to e_6: *Conflict resolution failed*. This discrepancy merely reflects the fact that keeping the routes clear for aircraft is the mission of the ATC [1]. Accordingly, the TCAS is a last resort that should resolve last-minute situations, and hence, causally, a failure by the ATC *preempts* a failure by the TCAS.

Understandably, a large part of the WBG focused on the factors of the late intervention of the ATC (e_{49}). Five direct factors coincided and led to the state of late intervention, namely: e_{52} *Control strips do not warn of crossing routes on the radar*, e_{53} *No visual warning from STCA*, e_{55} *21:35:00 Acoustic STCA signal was not detected in control room*, e_{56} *Heavy load on the ATC*, and e_{62} *Crossing routes*. Each factor can be thought of as a sufficient cause of e_{49}; however, when thinking of actual causes, people tend to consider *exceptional* events to be the probable causes [7]. Then, we argue that the exceptional heavy load on the ATC (due to another a late landing in a nearby airport, and the faulty phone system) is more influential than the normal technical problems with STCA system (e_{52}, e_{53}, and e_{55}). Also, having a potential route crossing (e_{68} B757 3 minutes past expected time, e_{69} Tu154 2 minutes ahead of expected time) is plausible in aviation. Accordingly, we added preemption relations among these events. Our sole aim from these steps is to show that a causal modeling methodology, like WBA, quickly yields comparatively large models. In this paper, we focus on the technical part of efficiently checking causality using such models. In some cases, the steps above seem as biased or forced, but they explicitly represent the investigators knowledge as documented in their report [1].

Results and Findings With the knowledge of causal factors made explicit using a model, we now can use the actual causality definition to check for cause(s). It is worth noting that, at least according to HP, multiple causes of an effect are possible [7]. Other concepts can then be used to compare such causes, like responsibility [4] and normality [7]. As a first causal check, we used a simple causal model that expresses all the relations as disjunctions, i.e., assumes that any factor is enough to cause the other side of the connection. Especially in confusing situations (in which events have coincided), such a model is not conclusive or over-determined. This check replaces the manual verification step of the Why-Because analysis [13], in which the model is checked against a sufficiency test to verify that the effect eventually happens, given that all the root causes (graphs leaves) occurred. We performed this check: Q_1*: Is* $\vec{X}$ *a cause of* e_5 *(collision)?*, where $\vec{X}$ is the set of 31 leaf events. The check passed the three HP conditions. This check shows that HP can be a part of the WBA methodology.

The interesting checks were performed on the *edited* model (with preemption and logical combination). The first check was the same as Q_1 (effect is the collision, and the cause is a set of 31 root causes). The result was a violation of AC3, i.e., the cause is not minimal. A minimal cause of 14 variables was returned by our solver. These are the details that resulted in the late intervention of ATC. This actual cause conforms with the immediate cause reported by the BFU [1]. However, this check is fine-grained. For example, one of the root causes in the check is e_{85} *attempt to retrieve timetable*, which is assumed to have delayed the take-off of the DHL flight. Thus, we checked the actual causality to find a

