Contents lists available at ScienceDirect

Safety Science

journal homepage: www.elsevier.com/locate/safety

A probabilistic model of belief in safety cases

Damir Nešić[a,*], Mattias Nyberg[a], Barbara Gallina[b]

[a] KTH Royal Institute of Technology, Brinellvägen 83, 100 44 Stockholm, Sweden
[b] Mälardalen University, Högskoleplan 1, 722 20 Västerås, Sweden

ARTICLE INFO

Keywords:
Safety case
Safety-case representation
Reasoning under uncertainty
Model Theory
Bayesian Network

ABSTRACT

A safety case is a hierarchical argument supported by evidence, whose scope is defined by contextual information. The goal is to show that the conclusion of such argument, typically "the system is acceptably safe", is true. However, because the knowledge about systems is always imperfect, the value true cannot be assigned with absolute certainty. Instead, researchers have proposed to assess the belief that a conclusion is true, which should be high for a safe system. Existing methods for belief calculations were shown to suffer from various limitations that lead to unrealistic belief values. This paper presents a novel method, underlined by formal definitions of concepts such as conclusion being true, or context defining the scope. Given these definitions, a general, probabilistic model for the calculation of belief in a conclusion of an arbitrary argument is derived. Because the derived probabilistic model is independent of any safety-case notation, the elements of a commonly used notation are mapped to the formal definitions, and the corresponding probabilistic model is represented as a Bayesian Network to enable large-scale calculations. Finally, the method is applied to scenarios where previous methods produce unrealistic values, and it is shown that the presented method produces belief values as expected.

1. Introduction

> *Belief* - "Conviction of the truth of some statement [...] especially when based on examination of evidence."
>
> www.merriam-webster.com

Safety cases are being increasingly adopted to express the arguments about why a system is acceptably safe to operate in a particular environment. From their early days in high-risk industries in the United Kingdom, e.g. oil and gas (UK GoV, 1992), and railway (UK GoV, 1994), safety cases became the recommendation of international standards in domains such as railway (Anon, 2003), automotive (International Organization for Standardization, 2018), and others. The culmination of this trend is reflected in the recent, first-ever standard for *safety of autonomous systems* (UL4600 Task Group, 2020), where a *safety case* is the *central artifact* throughout the safety lifecycle.

The core of a safety case is a *hierarchy of arguments*. Each argument consists of *claims*, where one is designated as the *conclusion*, and the remaining ones as *premises*. The premises are in turn conclusions of other arguments, thus forming the hierarchy. The claims at the bottom of the hierarchy are *supported by evidence* whose purpose is to show that these claims are *true*. The idea is that if the conclusion of each argument *follows* from the corresponding premises, and if the claims at the bottom of the hierarchy are *true*, then the conclusion of the top argument will also be *true*. This is the purpose of a safety case, to show that the claim at the top of the hierarchy is *true*. Typically, this claim is *"the system is acceptably safe"* and to define the scope in which the truthfulness of such claim is evaluated, safety cases contain *contextual information* (Origin Consulting (York) Limited, 2018).

It is nowadays widely acknowledged that regardless of the exact meaning of terms *supported by, true, follows* and *context*, showing that a claim is true *with absolute certainty* is not possible (Duan et al., 2017). The reason is the *inherent uncertainty*, both about whether a conclusion follows from the premises, and about whether an evidence supports a claim. For example, it cannot be said in general that the claim *"software component implements a requirement"* is true given the evidence of *"successful testing"* because testing is not an exhaustive verification technique, the testing tool–set might produce false positives, the result might be interpreted incorrectly by humans etc. Also, unless the typically natural-language conclusions and premises can be formalized and then proven that the premises imply the conclusion, in general, it cannot be said that a conclusion follows from the premises.

Because generally it is not possible to decide if a safety-case claim is true or false, a number of researchers have proposed to quantify the

[*] Corresponding author.
E-mail addresses: damirn@kth.se (D. Nešić), matny@kth.se (M. Nyberg), barbara.gallina@mdh.se (B. Gallina).

https://doi.org/10.1016/j.ssci.2021.105187
Received 16 July 2020; Received in revised form 29 December 2020; Accepted 17 January 2021

belief, also called *confidence*, that safety-case claims are true (Cyra and Górski, 2011; Wang et al., 2019, 2018; Denney et al., 2011; Zhao et al., 2012; Bishop et al., 2011; Guiochet et al., 2015; Ayoub et al., 2013; Hobbs and Lloyd, 2012). The idea is that if the calculated *belief* is higher than a predefined threshold, then the claim can be *considered to be true*. The intuition is that belief calculation corresponds to the process of *safety-case assessment* (Kelly, 2007) in which an assessor analyses safety-case arguments, and decides whether to accept them, or to require additional information to increase own belief in the conclusions of these arguments.

Despite a significant number of methods for belief calculations, a recent survey (Duan et al., 2017) has identified a number of open questions that hinder wider adoption. Moreover, a recent comprehensive replication study (Graydon and Holloway, 2016, 2017) has analyzed twelve methods by reproducing the considered safety cases, and the corresponding belief calculations. Then, the considered safety-cases were subjected to various modifications to verify that the belief values change as expected. In general, for each of the methods whose original examples were possible to reproduce, Graydon and Holloway (2016, 2017) has identified modifications which lead to *unrealistic belief values*. Based on Graydon and Holloway (2016, 2017), the following list summarizes the main reasons for unrealistic belief calculations:

I. None of the methods contains definitions of claim being true, of conclusion following from the premises, of evidence supporting a claim, or of contextual information. In the absence of such definitions it is unclear what the source of the modeled uncertainty is, and consequently it is unclear if the uncertainty is modeled consistently for different types and structures of arguments. Also, whenever belief values are assigned by experts, these beliefs reflect the subjective definition of *true, or follows from premises*, thus the result of belief calculations are subjective and not comparable.

II. Some unrealistic calculations are a consequence of the built-in properties of the underlying frameworks for reasoning about uncertainty.

   (a) A number of methods (Guiochet et al., 2015; Wang et al., 2019, 2017; Cyra and Górski, 2011; Ayoub et al., 2013) are based on the *Dempster–Shafer* (D–S) theory (Shafer, 1976). As convincingly shown in Dezert et al. (2012), and in line with the analysis in Graydon and Holloway (2016, 2017), the rules for combining beliefs in D–S theory may lead to unrealistic calculations when many beliefs are combined. The method in Wang et al. (2019) improves one of D–S-based methods analyzed in Graydon and Holloway (2016, 2017) by minimizing the number of beliefs being combined. Namely, Wang et al. (2019) accepts safety-case arguments with at most two premises, but it is unclear if it is realistic to enforce such constraint in real-world safety cases.

   (b) The second class of methods (Denney et al., 2011; Zhao et al., 2012; Hobbs and Lloyd, 2012) are based on *Bayesian Networks* (BN) (Nielsen and Jensen, 2009), which are based on classical *probability theory* (Jaynes, 2003; Athreya and Lahiri, 2006). These methods typically encode a conclusion q and premises $p_1, \ldots, p_n$ as discrete random variables, both with states $true$ and $false$. Then, $P(q = true)$ is calculated from the joint probability distribution $P(q, p_1, \ldots, p_n)$. However, according to the laws of probability theory, the value $P(q = true)$ corresponds to the *sum* of the conditional probability that the conclusion is *true* given that the premises are *true*, and the conditional probabilities that the conclusion is *true* given that *one or more premises are false*. Assigning probability values to cases when one or more premises are known to be false is at best difficult, and typically impossible to estimate reliably. Therefore, as noted by Graydon and Holloway (2016, 2017), this leads to scenarios where even if a crucial premise of an argument is known to be *false*, the belief in the conclusion can still be rather high.

III. None of the methods state *well-formedness constraints* that define for which safety cases can the belief be calculated reliably. As observed in literature, such constraints can possibly be related both to *the syntactic structure* of a safety case, but also to the actual safety case *content* (Chowdhury et al., 2019). The absence of well-formedness constraints means that any contradiction, incompleteness, or ambiguity within a safety case may be propagated into the corresponding uncertainty model, thus making the belief calculations difficult or unrealistic. Defining and enforcing such constraints allows for a sanity check before the belief calculations, thus maximing the chance for *reliable* belief calculations.

1.1. Paper contribution

Guided by the issues faced by previous methods, the present paper presents a novel method for calculating the belief in safety-case claims. The method consists of three parts, which are simultaneously the three contributions. To avoid the issues from (I), and ensure a clear definition of the belief being calculated, the first contribution is a formalization of the common safety-case elements, that is *independent* of the structure of arguments, and of any concrete safety-case notation. The formalization is based on the *assumption* that each concrete safety case *can be represented* by formulas of a *formal language* $\mathcal{L}$. Because safety cases are typically written in natural language, the purpose of such language is not to be used for actual safety-cases, but rather to act as *a formal proxy* for an arbitrary safety case in the context of the present paper. Assuming such representation allows rigorous definitions of concepts such as *claim*, claim being *interpreted in a context*, claim being *supported by evidence* etc. These definitions are then an input for the derivation of a *general, probabilistic model* for belief calculations that *uniformly and consistently* captures the uncertainty within an arbitrary safety case.

Based on the representation of a safety case in terms of formulas of a language $\mathcal{L}$, the second contribution is a *general probabilistic model* for *the lower limit of belief* in the conclusion of an arbitrary safety-case argument. The reason for calculating the lower limit, i.e. the *worst-case* belief value, is to avoid the issues from (II.b) where it is necessary to reason about the probability that a conclusion of an argument is true even if some premises are false. Such reasoning is not of practical relevance because it means acknowledging that there are more premises in the argument than necessary, and this is one of the well-known errors in argumentation, namely the *irrelevant premise fallacy* (Greenwell, 2006).

The third contribution is more practical in nature, and it presents a step-by step process for belief calculation given a safety case in *Goal-Structuring Notation* (GSN) (Origin Consulting (York) Limited, 2018). Firstly, to avoid the issues from (III), a number of *well-formedness* constraints are defined over the standard GSN format, to ensure that a safety case is of sufficient quality for reliable belief calculations. Then, the safety-case elements of the GSN format are mapped to the safety-case elements defined in terms of formulas of a language $\mathcal{L}$. This mapping allows the creation of a probabilistic model according to contribution two, for an arbitrary safety case in GSN format whose claims are in natural language. Finally, the resulting probabilistic model, according to the second contribution, is encoded as a Bayesian Network to enable tool-supported, large scale, belief calculations.

Besides the three contributions, the paper evaluates the proposed method against the same safety cases from Graydon (Graydon and Holloway, 2016, 2017). More precisely, the same modifications from Graydon and Holloway (2016, 2017), which led to unrealistic belief values in previous methods, are used to test if the proposed method is robust with respect to these modifications. The results of the evaluation show that the proposed method produces belief values as expected, unlike the twelve methods analyzed in Graydon and Holloway (2016, 2017).

1.1.1. Paper structure

In Section 2 the relevant background about probability theory, Bayesian Networks, model theory, and Goal-Structuring Notation is presented. Section 3 presents the methodology that underlines the presented research study. Section 4 formalizes the common elements of safety-case arguments in the framework of model theory. Section 5 defines the *belief in a claim* and derives a probabilistic model for the lower limit of belief in a conclusion of an arbitrary argument. Section 6 considers the GSN notation as the concrete syntax for safety cases, defines the well-formedness constraints for GSN, maps the GSN elements to elements defined in model theory, and creates the corresponding Bayesian Network. In Section 7 proposed method is evaluated by using the safety cases from Graydon and Holloway (2016, 2017). Section 8 discusses the benefits, limitations, and practical concerns related to the proposed method. Section 9 presents related work and is followed by Section 10, which concludes the paper.

2. Preliminaries

This section presents the relevant background related to *probability theory* with an emphasis on *Bayesian networks*, followed by background about *model theory*, and *Goal-Structuring Notation*.

2.1. Probability theory and Bayesian Networks

Probability theory reasons about *random variables* whose *state* is *uncertain*, e.g. because it represents the outcome of a still unperformed experiment, or because it represents a statement for which there is a lack of knowledge to decide if it is truthful or not. In the current paper, the later case is of interest and such reasoning is sometimes referred to as *plausible reasoning* (Jaynes, 2003).

Random variables are underlined by the concepts of a *sample space* and *events*. For a process whose outcome is *uncertain*, the sample space is the set of possible outcomes, which are *mutually exclusive* and *exhaustive*. An *event* is a subset of the sample space. An example is the sample space $\{1,2,3,4,5,6\}$ for rolling a six-sided dice, where an event A is $\{2,4,6\}$, i.e. an *even* number. Typically, the events are of less interest compared to some *function* of these events. A *random variable* X is *function* from the set of events over a sample space, to a set of random variable *states*. The set of possible states $x_1,\dots,x_n$, which are *mutually exclusive and exhaustive*, of a random variable X is called the *state space*. For example, the state space $\{win, lose\}$ of X could model winning a bet depending on the outcome of rolling a die. To measure the degree of uncertainty about the occurrence of events, and consequently about the state of a random variable, function P assigns a *probability value* from $[0,1]$ to each event over the sample space.

The law of *total probability* states that for a set of *pairwise disjoint* events $A_1,\dots,A_n$ such that their union is a *sample space*, given an event B over the same sample space, it holds that

$$P(B) = \sum_{i=1}^{n} P(B, A_i).$$

In other words, the probability of B is the sum of probabilities that B occurs jointly with each of the events A_i. To calculate the probability of joint events, the *product rule* is used. According to the product rule, the joint probability distribution $P(A_1,\dots,A_n)$ can be calculated as

$$\begin{aligned} &P(A_1,\dots,A_n) = P(A_1|A_2,\dots,A_n) \\ &P(A_2|A_3,\dots,A_n)\cdots P(A_{n-1}|A_n)P(A_n), \end{aligned} \tag{1}$$

where the notation $P(\cdot|\cdot)$ is called *conditional probability*. The product rule states is that the probability $P(A_1,\dots,A_n)$ is equal to the probability of A_1 given events $A_2,\dots,A_n$, times the probability of A_2 given events $A_3,\dots,A_n$ etc. Conditional probabilities are also defined for random variables, but because variables X and Y have their respective state spaces, writing $P(X|Y)$ means that a probability value from $[0,1]$ is assigned to each pair in the Cartesian product of state spaces of X and Y.

If information about an event C does not change the probability of event an B, given the information about some event A, i.e. $P(B|A,C) = P(B|A)$, we say that B and C are *conditionally independent* given A. The same concept applies to random variables, i.e. two random variables X and Y are *conditionally independent* given a variable Z, if $P(X|Y,Z) = P(X|Z)$. Typically, when probability theory is used to model a physical phenomena, a number of conditional independence assumptions are encoded into the probabilistic model in order to faithfully model the physical phenomenon. A graphical model which effectively encodes conditional independence assumptions for large joint probability distribution is a *Bayesian Network*.

Definition 1 (*Bayesian Network*). A Bayesian network consists of the following:

(i) a set $\mathcal{X}$ of random variables, and a set of edges $\mathcal{E}$ between the variables,
(ii) each variable has a finite state space,
(iii) variables and edges form an *acyclic directed graph* G,
(iv) to each variable $X \in \mathcal{X}$ with parents $Y_1,\dots,Y_n$ in G, a *conditional probability table* $P(X|Y_1,\dots,Y_n)$ is attached. □

More precisely, a Bayesian network graphically represents the joint probability distribution $P(X_1,\dots,X_n)$ which can be factorized according to the *product rule* as

$$P(X_1,\dots,X_n) = \prod_{j=1}^{n} P(X_j|pa(X_j)), \tag{2}$$

where $pa(X_j)$ denotes the parent random variables in the graph G. The main use of Bayesian Networks is to *update* probability values based on *received evidence*, where the evidence is an observation that a random variable X is in a particular state x_i, i.e. $P(X = x_i) = 1$.

2.2. Model theory

This section recalls the basics of model theory. Note that the following description is a less formal summary compared to the typical definitions that can be found in Huth and Ryan (2004), Marker (2006) and Doets (1996).

Central concepts in model theory are a formal *language* $\mathcal{L}$, and a *model* that may *satisfy* the *formulas* of language $\mathcal{L}$. A language $\mathcal{L}$ is defined through a set of *symbols* and *rules*, which define how symbols can be combined into *well-formed formulas*. The set of symbols is partitioned into two subsets. The first one is the set of *logical symbols*, e.g. *conjunction*, *disjunction* denoted $\wedge$ and $\vee$, *existential* quantification denoted $\exists$, *until* operator denoted $\mathcal{U}$, etc. In addition, the logical symbols also include symbols which express *variables*, and these will typically be denoted as x, y, z. The non-logical symbols either express relations Q of arity $n > 0$, typically referred to as *predicates*, or express *functions* g with arity $n > 0$, or express specific entities, typically referred to as *constants* c. The set of non-logical symbols is called the *signature* of a language, denoted $\Sigma_{\mathcal{L}}$, and the signature is chosen to support creation of formulas for a particular subject topic. For example, a part of the signature for the language of set-theory could be $\{\in, \emptyset, \cup, \cap\}$, where $\in, \cup$ and $\cap$ are binary predicates over symbols that represent sets, and $\emptyset$ is a constant that represents an empty set.

Given the symbols of a language $\mathcal{L}$, the *rules* to form *well-formed formulas* are defined as follows:

- Certain language symbols are declared to be *terms*, denoted t. These include *variables*, *constants*, and possibly *functions*. Grammar $\mathcal{G}_t$ defines the allowed combinations of terms.
- *Atomic* well-formed formulas are $Q(t_1,\dots,t_n)$, where Q is an n-ary predicate and $t_1,\dots,t_n$ are terms.

- *Grammar* $\mathcal{G}_f$ defines the *composite* well-formed formulas as combinations of atomic formulas and logical symbols.

