SRI International

CSL Technical Report SRI-CSL-25-01R2 • 29 December 2025

Quantifying Confidence In Assurance 2.0 Arguments

Robin Bloomfield (City St George's, University of London) and
John Rushby (SRI)

SRI Project 101425 in support of DARPA ANSR Program.
Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

Computer Science Laboratory • 333 Ravenswood Ave. • Menlo Park, CA 94025 • (650) 326-6200 • Facsimile: (650) 859-2844

Abstract

Confidence is central to safety and assurance cases: how much confidence a decision requires and how much the argument actually provides are both important questions. We present a new method for assessing probabilistic confidence in assurance case arguments that is simple, systematic and sound.

It exploits the ways claims are decomposed in a structured argument and provides different approaches according to the different degrees of (in)dependence and diversity among subclaims and the way they eliminate concerns that undermine confidence in their parent claims. The method uses only elementary probabilistic constructions that are well-known in other contexts (e.g., Fréchet bounds) but we interpret and apply them in a manner that is specifically focused on assurance arguments and requires no background in probabilistic analysis.

We show that the method is not susceptible to the counterexamples that Graydon and Holloway exhibit for other approaches to quantifying confidence [18] and we recommend it as an additional tool in assessment of Assurance 2.0 arguments. The primary evaluation criteria for Assurance 2.0 remain logical indefeasibility and dialectical examination, but probabilistic assessment can be useful in evaluating cost/confidence tradeoffs for different risk levels, and the overall balance of confidence across a structured argument.

Contents

List of Figures

List of Tables

1 Introduction

We have previously described overall assessment of confidence in an assurance case [10] within the methodology that we call Assurance 2.0 [8], with more details provided in a supporting technical report [9]. The primary positive assessment is *logical soundness*, and this should be subjected to Socratic/dialectical challenge by exploration of potential *defeaters* [6]. For overall confidence, all defeaters must themselves be defeated or else accepted as *residual doubts* that have been shown to pose negligible risk, and soundness must be established *indefeasibly* [38], meaning no credible new information would change the assessment.

But indefeasible assurance does not amount to certainty: there can always be some *aleatoric uncertainty* (uncertainty *in* the world, such as how many sensors fail during a mission) and some *epistemic uncertainty* (uncertainty *about* the world, such as whether our evidence for some attribute is really conclusive). These can be reduced (at increased cost) but not eliminated, so we set a threshold that balances cost and confidence. Often, this balance is struck and stated informally (if stated at all), but sometimes it can be useful to attempt to quantify and express it numerically.

Some authors are dubious of quantification because they consider that it conveys a false sense of precision [18]. A contrary opinion asserts that quantification can support explicit reasoning and structured decision-making. Decision makers need to evaluate choices among the alternatives with care, cognizant of their strengths and weaknesses. For example, Spiegelhalter [41] notes that in decision-making for the Bay of Pigs Invasion there were different interpretations for the qualitative assessment of "fair chance" that contributed to a major strategic failure. But there are also instances of egregious decision making based on unjustifiably precise quantified risk assessments. The Challenger disaster is an example: engineers had serious qualitative concerns about O-ring resilience at low temperatures, yet management relied on quantitative risk estimates that underestimated epistemic uncertainty and ignored the probability of catastrophic failure. These examples illustrate the central tension in quantified statements of assurance: used carelessly or without adequate grounding, numbers can suppress judgment, but when used appropriately they can also sharpen thinking and assist decision making.

Quantification can be particularly valuable in situations where a graduated treatment of confidence is needed. This arises in systems where different items pose different risks, and is exemplified by the Design Assurance Levels (DALs) A–E of DO-178C for software in commercial aircraft [35], the Safety

Integrity Levels (SILs) 4–1 of IEC 61508 for Programmable Electronic Systems [27], and the Automotive SILs (ASILs) D–A of ISO 26262 for cars [28] (in each case we list the levels in descending order). In Assurance 2.0 we insist on logical soundness, even for items that pose less risk, but to reduce the cost of their assurance we may reduce some thresholds on evidence and rigor. To make the reductions in a rational and justifiable manner, it is useful to assess some numerical estimate of confidence in an assurance case, and that is the topic of this report.

Subjective probability [29] provides a natural measure for numerical assessment of confidence: that is, confidence in a claim `X` is our subjective estimate that is it true, expressed as a probability, `P(X)`. Hence, our numerical assessment of confidence will use ideas and techniques from probability theory. Note that is fundamental that confidence assessments for a semi-formal assurance case are subjective judgments, preferably achieved through consensus using rational methods of assessment and calculation: they are not frequentist or other objective measurements but can be combined with such measures (e.g., absence of failure in system testing or early deployment) to yield credible worst-case estimates of long-term reliability [42].

We expect the numerical assessment to be *compositional*: that is, built up step by step, ascending the structured assurance case argument from its leaves (typically, evidence) to its top claim (the conclusion). In such cases, numerical assessment can also be used to examine the distribution of confidence across the argument, the sensitivity of confidence in the conclusion to that of evidence and intermediate steps, and to compare one argument with another.

As explained previously [8], we have two ways of organizing assessment of probabilistic confidence.

- There are circumstances where some probabilistic measure (e.g., probability of failure on demand, *pfd*) is an explicit part of the top claim (some nuclear systems are like this). In this case, probabilistic measures will also appear in the evidence assembled and in the internal claims developed within the argument, and these measures will be propagated by specific theories that are cited by the steps of the argument. We refer to this as *internal* probabilistic assessment.

- In other cases, the top claim will be unconditional (e.g., "the system is correct"), and we develop a separate *external* probabilistic assessment of our subjective confidence (i.e., belief) in the truth of this claim and of those claims and evidence that support it.

(It is plausible that these methods could be used in combination: for example, aleatoric uncertainty could be treated internally while epistemic uncertainty is assessed externally.)

Here, we are concerned with external assessments, where we and many others have proposed methods for calculating probabilistic or other numerical assessments. Most of these methods assume that assurance cases are organized around a structured argument that relates evidence about the system to significant claims about it (others use Toulmin arguments, Bayesian Belief Networks, or other diagrammatic representations). The argument must pass some threshold for logical validity, soundness, and persuasiveness but the rigor of this assessment varies according to the method. Most methods other than Assurance 2.0 apply only informal assessments of logical correctness and their techniques for numerical assessment therefore carry some, or much, of the burden for overall assessment of veridity. In Assurance 2.0 on the other hand, we assess logical soundness rigorously, separately, and prior to, numerical assessment. This allows our numerical assessment to focus on providing a view that is supportive and complementary to the logical one, and enables simple probabilistic models because they serve only a single purpose. However, notice that because it provides a different view, numerical assessment contributes to dialectical examination and can raise issues that cause logical assessment to be revisited. For example, we may initially make deterministic aleatoric assumptions about failure modes or event occurrences where a probabilistic treatment shows a more nuanced approach is to be preferred.

Our previous probabilistic treatments [9, 10] used very elementary and conservative calculations. We briefly summarize these treatments in the following section and then, in Section 3, we introduce our new method and its application to several different kinds of *reasoning* steps within an argument. Our new method is very simple and uses only classical probabilistic reasoning. Section 3 is the main novel contribution in this report. However, in its final example, we show that our previous treatment is the correct choice for certain reasoning steps and from there we segue in Section 4 to a treatment for similar but more complex examples using Bayesian Belief Networks (BBNs). We have covered this topic before [37], but the presentation here is integrated into our larger narrative. In Section 5 we compare our new treatment with some other methods for quantifying confidence in assurance cases. These use different probabilistic models, including those derived from theories of evidence such as Dempster-Shafer, and alternative applications of BBNs. In Section 6 we consider the critique provided by Graydon and

Holloway [17, 18] for some of these other methods and explain why ours are not susceptible to them. We provide conclusions in Section 7.

We conclude this introduction with a brief overview of Assurance 2.0 for those who have not yet read the earlier documents.