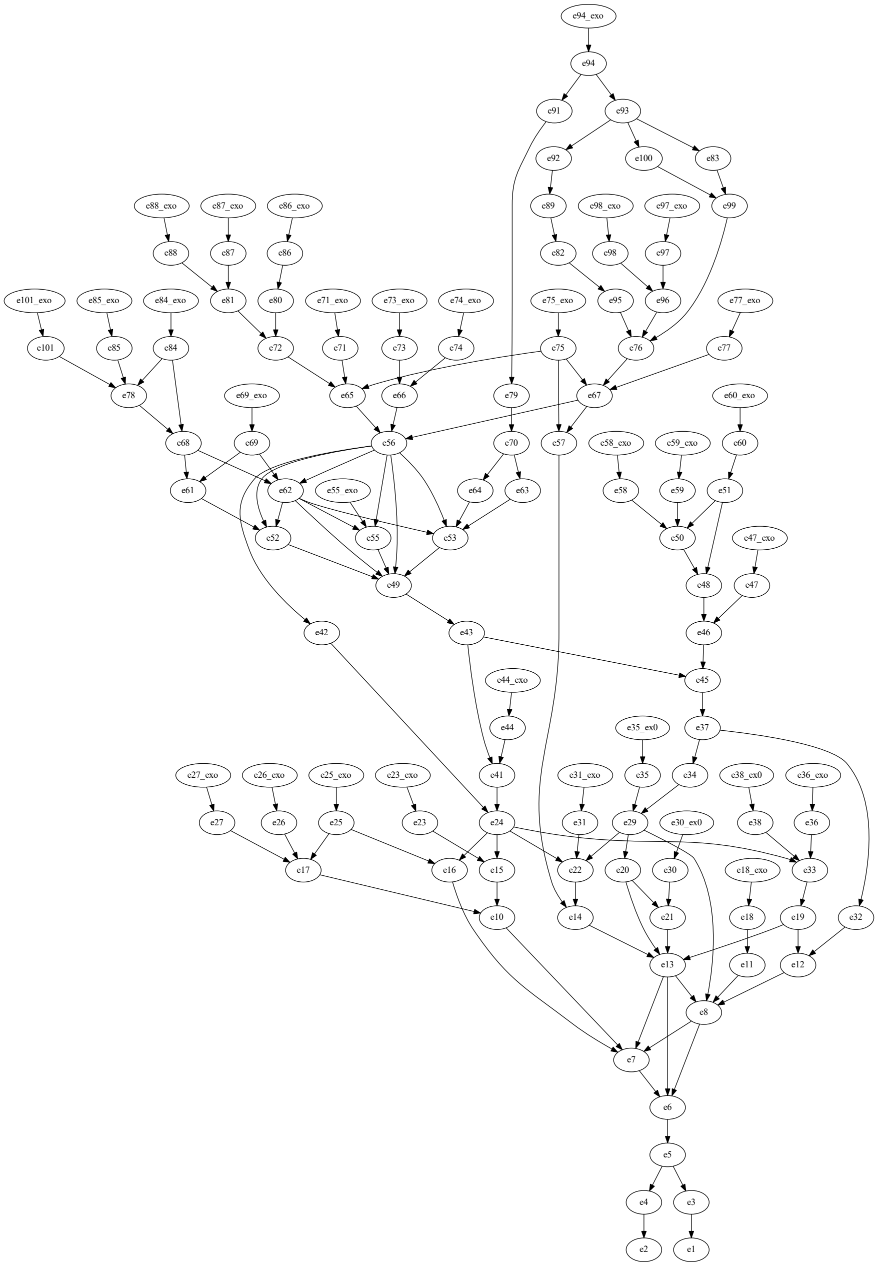


Fig. 8. Causal Network of the Ueberlingen accident [18]; events are described in Table 3

minimal cause on a coarse-level. The question this time is Q_2*: Is* $\{e_{13}, e_{70}, e_{74}\}$ *a cause of* e_5 *(collision)?*; the set of causes are chosen arbitrarily to represent different levels of granularity about the ATC where e_{13} *Conflict resolution by ATC failed*, e_{70} *MV9800 computer is not available*, e_{74} *ATC was not aware that he had an assistant at his disposal*. The result was that this is not a minimal cause and the minimal one was only e_{13}. Thus, at a coarse level, we conclude the failure of the ATC as an actual cause; this can be further explained into detailed events as we saw in Q_1. Please note that we omitted the $\vec{W}$ set for simplicity's sake in this description. Similarly, we conducted checks focusing on an intermediate event e_{49}: *Air traffic controller detects crossing routes late*, as an effect. On the fine-grained level, we found an actual cause comprised of 11 events (root causes) that conformed with the BFU systematic causes of the accident.

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. 9 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 of nodes in a full binary tree is $n = 2^h - 1$ with height $h \in \mathbb{N}$[4] [5], the generation of causal models with a very high number of nodes is simple, which makes it even more interesting for benchmarking.