To simplify the rest of the text, we will assume that all formulas are well-formed, and simply write *formula*. Moreover, arbitrary formulas will be denoted ϕ, φ, and ψ. For some intuition, an example of an atomic formula is $\texttt{Person(x)}$, where $\texttt{Person}$ is an unary predicate, and $\texttt{x}$ is a variable. An example of a composite formula is $\forall \texttt{x.Person(x)} \rightarrow \texttt{Human(x)}$ where x is *bound* to a *universal quantifier* and where the formula states that all things that are a person are also a human. Note that if *all variables* within a formula are bound to a quantifier, then it said that this formula has no *free variables* and it is referred to as a *sentence*.

While logical symbols have a standard meaning, non-logical symbols do not, and their meaning is defined over a *model* $\mathcal{M}$.

Definition 2 (*Model*). A *model* $\mathcal{M}$ is given by the following data:

(i) a non-empty set M of concrete values, called the *universe*,
(ii) an element $c^{\mathcal{M}} \in M$ for each constant c,
(iii) a function $g^{\mathcal{M}} : M^n \rightarrow M$ for each function g of arity n,
(iv) a set $Q^{\mathcal{M}} \subseteq M^n$ for each predicate Q of arity n.

The set of possible models for a language $\mathcal{L}$ is denoted $\mathbf{M}_{\mathcal{L}}$ and $c^{\mathcal{M}}, g^{\mathcal{M}}, Q^{\mathcal{M}}$ are referred to as *interpretations* of non-logical symbols c, g, Q. The universe of a model contains *concrete entities* over which interpretation is performed, e.g. a universe over which predicates $\texttt{Person}$ and $\texttt{Human}$ would be interpreted would typically contain a number of individuals. Given the concept of a model, model theory inductively defines whether a model *satisfies* a formula, for each type of formula defined by the grammar $\mathcal{G}_f$. Because for the purpose of the present paper *the exact* syntax of language $\mathcal{L}$ is not relevant, we introduce the function $eval_{\mathcal{L}}$ for language $\mathcal{L}$ that, given a model from $\mathbf{M}_{\mathcal{L}}$ and a formula of $\mathcal{L}$, returns the value *true* or *false*.

Definition 3 (*Model Satisfies a Formula*). Let $eval_{\mathcal{L}}$ be a function that given a model $\mathcal{M} \in \mathbf{M}_{\mathcal{L}}$ and a formula ϕ of $\mathcal{L}$, returns the Boolean value true or false. If $eval(\mathcal{M}, \phi) = true$, we say that $\mathcal{M}$ *satisfies* ϕ, denoted $\mathcal{M} \vDash \phi$. We also say that a formula ϕ *evaluates to true* for the model $\mathcal{M}$.

2.3. Goal-Structuring Notation

This section recalls the definition of the *Goal-Structuring Notation* (Origin Consulting (York) Limited, 2018), which is used as the concrete safety-case syntax in the evaluation section. Moreover, this section highlights the fact that other safety-case notations fundamentally rely on the same concepts, which means that regardless of the concrete syntax, a common meaning of safety-case arguments can be defined. A complete, formal definition of GSN notation can be found in Denney and Pai (2018), while a less formal summary follows.

Definition 4 (*GSN Argument*). A *GSN argument* is a labeled *directed acyclic graph*, where N is the set of nodes, A is the set of *arcs*, and l_t, l_d are functions that label nodes with a *type* and a *description*. Function l_d is defined as $l_d : N \rightarrow \text{string}$. Function l_t is defined as $l_t : N \rightarrow \{\text{goal}, \text{strategy}, \text{solution}, \text{context}, \text{justification}, \text{assumption}\}$. Furthermore, a GSN argument satisfies the following conditions:

(i) The type of the root node n is goal,
(ii) Arcs start from nodes of type goal or strategy,
(iii) Nodes of type goal cannot simultaneously connect to nodes of type solutions and strategy,
(iv) Nodes of type strategy cannot connect to nodes of type strategy, nor of type solution,
(v) Nodes of type solution cannot connect to any other node. □

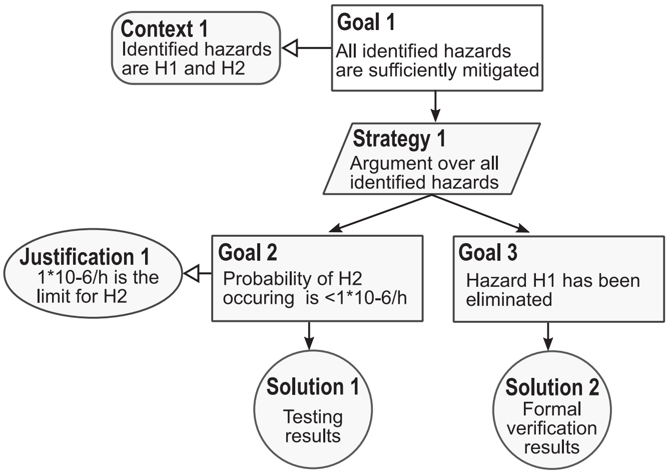
Context 1
Identified hazards are H1 and H2
Goal 1
All identified hazards are sufficiently mitigated
Strategy 1
Argument over all identified hazards
Justification 1
1*10-6/h is the limit for H2
Goal 2
Probability of H2 occurring is <1*10-6/h
Goal 3
Hazard H1 has been eliminated
Solution 1
Testing results
Solution 2
Formal verification results

Fig. 1. A safety-case fragment in GSN format.

Fig. 1 shows a fragment of a safety case from Ayoub et al. (2013), expressed in *GSN* notation. This example will be used as the running example in the following sections. The overall claim is captured by Goal 1, which states that *"all identified hazards are sufficiently mitigated"*, and this should follow from Goal 2 which states that *"hazard H1 has been eliminated"*, and Goal 3 which states that the *"probability of H2 occurring is* $< 1 \times 10^{-6}h^{-1}$*"*. The two premises are supported by two different solution nodes, namely Solution 1 that references *"testing results"*, and Solution 2 that references *"formal verification results"*. The reasoning as to why claim in Goal 1 follows from the ones in Goal 2 and Goal 3, is expressed by the Strategy 1 node. Finally, the contextual information that is needed to draw the overall conclusion is captured by Context 1 stating that *"identified hazards are H1 and H2"*, and Justification 1 stating that $1 * 10^{-6}h^{-1}$ *is the limit for H2*.

As can be seen from Fig. 1, and as defined by the GSN standard (Origin Consulting (York) Limited, 2018), goal nodes represent claims about a system, strategy nodes represent inference rules that are used to infer a claim from subclaims, solution nodes are references to evidence, assumption nodes express claims that are assumed to be true, justification nodes express the rationale for why a certain inference rule or a claim is considered true, and context nodes express contextual information in which a claim is interpreted.

An inspection of other popular safety-case notations such as CAE (Adelard LLP, 2020), SACM (Anon, 2020), or NOR-STA (Górski et al., 2012) shows that despite the different naming conventions and graphical representations, they effectively support safety-case arguments that have the same elements and same structure as the GSN notation. It should also be noted that all of these notations are just that, a concrete syntax for a safety case whose semantics is however undefined. In summary, regardless of the notation, safety-case arguments are comprised of *claims, inference rules, evidence*, and one or more types of *contextual information*.

2.3.1. Fallacious safety-case arguments

Fallacious arguments are arguments that encode some form of *faulty reasoning*. The work in Greenwell (2006) presents the *taxonomy of fallacies in system safety arguments*, which contains 33 different fallacies, grouped into eight major categories. The four categories of fallacies that will be considered in the present paper are shown in Table 1.

To provide some intuition, consider the argument with the conclusion *"software implements allocated safety requirements"*, given the premise that *"the source code was written by senior developers"*. If we acknowledge the fact that even the most senior developers can accidentally introduce faults into the source code, this is an example of a fallacy from the category *anecdotal arguments*. Indeed, there exists a

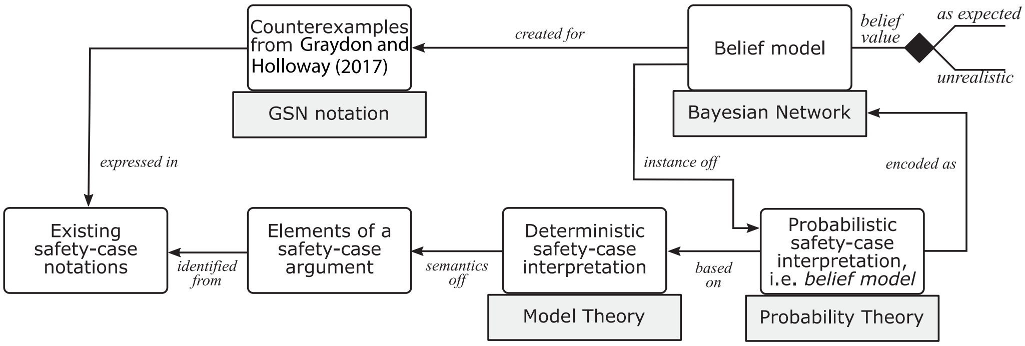
Counterexamples from Graydon and Holloway (2017)
GSN notation
created for
Belief model
Bayesian Network
belief value
as expected
unrealistic
expressed in
instance off
encoded as
Existing safety-case notations
identified from
Elements of a safety-case argument
semantics off
Deterministic safety-case interpretation
based on
Probabilistic safety-case interpretation, i.e. belief model
Model Theory
Probability Theory

Fig. 2. Elements of the study and their relations.

Table 1
Categories of considered fallacies from Greenwell (2006).

Fallacy category	Explanation
Anecdotal arguments	Conclusion does not follow from the premises
Diversionary arguments	High number of premises distracts from the fact that the conclusion does not follow from the premises
Omission of key evidence	Key evidence is missing or is even counter-evidence
Linguistic fallacies	The used language is vague and ambiguous

correlation between the seniority of a developer, and the quality of the source code. However, there is no causation between these two claims, thus the conclusion does not follow from the premises. The importance of detecting and removing fallacious arguments is that a method for belief calculation will *necessarily* produce *misleading belief values* given a fallacious argument.

3. Methodology

This section describes the methodology that underlines the performed research study, and motivates the choices of different notations and formalisms. Fig. 2 visualizes the different elements of the study and the relations between them.

The starting point for the present study is the work in Graydon and Holloway (2016, 2017), which presents an analysis of previously methods for the quantification of belief in a safety case. The work in Graydon and Holloway (2016, 2017) defines so-called *counterexamples*, which are safety cases that lead to *unrealistic belief values* for twelve previously-proposed methods for belief quantification. The purpose of the present study was to develop a method for the calculation of belief in arbitrary safety cases that also produces belief values as expected for the counterexamples from Graydon and Holloway (2016, 2017).

As Fig. 2 shows, to ensure that the belief can be calculated for arbitrary safety-cases, *elements of safety-case arguments* are *identified from* different, *existing safety-case notations*. As discussed in Section 2.3, different safety-case notations are fundamentally based on the same concepts and this study focuses on these fundamental concepts instead on their usage in a particular notation. Because existing safety-case notation define the syntax but not the semantics of safety cases, i.e. it is undefined when a safety-case claim is true, the first contribution is a *deterministic safety-case interpretation*, which is effectively a deterministic *semantics off* the elements of safety-case arguments. This deterministic interpretation is developed based on the principles of *model theory* (Marker, 2006) and Section 3.1 motivates this choice.

Given a clear definition of when a safety-case claim is true, the second contribution is a generalization into a *probabilistic safety-case interpretation*, referred to as the *belief model*. This is effectively a probabilistic safety-case semantics that also considers the cases when the available knowledge is insufficient to use the deterministic safety-case interpretation. The choice of probability theory over other frameworks for reasoning under uncertainty is motivated by the maturity and generality of probability theory, and also by the fact that other frameworks such as *belief theory* are shown to be unsound in certain scenarios (Dezert et al., 2012). The consequence of the proposed belief model is that for large, real-size safety cases, the corresponding belief model may become a large joint-probability distribution. To enable practical manipulation and analysis of large joint-probability distributions, the belief model is *encoded as* a *Bayesian Network* which can then be created and analyzed with off-the-shelf tool support.

As shown in Fig. 2, to verify whether the developed belief-model overcomes the limitations of belief-models from previous methods, the study uses the same counterexamples from Graydon and Holloway (2016, 2017) to evaluate the developed belief model. The *counterexamples*, which were expressed in GSN notation, were taken directly from Graydon and Holloway (2016, 2017) and to help presentation, they were recreated with the help of an open-source, GSN notation editor called *D-Case* (Matsuno et al., 2010). The corresponding Bayesian Networks and the belief calculations were created with the help of the academic version of *GeNIe Modeler* (Bayes Fusion). The criteria to judge whether the developed belief-model is superior compared to the previous ones was that for all counterexamples, the developed belief-model produces belief values as expected.

3.1. The choice of model theory

Because safety cases do not have a broadly accepted semantics (Langari and Maibaum, 2013), defining its semantics required the selection of a suitable framework for this task. Before selecting *model theory*, several other frameworks were considered. One candidate was the *argumentation framework* (Bench-Capon and Dunne, 2007; Dung, 1995; Riveret et al., 2018) from the artificial intelligence community. However, these frameworks *abstract away* the inner structure of arguments, while for safety cases, the structure is very much emphasized. Also, these frameworks are primarily used to reason about the *relative strength* between arguments and *not* about whether the conclusions of arguments are true or false. Another candidate was the so-called *informal logic* (Walton, 1996, 2008), which builds on the work in (Toulmin, 2003; Wigmore, 1931), and which comes from the domain of philosophy. The purpose of *informal logic* is to enable *critical thinking* but because it has been developed primarily to support everyday arguments, e.g. legal or rhetoric in general, it stay on an informal and abstract level. Finally, *model-theory* framework comes from the domain of *mathematical logic* and it studies formal languages and the scenarios in which the formulas of these languages evaluate to *true* or *false*. In model theory an argument consists of several formulas, i.e. the inner structure of arguments is explicit, and the semantics of formulas is well-defined. Given that the goal in the present paper is to be able to assign a true or false value to claims, to capture the structure of each argument, and to have a formal representation which allows unambiguous reasoning, model-theory was adopted to define the deterministic interpretation of safety cases.

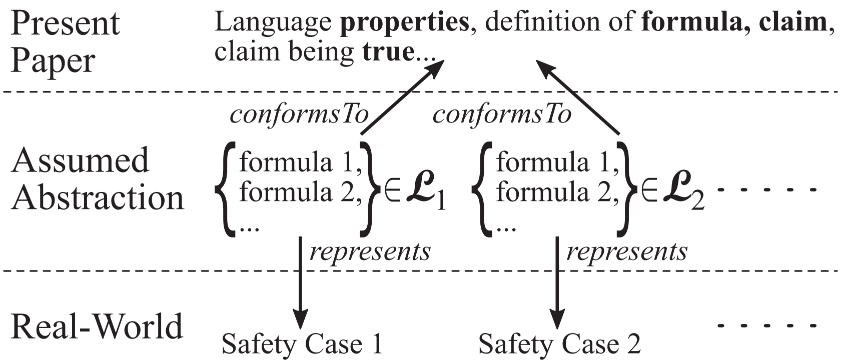

Fig. 3. The purpose of formal languages $\mathcal{L}$.

4. Deterministic interpretation of safety cases

To formally reason about the content of a safety case, in this section we develop an appropriate formal representation, which is the first contribution of the paper. The primary goal is to define a representation that is sufficiently detailed to define what it means for a claim to be true or false, but also sufficiently general so that arbitrary, typically natural-language safety-cases can be represented.

Due to the choice to rely on the principles of model theory, *we assume* that the content of each, typically natural-language safety case, can be represented by a set of *formulas* of a *formal language* $\mathcal{L}$. Fig. 3 illustrates the underlying idea where each *real-world* safety case is *represented* by a set of formulas of a formal-language.

If the differences between the claims of two safety cases are drastic, e.g. between a safety case from the automotive and aerospace domain, then the formulas that represent these two safety cases will probably belong to two different languages, e.g. $\mathcal{L}_1$ and $\mathcal{L}_2$ in Fig. 3. However, a language $\mathcal{L}_i$ might be sufficient to represent several safety cases, e.g. different versions of a safety case within a single company, or even safety cases from different companies. The crucial thing to note is that because our goal is to reason about arbitrary safety cases, we do not focus on a particular language $\mathcal{L}_i$ and the set of safety cases this language can represent. Rather we define the *properties of an arbitrary* language $\mathcal{L}$, and define how the formulas of an arbitrary language can be used as *claims, inference rules, evidence* etc. Although the *assumed abstraction*, i.e. the formulas of a language $\mathcal{L}$, are not expected to be explicitly created, they are used within the scope of the paper as a *formal proxy* to rigorously define the elements of *real-world* safety cases, and the relations between them. The purpose of these definitions is to provide the foundation for the subsequent derivation of a general probabilistic-model of belief for an arbitrary safety case, independent of the concrete notations, e.g. the ones in Section 2.3.

The following subsections introduce the definitions of safety-case elements based on model-theory.

4.1. Claims and inference rules

Given the assumptions that formulas of $\mathcal{L}$ represent the content of a safety case, i.e. for each element of a safety case there exists a semantically equivalent formula of $\mathcal{L}$, we define *claims*, *arguments* etc., in terms of formulas of $\mathcal{L}$. First, the definition of a *claim* is introduced.

Definition 5 (*Claim*). A *claim* is a sentence of $\mathcal{L}$. □

As an example, consider the claim from Goal 3 in Fig. 1, which states that *"Hazard H1 has been eliminated"*. An equivalent representation as a sentence of a language $\mathcal{L}$ is Eliminated(H1), where H1 is a constant and Eliminated is a unary predicate. An example of a composite formula is a representation of the claim from Goal 1 in Fig. 1, namely

$$\forall \mathtt{x}.\mathtt{Hazard(x)} \rightarrow \mathtt{SuffMitigated(x)}, \tag{3}$$

where x is a variable, and Hazard and SuffMitigated are two unary predicates. In the remainder of the text, claims will be typically denoted p and q. As discussed previously, claims are the building blocks of arguments.