We model an assurance case as a network of argument steps over propositions called *claims*, represented diagrammatically as *claim nodes*, connected by *argument nodes* or *steps*. A claim is intended to denote an atomic proposition (typically expressed in controlled natural language) or a conjunction or disjunction of these. An argument step relates a parent claim to a set of supporting *subclaims* that function as premises, together with an optional *sideclaim* that records any explicit conditions required for the step to be sound. The combination of an argument step and its parent claim, subclaims, and sideclaim is called an *argument block* (as in building block). Assurance 2.0 has just six different types of block [7].[1]

We distinguish subclaims, which are the principal supporting premises offered for the parent claim, from the *sideclaim*, which captures the *applicability conditions* for the warrant or *justification* for the argument scheme used in the step (e.g., assumptions about scope, model fidelity, tool qualification, independence, or completeness etc.). This distinction is methodological rather than logical—sideclaims could be represented as ordinary premises—but it is useful because such conditions on the warrant are often the main source of residual doubt and are best scrutinized explicitly.[2]

In Assurance 2.0, we treat the assessment process as layered: (i) logical scrutiny addresses coherence and logical validity, then evidence and warrant suitability justify soundness; (ii) dialectical scrutiny examines potential defeaters; and only then (iii) a quantitative confidence layer examines remaining epistemic uncertainty by assigning assessor confidence to claims and (where relevant) to inference strength. Confidence is represented as a subjective probability, and the aim is not to turn assurance into statistical sampling, but to make residual uncertainty explicit and compositional, enabling sensitivity analysis and principled trade-offs between evidence-gathering cost and achieved confidence.

[1] Exact defeater blocks are an addition to the original five.

[2] Historically, we have spoken of residual *doubts*, but in a probabilistic context we prefer to speak of residual *concerns* as *doubt* is used for the probabilistic function $1 - P(a)$.

2 Our Previous Method

Our previous treatments [9, 10] used very elementary calculations based on probabilistic logic. The logical interpretation of an Assurance 2.0 argument treats each block as a subargument with its parent claim as the conclusion and its subclaims and sideclaim (if any) as the premises. The premises are implicitly conjoined and the standard interpretation of conjunction in probabilistic logics [1, 34] is a product: `P(A & B) = P(A)` $\times$ `P(B | A)`, which can be simplified, if `A` and `B` are independent, to `P(A)`$\times$`P(B)`. We argued that because all claims and subclaims must be true in an assurance argument, the simple form is sound and probabilistic confidence can therefore be propagated upwards from the leaf nodes (i.e., from confidence in evidence) by this *product calculation*, where confidence in each claim is the product of confidence in its subclaims (and sideclaim, if any). Hence, when `conf(X)` denotes confidence in claim `X`, the general product expression (neglecting any sideclaim) for a decomposition block with n subclaims is

$$\texttt{conf(parent)} \geq \prod_{i=1}^{n} \texttt{conf(subclaim}_i\texttt{)}.$$

We also showed that a more conservative calculation, which applies with correlated claims, propagates confidence as the *sum of doubts*, where `doubt(X) = 1 - conf(X)` and doubt in each claim is calculated as the sum of doubts in its subclaims. Thus, the general expression for a decomposition block with n subclaims (again, neglecting any sideclaim) is

$$\texttt{doubt(parent)} \leq \min\left(1, \sum_{i=1}^{n} \texttt{doubt(subclaim}_i\texttt{)}\right).$$

This can also be written as

$$\texttt{conf(parent)} \geq \max\left(0, \sum_{i=1}^{n} \texttt{conf(subclaim}_i\texttt{)} - (n-1)\right)$$

where the right hand side is known as the *Fréchet lower bound* for intersection [45].

Both the product and the sum of doubts calculations produce highly conservative values that decrease as they go higher in the argument tree (so calculated confidence in the top claim is always very low) and are largely indifferent to its shape (i.e., to the actual argument). Consequently, they are of limited utility and we no longer recommend them for general use, although they are appropriate in some circumstances (e.g., see Section 3.2.6).

This limited utility is not surprising in retrospect: logic and probability have different purposes and model the world differently, so no combination of the two works well in all circumstances. There can be many reasons why an argument step has several subclaims: for example, different subclaims could address different circumstances, or they could address the same circumstance in diverse ways (so that confidence may increase as we ascend the argument) and a probabilistic assessment should treat each according to its reasons, whilst logical assessment always lumps all subclaims together as a conjunction and simply requires that each is `true`. Hence, in the new treatment, and unlike our previous ones, we largely separate logical and probabilistic interpretations although the latter assumes the former has been applied satisfactorily.

3 The New Method

Our new method for assessing probabilistic confidence in an Assurance 2.0 case works compositionally, claim by claim, on the argument of the case from bottom to top.

The leaf nodes at the bottom of a completed assurance case argument can be references to external subcases, assumptions, evidence, or residual doubts.[3] External subcase references need to be expanded in place (like a macro), or separately evaluated and represented by the confidence assessed for their top claim (like a subroutine). Confidence in assumptions is assigned by assessors or by Subject Matter Experts (SMEs) and is recorded as the estimated probability that each assumption is true.[4] The treatment of residual concerns[5] considers number of occurrences (e.g., are there tens of minor warnings from a static analyzer or hundreds?) as well as likelihood and severity of associated faults, and we postpone its consideration to Section 3.3. Here, we first deal with confidence in evidence.

[3] An incomplete argument may also contain defeaters, or explicitly unfinished nodes. These need to be resolved and eliminated prior to probabilistic assessment.

[4] Some of these estimates may be purely subjective, while others (e.g., likelihood of more than n sensor faults) may be based on historical experience or a combination of reliability analysis, data, and judgment that should perhaps be expanded into a subargument.

[5] Historically, we have spoken of residual *doubts*, but in a probabilistic context we prefer residual *concerns* as *doubt* is used for the probabilistic function $1 - P(a)$.

3.1 Confidence in Evidence

Evidence is usually provided to an Assurance 2.0 argument in two steps, as portrayed in Figure 1. In the first step, reference to the actual evidence is supplied to an *Evidence Incorporation* block. That (building) block or argument step supports a claim about *something measured*: that is, it tells us what the evidence *is*. In Figure 1 we see an example where the evidence is provided by requirements-based testing and the measurement claim states that this achieved some specific level (e.g., MC/DC [22]) of structural coverage. In a full assurance case, the evidence node and the measured claim will reference (perhaps via hyperlinks in their labels) detailed accounts of the testing performed and the coverage obtained, and the evidence incorporation node will reference a detailed justification that the evidence indeed supports the claim. The justification may cite an established theory for the test protocol involved, with necessary assumptions referenced in a sideclaim, or it may be decided that evidence incorporation is premature and that a subargument is needed to "make the case" that the test protocol indeed delivers the measured claim, thereby elaborating the overall argument beyond that shown in Figure 1.

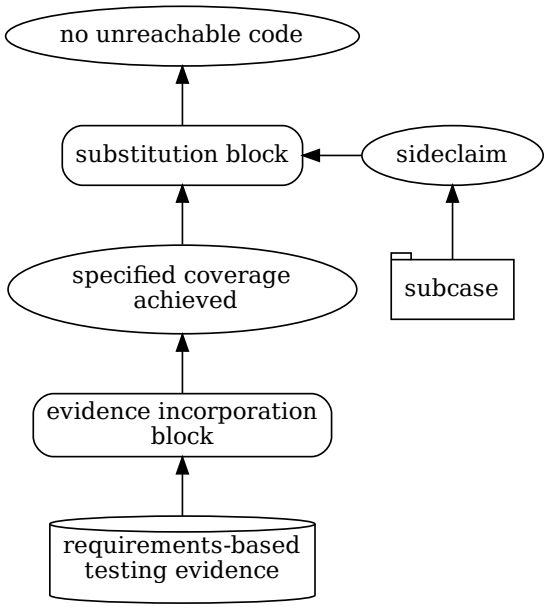

Figure 1: Evidence Example

In the second step, the measured claim is transformed, using a *substitution* block, into a claim about *something useful* that tells us what the evidence *means* in the context of this assurance case. In the example of Figure 1 the useful claim is that the software contains no unreachable code (e.g., debugging code that is disabled in operation; this is considered hazardous because experience shows that a fault may cause it to become reachable, with unpredictable consequences). The reasoning for this is documented and justified in the substitution block, citing a suitable theory of testing and reachability.