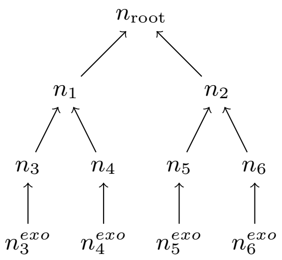


Fig. 9. Causal Graph of one Variant of the Binary Tree Example (Source: Own Figure)

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

[4] A tree consisting of one node only has height 1. In Fig. 9, the binary tree out of which the causal graph was created has height 3.

causal models, the Abstract Model 1 and a Binary Tree with $h = 12$. The causal graph in Fig. 10 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.

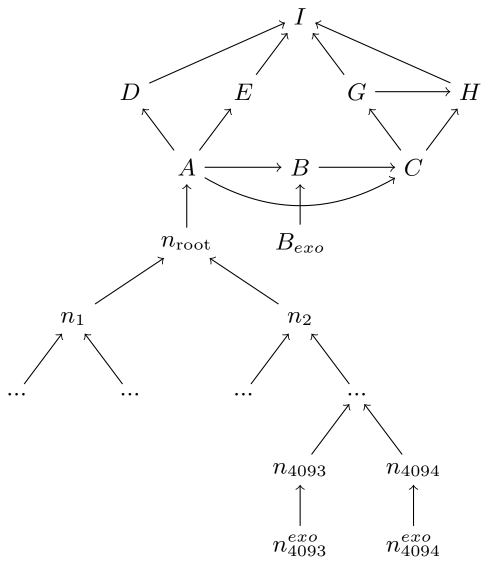


Fig. 10. Causal Graph of Abstract Causal Model 1 Combined with Binary Tree (Source: Own Figure)

2.11 Steal Master Key

The Steal Master Key example comes from an industrial partner and was originally represented as *attack tree* [15], which is shown in Fig. 12. Basically, it covers the steps an insider 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. 11 and the following equations are part of the causal model:

- $FS_{U_i} = FS^{exo}_{U_i}$ (“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”)
- $GP_{U_i} = FS_{U_i} \vee FN_{U_i}$ (“Get the Passphrase U_i”)

- $GK_{U_i} = FF_{U_i} \vee FDB_{U_i}$ (“Get the Key U_i”)
- $KMS_{U_i} = AU_i \wedge AD_{U_i}$ (“From Key Management Service U_i”)
- $DK_{U_1} = GP_{U_1} \wedge GK_{U_1}$ (“Decrypt the Key U_1”)
- $DK_{U_2} = GP_{U_2} \wedge GK_{U_2} \wedge \neg DK_{U_1}$ (“Decrypt the Key U_2”)
- $DK_{U_3} = GP_{U_3} \wedge GK_{U_3} \wedge \neg DK_{U_1} \wedge \neg DK_{U_2}$ (“Decrypt the Key U_3”)
- $SD_{U_1} = KMS_{U_1}$ (“Steal Decrypted U_1”)
- $SD_{U_2} = KMS_{U_2} \wedge \neg SD_{U_1}$ (“Steal Decrypted U_2”)
- $SD_{U_3} = KMS_{U_3} \wedge \neg SD_{U_1} \wedge \neg SD_{U_2}$ (“Steal Decrypted U_3”)
- $DK = DK_{U_1} \vee DK_{U_2} \vee DK_{U_3}$ (“Decrypt the Key”)
- $SD = SD_{U_1} \vee SD_{U_2} \vee SD_{U_3}$ (“Steal Decrypted”)
- $SMK = DK \vee SD$ (“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. 12.

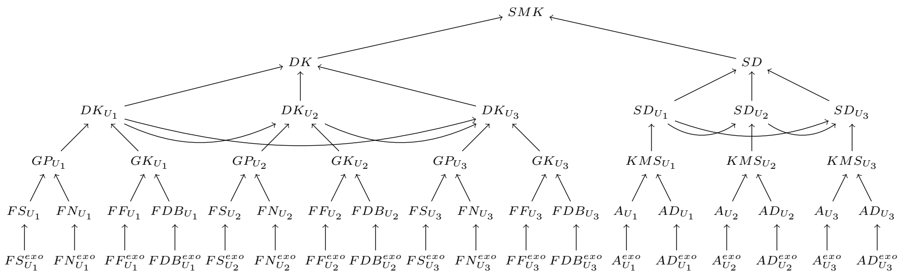


Fig. 11. Causal Graph of Steal Master Key Example (Source: Own Figure)

In addition to the standard model with three insiders, we created another 13 other variant of the Steal Master Key model with varying number of users ranging from 50 to 650 (based on the employees of the partner). The general structure and semantics remain the same, but we now have models of $11 \times n$ endogenous variables (and $10 * n$ exogenous variables), where n is the number of employees. 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.