Definition 6 (*Argument*). An *argument* is a pair $(\{p_1, \ldots, p_n\}, q)$, denoted $p_1, \ldots, p_n \vdash q$, where $\{p_1, \ldots, p_n\}$ is a non-empty set of claims called *premises*, and q is a claim called *conclusion*. □

An argument corresponds to a statement that a conclusion follows from the given premises. Off course, a conclusion of one argument can be a premise of another argument, thus forming a *hierarchy* of arguments. For example, the argument from Fig. 1 where Goal 1 is the conclusion and Goal 2 and Goal 3 are premises, can be encoded as

$$\begin{aligned}&\mathtt{Eliminated(H1)},\ \mathtt{Pr(H2)} < 1\times10^{-6}h^{-1} \vdash\\ &\forall \mathtt{x}.\mathtt{Hazard(x)} \rightarrow \mathtt{SuffMitigated(x)}\end{aligned} \tag{4}$$

and it formally represents the argument *"all hazards are sufficiently mitigated since hazard H1 is eliminated and hazard H2 occurs with a frequency less than* $1\times10^{-6}h^{-1}$*"*. To understand why this conclusion was asserted, the used *inference rule* must be stated.

Definition 7 (*Inference Rule*). An *inference rule* is a pair $(\{\phi_1, \ldots, \phi_k\}, \varphi)$, denoted $\langle\phi_1, \ldots, \phi_k \therefore \varphi\rangle$, where $\{\phi_1, \ldots, \phi_k\}$ are formulas called *premises*, and φ is a formula called *conclusion*. □

Since the argument in (4) is a representation of the overall conclusion from Fig. 1, the corresponding inference rule from Fig. 1 reads *"argument over all identified hazards"*. This inference rule actually conflates several inference rules and the language $\mathcal{L}$ representation of one of them is

$$\begin{aligned}&\langle\mathtt{Hazard(t)},(\mathtt{Eliminated(t)} \vee \mathtt{Pr(t)} < \mathtt{Limit(t)})\\ &\therefore \mathtt{SuffMitigated(t)}\rangle.\end{aligned} \tag{5}$$

The inference rule in (5) can be read as: if something represented by t is a hazard, and if it is either *eliminated* or the *probability of its occurrence* is lower than a predefined *limit*, then we can infer that t is *sufficiently mitigated*. The inference rule in (5) allows inferring that a single t is sufficiently mitigated, but the conclusion in (4) states that *all* hazards are sufficiently mitigated. The additional inference rules needed to infer the conclusion in (4) are *natural deduction rules* (Huth and Ryan, 2004), namely the rule to infer an implication, called *implication introduction*, and the rule to *generalize* the conclusion to all hazards, called *universal quantification introduction*. Moreover, the definition of $\mathtt{Limit}(t)$ is also needed, but we defer this discussion to a later part of Section 4.

Here we draw attention to a relation between inference rules and arguments. In the example argument (4), constants $\mathtt{H}_1, \mathtt{H}_2$ replace the arbitrary term t from (5). This highlights the point that an argument can rely on an inference rule *only if* they *syntactically match*. As the example shows, syntactic matching means that for each formula within the inference rule, there exists a formula within the argument such that their syntactic structure is equivalent with respect to the grammar $\mathcal{G}_f$ of a language $\mathcal{L}$ (c.f. Section 2.2). The difference between the two is that the inference rule allow arbitrary terms, while arguments necessarily contain concrete terms.

Before proceeding to the next section we note that a common alternative way to express an inference rule $\langle \phi_1, \dots, \phi_k \therefore \varphi \rangle$ is as an *implication formula* $\forall * .\phi_1 \wedge \cdots \wedge \phi_n \rightarrow \varphi$ (Galton, 1990, p. 201), where $\forall *$ denotes universal quantification over all terms within $\phi_1, \dots \phi_n$ and φ. This notation will be used when probabilistic models are considered, and such implication formulas will be denoted ψ. A consequence of using this notation is that *the grammar $\mathcal{G}_f$ must contain a rule for implication*.

4.2. Evidence and sound inference rules

When assessing a safety case, the ideal scenario is that for each claim p within a safety case, it holds that $\mathcal{M} \models p$ according to Definition 3, i.e. each claim p is *true*. While the next section discusses why this is often not possible, here we consider the type of claims for which $\mathcal{M} \models p$ is the case because $\mathcal{M}$ was *explicitly analyzed* with respect to p. This is the case for claims that are declared to be *evidence*.

Firstly, note that evidence-elements in different safety case notations (c.f. Section 2.3) are *not claims*, but *references* to claims within engineering artifacts. For example, the solution node from Fig. 1 is a reference to the claim p within the *"testing result"*, which could state that "`sw1` *passed testing against test suite for* `H2`", or that "`sw1` *failed testing against test suite for* `H2`". Because formulas of $\mathcal{L}$ are a representation of a concrete safety case, we *assume* that all references are *dereferenced*, and that each piece of evidence is a claim. Secondly, and more importantly, these claims encode the results of verification activities that *explicitly analyze* a model $\mathcal{M}$, i.e. the relevant part of the universe of $\mathcal{M}$, to establish that $\mathcal{M}$ satisfies some property. While Section 6.1.1 discusses the *uncertainty* about the *analysis process itself*, the claim that is the result of an *explicit analysis* of $\mathcal{M}$ is considered to be true regardless of what p states.

Definition 8 (*Evidence*)**.** An *evidence* is a claim, denoted e, for which it holds that $\mathcal{M} \models e$. The set of all evidence is denoted E. □

As discussed in Section 1, the idea of a safety-case is that given the available evidence-claims, the intermediary and ultimately the overall claim of a safety case is true. Whether this is the case depends on the used inference rules. Ideally, an inference rule should be such that whenever its premises are true, also the conclusion is true. Inference rules with such a property ensure that when a specific argument conclusion is inferred from the argument premises *that are known to be true*, then the conclusion will also be true. In other words, the use of such inference rules ensures that false conclusions are *impossible* to infer from premises which are true. This property of inference rules is called *soundness*.

Definition 9 (*Sound Inference rule*)**.** An inference rule $\langle \phi_1, \dots, \phi_k \therefore \varphi \rangle$ is *sound* if for all $\mathcal{M} \in \mathbf{M}_{\mathcal{L}}$ it holds that $\mathcal{M} \models \phi_1, \dots, \mathcal{M} \models \phi_k$ implies $\mathcal{M} \models \varphi$. □

Note that if an inference rule is expressed as an implication formula ψ, then a sound ψ is one that is satisfied by all models in $\mathbf{M}_{\mathcal{L}}$. For some intuition, recall the previously discussed inference rule *implication introduction*, for which there exists a formal proof of soundness (Huth and Ryan, 2004), and this is a reason why it is commonly used. The example argument in Section 2.3.1, related to the seniority of the developers is fallacious *exactly because* the used inference rule in *not sound*. Also, the example inference rule in (5) is *not sound*, because there could be models in $\mathbf{M}_{\mathcal{L}}$ where the predicate `Eliminated` has no interpretation. However, if the set of possible models is *restricted* to a subset of $\mathbf{M}_{\mathcal{L}}$, such that this predicate always has an interpretation, then the inference rule may have a *restricted form of soundness* in the scope of a subset of $\mathbf{M}_{\mathcal{L}}$. In other words, the scope in which formulas are evaluated is defined, and in safety cases this is the role of contextual information. Contextual information typically contains definitions such as `Limit(t)`, which was missing to draw the conclusion of the argument in (4).

4.3. Contextual information

A subset of $\mathbf{M}_{\mathcal{L}}$ is defined by a set of formulas Γ called a *theory* (Marker, 2006), such that each formula in Γ is satisfied by each $\mathcal{M}$ in this subset of $\mathbf{M}_{\mathcal{L}}$. Because the set Γ can contain many formulas, often a smaller, finite set of formulas $\mathcal{A}$ is defined, called *axioms*, and it is shown that all models that satisfy each $\alpha \in \mathcal{A}$ also satisfy each $\gamma \in \Gamma$. If this is the case, we say that the theory Γ is a *logical consequence* of the axioms $\mathcal{A}$. For example, if a company decides to develop their systems according to the theory of *contract-based design* (Benveniste et al., 2018; Nešić et al., 2019), then the conclusions of arguments, which rely on the theorems of contract-based design will be true, as long as the models generated by the engineering process satisfy the axioms of contract-based design.

In the rest of the paper, we assume that there exists a set of axioms $\mathcal{A} = \{\alpha_1, \dots, \alpha_m\}$, which defines the set of models $\mathbf{M}_{\mathcal{A}} \subseteq \mathbf{M}_{\mathcal{L}}$, over which the formulas of $\mathcal{L}$ are evaluated. Moreover, we assume that each α is a formula of language $\mathcal{L}$. The axioms can be domain or company-specific definitions and rules, e.g. naming conventions, definitions of engineering artifacts etc., but also general axioms of mathematics, physics etc. In other words, axioms in $\mathcal{A}$ are most frequently the definitions of non-logical symbols of language $\mathcal{L}$. For example, recall the node `Context1` from Fig. 1 that captures the definition of *identified hazards*, which can be expressed as a formula `∀x.Hazard(x) → x = H1 ∨ x = H2`. Deciding if the conclusion of (4) is true can only be done by considering that only identified hazards are `H1` and `H1`. Therefore, different kinds of *contextual information* within safety cases are interpreted as axioms in $\mathcal{A}$.

Definition 10 (*Context*)**.** A *context* is an axiom $\alpha \in \mathcal{A}$ and each model $\mathcal{M} \in \mathbf{M}_{\mathcal{L}}$ is such that $\mathcal{M} \models \alpha$. □

Definition 10 concludes the list of definitions that capture the elements and relations between the elements of a safety case. Before proceeding to the next section, we consider how the formal definitions can be used to detect various kinds of fallacious arguments.

A definition of symbols and formulas of a language $\mathcal{L}$ directly reduces the risk of *linguistic fallacies* in Table 1, e.g. if a safety-case claim cannot be represented by a formula of $\mathcal{L}$ then this claim probably contains ambiguities or inconsistencies. Requiring *syntactic matching* is a way to avoid the *omission of key evidence* and *diversionary arguments* fallacies from Table 1, where in both cases the argument premises do not match, or are absent, with respect to the used inference rule. Finally, using sound inference rules in safety cases ensures that *anecdotal arguments* fallacies from Table 1 are avoided, where the conclusion does not follow from the premises.

In this section, a formula of $\mathcal{L}$, and consequently the safety-case claim it represents, was always either true or false. The next section considers the more general case when the available knowledge is insufficient to decide if a claim is true or false, thus leading to the concept of *belief* in a claim.

5. Belief in safety-case arguments

This section presents the second contribution, namely a definition of the concept of *belief* in arbitrary claims, inference rules, and finally conclusions of arbitrary arguments.

Ideally, a safety case should be a hierarchy of arguments, according to Definition 6, where the premises and conclusions are satisfied by a model $\mathcal{M}$, according to Definition 3, that is associated with a particular engineering process. Depending on the type of argument claims, function $eval_{\mathcal{L}}$ from Definition 3 will require a model $\mathcal{M}$ according to Definition 2, of different content, size, and granularity, in order to return either the value *true* or *false*. For example, in the context of ISO 26262 (International Organization for Standardization, 2018) Part 3, it is necessary to assert the claim that *"each hazard is either prevented or sufficiently mitigated"* (c.f. Fig. 1). Then the *universe* of the model $\mathcal{M}$ must contain a representation of a real hazard log with identified hazards, of the corresponding specification document with safety goals, and of the corresponding traceability information. Moreover, the model should contain the interpretation for constants that represent *hazards* and *safety goals*, and for the predicates *prevented* and *sufficiently mitigated*. Given such model, and the mentioned claim, function $eval_{\mathcal{L}}$ can return a value true or false.

However, claims within a safety case are often about complex properties such as the ones about the implementation of the system being assured, or about the environment in which the system should operate. Examples are claims about software implementing a requirement, or successfully processing all inputs from the environment. For the function $eval_{\mathcal{L}}$ to return a value true or false for such claims, the universe of the required model would have to contain the *complete state-space of a software*, or a model of the intended operating-environment behavior. However, because the number of software states is typically astronomical (David et al., 2011), and the behavior of the intended environment is random (Hauer et al., 2019), in such cases *it is often not possible, or practically feasible* to construct the required model (Bishop et al., 2011; Bloomfield et al., 2007). In general, this means that for some claims within a safety case, and because the model $\mathcal{M}$ *is partly unknown*, the function $eval_{\mathcal{L}}$ cannot return a value true or false. Other methods have previously discussed a general lack of knowledge in the context of safety-cases (Rushby, 2017), or informally introduced uncertainty about the *"contribution of evidence to a claim"* (Wang et al., 2019), or uncertainty about the *"appropriateness and trustworthiness of the context"* (Denney et al., 2011). In the present paper, the fact that the model $\mathcal{M}$ underlies the definition of a claim being true or false allows formally expressing the uncertainty about whether a claim is true or false as the uncertainty about the model $\mathcal{M}$. This observation leads to a *formal definition* of the *belief in a claim*.

5.1. Belief in a conclusion of an argument

As discussed in the previous section, because the model $\mathcal{M}$ is *partially unknown*, it is not possible to decide if for some conclusion q it holds that $\mathcal{M} \vDash q$. Consequently, the present paper estimates the probability that $\mathcal{M} \vDash q$, i.e. $P(\mathcal{M} \vDash q)$. This means that the model $\mathcal{M}$ is treated as a *discrete random variable* with the state space containing states $\mathcal{M} \vDash q$ and $\mathcal{M} \nvDash q$, i.e. $\mathcal{M}$ satisfies q, $\mathcal{M}$ does not satisfy q.

A crucial thing to note is that the model $\mathcal{M}$ is not completely unknown. Firstly, $\mathcal{M}$ is a member of the set $\mathbf{M}_{\mathcal{L}}$, but more importantly, the safety-case evidence and the asserted axioms represent observations that the model satisfies each $e \in E$ and each $\alpha \in \mathcal{A}$, respectively. In other words, instead of calculating the *marginal probability* $P(\mathcal{M} \vDash q)$, a conditional probability can be calculated, namely

$$P(\mathcal{M} \vDash q \mid \mathcal{M} \vDash \mathcal{A}, \mathcal{M} \vDash E). \tag{6}$$

The expression in (6) is the *belief* in a claim q. As a shorthand, from now on we will simply write $P(q|\mathcal{A}, E)$ or $P(\neg q|\mathcal{A}, E)$ as a shorthand for $P(\mathcal{M} \nvDash q|\mathcal{M} \vDash \mathcal{A}, \mathcal{M} \vDash E)$. The value $P(q|\mathcal{A}, E)$ is best understood as a special case of *plausible reasoning* (Polya, 1990; Jaynes, 2003). Because in general it is not possible to determine if $\mathcal{M} \vDash q$, only the *plausibility* of $\mathcal{M} \vDash q$ can be determined, given all available information. In other words, the value $P(q|\mathcal{A}, E)$ represents the state of knowledge about $\mathcal{M}$, encoded in $\mathcal{A}$ and E, with respect to the claim q. Whenever the set of evidence is updated to E' with additional information about $\mathcal{M}$ that supports the conclusion $\mathcal{M} \vDash q$, then it holds that $P(q|\mathcal{A}, E) < P(q|\mathcal{A}, E')$.

In the most general sense, expression (6) is sufficient to calculate the belief in the top safety-case claim. However, because q is typically an abstract claim, e.g. *system is acceptably safe*, while the evidence is very specific, e.g. *positive review of SW specification*, estimating this value directly would be very difficult. But the power of safety cases lies exactly in the fact that q should be possible to deduce from given *hierarchy of arguments* and not only the evidence. More precisely, to calculate the belief in the top conclusion, the belief in the corresponding premises and inference rule can be taken into account. This means that given an argument $p_1, \dots, p_n \vdash q$ based on an inference rule ψ, and according to the *law of total probability* (c.f. Section 2.1), the belief in q can be calculated as

$$\begin{aligned} P(q|\mathcal{A}, E) = &P(q, \psi, p_1, \dots, p_n|\mathcal{A}, E) \\ &+ P(q, \psi, p_1, \dots, \neg p_n|\mathcal{A}, E) + \cdots \\ &+ P(q, \neg\psi, \neg p_1, \dots, \neg p_n|\mathcal{A}, E). \end{aligned} \tag{7}$$

The only term in Eq. (7) that is of interest is the first one, because we assume that this term is *much larger* than the other ones. This is a reasonable assumption because in all other cases at least one of the premises, or the inference rule, is not satisfied by the model $\mathcal{M}$, thus the probability that $\mathcal{M} \vDash q$ must very low. Consequently, all terms but the first one are omitted from (7), and (7) becomes an inequality which defines a *tight lower limit of the belief* in conclusion q. More formally, it follows that

$$P(q|\mathcal{A}, E) \geq P(q, \psi, p_1, \dots, p_n|\mathcal{A}, E). \tag{8}$$

By using the *product rule* of probability theory, (8) becomes:

$$\begin{aligned} P(q|\mathcal{A}, E) \geq &P(q|\psi, p_1, \dots, p_n, \mathcal{A}, E) P(\psi|p_1, \dots, p_n, \mathcal{A}, E) \\ &P(p_1|p_2, \dots, p_n, \mathcal{A}, E) \cdots P(p_n|\mathcal{A}, E). \end{aligned} \tag{9}$$

Since each of the factors in the product from (9) is conceptually different, one by one factor is now analyzed.

First consider the factor $P(\psi|p_1, \dots, p_n, \mathcal{A}, E)$. This is the probability that a model satisfies the implication formula of the inference rule, given that a model satisfies the premises, axioms and evidence. According to Definition 9 and Section 4.3, if an inference rule is *sound*, then it is satisfied by all possible models, and if it is not sound, then it might be a *logical consequence* of axioms $\mathcal{A}$. In either case, the premises of a particular argument, and evidence of a particular safety case are irrelevant. Therefore, according to the definition of conditional independence it holds that $P(\psi|p_1, \dots, p_n, \mathcal{A}, E) = P(\psi|\mathcal{A})$, and we call this factor the *belief in an inference rule*.