In previous papers and reports [9,10,36] we describe in detail how claims should be evaluated against evidence by examining *confirmation measures* constructed from various estimated conditional probabilities, such as Good's measure

$$\log \frac{P(E \mid C)}{P(E \mid \neg C)},$$

where C represents the useful evidential claim and E the measured claim. Confirmation measures help us assess the discriminating power of our evidence, but once this has been done and accepted, we need a numerical assessment of confidence in the useful claim, given the evidence, and for this the simple posterior probability estimate $P(C \mid E)$ is appropriate. In our notation, this is

```
P(useful claim) = P(useful claim | measured claim).
```

Note that we are assuming `P(measured claim | evidence)` is 1 because the measured claim should simply state what the evidence is, but it can be separately estimated and added as an additional factor if desired.

Argument blocks in Assurance 2.0 generally have a *sideclaim* recording conditions that may be necessary to ensure their justification is sound. Here, it might specify that coverage must be measured on executable object code (EOC), not source code. For reasons that are explained in the following section, confidence in the useful claim is then the product of the conditional probability stated above and confidence in the sideclaim:

```
conf(useful claim) =
  P(useful claim | measured claim) × conf(sideclaim).
```

The first factor on the right of this assignment must be estimated by the assessors or other SMEs while the second, `conf(sideclaim)`, is evaluated by recursive application of the techniques being developed here.

It is also possible to have a sideclaim on the evidence incorporation block, in which case its confidence will be included as a further factor in the product above.

3.2 Confidence in Reasoning Steps

Above the leaf nodes, an Assurance 2.0 argument consists of reasoning steps that each justify a parent claim on the basis of one or more subclaims and, possibly, a sideclaim. The reasoning steps comprise concretion, substitution, and exact defeater (aka. negation) blocks, which each have a single subclaim, and decomposition and calculation blocks that each have two or more subclaims [7].

3.2.1 Blocks with a Single Subclaim

The substitution blocks that are used in combination with evidence incorporation blocks have already provided simple examples of argument blocks with a single subclaim. The general case for blocks with a single subclaim is portrayed in Figure 2.

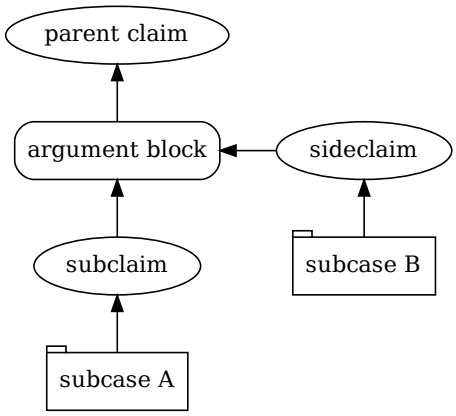

Figure 2: Argument Block with a Single Subclaim

Our task is to estimate probabilistic confidence in the truth of the parent claim given previously estimated confidence in the components of its supporting argument block, namely its subclaim, sideclaim, and the inference used in its argument.

In logical assessment, such as described in [10], we explain that the inference is assumed to be deductive and the sideclaim states conditions for it to be sound, which includes the requirement for subclaim(s) to be sound premises of its argument step. The logical interpretation of an argument block is then `sideclaim` $\supset$ `(subclaim` $\supset$ `parent)`, which is equivalent to `sideclaim` $\wedge$ `subclaim` $\supset$ `parent` and so confidence in the parent claim is based on that in `sideclaim` $\wedge$ `subclaim`.[6]

We prefer that the argument step is deductive in numerical assessment also, which can often be achieved by internalizing confidence into claims (see

[6] We use $\supset$ for material implication, and $\wedge$ for conjunction.

introduction) and explicitly addressing confidence in its inference. Nevertheless, this is not always feasible so we need an approach that allows some uncertainty and external assessment of confidence.

Here, confidence in the parent is some function of the argument step applied to confidence in the subclaim and sideclaim. That is,

```
conf(parent) = f(argstep)(conf(subclaim ∧ sideclaim)).   (1)
```

Where the (higher-order) function `f` depends on the kind of reasoning performed in the argstep (e.g., substitution) and its specific instance (e.g., from structural test coverage to reachable code).

Confidence is a subjective probability, so we can apply probability laws. By the chain rule

```
conf(subclaim ∧ sideclaim)
   = conf(subclaim | sideclaim) × conf(sideclaim)
```

The first term in the product is confidence in the subclaim under the conditions stated by the sideclaim, which may seem rather difficult to assess. However, we generally expect the subclaim to be conditionally (though not logically) independent of the sideclaim and so the term reduces to `conf(subclaim)` which is assessed by applying the methods being described here to the subclaim's subcase.

For example, in the previous section the measured claim (i.e., subclaim) will state that MC/DC coverage was achieved using some testing and measurement procedure and confidence in the claim will concern these procedures and how they were performed. The sideclaim will state (possibly among other things) that coverage must be achieved and measured on EOC. Confidence in this sideclaim will be 1 if the measurement actually was on EOC and 0 otherwise (e.g., if it was measured on source code), but this has no impact on confidence in the measured claim. Other examples may have more graduated confidence in the sideclaim, but this should generally have no impact on the subclaim; in cases where it does, the conditional confidence should be assessed explicitly. Thus, assuming conditional independence,

```
conf(subclaim ∧ sideclaim)
   = conf(subclaim) × conf(sideclaim).
```

Substituting back in `(1)`

```
conf(parent) = f(argstep)(conf(subclaim) × conf(sideclaim)).
```

We assert that for the functions `f` we are interested in, the product can be decomposed, so that

$$\texttt{conf(parent)} \quad \texttt{= f(argstep)(conf(subclaim))} \times \texttt{conf(sideclaim).} \tag{2}$$

Applying this formula to Figure 2, if our confidence in the subclaim, given its subcase A is 0.8, confidence in the sideclaim, given its subcase B, is 0.9, and confidence in the inference from test coverage to reachable code—that is `f(argstep)(a)`—is `0.95` $\times$ `a`, then confidence in the parent claim is the product of these three numbers: 0.684.

3.2.2 Blocks with Multiple Subclaims

The analysis for argument blocks having multiple subclaims proceeds as in the previous section, except that subclaim becomes plural in `(2)` so that we have

$$\texttt{conf(parent)} \quad \texttt{= f(argstep)(conf(subclaims))} \times \texttt{conf(sideclaim).} \tag{3}$$

However, the function `f(argstep)` is now based on two components: one concerns the inference, as before, and the other concerns how the multiple subclaims combine to form the premise to that inference. We will suppose these elements compose, so that

$$\texttt{conf(parent)} \quad \texttt{= f(argstep)(h(argstep)(conf(subclaims)))} \times \texttt{conf(sideclaim)} \tag{4}$$

where, as before, `f(argstep)` is the confidence function due to the inference, as before, and `h(argstep)` is that due to the combination of subclaims. We generally expect blocks to employ just one or the other of these elements so that one of these functions will be the identity (or multiplication by a simple factor). For example, we often have a substitution block that performs inference (but no combination) above a decomposition block that performs combination (but no inference), rather than a single block that performs both of these.

In the following, we concentrate on combination and will assume `f(argstep)` is some simple constant factor `k` that defaults to 1, and we will initially postpone consideration of the sideclaim, so (4) becomes

$$\texttt{conf(parent) = h(argstep)(conf(subclaims))} \tag{4a}$$

and our task is to estimate `h(argstep)` for various kinds of reasoning steps.

We focus on decomposition blocks, which can be used in several different ways, possibly requiring different probabilistic interpretations and therefore different calculations for `h(argstep)`.

3.2.3 Decomposition Blocks, Diversity Case

Let us return to the evidential step of Section 3.1 and develop it further. Suppose requirements-based testing did not quite achieve the desired coverage. We could try to provide a more complex justification for the same argument using additional testing, or we could regard the testing evidence as inadequate and either replace it or reinforce it with something else. Testing is attractive because it provides concrete evidence, so we decide to retain it and buttress it with static analysis: the idea being that testing and static analysis build on different foundations and are therefore "diverse" and may be supposed to fail independently [5]. This diversity argument is represented in the decomposition block shown in Figure 3.