2.12 Samples of the Scenarios Presented in the Paper

We show representative scenarios from our evaluation, shown in Table 4. Each query is identified by a *model name*, its size, and an *ID*; those are shown in the first three columns. For the query, we show, in the fourth column, the size of the cause $|\vec{X}|$. The results are displayed in columns AC1-AC3. The sizes of $|\vec{W}|$, and the minimal cause $|\vec{X}_{min}|$ are displayed in the eighth and ninth columns. For each approach, the execution time, in seconds (s), and the memory allocation, in gigabytes (GB), are shown.

References

1. of Aircraft Accident Investigation, G.F.B.: Investigation report ax001-1-2/02 (2004), https://www.bfu-web.de/EN/Publications/Investigation%20Report/2002/Report_02_AX001-1-2_Ueberlingen_Report.html
2. 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)
3. 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 (2018)
4. Chockler, H., Halpern, J.Y.: Responsibility and blame: A structural-model approach. J. Artif. Intell. Res. **22**, 93–115 (2004). https://doi.org/10.1613/jair.1391
5. 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)
6. 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)
7. Halpern, J.Y.: Actual causality. The MIT Press, Cambridge, Massachussetts (2016)
8. 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)
9. 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
10. Hopkins, M.: Strategies for determining causes of events. In: AAAI/IAAI (2002)
11. Ibrahim, A., Kacianka, S., Pretschner, A., Hartsell, C., Karsai, G.: Practical causal models for cyber-physical systems. In: NASA Formal Methods. pp. 211–227 (2019)
12. Ladkin, P., Loer, K.: Why-because analysis: Formal reasoning about incidents. Bielefeld, Germany, Document RVS-Bk-98-01, Technischen Fakultat der Universitat Bielefeld, Germany (1998)

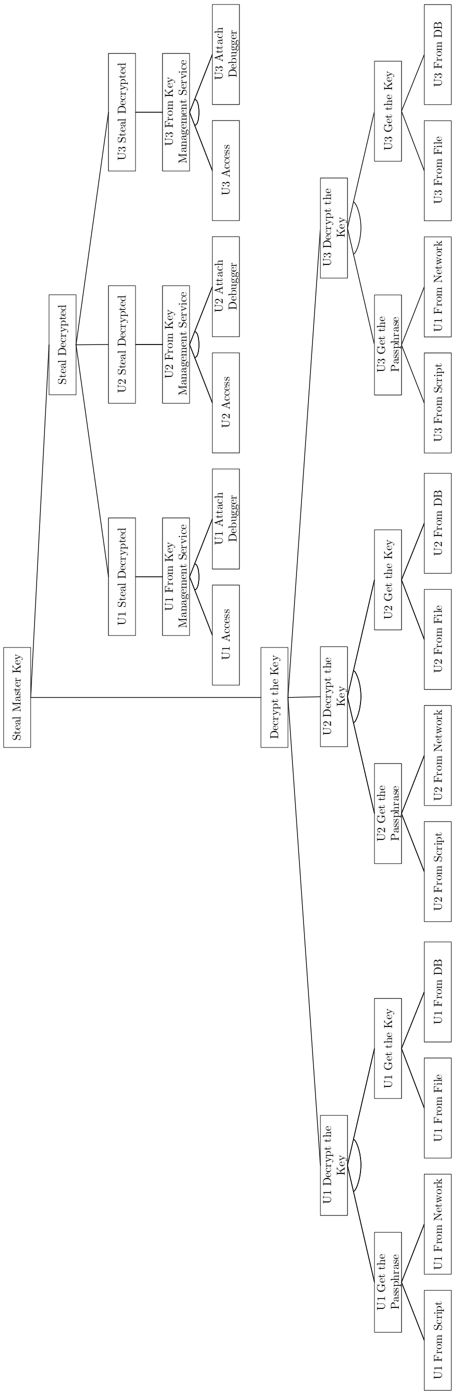


Fig. 12. Steal Master Key Attack Tree (Source: Industrial Partner)