Next factors to consider is the set of factors related to the premises. Within the majority of safety-case arguments proposed in the past 20 years (Szczygielska and Jarzębowicz, 2018), the premises of arguments are considered independent. Another indication that such independence is assumed, is the fact that standardization documents, and each formalization of safety case notation from Section 2.3 either explicitly forbid such dependencies, or such dependencies are never used. Because in the general case, two premises may be dependent, this common independence assumption is interpreted as *conditional independence* between the premises *given the evidence*, and the factors $P(p_1|p_2, \dots, p_n, \mathcal{A}, E) \cdots P(p_n|\mathcal{A}, E)$ reduce to $P(p_1|\mathcal{A}, E) \cdots P(p_n|\mathcal{A}, E)$.

Finally, the first factor is the probability that a conclusion is satisfied, given that the premises, the inference rule as implication formula ψ, the axioms, and the evidence are satisfied. This term represents the concept of *syntactic matching* from the previous section, i.e. this factors models the belief that q can be inferred from $p_1, \dots, p_n$ based on the inference rule ψ. An important observation is that this factor is conditionally independent of E given the premises $p_1, \dots, p_n$. However, this factor is *not* in general independent of $\mathcal{A}$. If we recall the example from Fig. 1, concluding that *all identified hazards are sufficiently mitigated* is not possible without the definition of identified hazards, i.e. an axiom. Therefore, $P(q|\psi, p_1, \ldots, p_n, \mathcal{A}, E)$ reduces to $P(q|\psi, p_1, \ldots, p_n, \mathcal{A})$. Given the simplifications of each of the terms from (8), inequality (8) becomes

$$P(q|\mathcal{A}, E) \geq P(q|\psi, p_1, \ldots, p_n, \mathcal{A})P(\psi|\mathcal{A})P(p_1|\mathcal{A}, E)\cdots P(p_n|\mathcal{A}, E). \tag{10}$$

Inequality, (10) is a *general probabilistic model of the lower limit of belief* in the conclusion of an *arbitrary safety-case argument*. Depending on the particular claims, inference rule, and evidence, (10) can be modified in different ways, and such different scenarios are discussed in the following section.

5.2. Argument-specific modifications of (10)

This section shows how the probabilistic model from (10) can be modified, or how some factors can be assigned values, for different kinds of arguments.

5.2.1. Matching of arguments and inference rules

In the case where an argument is such that the claims within the argument *syntactically match* the formulas within the inference rule, as described in Section 4.1, then it holds that $P(q|\psi, p_1, \ldots, p_n, \mathcal{A}) = 1$ and (10) becomes

$$P(q|\mathcal{A}, E) \geq P(\psi|\mathcal{A})P(p_1|\mathcal{A}, E)\cdots P(p_n|\mathcal{A}, E) \tag{11}$$

An example of such argument is

$\mathtt{Hazard(H1)}, \mathtt{Eliminated(H1)} \vdash \mathtt{SuffMitigated(H1)}$

and the corresponding inference rule is

$\langle \mathtt{Hazard(t)}, \mathtt{Eliminated(t)} \therefore \mathtt{SuffMitigated(t)} \rangle.$

5.2.2. Arguments with evidence as premises

For an argument where $k \leq n$ premises are evidence $e_1, \ldots, e_k$, inequality (10) is actually

$$P(q|\mathcal{A}, E) \geq P(q|\psi, p_1, \ldots, p_{n-k}, e_1, \ldots, e_k, \mathcal{A})P(\psi|\mathcal{A}) P(p_1|\mathcal{A}, E)\cdots P(p_{n-k}|\mathcal{A}, E)P(e_1|\mathcal{A}, E)\cdots P(e_k|\mathcal{A}, E).$$

Because each e_i is an element of E, it follows that each $P(e_i|\mathcal{A}, E) = 1$ and (10) reduces to

$$P(q|\mathcal{A}, E) \geq P(q|\psi, p_1, \ldots, p_{n-k}, e_1, \ldots, e_k, \mathcal{A})P(\psi|\mathcal{A}) P(p_1|\mathcal{A}, E)\cdots P(p_{n-k}|\mathcal{A}, E). \tag{12}$$

An interesting scenario is when an argument contains evidence that is a *counter-evidence*. Although a widely accepted definition of counter-evidence does not exist, multiple sources describe it as an *evidence that refutes, or undermines a claim* (Greenwell, 2006; Nemouchi et al., 2019; Origin Consulting (York) Limited, 2018). Because an evidence is always a premise of an argument, then it must be that a counter-evidence *refutes or undermines* the claim that is the conclusion of the corresponding argument. Also, because conclusions are inferred from the corresponding premises *based on an inference rule*, it follows that counter-evidence hinders the intended inference.

To interpret the concept of counter-evidence, recall that according to Definition 8, an evidence is a claim e and it holds that $\mathcal{M} \models e$. Given an argument where evidence are premises, according to Section 4.1, the conclusion can be inferred from the premises if there exists *syntactic matching* between the claims of the argument and the formulas of the corresponding inference rule. However, a *counter-evidence* is a premise that is *opposite* to the one that achieves syntactic matching. For example, consider an argument whose conclusion should be $\mathtt{SuffMitigated(H2)}$, and the intended inference rule to infer this conclusion is

$\langle \mathtt{Hazard(t)}, \mathtt{Pr(t)} < \mathtt{Limit(t)} \therefore \mathtt{SuffMitigated(t)} \rangle.$

To perform the inference according to the inference rule, and to achieve syntactic matching, a premise of an argument must be evidence that $\mathtt{Pr(H2)} < \mathtt{Limit(H2)}$. A counter-evidence would be a premise that is the opposite, i.e. the evidence $\mathtt{Pr(H2)} \geq \mathtt{Limit(H2)}$. As can be seen, an evidence can be a counter-evidence *only in relation* to a particular inference rule. In general, counter-evidence is a special case of a premise that does not syntactically match the inference rule. This case is opposite to the case considered in Section 5.2.1.

Still, inequality (12) applies but the value of factor $P(q|\psi, p_1, \ldots, p_{n-k}, e_1, \ldots, e_k, \mathcal{A})$, which represents the degree of syntactic matching between the inference rule and the argument, must be set accordingly. In the case with *matching premises*, including evidence, this factor will be set to a rather high value, i.e. similar to the case in Section 5.2.1. In the case with premises that do not match the inference rule, e.g. for *counter-evidence*, this value must be set to a value close to 0.

5.2.3. Values of belief in inference rules

In the not so common, but highly desirable case when there exists a proof that the inference rule is a consequence of the adopted axioms, the term $P(\psi|\mathcal{A})$ is equal to 1, and (10) can be reduced to

$$P(q|\mathcal{A}, E) \geq P(q|\psi, p_1, \ldots, p_n, \mathcal{A})P(p_1|\mathcal{A}, E)\cdots P(p_n|\mathcal{A}, E). \tag{13}$$

Moreover, if an inference rule is sound, i.e. the implication formula ψ is satisfied by each model in $\mathbf{M}_{\mathcal{L}}$, then it holds that $P(\psi|\mathcal{A}) = P(\psi) = 1$ and inequality (13) still applies. The belief in an inference rule can be set to 1 also when an inference rule is *prescribed* by a standard and is used within a safety case that shows compliance with this standard. In all other cases, the belief in an inference rule must be strictly less than 1. Assigning a high belief in an inference rule, for which there is little proof that it follows from the adopted axioms, corresponds to the *anecdotal arguments* fallacy from Table 1.

6. From a safety case to a Bayesian Network

This section presents the third contribution of the paper, which is oriented towards the practical use of the proposed method. Namely, inequality (10), and its modifications (11)–(13) are derived for a single argument based on a representation of a safety case in terms of formulas of a language $\mathcal{L}$. However, a safety case is typically expressed in natural language and it contains many inter-dependent arguments, which rely on particular, possibly common, inference rules and premises. This means that for a complete safety case, an inequality such as (10) would correspond to a large joint probability distribution with many conditional independence assumptions that reflect the structure of the argument within a safety case. To enable practical, tool-supported reasoning about such probability distributions, inequality (10) can be encoded as a Bayesian Network. However, because inequality (10) is derived independently of any concrete safety-case notation, first it is necessary to *map* the types of elements of a particular notation to the argument elements in terms of formulas of a language $\mathcal{L}$.

Defining such mapping, and encoding (10) into a Bayesian Network is the topic of this section. Going from a concrete, natural-language safety in one the notations from Section 2.3, to a corresponding Bayesian Network for belief calculations is done in three steps:

- Mapping the types of elements of a concrete safety-case notation to the concepts of *claim*, *argument*, *inference rule*, *evidence*, and *axioms* as defined in Section 4. The considered safety-case notation is GSN and by establishing this mapping, a probabilistic model (10) can be obtained for an arbitrary safety case in GSN format.
- An encoding of the probabilistic model (10), into a Bayesian Network. By creating a Bayesian Network, tool-supported belief calculations for real-size safety cases are enabled.

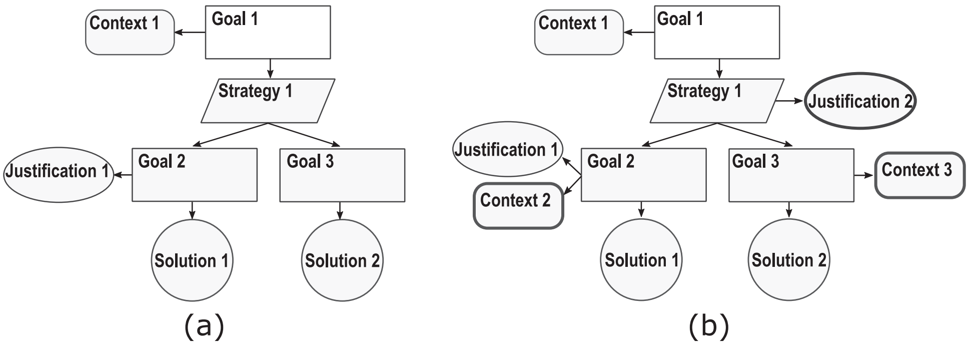

Fig. 4. (a) GSN argument from Fig. 1, and (b) its modification to conform with Definition 11.

- An assignment of values to the *conditional probability tables* (CPTs) that are associated with the random variables within the Bayesian Network.

Before defining the mapping in step one, we define the *well-formedness* constraints to ensure that a safety case in GSN format is sufficiently complete and non-fallacious, thus facilitating reliable belief calculations.

6.1. *Well-formedness constraints over GSN arguments*

The GSN format as defined in Definition 4 is envisioned as a flexible notation, where many argument elements are optional. For example, elements that represent inference rules can be omitted, axioms from which the inference rule should follow can be omitted, a claim can be supported both by evidence and claims simultaneously etc. The following definition is in line with Definition 4, but it introduces additional constraints.

Definition 11 (*Constrained GSN Argument*)**.** A *constrained GSN argument* is a GSN argument that satisfies the following conditions:

(i) Nodes of type goal cannot connect to other nodes of type goal,
(ii) Each node of type goal connects either to exactly one node of type strategy, or to at least one type solution,
(iii) Each node of type strategy connects to a node of type justification,

To exemplify the differences between a *constrained* GSN argument and a *regular* GSN argument, Fig. 4(a) shows the structure of the GSN argument from Fig. 1, and Fig. 4(b) shows its modification in order to conform with Definition 11 (nodes Context 1 and Context 2 are explained later).

Firstly, although (Denney and Pai, 2018; Origin Consulting (York) Limited, 2018) allow goals to connect directly to other goals, condition (i) in Definition 11 prohibits this in order to ensure that there exists a strategy node that represents the inference rule of the argument, e.g. Strategy1 in Fig. 4. Secondly, without loss of generality, condition (ii) allows goals to connect to *only one strategy node*. This means that a conclusion represented by a goal should follow from the premises represented by other goals, based on a *single inference rule*. Thirdly, condition (iii) requires that each strategy is connected to a justification node that justifies the use of the particular strategy, e.g. J1 in Fig. 4. Such justification node should express the axioms whose logical consequence is the inference rule.

Besides the constraint of Definition 11, we *assume* that the description of context nodes contain the *definitions of entities* within the connected goal or strategy nodes. For example, the context node Context 1 from Fig. 1 contains the definition of *identified hazards*. This use of context nodes is the most common scenario and it is referred to as *explication* (Graydon, 2014).

6.1.1. *GSN arguments with evidence as premises*

GSN arguments with evidence as premises deserve special consideration. One reason for this is of general nature, because the sources of uncertainty related to evidence are specific. The second reason for special treatment is a built-in limitation of GSN to express such uncertainties.

In the general setting of model theory, Definition 8 and consequently Sections 4 and 5 have treated evidence simply as the claims that are *absolutely true*, i.e. as facts from which further inferences can be made. However, within a concrete safety case, generated by a concrete engineering process, typically there is *uncertainty* about whether the evidence is really true. For example, although an evidence might state that *"SW1 passed unit testing"*, to be certain that this evidence is true, one must *eliminate the possibility* that the testing tool-set has produced a false positive, or that the test-cases exercised a very small portion of the software state space. Previous methods, e.g. Wang et al. (2019) and Denney et al. (2011), *conflate* all such uncertainties into *a single value*.

In the present paper we take a different approach and instead of modeling the belief in evidence, we *require* that the sources of uncertainty related to the evidence are *explicitly argued about in the safety case*. The reason for doing so is twofold. Firstly, safety standards such as ISO 26262 International Organization for Standardization (2018) and IEC 61508 The International Electrotechnical Commission (2010), already require that the quality of the used tools is justified, that testing coverage is justified etc. Thus, safety cases *already contain such claims*, and they can be included in arguments where evidence are premises. Secondly, requiring that such uncertainties are made explicit results in *more systematic and thorough* safety cases, and minimizes the room for introducing *confirmation bias* (Leveson, 2011).

To achieve this, we require that for arguments with evidence as premise, context nodes are used to according to a common pattern called *implicit premise* (Graydon, 2014). Context nodes that are implicit premises are *references* to other goal nodes within the safety case. For example, in Fig. 4(a) Goal 3 is supported by Solution 2. A context node Context 3 is added in Fig. 4(b) to act as an *implicit premise* to the argument $\mathtt{Solution2} \vdash \mathtt{Goal3}$, i.e. the argument becomes $\mathtt{Context3}, \mathtt{Solution2} \vdash \mathtt{Goal3}$. Node Context 3 is a reference to a goal node that argues that a particular source of uncertainty related to evidence Solution 2 has been eliminated. This constraint is captured by the following definition, and this definition is the reason for nodes Context 2 and Context 3 in Fig. 4(b).

Definition 12 (*Complete GSN Argument*)**.** A *complete* GSN argument is a constrained GSN argument such that for each $(n_1, n_2) \in A$ where $l_t(n_1) = \mathsf{goal}$ and $l_t(n_2) = \mathsf{solution}$, there exists at least one node $n_3 \in N$ where $l_t(n_3) = \mathsf{context}$, $(n_1, n_3) \in A$ and n_3 is an *implicit premise*. □

The second consideration about GSN arguments with evidence as premises stems from a limitation of the GSN format. Namely, according to Definition 11, but also definitions in Denney and Pai (2018) and Origin Consulting (York) Limited (2018), an argument in GSN

Table 2
The mapping of GSN types of nodes to concepts based on formulas of language $\mathcal{L}$.

GSN node	Concept based on $\mathcal{L}$	Clarification
goal	*claim* as per Definition 5	
strategy	*inference rule* as per Definition 7	
solution	*evidence* as per Definition 8	
assumption	*axiom* $\alpha \in \mathcal{A}$	Information whose truth is assumed
justification	*axiom* $\alpha \in \mathcal{A}$	
context	*axiom* $\alpha \in \mathcal{A}$ *claim* as per Definition 5	Definitions of entities in connected nodes, i.e. *explication* Reference to nodes of type goal, i.e. *implicit premise*

Table 3
Encoding inequality (10) as a Bayesian Network.

Concept based on formulas of $\mathcal{L}$	Random variable (s)	State space	Parent random variable (s)
evidence e	X_e	$sat, notSat$	No parents
axiom α	X_α	$sat, notSat$	No parents
inf. rule ψ	X_ψ	$sound, notSound$	X_α
premises $p_1, \ldots, p_n$	$X_{p_1}, \ldots, X_{p_n}$	$sat, notSat$	Depending on arguments where $p_1, \ldots, p_n$ are conclusions
conclusion q	X_q	$sat, notSat$	$X_\psi, X_{p_1}, \ldots, X_{p_n}, X_\alpha$
Bayesian Network Evidence: $P(X_\alpha = sat) = 1,\ P(X_e = sat) = 1$			

format where the premises are evidence, *are not allowed* to have a strategy node, i.e. an inference rule, explicitly declared. For example, in Fig. 4(a) this is the case for the argument $\texttt{Solution2} \vdash \texttt{Goal3}$. Although in some scenarios the evidence *directly supports* the corresponding conclusion, thus there is no need to state the inference rule, in other scenarios this is not the case. For example, it cannot be said directly that the claim *"Probability of H2 occurring is* $< 1 \times 10^{-6}/h$*"* is *true*, given the evidence about successful testing. This means that although omitted, *implicit inference rules exist*, for which the belief must be estimated. The fact that in general an argument always relies on an inference rule is also visible in inequality (10). Only in the special case when the inference rule is *sound*, or is a *logical consequence* of the axioms, the belief in the inference rule disappears from (10) and reduces to (13). To circumvent this limitation of GSN, when defining the procedure to create a Bayesian Network given a safety case, all *implicit inference rules will be made explicit.*

Note that because implicit inference rules exist only for GSN arguments where evidence are premises, it follows that these can only state that a particular type of evidence, *implies* a particular type of conclusion. Such inference rules correspond exactly to recommendations from various standards about which evidence to produce in order to ensure a certain property. For example, ISO 26262, part 4 Table 9, gives ASIL-dependent recommendation which type of evidence should be produced in order to support the claim that a *"system correctly implements functional and technical safety requirements"*.