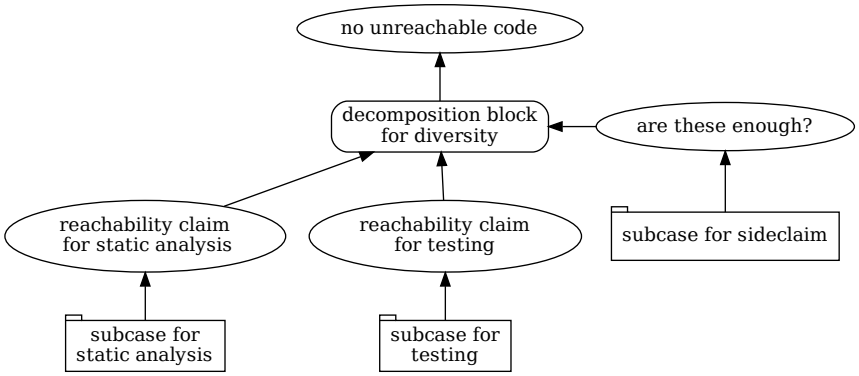

Figure 3: Decomposition Block for Argument by Diversity

In some discussions of assurance cases, diversity arguments of this kind are regarded as disjunctions, the idea being that "if you don't like one subcase, you can use the other." This is not our interpretation: we think of one subclaim eliminating some concerns about the parent claim and the other subclaim eliminating other concerns, and we conclude the parent claim only when *both* subclaims have been assessed as sound. Hence, the logical interpretation of a decomposition argument has to be conjunction and the adequacy of any specific decomposition (i.e., are these specific subclaims adequate?) is stated as a sideclaim and supported by its own subcase.

Logical assessment is holistic, but when we assess confidence we are interested in the *quantity* of concerns that have been eliminated and their impact on the parent claim. This interpretation is illustrated in the Venn diagrams of Figure 4. The blue shape in the left hand diagram indicates the probability mass of concerns eliminated by testing and the background circles indicate those concerns that remain. The area of the blue shape

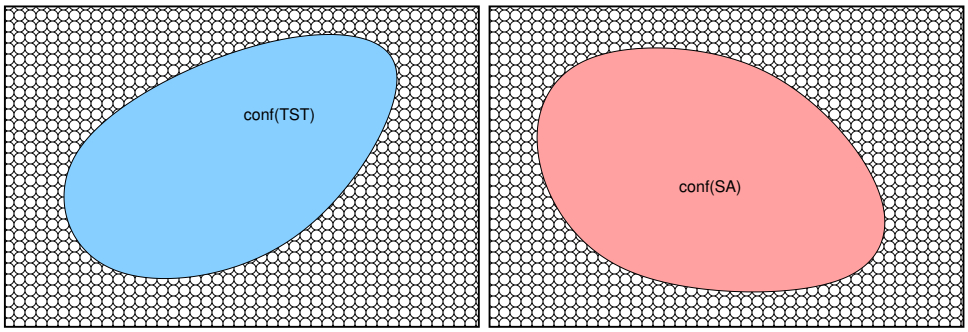

Figure 4: The background circles represent concerns about the parent claim, and the colored shapes represent those eliminated (i.e., covered) by subclaims for testing (left) and static analysis (right)

represents the confidence delivered by the testing subcase: `conf(TST)`. (Of course, in practice we want useful evidence to eliminate almost all concerns, but we reduce this for pictorial clarity.) Likewise, we indicate the concerns eliminated through static analysis by the pink shape on the right of Figure 4, where the area of the shape represents our confidence in the subcase for static analysis: `conf(SA)`.

More precisely, we interpret the Venn diagrams over an *epistemic possibility space*: that is, the (unknown) set of scenarios, consistent with scope assumptions and background knowledge recorded elsewhere in the case, that determine whether the parent claim is `true` or `false`. The intent is epistemic rather than frequentist: it is a space of possibilities, not a population from which we sample.

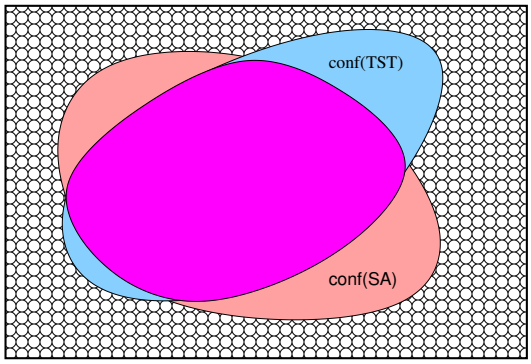

Figure 5: Concerns Eliminated by Combination of Diverse Subclaims

Conceptually, we assign subjective probabilities to scenarios based on our judgment about their likelihood. Confidence in a claim `C`, denoted `conf(C)`, is the sum of probabilities across all scenarios where `C` is `true`; conversely,

its `doubt` is the sum across scenarios where it is `false`. When we remove $x\%$ of doubt, we gain $x\%$ in confidence.

A *concern* about claim `C` is a specific scenario in which `C` is `false`. In the diagrams, the small background circles represent such concerns and the rectangles correspond to the totality of these (i.e., doubt in the parent claim). Colored regions represent sets of scenarios (i.e. events), where the sizes correspond to the sum of probabilities, the probability mass, associated with those scenarios.

The concerns eliminated by the combination of requirements-based testing and static analysis are shown in Figure 5 where their probability mass corresponds to the union of the blue and pink shapes. Thus, confidence in the parent claim delivered by the two subclaims corresponds to the area of this union.[7] This area can be calculated as the sum of the areas for testing and for static analysis, less their overlap (shown in magenta), which would otherwise be counted twice. The overlap corresponds to the intersection of the testing and static analysis subclaims. Thus, using `NUC`, as an abbreviation for "no unreachable code," we have

$$\texttt{conf(NUC)} = \texttt{conf(SA)} + \texttt{conf(TST)} - \texttt{conf(SA} \cap \texttt{TST)}$$

where the right hand side corresponds to `h(diversity)(conf(subclaims))` in (4a), and we need some way to estimate the final term.

We know that static analysis covers a portion of the total "doubt space" equal to `conf(SA)` so, assuming independence, it also covers the same proportion of the space covered by testing, which itself covers `conf(TST)` of the total space. Hence, our estimate for their overlap `conf(SA` $\cap$ `TST)` is `conf(SA)` $\times$ `conf(TST)`[8] and therefore

$$\texttt{conf(NUC)} = \texttt{conf(SA)} + \texttt{conf(TST)} - \texttt{conf(SA)} \times \texttt{conf(TST)}.$$

Recalling that `doubt(x)` represents `1 - conf(x)`, this becomes

$$\texttt{conf(NUC)} = 1 - \texttt{doubt(SA)} \times \texttt{doubt(TST)}, \text{ and therefore}$$
$$\texttt{doubt(NUC)} = \texttt{doubt(SA)} \times \texttt{doubt(TST)}.$$

[7] Notice how far we have departed from the standard interpretation of probabilistic logic: *logical* assessment *whether* the parent claim is true requires the *conjunction* of subclaims, whereas *confidence* assessment for *how strongly* we should believe it requires their union (traditionally associated with *disjunction*).

[8] These and several other formulas are well-known probabilistic tautologies; we spell them out so that later derivations are easier to follow.

We refer to this calculation as the *product of doubts* and it generalizes to arguments with n subclaims in the obvious way:

$$\texttt{doubt(parent)} = \prod_{i=1}^{n} \texttt{doubt(subclaim}_i\texttt{)}. \tag{5}$$

For notational brevity we have omitted the sideclaim in the calculations above but we recall from (4) that the confidence that is propagated is the product of confidence in the inference from subclaims to parent claim (i.e., the calculation described above) and confidence in the sideclaim. Additionally, a function `f` or constant factor `k` can optionally be applied if the developer or assessor considers it desirable.

So, in the example above, if we are 95% confident that testing ensures no unreachable code and 90% confident that static analysis does the same, then the doubts are 5% and 10% respectively, and their product is 0.5%. Hence, confidence in the combination is 99.5%. If we have 90% confidence in the sideclaim that the subclaims are adequately diverse and independent, then overall confidence is the product of these two, which yields 89.5% (indicating that diversity adds little when there is low confidence in the sideclaim).