13. Ladkin, P.B.: Causal reasoning about aircraft accidents. In: International Conference on Computer Safety, Reliability, and Security. pp. 344–360. Springer (2000)
14. Sanders, D.I.J.: Introduction to why-because analysis. Dipl.-Inform, February (2012)
15. Schneier, B.: Attack trees. Dr. Dobb's journal **24**(12), 21–29 (1999)
16. Stuphor, J.: Handout of the 2002 ueberlingen mid-air, `https://rvs-bi.de/Bieleschweig/5.5/Stuphorn_Ueberlingen_handout.pdf`
17. Stuphor, J.: Kausale untersuchung der kollision zweier verkehrsflugzeuge über dem bodensee, 1. juli 2002, `https://rvs-bi.de/Bieleschweig/5.5/Stuphorn_Ueberlingen_ListOfFacts.pdf`
18. Stuphor, J.: The wbg of 2002 ueberlingen mid-air, `https://rvs-bi.de/Bieleschweig/5.5/Stuphorn_Ueberlingen_WBA.pdf`
19. Vesely, W., Goldberg, F., Roberts, N., Haasl, D.: Fault tree handbook (1981)

ID	Description	ID	Description
(2)	Crash B757	(63)	optical STCA not active
(4)	B757 vertical tail destroyed	(64)	no correlation of flight plan data and radar data
(1)	Crash Tu154M	(70)	MV9800 computer is not available
(3)	Fuselage Tu154 severed	(79)	Radar system in fallback mode
(5)	Collision	(91)	System work in the ADAPT system
(6)	Conflict resolution failed	(41)	ATC must restore separation
(13)	Conflict resolution by air traffic controllers failed	(44)	The role of the air traffic controller to ensure separation
(7)	Resolution of conflict by crews failed	(35)	TCAS training DHL
(8)	Conflict resolution by TCAS failed	(22)	B757 TCAS RA only reports 23 sec after RA
(12)	Tu154 controls against TCAS RA	(31)	Only one radio channel for Tu154 and B757
(29)	B757 complies with TCAS RA	(57)	Radio messages from AeroLloyd 1135 to ACC Zurich
(11)	TCAS does not reverse RA	(75)	Land approach from AeroLloyd 1135 to Friedrichshafen
(18)	Avoidance of unnecessary RAs	(71)	Technical limitations of the workstation in relation to radio
(14)	ATC not responding to B757 radio message	(77)	Handover procedure to Friedrichshafen telephone based
(21)	ATC not responding to B757 manoeuvre	(76)	Pilot does not successfully use any of three telephone systems
(19)	Avoidance maneuver of the Tu by sinking	(61)	Requirements for warning of crossing routes not met
(20)	Evasive maneuver of the B757 by sink	(87)	Night staffing
(30)	Radar system does not display transponder S information	(85)	Assumption: attempt to retrieve timetable
(33)	PIC encourages PF to descend	(101)	ATC approves deviation from course
(32)	TCAS Tu154: climb	(37)	TCAS detects danger of collision
(62)	crossing routes	(45)	Criteria for TCAS RA fulfilled
(69)	Tu154 2 minutes ahead of expected time (control strip)	(43)	Aircraft below staggering
(68)	B757 3 minutes past expected time (control strip)	(46)	Further approximation of aircraft
(84)	delayed departure	(47)	B757 follows original flight path
(78)	Deviation from planned exchange rate	(48)	Tu154 maintains rough flight direction
(38)	Training PIC BTC	(50)	approx. 21:35 Tu154 starts right turn
(36)	Decision-making of the PIC	(51)	approx. 21:33 Tu154 changes course to left
(34)	TCAS B757: descending	(59)	Assumption: Crew mainly concerned with conflict situation
(72)	Understaffing in the ACC, only one instead of 4 controllers	(58)	Autopilot switched off
(80)	Rest of the second controller	(60)	unknown cause
(81)	2 controllers in control room instead of 4 during day shift	(10)	Visual identification of conflict traffic does not solve conflict
(94)	Sectorisation work	(16)	B757 does not understand the conflict traffic message.
(92)	Replacement telephone had to be switched	(15)	contradictory conflict information in the Tu154
(93)	Changeover to telephone system	(17)	B757 does not recognize flight maneuvers of conflict traffic
(82)	Replacement telephone of ATC not in working order	(27)	Perceived size of conflict traffic
(89)	faulty switching of the replacement telephone	(26)	Night flight affects visibility and accuracy
(83)	no release of the service telephone after conversion	(25)	B757 visually identifies conflict traffic at 2am
(73)	ATC had a System manager at their disposal as an assistant	(23)	Tu154 visually identifies conflict traffic at 10am
(74)	ATC was not aware that he had an assistant at his disposal.	(24)	ATC: ”rapidly sink to FL350, conflict traffic 2 o’clock.”
(86)	Common practice in ACC Zurich	(88)	Usually low traffic during the night
(49)	ATC detects crossing routes too late	(42)	Misinterpretation of the radar image by air traffic controllers
(56)	Heavy load on the air traffic controller	(95)	Bypass system cannot be used
(53)	No visual warning from implement	(96)	Mobile phone not used
(55)	acoustic STCA signal was not detected in control room	(99)	Telephone Switch-02 is not used
(52)	Control strips do not warn of crossing routes	(100)	Telephone Switch-02 not operational
(66)	ATC does not request support from System manager	(98)	Emergency manual lists 3 telephone systems available
(65)	ATC checks at two workplaces	(97)	ATC not aware of mobile phone availability
(67)	Transfer of landing approach from AeroLloyd 1135 to Friedrichshafen not successful		