6.1.2. Mapping GSN nodes to concepts in terms of formulas of $\mathcal{L}$

Definitions 11, Definitions 12, explicit modeling of implicit inference rules, and the use of context nodes for *explication*, are the *well-formedness* constraints imposed on standard GSN arguments. Given these, we define the first step towards creating a Bayesian network for belief calculations for a safety case in GSN format. Namely, the types of nodes from GSN notation are mapped to the elements of safety-case arguments that are defined in terms of formulas of language $\mathcal{L}$. Table 2 shows the mapping. Note that the mapping is independent of the description of GSN nodes, i.e. there is no requirement that the descriptions of GSN nodes must be formulas of a formal language. Table 2 states that whatever the description of a goal node is, it is considered to be a *claim*, the description of strategy node is considered to be an inference rule etc. In this way, the typically natural-language content of a safety case populates the probabilistic belief-model (10).

6.2. Building the Bayesian Network

To enable practical, tool-supported analysis of a belief model for a *complete safety case*, we encode inequality (10) into a Bayesian Network. Table 3 shows the mapping between the elements of a safety case in terms of formulas of a language $\mathcal{L}$, to *discrete random variables* and their corresponding *conditional probability tables* (CPTs). Table 3 also sets some of the *Bayesian Network evidence*.

The mapping is defined for an argument $p_1, \ldots, p_n \vdash q$, the corresponding inference rule ψ, and the axioms connected to q and ψ. If the premises $p_1, \ldots, p_n$ are evidence, and despite the GSN format (c.f. Section 6.1.1), it is *assumed* that an inference rule is declared and a corresponding random variable is created. Note that the mapping should be recursively applied to each premise p_i, that is a conclusion of another argument until the complete GSN argument is converted into a Bayesian Network.

To illustrate that by using the encoding in Table 3, the resulting Bayesian Network captures the same factorization as inequality (10), consider the GSN argument in Fig. 5(a) and the corresponding Bayesian Network in Fig. 5(b).

The GSN argument encodes three arguments according to Definition 6, namely

$$\texttt{G2}, \texttt{G3} \vdash \texttt{G1}, \quad \texttt{Sn1} \vdash \texttt{G2}, \quad \texttt{Sn2} \vdash \texttt{G3}.$$

The first argument relies on the explicit inference rule S1, and for the two latter arguments the *implicit inference rules* are inferred when the Bayesian network is created, namely S2 and S3. Besides the three arguments, the toy safety case encodes the axioms $\mathcal{A} = \{\texttt{C1}, \texttt{J1}, \texttt{A1}, \texttt{C2}, \texttt{C3}\}$ and evidence $E = \{\texttt{Sn1}, \texttt{Sn2}\}$. Given the three arguments in Fig. 5(a), the corresponding inference rules, axioms and evidence, the Bayesian Network in Fig. 5(b) is the result of applying the encoding from Table 3. The Bayesian Network captures the joint probability distribution

$$\begin{aligned} &P(X_{G1}, X_{S1}, X_{J1} = sat, X_{G2}, X_{S2}, X_{J1} = sat, X_{Sn1} = sat, \\ &\quad X_{Gx}, X_{G3}, X_{S3}, X_{Sn2} = sat, X_{Gy}) \end{aligned}$$

which corresponds to the following factorization:

$$\begin{aligned} &P(X_{G1} | X_{S1}, X_{C1} = sat, X_{G2}, X_{G3}) P(X_{S1} | X_{J2} = sat) \\ &P(X_{G2} | X_{S2}, X_{Gx}, X_{Sn1} = sat, X_{J1} = sat) \\ &P(X_{G3} | X_{S3}, X_{Sn2} = sat, X_{Gy}) \\ &P(X_{C1} = sat) P(X_{J2} = sat) \\ &P(X_{J1} = sat) P(X_{Sn1} = sat) P(X_{Sn2} = sat) \end{aligned} \tag{14}$$

A comparison to (10) shows that expression (14) contains the same kind of factors as (10). The difference between (10) and (14) is that (10) captures *the lower limit of the belief* in claim, while the CPTs of random variables in the Bayesian Network also encode the probabilities for the conclusion being true if either of the premises, or the inference rule are not satisfied. To adjust the Bayesian Network to encode only the calculation for the lower limit of beliefs, but also to enable the

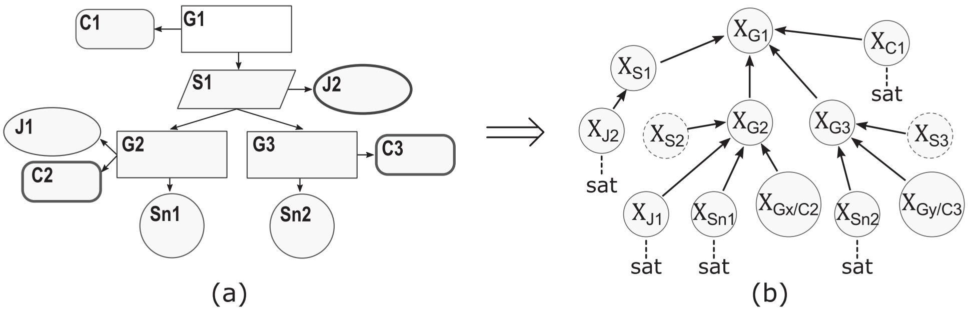

Fig. 5. A GSN structure (a), and the corresponding Bayesian Network (b). Nodes with dashed outline represent implicit inference rules.

actual belief calculations, the CPTs of the Bayesian Network must set accordingly.

6.3. Setting probability values of CPTs

The final step in the three-step process to create a Bayesian Network, is to populate the Bayesian Network CPTs. In general, there are three different types of values to assign.

6.3.1. Type I CPT values

The first type are the values for the CPTs of random variables X_q, that represent a conclusion of an argument, whose parents are random variables $X_{p_1}, \dots, X_{p_n}$, X_ψ, and possibly X_α. The CPTs of such X_q encode the probability that $X_q = sat$ or $X_q = notSat$, for all combinations of states of parent variables. Because we are only interested in the case when $X_q = sat$, each $X_{p_i} = sat$, $X_\psi = sat$, and $X_\alpha = sat$, and under the assumption that the inference rule and the argument *syntactically match* (c.f. Section 5.2.1), the value $P(X_q = sat | X_q = sat, X_{p_1} = sat, \dots, X_{p_n} = sat, X_\psi = sound, X_\alpha = sat)$ is set to 1. For all other combinations of states of parent variables $P(X_q = sat | \dots) = 0$.

6.3.2. Type II CPT values

The second type are the values for the probability that an explicitly declared inference rule, is sound or a logical consequence of the asserted axioms. An example from (14) is $P(X_{S1} = sound | X_{J2} = sat)$. In general, this value must be set manually, but as discussed in Section 5.2 there are two special cases in which setting this value is trivial, i.e. $P(\psi|\mathcal{A}) = 1$. The first case is when there exists a formal proof that the inference rule follows from the axioms. The second case is when a definition from a standard is used as an inference rule within a development process alligned with that standard. For example, ISO 26262 Part 8, Section 6 defines a *correctly specified* safety requirement as a requirement that is *unambiguous*, *comprehensible*, *atomic* etc. This definition can also be interpreted as an inference rule stating that if a safety requirement is *unambiguous*, *comprehensible*, *atomic* etc., then a safety requirement is *correctly specified*. If such inference rule is used within a safety case that argues safety with respect to ISO 26262, the belief in this inference rule can be set to 1.

6.3.3. Type III CPT values

The third type are the values for the belief in the *implicit inference rules*. Again, in the general case, such values must be set manually. Fortunately, because the belief in such inference rule represents the probability that a certain type of evidence implies a certain type of conclusion, the literature from the *evidence-based software engineering* community can provide some realistic values. For example, contributions in Graves et al. (2001) and Juristo et al. (2004) quantify the effectiveness of various testing techniques for fault detection, contributions in Zheng et al. (2006) and Wedyan et al. (2009) quantify the effectiveness of *static-analysis* for fault detection, contributions in Edmundson et al. (2013) and Runeson et al. (2006) evaluate and compare the effectiveness of various types of testing and manual code-review methods for fault detection, etc.

7. Evaluation

In this section the proposed method is evaluated. The evaluation is structured around the work in Graydon and Holloway (2016, 2017). As discussed in the introduction, the work in Graydon and Holloway (2016, 2017) has reproduced the safety cases and corresponding belief calculations from twelve different publications, and then subjected the safety cases from these publications to various modification to verify that belief values change as expected. Whenever a modification led to an unrealistic change in belief value, this modification was declared to be a *counterexample*. Also, for each counterexample, the *expected change* of belief values was defined. The work in Graydon and Holloway (2016, 2017) has identified *two general* counterexamples which led to unrealistic belief calculations for six methods, and *three method-specific* counterexamples which led to unrealistic belief calculations.

The remainder of this section is an evaluation against the counterexamples from Graydon and Holloway (2016, 2017). More precisely, the evaluation is performed as follows:

- We recreate a safety case that was analyzed in Graydon and Holloway (2016, 2017). The safety case to recreate is chosen in a way such that the largest number of counterexamples can be applied. For the chosen safety case, four counterexamples are applicable, both general ones, and two method-specific.[1]
- We create the Bayesian Network that corresponds to the recreated safety case, and calculate the belief in the top claim to ensure that the value is similar to the methods analyzed by Graydon.
- We modify the safety case according to each of the four applicable counterexamples, recalculate the belief values, and compare the change of belief value to the *expected change* as defined in Graydon and Holloway (2016, 2017). The goal is to show that the proposed method produces belief values as expected, unlike the methods analyzed in Graydon and Holloway (2016, 2017).

7.1. The considered safety case and the corresponding Bayesian Network

Fig. 6 shows the safety-case fragment that is used for the evaluation of the proposed method, and which is created with the open-source GSN editor called *D-Case* (Matsuno et al., 2010). The safety-case fragment is taken from Figure J1 in Graydon, but nodes J1, J2, C2 – C5 and C_7 were added to conform to the *well-formedness* constraints from Section 6.1. As will be seen later, because the counterexamples modify only the number and the structure of *goal and evidence nodes*, the additional context and strategy nodes do not preclude the use of the counterexamples. Also note that the identifiers of the *implicit inference rules* are overlaid over the GSN argument, namely S3 – S5.

The argument structure in Fig. 6 argues that a system is acceptably safe, since the significant hazards have been identified, and since

[1] The third method-specific counterexample is conceptually a special case of counterexample 1 in Section 7.2.

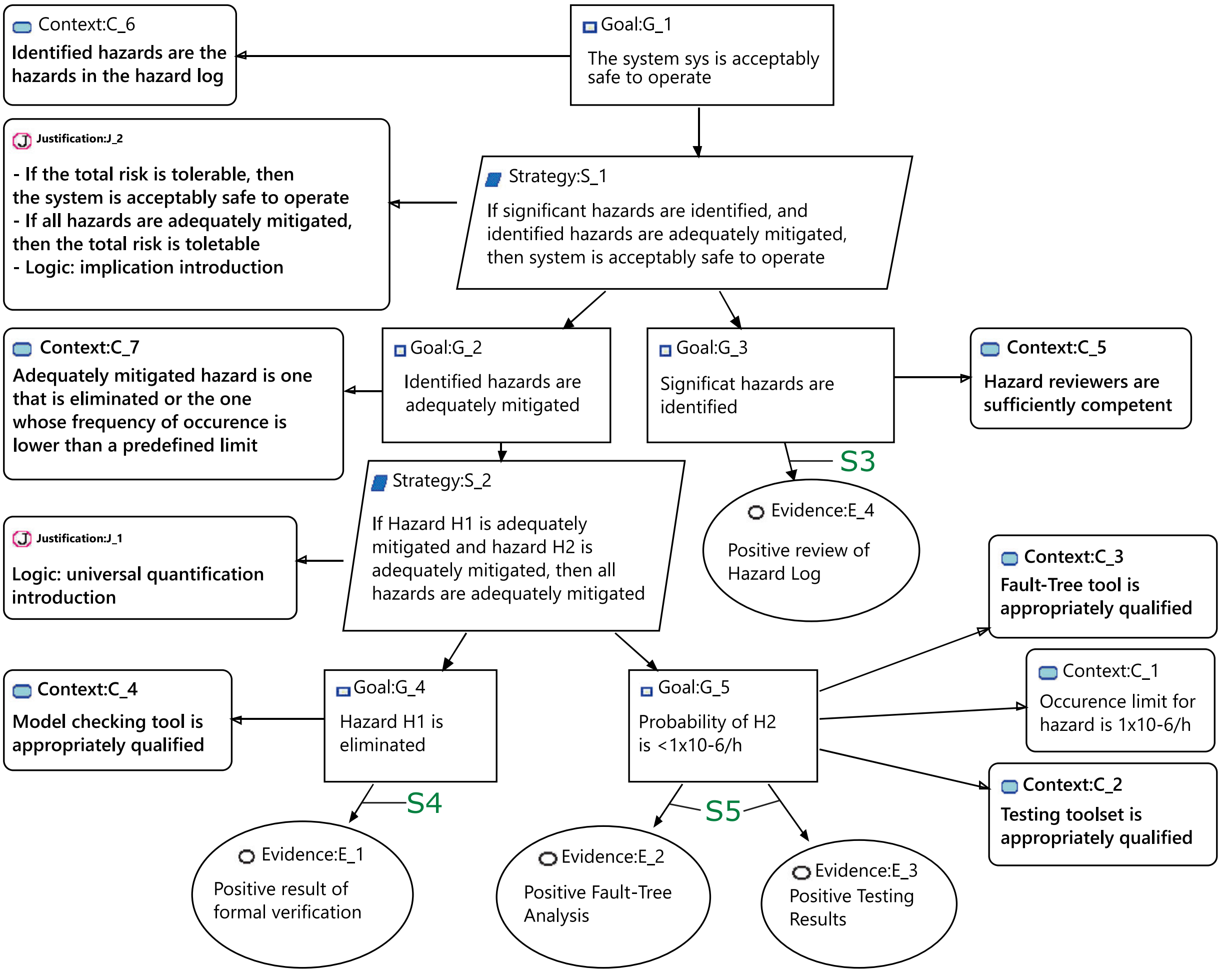
Context:C_6
Identified hazards are the hazards in the hazard log
Goal:G_1
The system sys is acceptably safe to operate
Justification:J_2
- If the total risk is tolerable, then the system is acceptably safe to operate
- If all hazards are adequately mitigated, then the total risk is toletable
- Logic: implication introduction
Strategy:S_1
If significant hazards are identified, and identified hazards are adequately mitigated, then system is acceptably safe to operate
Context:C_7
Adequately mitigated hazard is one that is eliminated or the one whose frequency of occurence is lower than a predefined limit
Goal:G_2
Identified hazards are adequately mitigated
Goal:G_3
Significat hazards are identified
Context:C_5
Hazard reviewers are sufficiently competent
S3
Strategy:S_2
If Hazard H1 is adequately mitigated and hazard H2 is adequately mitigated, then all hazards are adequately mitigated
Evidence:E_4
Positive review of Hazard Log
Justification:J_1
Logic: universal quantification introduction
Context:C_3
Fault-Tree tool is appropriately qualified
Context:C_1
Occurence limit for hazard is 1x10-6/h
Context:C_4
Model checking tool is appropriately qualified
Goal:G_4
Hazard H1 is eliminated
Goal:G_5
Probability of H2 is <1x10-6/h
Context:C_2
Testing toolset is appropriately qualified
S4
S5
Evidence:E_1
Positive result of formal verification
Evidence:E_2
Positive Fault-Tree Analysis
Evidence:E_3
Positive Testing Results

Fig. 6. Safety-case fragment in GSN format which is used for evaluation.

each of the identified hazards has been mitigated. The first hazard is mitigated by showing that it is completely eliminated, and the second one is mitigated by showing that the frequency of occurrence is lower than a predefined limit. Context nodes express the definitions, such as *adequately* mitigated, or refer to goal-nodes that claim that a particular tool is appropriately qualified $C_2 - C_4$, and that the involved personnel is sufficiently competent C_5. The justification nodes J_1, J_2 refer to the *natural deduction* rules of logic, and to the slightly adapted, but generally accepted definitions of *tolerable risk* and *safety* from the ISO 26262 standard.

Given the argumentation structure from Fig. 6, Fig. 7 shows the corresponding Bayesian Network, created according to Table 3. Note the three random variables that represent the implicit inference rules of arguments whose premises are evidence, namely InferenceRuleS3–InferenceRuleS5 in Fig. 7. The Bayesian Network is manually created with the tool *GeNIe Modeler* (Bayes Fusion).

Once the Bayesian Network is created, the CPTs are populated and the overall belief is calculated. According to Table 3, Bayesian Network evidence is asserted for random variables that represent safety-case evidence and axioms, namely for evidence EvidenceE1-EvidenceE4, and for axioms AxiomC1,AxiomsC6,AxiomsC7, AxiomJ1-AxiomJ2. In Fig. 7, the *underlined state* of a random variable denotes that this is the states for which evidence is asserted.

The CPT values for random variables ClaimG1-ClaimG5 are of Type I from Section 6.2. The CPT values for random variables InferenceRule1 -InferenceRule2 are of Type II from Section 6.2. Fortunately, because the asserted axioms and inference rules from Fig. 7 can be expressed as formulas of *predicate logic*, and because it can be proven that the asserted inference rules are *logical consequences* of the asserted axioms, then the *belief* in these inference rules is equal to 1. The CPT values for random variables InferenceRule3-InferenceRule5 are of Type III from Section 6.2 and must be set manually. Because our method differs from ones analyzed in Graydon and Holloway (2016, 2017), these values do not directly correspond to values in Graydon and Holloway (2016, 2017). Because the methods analyzed by Graydon and Holloway (2016, 2017) set the vast majority of the values arbitrarily, and also to rather high values, we set the Type III values in a way to obtain a similar belief in the top claim as in Graydon and Holloway (2016, 2017). Since random variables InferenceRule3 – InferenceRule5 represent the effectiveness of increasingly rigorous verification techniques, namely review, testing, and formal verification, the beliefs are set to 0.9, 0.95, and 0.99, respectively. Finally, if the argument structure from Fig. 6 would be a complete safety case, the CPT values for ClaimC2-ClaimC5 would be of Type I. However, because Fig. 6 contains just a fragment, we manually set these values to 0.99. The impact of particular probability values on the belief calculation is discussed in detail in Section 8.