Product of doubts is accurate when the independence assumption is valid and less so when it is not. There is more opportunity for violation of independence when confidence in the individual subclaims is low. For example, if we are 49% confident in each of two subclaims, the product of doubts calculation gives 74% confidence in their combination whereas in reality it could range from 49% (total overlap) to 98% (completely disjoint). The latter may be a deliberate, alternative, strategy to diversity, and we consider it next.

3.2.4 Decomposition Blocks, Partitioned Case

In contrast to the previous example where, in the Venn diagram representation, the subclaims substantially overlap, we now consider the case where they are disjoint. This arises in decompositions over different hazards, or over the components of a larger system, or over different operating conditions, etc. The idea is that the subcases partition the "concern space" and the subclaims are each local to a separate partition.

This case is portrayed in Figure 6. To make it concrete, we could suppose we are decomposing the argument over two hazards. This might suggest two partitioned subcases, one for each hazard, but reflection suggests this is insufficient, we really need three: one for Hazard 1 alone, another for Hazard 2 alone, and a third for when they occur together. It is even conceivable that

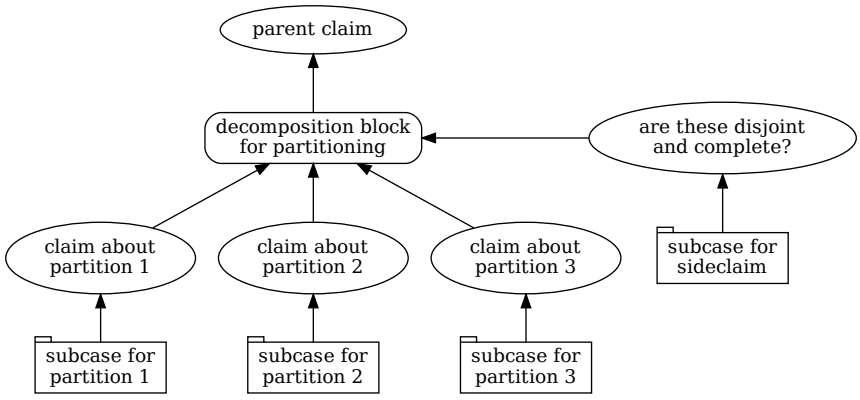

Figure 6: Decomposition Block for Partitioned Argument

there should be further partitions for cases such as those where a second instance of a hazard arises while the system is still dealing with a first instance of the same hazard. However, we will suppose three partitions are sufficient (e.g., by interpreting the third as "everything else") and that this is justified in the sideclaim to the decomposition.

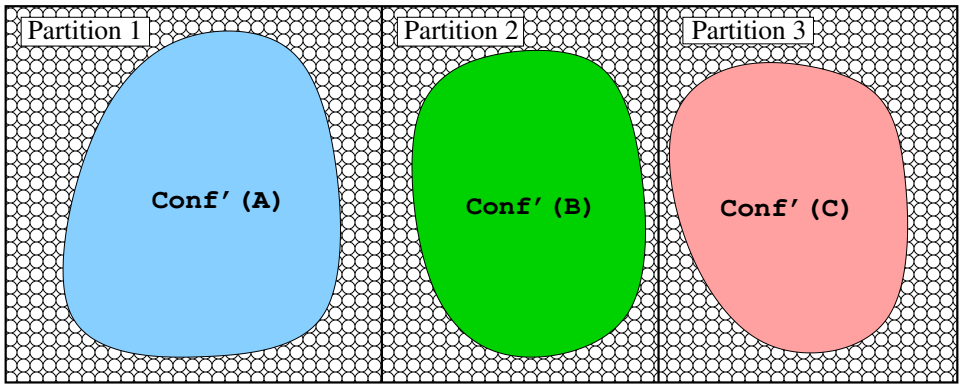

Figure 7: Partitioned Subclaims

The Venn diagram corresponding to this argument is shown in Figure 7 where the background rectangle represents the total concern space partitioned into the three subcases. The partitions are of different sizes because we may have more or less concern about them: `weight(x)` indicates the fraction of the total concerns or risks, depending on the claim, associated with partition `x` (so the `weight`s sum to one). The colored shapes represent our confidence in the subcase for their associated partition: `conf′(x)` (we use the prime ′ to indicate this confidence is assessed relative to its

partition, not the whole space, which is denoted by plain `conf(x)` where `conf(x)` = `conf′(x)`×`weight(x)`). Our confidence in the overall decomposition, given the sideclaim, is then the weighted sum of these (this is just a Total Probability calculation):

```
conf(parent) = conf′(A) × weight(1)
             + conf′(B) × weight(2)
             + conf′(C) × weight(3).
```

Notice that if the `weight`s are equal, this is simply the arithmetic mean of the subclaim confidences; we call this the **averaging case**.

As in the previous cases, the confidence propagated from this argument step is the product of confidence in the parent claim, calculated as above, and confidence in the sideclaim. Thus if we assess that 60% of the risks are associated with Hazard 1 alone, 30% with Hazard 2 alone, and 10% with other cases, and that our confidence in the subclaims associated with these are 90%, 95% and 80%, respectively, then basic confidence in the parent claim is $90 \times 60 + 95 \times 30 + 80 \times 10 = 90.5\%$ and this needs to be multiplied by confidence in the sideclaim, say 99%, to yield an overall confidence of 89.6%. As in the previous sections, a function `f` or constant factor `k` can optionally be applied to the basic calculation if the developer or assessor considers it desirable.

The formula above obviously generalizes to decompositions with n partitioned subclaims as follows.

$$\texttt{conf}(\texttt{parent}) = \sum_{i=1}^{n} \texttt{conf}'(\texttt{subclaim}_i) \times \texttt{weight}(\texttt{subclaim}_i) \tag{6}$$

where the right hand side corresponds to `h(partition)(conf(subclaims))` in (4a).

Despite its different derivation, this is essentially equivalent to Jeffrey's rule of combination [25], which is (implicitly) used in some Dempster-Shafer treatments of quantified confidence in assurance cases [3, Section 3.2]. Notice that it does not matter if some of the concerns addressed by one subclaim are located in the partition of another, provided assessment of confidence in the subclaim is confined to its own partition.

Also note that since

$$\texttt{conf}'(\texttt{subclaim}_i) \times \texttt{weight}(\texttt{subclaim}_i) = \texttt{conf}(\texttt{subclaim}_i),$$

(6) is equivalent to

$$\texttt{conf}(\texttt{parent}) = \sum_{i=1}^{n} \texttt{conf}(\texttt{subclaim}_i), \tag{7}$$

which corresponds to Fréchet's upper bound for union.

We can also derive this result from the analysis used for diversity cases. Recall formula 3.2.3 and imagine pulling apart the subclaim areas of the concern space so that they no longer overlap. But when there is no overlap, `conf(A ∩ B)` is zero. This corresponds to the partitioned case and we will obtain the same results under either the diversity or partitioned calculations, provided we take note that the subcase confidences are evaluated differently: in the diversity case they are relative to the whole concern space (i.e., Formula (7)), whereas in the partitioned case each is relative to its own partition within that space and is weighted according to the partition's importance or risk (i.e., Formula (6)).

3.2.5 Decomposition Blocks, Containment Case

We can reverse the mental experiment just described and imagine pushing the subclaims of a diversity case together until one is entirely within the other, as shown on the left of Figure 8.