Table 3. Accident’s Facts; Originally in German [18]

				Result					Execution Time (s)				Memory consumption (GB)			
Model	$\lvert\vec{V}\rvert$	ID	$\lvert\vec{X}\rvert$	AC1	AC2	AC3	$\lvert\vec{W}\rvert$	$\lvert\vec{X}_{min}\rvert$	SAT	ILP	MAX_SAT	ILP_{why}	SAT	ILP	MAX_SAT	ILP_{why}
SMK	91	11	3	Y	Y	Y	0	3	0.013	0.0024	0.002	0.062	0.008	0.0026	0.002	0.017
Ueberlingen	95	5	4	Y	Y	N	88	3	0.805	0.488	0.005	—	0.0018	0.0040	0.0021	—
BT_11	4095	35	4	Y	N	Y	N/A	4	7.37	3.24	0.22	8.56	2.03	1.05	0.07	1.79
ABT	4103	4	2	Y	Y	Y	4086	2	8.40	4.54	1.5	11.66	2.04	1.05	0.082	6.5
		5	5	Y	Y	N	4090	2	9.41	3.88	1.4	8.2	2.04	1.05	0.09	6.5
		6	10	Y	Y	N	4086	2	16.77	5.11	1.32	—	4.2	1.05	0.082	—
		7	11	N	Y	N	4086	2	26.29	5.26	1.35	—	4.18	1.05	0.082	—
		8	15	Y	Y	N	4086	2	N/A	5.108	1.37	—	N/A	1.05	0.082	—
		10	15	N	Y	N	4080	5	7301	5.17	1.35	—	9.5	1.05	0.09	7.8
		11	50	Y	Y	N	4079	5	N/A	4.80	1.44	—	N/A	1.05	0.082	—
ABT2	8207	1	11	Y	Y	Y	8161	11	N/A	22.8	5.67	58	N/A	4.0	0.22	13.88
		2	22	Y	Y	N	8191	11	N/A	24.8	6.67	63	N/A	4.0	0.16	13.88

Table 4. Discussed scenarios as part of the analysis