Given these values, and by using the GeNIe tool, the computed lower limit of the belief in the top claim is 0.81. Comparable value in Graydon and Holloway (2016, 2017) (Table 2) is also around 0.8. The following section applies the modifications from the four counterexamples to the safety case fragment in Fig. 6, and analyzes the changes of the belief in the top claim.

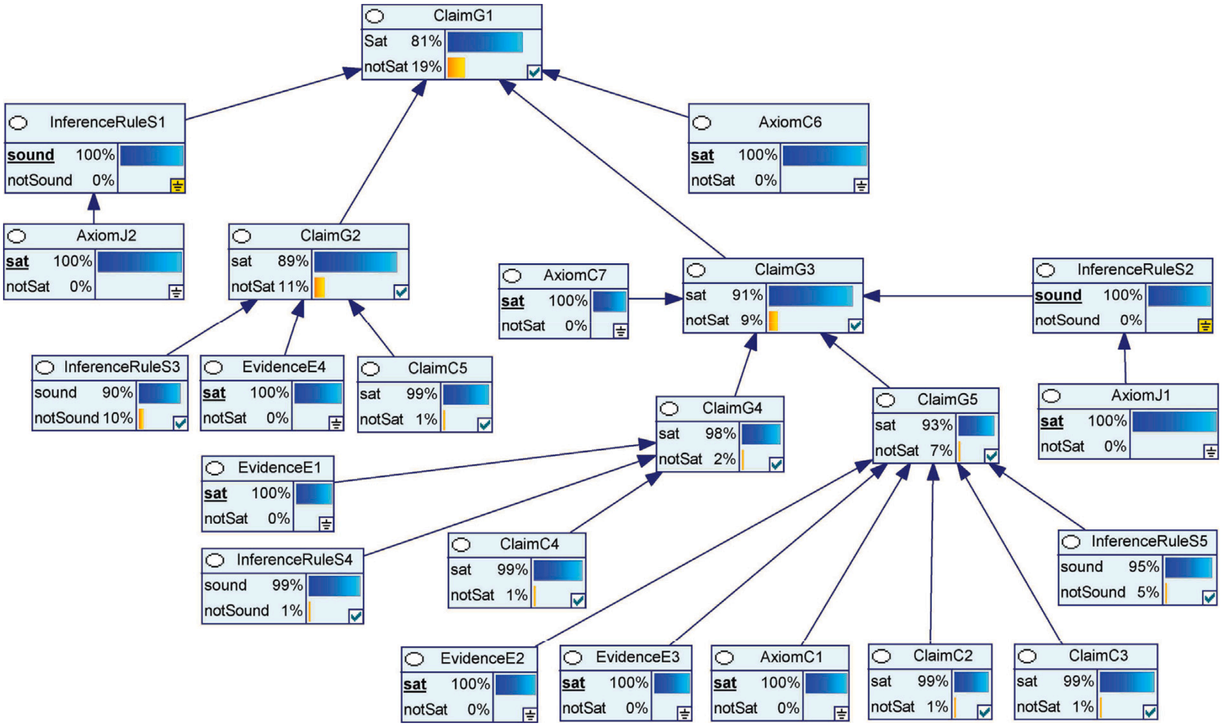

Fig. 7. The Bayesian Network for the GSN argument from Fig. 6, according to Tables 2 and 3.

7.2. Results

In this section we analyze the belief values produced by the Bayesian Network from Fig. 7 for various modifications of the safety case from Fig. 6, according to counterexamples from Graydon and Holloway (2016, 2017). As mentioned previously, four counterexamples are considered and the modifications for all four counterexamples are superimposed in the safety case in Fig. 8. The following four subsections consider the four counterexamples and each starts by describing the counterexample based on Figs. 6 and 8, then follows the modification of the Bayesian Network from Fig. 7 and new belief calculation, and then the comparison to the *expected change* of the belief. For quick reference, Table 4 shows the belief values calculated by the proposed method for different counterexamples, as well as the expected change in values according to Graydon and Holloway (2016, 2017).

7.2.1. Counterexample 1

The first counterexample is named *masked missing-evidence or counter-evidence* and it reveals unrealistic calculations in six of the twelve methods analyzed in Graydon and Holloway (2016, 2017). This counterexample first extends the safety case from Fig. 6 with 18 additional hazards, corresponding goals, and evidence. The additional hazards, except the identifier, are identical to H1. That is, goal G_4.1 is the same as goal G_4 in Fig. 6 and it claims that H1 is eliminated, while goals G_4.2-G_4.19 claim that the additional 18 hazards are also eliminated. Goal G_5 is the same as in Fig. 6. In the Bayesian Network for this counterexample, the CPTs of the additional nodes are populated with the same values as the node ClaimG4 from Fig. 7. Given the additional hazards, the counterexample shows that in the presence of *counter-evidence*, or *missing evidence*, the belief in the top claim is unrealistic. In the case with counter-evidence, newly added evidence E_5 shows that the probability of H2 *is greater* than $1\times10^{-6}/h$. In the case with missing evidence, the evidence referenced by E_1.19 is *missing*. In both cases, the belief in the top claim reduces just a few percent, while the *expected change* is a more significant reduction of the belief.

To establish a *reference belief value*, and without counter-evidence or missing evidence, we extend the Bayesian Network from Fig. 7 with nodes for the 18 additional hazards and compute the belief in the top claim. This is value 1 in Table 4, which is 0.56. The corresponding value in Graydon and Holloway (2016, 2017) is still around 0.8, but that is unrealistic. A system with two and 20 hazards, which are not mitigated with certainty, cannot have the same value for the belief in the claim that the system is acceptably safe. Therefore, the proposed method yields a more realistic value already for the reference value.

To model the case with counter-evidence we use the principle discussed in Section 5.2.2. Since counter-evidence means that there is no *syntactic matching* between the inference rule and the argument premises, the value

$$P(\mathtt{ClaimG5} = sat|\mathtt{InferenceRuleS5} = sound, \mathtt{EvidenceE1} = sat,$$
$$\mathtt{EvidenceE2} = sat, \mathtt{EvidenceE5} = sat, \dots)$$

is set to 0.1 as in Graydon and Holloway (2016, 2017). The case when E_1.19 is *missing*, is conceptually the same as the case with counter-evidence. Because the premise of the argument is *missing*, then there is no *syntactic matching* between the argument and the inference rule. i.e. the value

$$P(\mathtt{ClaimG4.19} = sat|\mathtt{InferenceRuleS4.19} = sound,$$
$$\mathtt{EvidenceE1.19} = sat, \mathtt{ClaimC4} = sat)$$

must be relatively low. Following the counterexample in Graydon and Holloway (2016, 2017), we set this value also to 0.1. Value 2 and 3 in Table 4 show the beliefs values for missing evidence E_1.19, and counter-evidence E_5.

As can be seen from Table 4, the lower limit of the belief in the top claim is much smaller for missing evidence and counter-evidence than for the reference value, as expected. Because Graydon and Holloway (2016, 2017) do not define explicitly how big should the reduction be, it could be argued that our method reduces the belief too much, because there is still high belief that the remaining 19 out of the 20 hazards are adequately mitigated. However, because the definition of safety in the safety case requires that *each hazard is adequately mitigated*, then even if only one identified hazards is not adequately mitigated, the belief in the claim that the system is acceptably safe must suggest that the claim should be rejected.

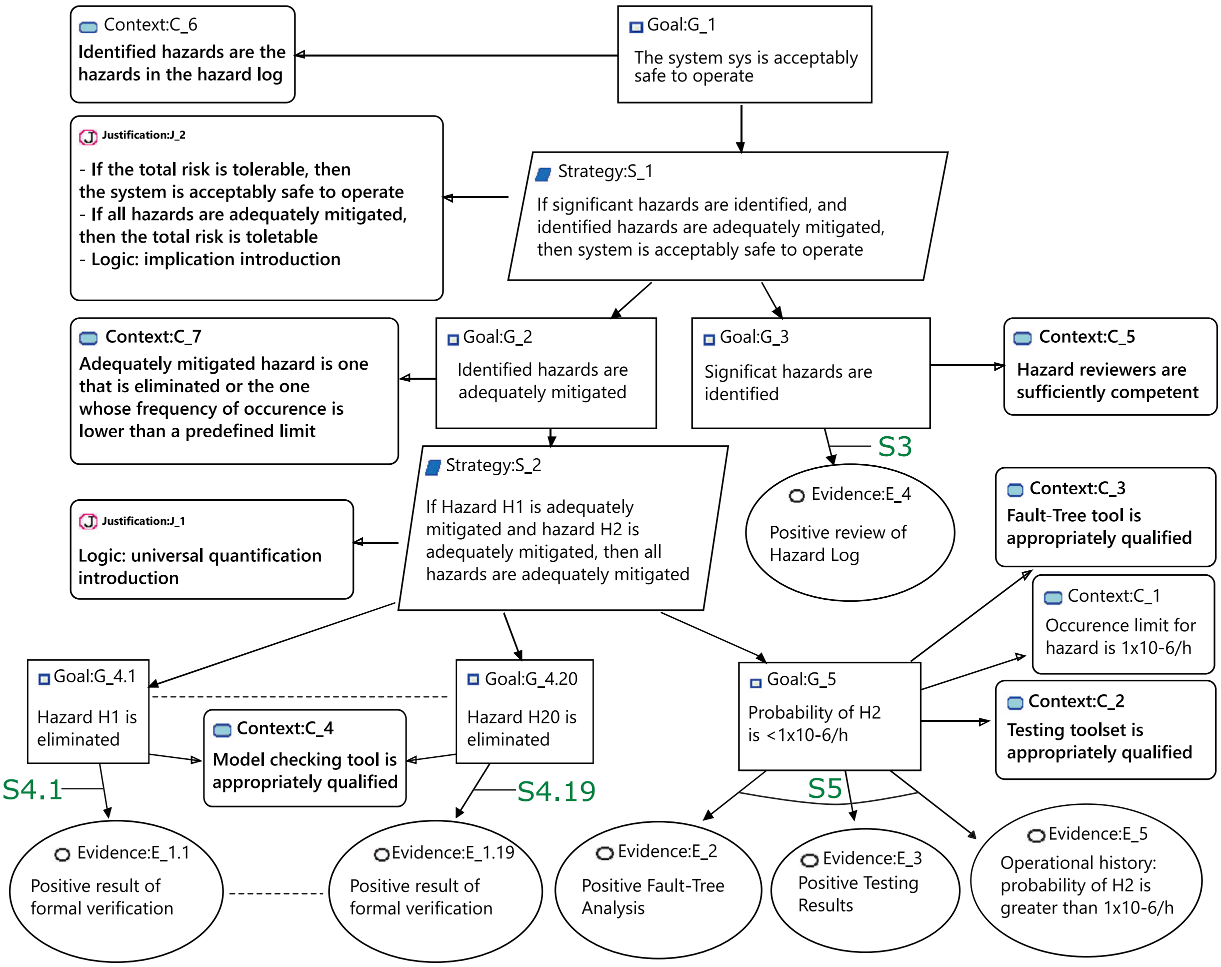
Context:C_6
Identified hazards are the hazards in the hazard log
Goal:G_1
The system sys is acceptably safe to operate
Justification:J_2
- If the total risk is tolerable, then the system is acceptably safe to operate
- If all hazards are adequately mitigated, then the total risk is toletable
- Logic: implication introduction
Strategy:S_1
If significant hazards are identified, and identified hazards are adequately mitigated, then system is acceptably safe to operate
Context:C_7
Adequately mitigated hazard is one that is eliminated or the one whose frequency of occurence is lower than a predefined limit
Goal:G_2
Identified hazards are adequately mitigated
Goal:G_3
Significat hazards are identified
Context:C_5
Hazard reviewers are sufficiently competent
S3
Strategy:S_2
If Hazard H1 is adequately mitigated and hazard H2 is adequately mitigated, then all hazards are adequately mitigated
Justification:J_1
Logic: universal quantification introduction
Evidence:E_4
Positive review of Hazard Log
Context:C_3
Fault-Tree tool is appropriately qualified
Context:C_1
Occurence limit for hazard is 1x10-6/h
Goal:G_4.1
Hazard H1 is eliminated
Goal:G_4.20
Hazard H20 is eliminated
Context:C_4
Model checking tool is appropriately qualified
Goal:G_5
Probability of H2 is <1x10-6/h
Context:C_2
Testing toolset is appropriately qualified
S4.1
S4.19
S5
Evidence:E_1.1
Positive result of formal verification
Evidence:E_1.19
Positive result of formal verification
Evidence:E_2
Positive Fault-Tree Analysis
Evidence:E_3
Positive Testing Results
Evidence:E_5
Operational history: probability of H2 is greater than 1x10-6/h

Fig. 8. Modification of the argument from Fig. 6 according to the counterexamples from Graydon and Holloway (2016, 2017).

7.2.2. Counterexample 2

The second counterexample is named *sensitivity to the arbitrary scope of hazards*, and it reveals unrealistic calculations in three of the twelve methods analyzed in Graydon and Holloway (2016, 2017). Namely, this counterexample reveals drastically different belief in the top claim depending on the number of identified hazards, and with a single missing evidence. More precisely, with only two identified hazards and one missing evidence, the belief in the top claim is high, while with 20 identified hazards and one missing evidence the belief is very low. Essentially, this counterexample compares the safety case fragment from Fig. 6 with one missing evidence, and the safety case fragment from Fig. 8 with one missing evidence (counter-evidence E_5 is removed).

To model this counterexample, we consider that the missing evidence is E_1 in Fig. 6, and E_1.1 in Fig. 8. The belief in the top claim for the case with 20 hazards and missing evidence has already been calculated for counterexample 1, and this is value 3 in Table 4. The belief in the top claim with only two identified hazards and missing evidence E_1 can be calculated from the Bayesian Network in Fig. 7 by setting

$$P(\texttt{ClaimG4} = sat|\texttt{InferenceRuleS4} = sound,$$
$$\texttt{EvidenceE1} = sat, \texttt{ClaimC4} = sat)$$

to 0.1 as in Graydon and Holloway (2016, 2017). Value 4 in Table 4 shows the belief in the top claim for this case.

As can be seen, and unlike the methods analyzed in Graydon and Holloway (2016, 2017), regardless of the number of identified hazards, missing evidence results in a low belief in the top claim because the definition of safety requires all identified hazards to be mitigated. The two beliefs differ slightly because if the mitigation of each identified hazard is somewhat uncertain, and if there are many hazards, then that is reflected as a slightly lower belief in the top claim.

7.2.3. Counterexample 3

The third counterexample from Graydon and Holloway (2016, 2017) is unnamed but it considers *optimistic* versus *pessimistic* scenarios. This counterexample reveals unrealistic calculations in four methods analyzed in Graydon and Holloway (2016, 2017). Namely, one or more of the manually asserted belief values are changed from *high belief values*, i.e. *optimistic values*, to *low belief values*, i.e. *pessimistic values*. The expected change is that in the optimistic case, the belief in the top claim is high, while in the pessimistic case the expected belief value is significantly lower. To model the counterexample, we follow Appendix L from Graydon and Holloway (2016, 2017) and change the belief in the completeness of hazard identification from very high to very low.

Because the manually asserted belief values in the CPTs for Fig. 8 are already very high, we consider that this is the optimistic variant, i.e. value 1 in Table 4 shows the belief in the top claim for the optimistic case. Unlike the two previous counterexamples, in this scenario the required evidence is present, which means that *syntactic matching* exists. The fact that evidence exists, but the belief in the conclusion of the argument is *pessimistic* means either that the process to produce the evidence is probably erroneous, or that this type of evidence does not directly support the corresponding type of conclusion. Because the sources of uncertainty for evidence are not distinguished in the analyzed methods, Graydon and Holloway (2016, 2017) does not specify the *source of the pessimism*. In our method, the uncertainty about

Table 4
Belief in the top claim of the safety case fragment from Fig. 8 for different counterexamples. Far-right column specifies the expected relations between the values.

Nr.	Counterexample	Belief in G_1	Expected change acc. to Graydon and Holloway (2016, 2017)
Value 1	Reference value	0.56	n/a
Value 2	Counter-evidence	0.057	$<$ Value 1
Value 3	Insufficient evidence	0.057	$<$ Value 1
Value 4	Hazard scope - 2 hazards	0.08	$\approx$ Value 3
Value 5	Pessimistic case	0.063	$\ll$ Value 1
Value 6	Imperfect A	0.6	$>$ Value 2, 3, 7, and $\gg$ 0.0
Value 7	Imperfect B	0.19	$>$ Value 2, 3, and $<$ Value 6

evidence is either related to the process of producing the evidence, modeled by context nodes as *implicit premises*, or to the *implicit inference rule*. For this scenario, we choose to model this counterexample by reducing the belief in the implicit inference rule. Consequently, the CPT value $P(\texttt{S3} = \mathit{sound})$ is set to 0.1, as in Graydon and Holloway (2016, 2017). The result would have been the same if the belief in claim C_5, which is an implicit premise, was set to 0.1.

Value 5 in Table 4 shows the lower limit of the belief that the top claim is satisfied for the pessimistic case. Unlike the methods analyzed in Graydon and Holloway (2016, 2017), where the belief in the top claim is reduced by just a few percent in the pessimistic case, Fig. 8 shows a significant difference, as expected. In words, if the belief in the completeness of the hazard identification is very low, then the belief in the safety of the system must also be very low.