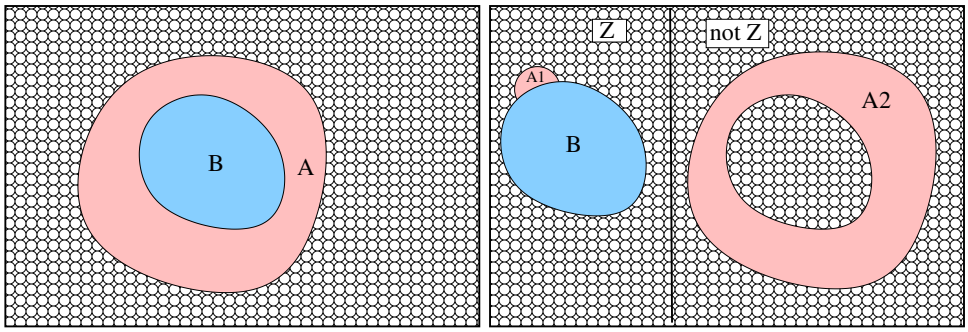

Figure 8: Contained Subclaim (on left, redrawn on right)

Ayoub and colleagues [3] call this a "containment" argument and treat it as a special case of what they call a "disjoint" argument and that we refer to as a partitioned case (recall the previous subsection). We will describe their treatment later, but first we analyze it as a special form of diversity case.

It is obvious in the diagram on the left of Figure 8 that the eliminated concerns are all covered by subcase A—and B adds nothing—so the value propagated should be simply `conf(A)`.

Nonetheless, we can attempt to apply our treatment for diversity cases, as formalized in (5). This gives

$$\texttt{conf(parent)} = \texttt{conf(A)} + \texttt{conf(B)} - \texttt{conf(A)} \times \texttt{conf(B)} \qquad (8)$$

but this treatment is indifferent to how much of B overlaps A: in particular, it does not propagate the correct value, namely `conf(A)`, when B is fully within A and we might suspect it is also incorrect when only a little of B is outside A.

Now (8) is derived from our treatment of diversity cases in Section 3.2.3, which starts from the observation, true in all cases, stated in (4). The challenge in applying (4) is to find a good estimate for `conf(A ∩ B)`. The treatment in (8) assumes that A and B are independent, so that `conf(A)` $\times$`conf(B)` is a good estimate for `conf(A ∩ B)`. However, in the case where B is within A, the subcases are positively correlated and that product is no longer a good estimate. But a conservative estimate is provided by the *Fréchet upper bound for intersection* [45]:

$$\texttt{conf(A} \cap \texttt{B)} \leq \min(\texttt{conf(A), conf(B)}).$$

In our containment case, the minimum is `conf(B)` and applying this in (4) gives `conf(parent)` $\geq$ `conf(A)` as required. More directly, (4) becomes

$$\texttt{conf(parent)} \geq \max(\texttt{conf(A), conf(B)}),$$

which corresponds to the *Fréchet lower bound for union*. For the general case of containment among n subclaims, we have

$$\texttt{conf}(\texttt{parent}) \geq \max_{i=1..n}(\texttt{conf}(\texttt{subclaim}_i)),$$

which comes from the general form of the Fréchet bound [45].

For completeness, we recall the *Fréchet lower bound for intersection*:

$$\texttt{conf(A} \cap \texttt{B)} \geq \max(\texttt{0, conf(A) + conf(B) - 1}).$$

When `conf(A)` and `conf(B)` are fairly large (i.e., sum to more than 1), substituting into (4) gives `conf(parent)` $\leq$ `1`, which is not very helpful, but when their sum is less than 1 (as will likely be so with largely disjoint subcases), it gives

$$\texttt{conf(parent)} \leq \texttt{conf(A) + conf(B)},$$

which is an alternative way of deriving the basis for our treatment of partitioned cases in (7).

We see that our general treatment of diversity cases is sound but the specifics of its application depend on whether the subcases are independent or not. If they are independent, the product of doubts calculation provides a good estimate; if they are positively correlated (i.e., largely nested) then

a calculation using the Fréchet lower bound for union provides a conservative estimate, while if they are negatively correlated (i.e., largely disjoint) the partitioned treatment (which corresponds to Fréchet's upper bound for union) is suitable. In practice, we may not know if our subcases are correlated, so we can do "what if" exercises to examine impact of the alternative possibilities, with further examination should it prove significant.

As mentioned earlier, Ayoub and colleagues [3] treat containment cases as a special form of disjoint cases, which are what we call partitioned cases. We next describe this treatment as it provides an alternative (but more contrived) route to the same conclusion.

The idea of their construction is to create two partitions: one corresponding to B and the other to everything else. We adjust this slightly, as illustrated by the diagram on the right of Figure 8, so that concerns covered by B are contained in a partition Z, and concerns covered by A are divided into two parts: A1 contains the concerns also covered by B (we show a little of A1 for clarity but it is really hidden under B), and A2 contains all the other concerns covered by A and is located in partition "`not Z`." By formula (6), we have

$$\texttt{conf(parent)} = \texttt{conf}'\texttt{(B)} \times \texttt{weight(Z)} + \texttt{conf}'\texttt{(A2)} \times \texttt{weight(not Z)}.$$

Now, $\texttt{conf}'\texttt{(B)} = \texttt{conf(B)} \;/\; \texttt{weight(Z)}$,
and $\texttt{conf}'\texttt{(A2)} = \texttt{conf(A2)} \;/\; \texttt{weight(not Z)}$, so

$$\texttt{conf(parent)} = \texttt{conf(B)} + \texttt{conf(A2)}.$$

But $\texttt{conf(A2)} = \texttt{conf(A)} - \texttt{conf(A1)}$ and $\texttt{conf(A1)} = \texttt{conf(B)}$, and thus $\texttt{conf(parent)} = \texttt{conf(A)}$ as desired.

This is the same (correct) result that we obtained previously using a diversity case adjusted to use Fréchet's upper bound for union. This is not surprising as all our calculations are variants on the same basic method, but it is gratifying to see that different routes through the constructions do converge on the same result.

Discussion of these constructions does invite the question why should we be interested in containment cases? We examined them because Ayoub and colleagues do but otherwise see no utility in them. They do, however, provide a conservative limiting case for diversity arguments that may be useful when unable to justify the independence assumptions needed for stronger results.

3.2.6 Decomposition Blocks, Cumulative Case

Barrett and colleagues [4, page 32] consider a case where the parent claim that a dangerous AI function is taken out of service within one week is supported by the following three subclaims:

Detection: Incident monitoring detects all novel cyber attacks within one day,

Revision: A revision protocol ensures the safety case is updated within five days of a novel cyber attack being detected,

Removal: The AI system will be taken offline within one day of the top-level safety case claim becoming false.

They represented the argument by a decomposition block similar to Figure 6 and estimated confidence in the detection, revision, and removal subcases to be 48%, 81%, and 56% respectively.

At the time they performed their study, the only probabilistic assessments offered for Assurance 2.0 were the "product" and "sum of doubts" calculations outlined in Section 2. The sum of doubts is conservative but robust (recall, it is equivalent to the Fréchet lower bound), while the product calculation has rather strong assumptions. Here, sum of doubts delivers confidence 0 in the parent claim, while the product calculation delivers 21.772%.

Now that we have developed more specific treatments it is worth reexamining these calculations to see if we can derive a more precise result. However, although the argument resembles Figure 6, we can see that the subcases are not independent (each is contingent on the success of its predecessors) so the partitioned treatment is not appropriate, and neither is diversity.

If we try to portray the concerns eliminated by each subcase, we obtain Figure 9, which looks like Figure 8 but we can see that it requires a different interpretation. Rather than the concerns eliminated by later subclaims being contained within those of earlier ones, they build support *cumulatively* and are all required. That is, by the standard probabilistic interpretation of conjunction (the chain rule):

```
conf(parent) = conf(detection & revision & removal)
             = conf(detection)
               × conf(revision | detection)
               × conf(removal | detection & revision).
```

Now, notice that the definitions of the `detection` and `removal` subclaims given earlier implicitly have this conditional form. Hence, the estimated subclaim confidences do correspond to the terms in the formula above and this is the same as the product calculation performed earlier—but now we can see that for this case it is exact, rather than an approximation.

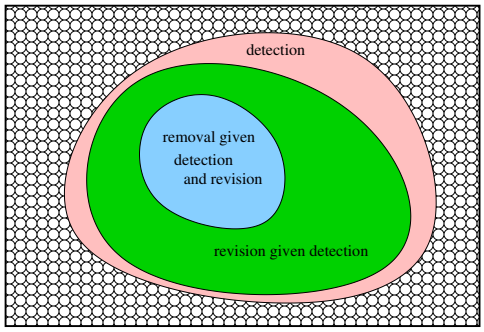

Figure 9: Cumulative Subclaims

This example is a simple illustration of an argument where there are conditional relationships among subclaims. Because the relationships in this example are simple, we can easily calculate (as the product) the confidence to be propagated to the parent claim (subject, as usual, to adjustment by a function `f` or factor `k` and multiplication by confidence in any sideclaim). In arguments where the conditional relationships among subclaims are more complex than illustrated here, it may be useful to employ BBNs and their tools, which are discussed in Section 4.

3.2.7 Decomposition Blocks, Other Cases

In Sections 3.2.3 (diversity), 3.2.4 (partitioned), and 3.2.5 (containment) we presented a repertoire of different probabilistic interpretations for the logical conjunction in a decomposition block. Furthermore, we have seen that partitioned and cumulative cases can be reduced to diversity cases, so that all are based on a single construction that also clarifies analysis of cumulative cases. We consider this uniformity to be a positive indication that our treatment is correct and we speculate that similar constructions can be developed for other novel cases.

Table 1 summarizes the various probabilistic formulas that we have introduced, along with calculated values for a simple example.

3.3 Residual Risks

An assurance argument may contain residual concerns (also called residual doubts, recall footnote 2): these are explicitly marked defeaters that we have been unable or have chosen not to eliminate or fully mitigate.

Residual concerns may be due to aleatoric uncertainty (i.e., uncertainty in the environment): for example, the system may be designed to withstand

Approach	**Confidence Formula**	**Example Value**
Sum of doubts (Fréchet intersection lower bound)	$1 - \min(1, d_1 + d_2 + d_3)$ or $\max(0, c_1 + c_2 + c_3 - (3-1))$	0.55
Most conservative; no dependency assumptions		
Confidence Product	$c_1 \times c_2 \times c_3$	0.612
Assumes independent subclaims		
Containment (Fréchet union lower bound)	$\max(c_1, c_2, c_3)$	0.9
Total dependence (perfect positive correlation among subclaims)		
Diversity (Product of Doubts)	$1 - d_1 \times d_2 \times d_3$	0.997
Strong assumption: doubts are independent		
Arithmetic Mean (Averaging Case)	$1/n \times \sum c_i'$	0.85
Equally weighted disjoint partitions (perfect negative correlation)		
Weighted Average (Partitioned Case)	$\sum w_i \times c_i'$; e.g., $0.5 \times 0.9 + 0.3 \times 0.8 + 0.2 \times 0.85$	0.865
Weighted disjoint partitions		
Conditional Chain (Cumulative Case)	$c_1 \times P(c_2 \mid c_1) \times P(c_3 \mid c_1 \wedge c_2)$	0.761
Used when subclaims build on each other in sequence		

Table 1: Summary of formulas for several approaches to probabilistic confidence assignments in Assurance 2.0. The example is a decomposition block with three subclaims having confidence c_1, c_2, c_3 with values 0.9, 0.85, 0.8, and conditional values $P(c_2|c_1) = 0.89$ and $P(c_3|c_1 \wedge c_2) = 0.95$. Doubts $(1 - c_i)$ are denoted d_1, d_2, d_3. Confidence within a partition $c_i \div w_i$ is denoted c_i' and the example values are reinterpreted as those c_i'.

a single sensor failure and historical evidence indicates this is sufficient, but it is always possible to encounter more. Or they may be due to epistemic uncertainty such as the fallibility of human review (e.g., human requirements tracing cannot be guaranteed to be free of error), or to limitations in automated analysis (e.g., automated static analysis may be unable to discharge some proof obligations, leading to possible false alarms that must be reviewed by humans, a potentially error-prone process). More generally, they may represent a missing or unknown subcase in a nondeductive (i.e., "inductive") reasoning step.[9]

In assessing logical soundness in an assurance case, we assume the consequences of residual concerns (i.e., residual risks) are insignificant and, on that basis, we ignore them. We thereby incur an obligation to justify this assumption and must consider the potential impact that a faulty assessment could have on failure (i.e., defeasibility) of the case. Specifically, for each residual concern, we must show that the likelihood of wrongly assessing it (as residual), combined with its worst possible consequences (i.e., its *risk*), is below some threshold for concern.

We usually assess this threshold in a qualitative manner, as described in the fourfold classification below, which is adapted from [10].

Significant: an individual residual concern poses a risk that is above the threshold for concern. In this case, the issue cannot be considered a merely "residual" risk, but must treated as a defeater and either eliminated or mitigated.

Minor: an individual residual concern poses a risk that is below the threshold for concern, but it is possible that many such might cumulatively exceed the threshold. An example could be static analysis, where we use human review to evaluate proof obligations that the automation cannot decide. These risks need to be managed explicitly: 10 might be OK, but not 100.

Manageable: a class of minor residual risks whose number and cumulative severity are below the threshold of concern.

Negligible: these are residual concerns where even multiple similar instances collectively pose a risk that is below the threshold for concern. This may arise when a source of concern occurs many times but is

[9] Residual concerns may also be used to represent nonspecific concerns that we are unable to localize more specifically.

adjudged to be trivial. An example (depending on local policy) might be "style" warnings from a static analyzer.

At final assessment, the only residual concerns remaining should be those whose risks are categorized negligible and those categorized minor but manageable.

In probabilistic assessment, we can follow the lead of logical assessment and ignore residual concerns on the basis that the classification above ensures they are truly negligible. Alternatively, we can attach a numerical assessment to each residual concern and propagate this through the argument in the usual way. The numerical assessment is our subjective confidence that the residual concern will have no impact and should be justified in an accompanying narrative that considers the likelihood of harm due to the identified concern, its maximum severity, and number of instances if it is a circumstance that may occur many times.

4 Bayesian Belief Networks

As noted in the Introduction, we have covered this topic before [37], but the presentation here is integrated into our larger narrative. In particular, it builds on Section 3.2.6 to show how confidence can be calculated for argument steps with complex dependencies among its subclaims.

The example in Section 3.2.6 has three subclaims that depend cumulatively on each other; this dependency is sufficiently straightforward that we can calculate confidence in their combination as a simple function (i.e., the product) of confidence in each of them individually. Here, we examine an example with more complicated dependencies and illustrate how tools for Bayesian Belief Networks can be used to perform the required calculations.

The example concerns assurance by testing for software correctness. The claim of correctness by testing depends on the actual correctness of the software (wrt. its specification), correctness of the specification (wrt. its informal requirements), correctness of the test oracle (wrt. the specification), coverage of the test generation process, a suitable theory of testing, and the outcome of the testing process. A skeletal assurance argument for this example is shown in Figure 10. The hexagonal node shape for `correct oracle` indicates that this is an assumption: there is no supporting evidence. To keep the presentation succinct we will ignore the test generation process (it would not illustrate any new topics), so its subcase is shown with dashed lines.

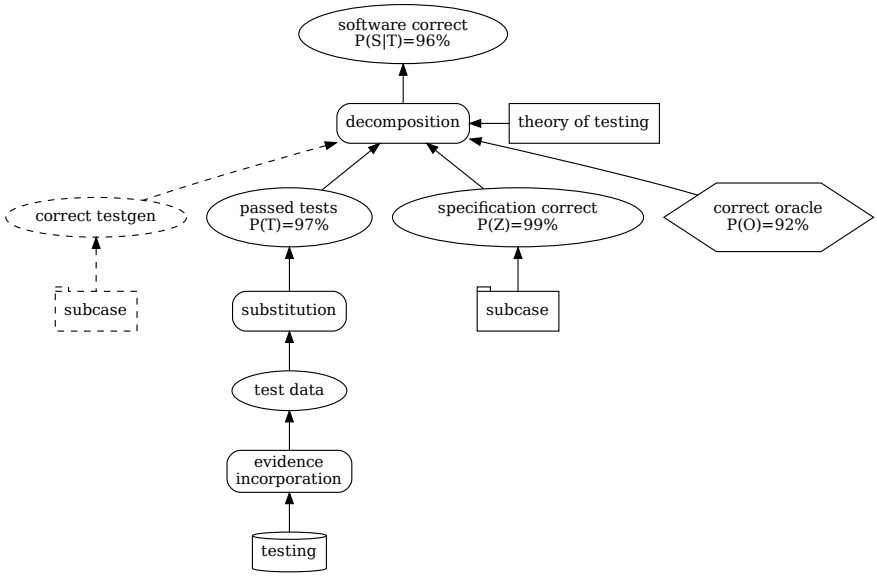

Figure 10: Argument for Correctness by Testing

The node labels of Figure 13 include confidence measures, where that for the decomposition block that delivers the top claim is calculated using the averaging method of Section 3.2.4 (and those for its subclaims are simply assumed). This calculation assumes that the subclaims to the decomposition are independent, whereas in reality the software passing its tests depends on its own correctness, the correctness of the test oracle, and the quality of the testing process (wrt. the theory employed). Correctness of software that has passed the tests then depends on correctness of its specification. These are fairly complex dependencies that are not represented in the Venn diagrams used in Section 3.2.4. Instead, we turn to Bayes' Theorem.