7.2.4. Counterexample 4

The fourth counterexample considers two cases of *imperfect evidence*, named *imperfect A* and *imperfect B*. This counterexample reveals unrealistic calculations for a single method analyzed in Graydon and Holloway (2016, 2017). Namely, for the argument structure from Fig. 8 with 20 hazards, the evidence E_1.2 – E_1.19 for the 18 added hazards (counter-evidence E_5 is removed) is *imperfect* to different degrees. What is meant by this is that for arguments where evidence E_1.2 – E_1.19 are premises, the belief in the corresponding conclusions has different values. For the case imperfect A, four evidence perfectly support the corresponding conclusions, 13 evidence are such that belief in the corresponding conclusion is *very high*, and one evidence is such that the belief in the corresponding conclusion is just *high*. For the case imperfect B, four evidence is perfect, one evidence is such that the belief in the corresponding conclusion is *very high*, and 13 evidence is such that the belief in the corresponding conclusion is just *high*. The counterexample reveals three issues with belief calculations. Firstly, the belief that the system is acceptably safe is *very low* for imperfect A, although the majority of hazards are mitigated with very high belief. Secondly, the belief in the top claim of imperfect A and B is similar, despite the 13 very high beliefs in imperfect A versus just one very high belief in imperfect B. Thirdly, the belief for imperfect A and B is similar to the cases with counter-evidence and missing evidence from counterexample 1, although this should not be the case.

To model this counterexample, as in counterexample 3, the CPTs of the variables that represents the implicit inference rules is adjusted. Because Graydon and Holloway (2016, 2017) contains only qualitative values for this scenario, we interpret *perfect* evidence as belief value 1, *very high* as value 0.99, and *high* as value 0.9. For imperfect A, we set the four beliefs for S_4.2 – S_4.5 to 1, the 13 beliefs for S_4.6 – S_4.18 to 0.99, and belief for S_4.19 to 0.9. For imperfect B, the same CPT values are modified but only with the ratio four with belief 1, one with belief 0.99, and 13 with belief 0.9. Values 6 and 7 in Table 4 show the belief in the top claim for imperfect A and imperfect B.

Regarding the first issue, and as expected by Graydon and Holloway (2016, 2017), our method calculates the belief value for imperfect A that is relatively high, and it certainly cannot be interpreted as a suggestion to consider the top claim false. Secondly, again as expected, the belief for imperfect A is significantly higher than the belief for imperfect B because imperfect B incorporates much more uncertainty. Finally, both the belief for imperfect A and imperfect B are much higher than the values for missing and counter-evidence. Therefore, with respect to all three issue, our method produces belief values in the top claim as expected.

8. Discussion

This section discusses the benefits and limitations of the proposed method with respect to decision-making based on the calculated belief values, and also the practical issues such as using other notations for the safety case, or working with large Bayesian Networks.

8.1. Decision making based on the belief values

The most important consideration is the use of the proposed method in terms of the numerical values it produces. As any other quantitative technique, the purpose of quantifying the belief in the top claim of a safety case is for *decision-making*. There are two types of decisions that can be made based on the proposed method. The first one is to decide if the lower limit of belief in the top safety-case claim is higher than a predefined threshold, and thus conclude that the top claim should be considered to be true. The second type of decision is which part of the safety case should be improved, e.g. by creating additional evidence or by changing an inference rule, in order to increase the belief in the top claim the most.

8.1.1. Change-impact analysis

First we consider the second type of decision-making. Identifying the beliefs whose increase would led to the greatest increase of the belief in the top claim can be achieved by performing *sensitivity analysis* (Castillo et al., 1997) of the Bayesian Network. Sensitivity analysis of Bayesian Networks is a mature analysis method and is often readily available in tools for editing Bayesian Network, e.g. in GeNIe Modeler. The goal of sensitivity analysis is to assess how sensitive particular probability values are, in GeNIE modeler called a *target node*, to small changes of other probability values. Fig. 9 shows the Bayesian Network from Fig. 7 with the CPT values from counterexample 2, namely the *sensitivity to scope of hazards*, where the target node is ClaimG1. The sensitivity to changes in other probability values is indicated by the heat map, where the highest sensitivity is indicated by red color and the lowest with gray color.

Because in counterexample 2 evidence E_1 is missing, as expected, Fig. 9 indicates that the probability values of the target node ClaimG1 are most sensitive to changes of probability values for node ClaimG4. The reason for this is that the CPT of node ClaimG4 encodes the value that represents the absence of syntactic matching between the used inference rule and the premises. It follows that the belief whose increase would lead to the biggest increase of the belief in ClaimG1, is the belief in ClaimG4, i.e. the missing evidence E_1 should be provided. The usefulness of sensitivity analysis is especially visible in large Bayesian Networks where the probability values in CPTs are not so drastically different such as in counterexample 2.

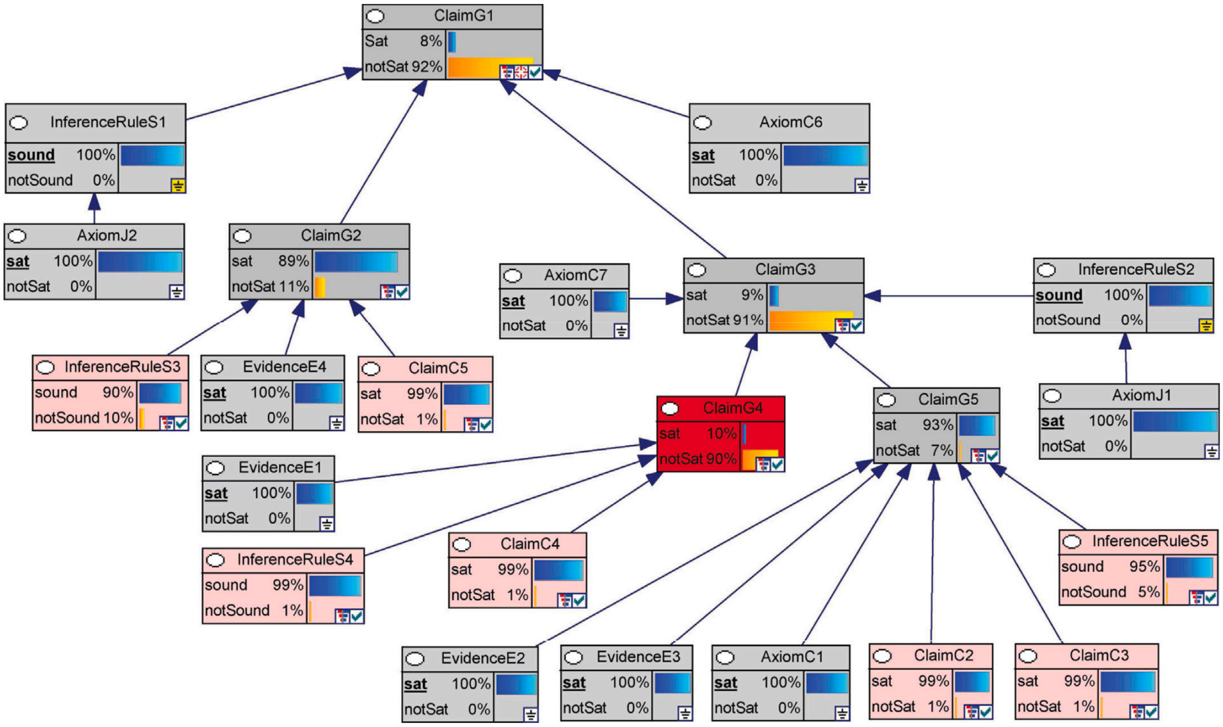

Fig. 9. Sensitivity analysis of the Bayesian Network for counterexample 2 with only 2 hazards (target node is ClaimG1.). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

8.1.2. Stop/continue development/deployment

The first type of decision that can be made based on the belief values is the decision whether the calculated belief in the top claim is sufficiently high to indirectly conclude that system is acceptably safe. The evaluation in Section 7 has shown that the belief in the top safety-case claim changes as expected for various realistic, and extreme changes of the underlying argument structure, and probability values. This result strongly indicates that the calculation of the belief is sound, and that if the CPTs are populated with reliable probability values, then the belief in the top claim can be trusted. Moreover, the description of the three types of CPT values in Section 6.3 has pointed to the different sources of reliable probability values.

This leaves the question of what is an appropriate threshold for the lower limit of the belief in order to decide that a claim should be considered true. Our conjecture is that such value is specific to each company or even a particular system, and moreover, this value is different in different stages of the system lifecycle. Therefore, identifying the parameters which impact the threshold value, and developing a model to calculate the value, is left as a topic of future work. However, here we present some preliminary ideas towards such model.

First and foremost, the belief in the top claim depends on the *exact numerical values* in the CPTs. More precisely, the majority of CPTs will be of Type I. The values of Type II and III will in some cases be possible to obtain from literature, standards, or by doing formal proofs. However, some values will have to be assigned manually. Depending on the choice of values that correspond to qualitative estimates such as *high*, *very high*, or *very low*, the threshold will have to be set differently. Secondly, and as evident by comparing the belief in the top claim for the safety case fragment with two identified hazards and for 20 identified hazards, the more random variables that represent belief less than 1, the lesser is the belief in the top claim. This behavior of the model also matches the intuition; the more sources of uncertainty, the lower the overall belief and this means that the threshold must depend on the complexity of the system and the corresponding safety case. Thirdly, the threshold value must be different in different stages of the safety lifecycle. Initially, the belief in the top claim will be very low and as more evidence is added the belief in the top claim will rise. However, it might be useful to judge if the belief in the *safety concept* is sufficiently high, before proceed to actual *system development* and in this case multiple different threshold values would be needed. Finally, the threshold value must be chosen such that the belief in the top claim of a safety case for a system that is known to be safe, is higher than the threshold. Therefore, it is realistic to assume that a *baseline* threshold value will be derived based on a safety case of a system that is shown to be safe, and then tuned for different companies, probability-value scales etc.

8.2. Practical aspects of using the method

The proposed method is independent of any concrete safety case notation. Just as Table 2 presents a mapping from a GSN argument structure into the representation in terms of formulas of a language $\mathcal{L}$, a mapping from other notations such as CAE, SACM, or NOR-STA could also be defined. Furthermore, the method is general in the sense that it is independent of any particular standard, domain, implementation technology, or engineering process. However, in a concrete usage scenario, the set of axioms $\mathcal{A}$ will have to be selected, and then the probability values in the Bayesian Network become specific for this scenario. For example, the argumentation structure in Fig. 6 adopts the axioms of ISO 26262 which makes the argument applicable to the automotive domain.

An additional benefit of the mappings in Tables 2 and 3, is that they can be used to build a tool that *automatically constructs* Bayesian Networks for a given safety case. Such tool is essential for industrial acceptance of the proposed method because real-world safety cases are huge artifacts. Such tool would also allow additional simplifications to be made, in order to bring the method closer to typical process of safety-case creation. Namely, the responsibility for enforcing some well-formedness constraints could be moved from the safety-case creator to such tool. For example, strategy nodes could be optional between goal nodes, but the tool would warn the user that it is considered that the omitted strategy node expresses an inference rules which is known to be *sound*.

When it comes to the practical aspects of working with large safety cases, and consequently large Bayesian Networks, there are several simplifications that can be employed. Firstly, as it can be seen from Fig. 9, the Bayesian Network nodes, which have received evidence, are irrelevant for the calculation of the belief in the top claim. Therefore, to avoid maintaining a large Bayesian Network, all nodes that represent evidence, and axioms can be removed, and the conditional-probability tables of their children nodes can be modified accordingly. In such scenario, the evidence and the axioms act as *background information* for the complete Bayesian Network. Also, if a Bayesian Network is still very large, Bayesian Network editors such as GeNIe allow partitioning a large model into several smaller ones.

9. Related work

Both the formal representation of safety-case content, and confidence assessment in the top claim of a safety case have been the subject of previous research. These two types of related work are reviewed in the following sections.

9.1. Formal representation of safety-case content

In recent years, a growing number of research contribution are proposing the formalization of safety cases in order to enable automated safety-case construction and analysis (Diskin et al., 2018; Nemouchi et al., 2019; Nešić et al., 2019; Prokhorova et al., 2015; Rushby, 2015, 2017). Work in Diskin et al. (2018) presents a method which is an alternative to notations in Section 2.3, called *model-transformation based assurance*. The central idea is that safety-case creation can be defined as a function f which takes as input engineering data, and outputs assurance data. When creating the assurance data, the function f must satisfy certain constraints, and conceptually this is the definition of *model-transformations*. Consequently, the assurance case is defined by the input engineering data, in the form of various *meta-models*, the function f, and the constraints that the function f must satisfy. The work in Nemouchi et al. (2019) presents a syntactic extension of the *Isabelle theorem prover* (Paulson, 1994) that allows manual creation of textual GSN arguments. By using the built-in Isabelle capabilities, it is ensured that a variety of structural constraints, such as the ones in Definition 11, are satisfied. Moreover, the method shows how claims about a software satisfying a requirement can be formalized and verified in Isabelle, and then the verification results can be referenced within GSN arguments. The method in Nešić et al. (2019) presents a formal structure, based on the *contract-based design* paradigm (Benveniste et al., 2018), which contains the *component-based architecture* and the corresponding *assume-guarantee specification* of a *configurable system*. Given such formal structure, the work in Nešić et al. (2019) defines the translation from this structure into a GSN argument, which claims that the configurable system satisfies the allocated requirements in all configurations of the system. In a conceptually similar way to Nemouchi et al. (2019), the work in Prokhorova et al. (2015) presents a methodology for formalizing functional safety-requirements and system models in *Event-B language* Abrial (2010). Then, by using *theorem proving*, the functional safety requirements are verified against the system model, and the verification results are the evidence for several GSN arguments.

The proposed method differs from each the discussed methods. The work in Nemouchi et al. (2019), Nešić et al. (2019) and Prokhorova et al. (2015) introduce a formal model which is a proxy for a small part of a safety case. In other words, these methods identify claims that can be formalized in particular formal theory, but do not provide a framework to formalize arbitrary claims. The work in Diskin et al. (2018) does not even maintain an explicit safety-case-like artifact, but rather places the focus on establishing the formal framework where the input engineering artifacts and a transformation function can be defined.

9.2. Confidence assessment

When it comes to methods for belief, or confidence, assessment in safety cases, the literature contains two types of methods; qualitative, and quantitative. The quantitative methods (Wang et al., 2019; Cyra and Górski, 2011; Denney et al., 2011; Zhao et al., 2012; Bishop et al., 2011; Hobbs and Lloyd, 2012) primarily differ with respect to the used framework for uncertainty modeling. The two most prominent ones are the *Dempster–Shafer theory* (D–S) (Shafer, 1976) and classical probability theory, typically in the Bayesian sense. Because the method in the present paper relies on probability theory, other such methods are analyzed in detail, but also some of the most notable D–S-based methods. Table 5 shows a comparison between the different methods.

When it comes to the methods based on the D–S theory, Cyra and Górski (2011), Wang et al. (2019) and Ayoub et al. (2013) are three of the most mature methods. Both the work in Cyra and Górski (2011), Ayoub et al. (2013), and the predecessor of Wang et al. (2019) in Guiochet et al. (2015), were included in the evaluation by Graydon and Holloway (2016, 2017), and each of the methods failed to handle one or more counterexamples. There are several major differences between (Cyra and Górski, 2011; Wang et al., 2019; Ayoub et al., 2013) and the method in the present paper. Firstly, Cyra and Górski (2011) and Wang et al. (2019) simply reuse the syntactic definitions of safety-case elements from NOR-STA or GSN notation, and do not define their semantics, e.g. when a *claim* is *true*. Instead, methods in (Cyra and Górski, 2011; Wang et al., 2019; Ayoub et al., 2013) differentiate between *types of arguments* based on the intended inference rule, and for each type of argument define a specific function to compute the belief in the corresponding conclusion. The calculated belief values depend on the user-provided belief in evidence, e.g. given a positive review that a document is correct, what is the belief that the document is correct. Consequently, the overall belief and its semantics is purely subjective. Also, because evidence in safety cases are artifacts produced by verification activities which *categorically verify* that a certain claim is *true* or *false*, the belief in evidence can only correspond to the question of whether the *verification activity, not the result, is flawless*. While in Cyra and Górski (2011), Wang et al. (2019) and Ayoub et al. (2013) this distinction is not made, in our method this is achieved by requiring the creation of *implicit premises* that refer to claims about the correctness of the *verification activity*. Finally, to avoid the issues of combining multiple beliefs with the rules of D–S theory, Wang et al. (2019) assumes that each argument within a safety is transformed into an argument with *at most two premises* while our approach accepts arguments with an arbitrary number of premises.

Methods in Denney et al. (2011), Zhao et al. (2012), Bishop et al. (2011), Ayoub et al. (2013), Hobbs and Lloyd (2012) and Rushby (2017) are all based on Bayesian Networks. The work in Denney et al. (2011) outlines a conceptually similar method to the one in the present paper where a GSN argument is transformed into a Bayesian Network, and the belief in each argument conclusion is a function of the belief in the argument premises. However, the method is introduced through an example and the procedure to construct the Bayesian Network for the given safety-case argument is not presented. Also, in the analysis by Graydon, the method in Denney et al. (2011) is insensitive to drastic drops in belief about critical claims because premises of arguments are weighted but it is not defined how and when to adjust the weights. The work in Zhao et al. (2012) approaches belief calculations similarly, with the difference that first an arbitrary GSN argument is transformed into a *Toulmin* argument Toulmin (2003), and then into a Bayesian Network. The belief model is based on six sources of uncertainty to be quantified for each argument. These sources of uncertainty come from the informal, argument-assessment criteria from Hitchcock (2005). However, as Graydon and Holloway (2016, 2017) shows, the method suffers from the same issue as Denney et al. (2011). The method in Bishop et al. (2011) does not consider safety-case arguments in general, but a specific scenario where with

Table 5
Properties of different methods for belief calculation.

Source	Def. of safety-case elements	Considered arguments	Belief-model based on	Computed belief function of
Cyra and Górski (2011)	NOR-STA defs.	Five argument patterns	Argument structure	Belief in evidence
Wang et al. (2019)	GSN defs.	Eight argument patterns	Argument structure	Belief in evidence
Denney et al. (2011)	GSN defs.	Specific argument	Argument structure	Belief in argument premises
Zhao et al. (2012)	GSN/CAE defs.	Arbitrary argument	Hitchcock criteria (Hitchcock, 2005)	Values for criteria from Hitchcock (2005)
Bishop et al. (2011)	NA	Specific argument	Argument structure	Amount of evidence
Ayoub et al. (2013)	NA	Four argument patterns	Argument structure	Belief in evidence
Hobbs and Lloyd (2012)	NA	Four argument patterns	Patterns from Marsh (1999)	Belief in evidence
Rushby (2017)	Claim/assumption as propositions, argument as implication	Arbitrary arguments	Argument structure	Belief in evidence
Present paper	All elements in model theory	Arbitrary argument	Argument structure and content	Belief in inference rules, process to produce evidence, matching of evidence and inference rules

high confidence it is claimed that the component's probability of failure on demand is θ. Given such claim, the goal of the method is to develop a probabilistic model which allows concluding *with near certainty* a weaker claim about θ, i.e. that component's probability of failure on demand is greater than θ. Obviously, the method has a slightly different focus than the present paper because the input is not a safety-case but a specific argument. Also, the computed belief values are a function of *the amount of positive evidence* which fits well for the specific claim being considered. The method in Hobbs and Lloyd (2012) proposes to express safety-case claims directly as discrete random variables within a Bayesian Network, where the random variables can be in states true or false. The method distinguishes between the use of Bayesian Networks to represent safety-case evidence, and the usage to represent the safety-case arguments where the possible arguments are defined through several possible patterns, referred to as *idioms*. However, the method is presented through a number of safety and non-safety-related examples, and no formal foundation for the method is provided. As a consequence, even reproducing the results is challenging, as shown by Graydon. The work in Rushby (2015, 2017) introduces the idea that an argument with non-evidence premises can be formalized in *propositional logic*, while arguments with evidence as premises can be formalized in *probability theory*, i.e. in as *Bayesian Networks*. This line of work has a similar intention as the method in the present paper, but the formalization uses propositional logic which cannot express the fine-grained structure of claims, and it does not consider all elements of safety-case arguments, e.g. inference rules or contextual information. Furthermore, the integration between the probabilistic reasoning and the propositional logic is not explicitly defined, thus it is unclear how to assess if a top safety claim should be accepted or rejected.

10. Conclusion

Because the knowledge about systems is typically *imperfect*, the corresponding safety cases are riddled with uncertainties, and therefore, safety cases cannot show *with absolute certainty* that systems are acceptably safe. To measure the degree of uncertainty, the *belief* that safety-case claims are true may be calcualted.

This paper has presented a *novel method* for probabilistic calculation of belief in a safety case. The *major result* is that unlike previous methods, the produced *belief values* are *realistic* for safety cases of different sizes, structure, and even for incomplete safety cases. This result is to a good extent the consequence of the first contribution, namely the *formal definitions of safety-case elements*, based on the principles of model theory, and independent of particular safety-case notations. The significance of these definitions is that the typically *omitted, implicit, or unclear* information, e.g. the underlying axioms or the structure of used inference rules, can be made *explicit*. Guided by these definitions, the second contribution is a *general, probabilistic model* of belief in conclusions of arbitrary safety-case arguments, where the uncertainty is captured in a *consistent and uniform* way across the different arguments. Consequently, the third contribution, namely the application of the method to typically natural-language safety cases in *Goal-Structuring Notation* (GSN), reveals the need for additional *well-formedness* constraints that make a GSN safety case sufficiently complete for *reliable* belief calculations. In this sense, the presented method can be seen as a method to *systematically* create safety cases, for which the belief is *high by construction*.

The benefit of calculating *an absolute value* of belief in a safety case is that it can be used to decide whether a system is acceptably safe. Although the presented method allows calculating the absolute belief value, defining the *threshold value* above which a system can be considered safe is left as future work because such value depends on the properties of particular systems. The main practical benefit of the presented method is the ability to analyze the belief in a safety case *relative* to different trade-offs. In scenarios when there is no time to produce all intended evidence, or when a supplier is replaced to optimize development costs, the presented method allows assessing the impact of such changes on the belief in safety-case claims. In this way, a system can be optimized with respect to various business and engineering criteria, while ensuring a high belief in the corresponding safety-case claims.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Damir Nešić and Mattias Nyberg were supported by the ECSEL PRYSTINE project (No 783190). Barbara Gallina was partially supported by Sweden's Knowledge Foundation via the SACSys project. The authors would like to thank the reviewers for their constructive feedback.

References

Abrial, J.-R., 2010. Modeling in Event-B: System and Software Engineering. Cambridge University Press.
Adelard LLP, 2020. Claim, argument, evidence. URL https://claimsargumentsevidence.org/.
Anon, 2003. EN 50129: Railway Applications - Communication, Signaling and Processing Systems - Safety Related Electronic Systems for Signaling, Standard. CENELEC.

Anon, 2020. Structured Assurance Case Metamodel, Standard. Object Management Group, Formal version. URL https://www.omg.org/spec/SACM/2.1, April.

Athreya, K.B., Lahiri, S.N., 2006. Measure Theory and Probability Theory. Springer Science & Business Media.

Ayoub, A., Chang, J., Sokolsky, O., Lee, I., 2013. Assessing the overall sufficiency of safety arguments. In: 21st Safety-Critical Systems Symposium, SSS.

Bayes Fusion, 0000. GeNIe modeler, version 2.3 academic. URL https://www.bayesfusion.com/genie/.

Bench-Capon, T., Dunne, P.E., 2007. Argumentation in artificial intelligence. Artificial Intelligence 171 (10), 619–641. http://dx.doi.org/10.1016/j.artint.2007.05.001.

Benveniste, A., Caillaud, B., Nickovic, D., et al., 2018. Contracts for system design. Found. Trends Electron. Des. Autom. 12 (2–3), 124–400. http://dx.doi.org/10.1561/1000000053.

Bishop, P., Bloomfield, R., Littlewood, B., et al., 2011. Toward a formalism for conservative claims about the dependability of software-based systems. IEEE Trans. Softw. Eng. 37 (5), 708–717. http://dx.doi.org/10.1109/TSE.2010.67.

Bloomfield, R., Littlewood, B., Wright, D., 2007. Confidence: Its role in dependability cases for risk assessment. In: International Conference on Dependable Systems and Networks, DSN. pp. 338–346.

Castillo, E., Gutiérrez, J.M., Hadi, A.S., 1997. Sensitivity analysis in discrete Bayesian networks. IEEE Trans. Syst. Man Cybern. A 27 (4), 412–423. http://dx.doi.org/10.1109/3468.594909.

Chowdhury, T., Wassyng, A., Paige, R.F., Lawford, M., 2019. Criteria to systematically evaluate (safety) assurance cases. In: 30th International Symposium on Software Reliability Engineering. ISSRE, IEEE, pp. 380–390. http://dx.doi.org/10.1109/ISSRE.2019.00045.

Cyra, L., Górski, J., 2011. Support for argument structures review and assessment. Reliab. Eng. Syst. Saf. 96 (1), 26–37. http://dx.doi.org/10.1016/j.ress.2010.06.027.

David, A., Larsen, K.G., Legay, A., Mikučionis, M., Wang, Z., 2011. Time for statistical model checking of real-time systems. In: Computer Aided Verification. CAV, Springer, pp. 349–355.

Denney, E., Pai, G., 2018. Tool support for assurance case development. Autom. Softw. Eng. 25 (3), 435–499. http://dx.doi.org/10.1007/s10515-017-0230-5.

Denney, E., Pai, G., Habli, I., 2011. Towards measurement of confidence in safety cases. In: International Symposium on Empirical Software Engineering and Measurement. ESEM, pp. 380–383. http://dx.doi.org/10.1109/ESEM.2011.53.

Dezert, J., Wang, P., Tchamova, A., 2012. On the validity of Dempster–Shafer theory. In: 15th International Conference on Information Fusion. ISIF, IEEE, pp. 655–660.

Diskin, Z., Maibaum, T., Wassyng, A., et al., 2018. Assurance via model transformations and their hierarchical refinement. In: 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems. MODELS, pp. 426–436. http://dx.doi.org/10.1145/3239372.3239413.

Doets, K., 1996. Basic Model Theory. In: Studies in Logic, Language and Information, CSLI.

Duan, L., Rayadurgam, S., Heimdahl, M.P.E., et al., 2017. Reasoning about confidence and uncertainty in assurance cases: A survey. In: International Symposium on Software Engineering in Health Care. SEHC, Springer, pp. 64–80. http://dx.doi.org/10.1007/978-3-319-63194-3_5.

Dung, P.M., 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77 (2), 321–357. http://dx.doi.org/10.1016/0004-3702(94)00041-X.

Edmundson, A., Holtkamp, B., Rivera, E., et al., 2013. An empirical study on the effectiveness of security code review. In: International Symposium on Engineering Secure Software and Systems. ESSoS, Springer, pp. 197–212. http://dx.doi.org/10.1007/978-3-642-36563-8_14.

Galton, A., 1990. Logic for Information Technology. John Wiley and Sons, Inc., USA.

Górski, J., Jarzębowicz, A., Miler, J., et al., 2012. Supporting assurance by evidence-based argument services. In: International Conference on Computer Safety, Reliability, and Security. SAFECOMP, Springer, pp. 417–426. http://dx.doi.org/10.1007/978-3-642-33675-1_39.

Graves, T.L., Harrold, M.J., Kim, J.-M., et al., 2001. An empirical study of regression test selection techniques. ACM Trans. Softw. Eng. Methodol. (TOSEM) 10 (2), 184–208. http://dx.doi.org/10.1145/367008.367020.

Graydon, P.J., 2014. Towards a clearer understanding of context and its role in assurance argument confidence. In: Computer Safety, Reliability, and Security. SAFECOMP, pp. 139–154. http://dx.doi.org/10.1007/978-3-319-10506-2_10.

Graydon, P.J., Holloway, C.M., 2016. An Investigation of Proposed Techniques for Quantifying Confidence in Assurance Arguments. Tech. Rep. NASA/TM-2016-219195, NASA.

Graydon, P.J., Holloway, C.M., 2017. An investigation of proposed techniques for quantifying confidence in assurance arguments. Saf. Sci. 92, 53–65. http://dx.doi.org/10.1016/j.ssci.2016.09.014.

Greenwell, W.S., 2006. A taxonomy of fallacies in system safety arguments. In: International System Safety Conference, SSS.

Guiochet, J., Do Hoang, Q.A., Kaaniche, M., 2015. A model for safety case confidence assessment. In: Computer Safety, Reliability, and Security. Springer International Publishing, Cham, pp. 313–327. http://dx.doi.org/10.1007/978-3-319-24255-2_23.

Hauer, F., Schmidt, T., Holzmüller, B., Pretschner, A., 2019. Did we test all scenarios for automated and autonomous driving systems? In: Intelligent Transportation Systems Conference, ITSC. pp. 2950–2955.

Hitchcock, D., 2005. Good reasoning on the Toulmin model. Argumentation 19 (3), 373–391.

Hobbs, C., Lloyd, M., 2012. The application of Bayesian belief networks to assurance case preparation. In: 20th Safety-Critical Systems Symposium. Springer, London, pp. 159–176. http://dx.doi.org/10.1007/978-1-4471-2494-8_12.

Huth, M., Ryan, M., 2004. Logic in Computer Science: Modelling and Reasoning about Systems. Cambridge University Press, http://dx.doi.org/10.1017/CBO9780511810275.

International Organization for Standardization, 2018. ISO 26262: Road vehicles - Functional safety.

Jaynes, E.T., 2003. Probability Theory: The Logic of Science. Cambridge University Press, http://dx.doi.org/10.1017/CBO9780511790423.

Juristo, N., Moreno, A.M., Vegas, S., 2004. Reviewing 25 years of testing technique experiments. Empir. Softw. Eng. 9 (1–2), 7–44. http://dx.doi.org/10.1023/B:EMSE.0000013513.48963.1b.

Kelly, T., 2007. Reviewing assurance arguments-a step-by-step approach. In: Workshop on Assurance Cases for Security-the Metrics Challenge, Dependable Systems and Networks, DSN.

Langari, Z., Maibaum, T., 2013. Safety cases: A review of challenges. In: International Workshop on Assurance Cases for Software-Intensive Systems, ASSURE. pp. 1–6.

Leveson, N.G., 2011. The Use of Safety Cases in Certification and Regulation. Massachusetts Institute of Technology, Engineering Systems Division.

Marker, D., 2006. Model Theory: An Introduction, Vol. 217. Springer Science & Business Media, http://dx.doi.org/10.1007/b98860.

Marsh, W., 1999. Safety and Risk Evaluation Using Bayesian NEts: SERENE. Tech. Rep. SERENE/5.3/CSR/3053/R/1, ERA Technol., Surrey, UK.

Matsuno, Y., Takamura, H., Ishikawa, Y., 2010. A dependability case editor with pattern library. In: 2010 IEEE 12th International Symposium on High Assurance Systems Engineering. IEEE, pp. 170–171.

Nemouchi, Y., Foster, S., Gleirscher, M., Kelly, T., 2019. Isabelle/SACM: Computer-assisted assurance cases with integrated formal methods. In: Integrated Formal Methods. IFM, Springer, pp. 379–398. http://dx.doi.org/10.1007/978-3-030-34968-4_21.

Nešić, D., Nyberg, M., Gallina, B., 2019. Constructing product-line safety cases from contract-based specifications. In: 34th ACM/SIGAPP Symposium on Applied Computing. ACM, pp. 2022–2031. http://dx.doi.org/10.1145/3297280.3297479.

Nielsen, T.D., Jensen, V., 2009. Bayesian Networks and Decision Graphs. Springer Science & Business Media, http://dx.doi.org/10.1007/978-0-387-68282-2.

Origin Consulting (York) Limited, 2018. GSN community standard version 2. URL https://scsc.uk/publications, Jan.

Paulson, L.C., 1994. Isabelle: A Generic Theorem Prover, Vol. 828. Springer Science & Business Media, http://dx.doi.org/10.1007/BFb0030541.

Polya, G., 1990. Mathematics and Plausible Reasoning: Patterns of Plausible Inference, Vol. 2. Princeton University Press.

Prokhorova, Y., Laibinis, L., Troubitsyna, E., 2015. Facilitating construction of safety cases from formal models in Event-B. Inf. Softw. Technol. 60, 51–76. http://dx.doi.org/10.1016/j.infsof.2015.01.001.

Riveret, R., Baroni, P., Gao, Y., Governatori, G., Rotolo, A., Sartor, G., 2018. A labelling framework for probabilistic argumentation. Ann. Math. Artif. Intell. 83 (1), 21–71. http://dx.doi.org/10.1007/s10472-018-9574-1.

Runeson, P., Andersson, C., Thelin, T., et al., 2006. What do we know about defect detection methods? IEEE Softw. 23 (3), 82–90. http://dx.doi.org/10.1109/MS.2006.89.

Rushby, J., 2015. The Interpretation and Evaluation of Assurance Cases. Tech. Rep. SRI-CSL-15-01, Computer Science Laboratory, SRI International.

Rushby, J., 2017. On the interpretation of assurance case arguments. In: New Frontiers in Artificial Intelligence. Springer, pp. 331–347. http://dx.doi.org/10.1007/978-3-319-50953-2_23.

Shafer, G., 1976. A Mathematical Theory of Evidence, Vol. 42. Princeton University Press.

Szczygielska, M., Jarzębowicz, A., 2018. Assurance case patterns on-line catalogue. In: International Conference on Dependability and Complex Systems. DepCoS-RELCOMEX, pp. 407–417. http://dx.doi.org/10.1007/978-3-319-59415-6_39.

The International Electrotechnical Commission, 2010. ISO 61508: Functional safety.

Toulmin, S.E., 2003. The Uses of Argument. Cambridge University Press, http://dx.doi.org/10.1017/CBO9780511840005.

UK GoV, 1992. The Offshore Installations (Safety Case) Regulations. National Legislation, Parliament of United Kingdom.

UK GoV, 1994. The Railway Installations (Safety Case) Regulations. National Legislation, Parliament of United Kingdom.

UL4600 Task Group, 2020. UL4600 - Standard for the Evaluation of Autonomous Products. Underwriters Laboratories (UL), URL https://ul.org/UL4600, April.

Walton, D.N., 1996. Argumentation Schemes for Presumptive Reasoning. Psychology Press.

Walton, D., 2008. Informal Logic: A Pragmatic Approach. Cambridge University Press.

Wang, R., Guiochet, J., Motet, G., 2017. Confidence assessment framework for safety arguments. In: Computer Safety, Reliability, and Security. SAFECOMP, Springer, pp. 55–68. http://dx.doi.org/10.1007/978-3-319-66266-4_4.

Wang, R., Guiochet, J., Motet, G., Schön, W., 2018. Modelling confidence in railway safety case. Saf. Sci. 110, 286–299. http://dx.doi.org/10.1016/j.ssci.2017.11.012.

Wang, R., Guiochet, J., Motet, G., Schön, W., 2019. Safety case confidence propagation based on Dempster–Shafer theory. Internat. J. Approx. Reason. 107, 46–64. http://dx.doi.org/10.1016/j.ijar.2019.02.002.

Wedyan, F., Alrmuny, D., Bieman, J.M., 2009. The effectiveness of automated static analysis tools for fault detection and refactoring prediction. In: International Conference on Software Testing Verification and Validation. ICST, IEEE, pp. 141–150. http://dx.doi.org/10.1109/ICST.2009.21.

Wigmore, J.H., 1931. The Principles of Judicial Proof: Or, the Process of Proof as Given by Logic, Psychology, and General Experience and Illustrated in Judicial Trials. Little, Brown, http://dx.doi.org/10.1017/S0008197300132313.

Zhao, X., Zhang, D., Lu, M., et al., 2012. A new approach to assessment of confidence in assurance cases. In: Computer Safety, Reliability, and Security. SAFECOMP, Springer, pp. 79–91. http://dx.doi.org/10.1007/978-3-642-33675-1_7.

Zheng, J., Williams, L., Nagappan, N., et al., 2006. On the value of static analysis for fault detection in software. IEEE Trans. Softw. Eng. 32 (4), 240–253. http://dx.doi.org/10.1109/TSE.2006.38.