Bayes' Theorem is the principal method for analyzing conditional subjective probabilities or beliefs: it allows a prior assessment of probability to be updated by new evidence to yield a justified posterior probability. It is difficult to calculate over complex conditional (i.e., interdependent) probabilities, but usually the dependencies are relatively sparse and can conveniently be shown by a graph or "network," giving rise to the term "Bayesian Belief Network" or BBN. Software tools based on Bayes' Theorem are able to exploit the sparseness and can calculate probabilities associated with various scenarios relevant to the problem modeled by a BBN.

A BBN for our testing example, adapted from [32] and [37], is shown in Figure 11. The nodes of the graph represent probabilistic judgments about

components of the argument and the arcs indicate dependencies between these.

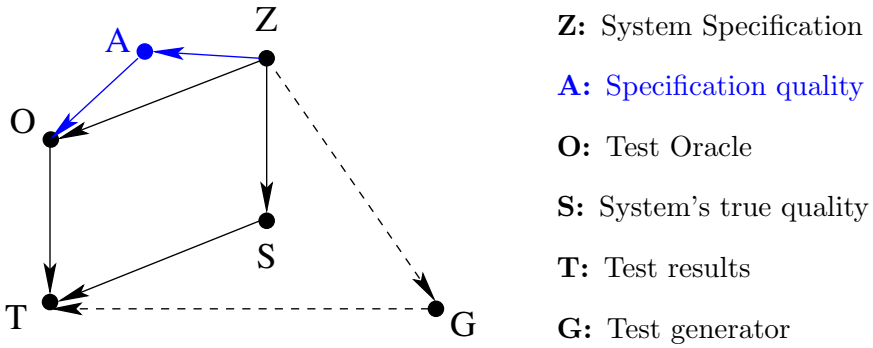

Figure 11: BBN for Testing Evidence

More precisely, the nodes of the graph represent random variables but we can most easily understand the construction of the graph by first considering the artifacts from which these are derived. Here, Z is the system specification; from this are derived the actual system S, the test oracle O (ignore, for the time being, the arcs associated with A and shown in blue) and the test generator G. Tests T are dependent on the test generator, the oracle, and the system. As before, we will ignore test generation and these arcs are shown as dashed lines.

A version of the BBN without test generation is shown represented inside the BBN tool *Hugin Expert* [24] in Figure 12. Here, the node labeled `specification` corresponds to Z, the random variable representing correctness of the system specification. It has two possible values: `correct` (i.e., it correctly represents the informal requirements established for the system) or `incorrect`. The assessor must attach some prior probability distribution to these; we will suppose 99% confidence that it is `correct`, vs. 1% that it is `incorrect`.

The node labeled `system` corresponds to S, the variable that represents the true (but unknown) quality of the system, stated as a probability of failure on demand (that is, failure wrt. the informal requirements). This probability depends on Z and is recorded in a joint probability table (we give an example below): we suppose that it is 99% if Z is `correct`, but only 50% if it is `incorrect`.

The node labeled `oracle` corresponds to O, the variable that represents correctness of the test oracle; this is derived in some way from the specification Z and its probability distribution will be some function of the

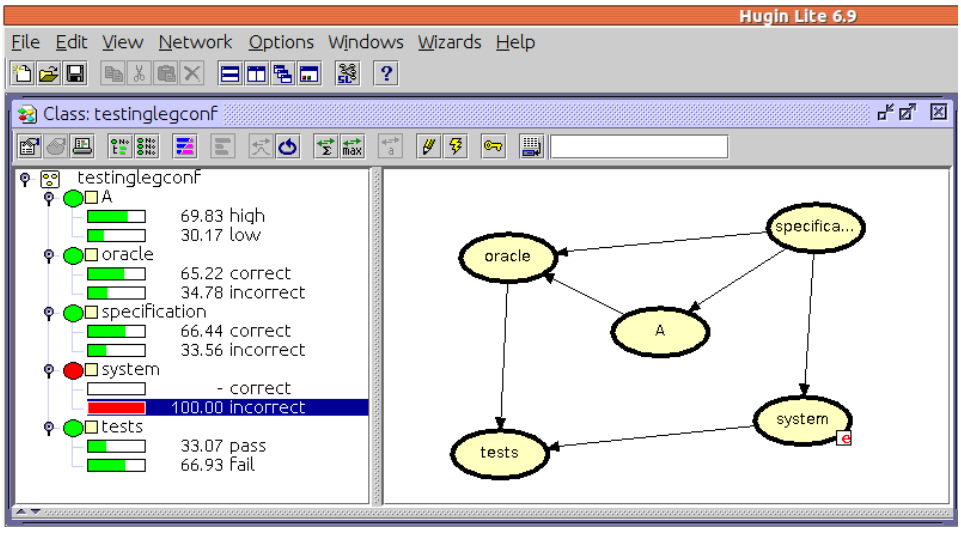

Figure 12: Hugin Analysis of BBN for Testing Evidence

correctness of Z; if Z is `correct`, we will suppose it is 95% probable that O is `correct`, but if Z is `incorrect`, then it is only 2% probable that O is `correct`

Finally, the node labeled `tests` corresponds to T, the Boolean variable that represents the outcome of testing. It depends on both the oracle O and the true quality of the system S. Its probability distribution over these is represented by a joint probability table such as the following, which gives the probabilities that the test outcome is judged successful.

S	Correct System		Incorrect System	
O	Correct Oracle	Bad Oracle	Correct Oracle	Bad Oracle
T	100%	50%	5%	30%

Using a BBN tool such as Hugin, it is possible to conduct "what if" exercises on this model. In particular, Hugin allows the user to manipulate the values of some variables and observe the impact on others. In Figure 12, we have hypothesized the system is incorrect (indicated by the red bar, and set by double-clicking on the value) and can see that the conditional probability that testing succeeds (which was earlier denoted abstractly as $P(E \mid \neg C)$) is 33.07%.

If the system is assumed correct, the probability that testing succeeds (i.e., $P(E \mid C)$) is 98.53%. We can also examine the probability of a correct

system, given that testing succeeds (i.e., $P(C \mid E)$, the usual measure of assurance), which evaluates to 99.49%, or given that it fails (i.e., $P(C \mid \neg E)$), which is 59.21%.

We see that in this model the assumed prior distributions are such that testing has rather poor evidential weight: it is disappointingly likely that an incorrect system will be accepted and that a rejected system is in fact correct. Further inspection and experimentation will show that part of the explanation is that the modeled test oracle is of low quality. The variable O has strong impact on the test outcome T but is not itself observed or evaluated (in Figure 10 it is simply assumed). We might suppose that reliability of the testing procedure would be improved if we could assess the quality of the test oracle and require this to exceed some threshold. However, it is not easy to see how this artifact can be assessed directly, so an alternative might be to assess the "testability" of the specification Z (meaning the likelihood that tests derived from Z will reveal its correctness wrt. the informal requirements), since this surely has a large impact on the quality of the oracle. Reasoning similar to this may implicitly underlie some of the DO-178C guidelines for software assurance in civil aircraft [35]: for the most critical software, DO-178C specifies 71 assurance "objectives" that must be accomplished and several of these concern attributes of requirements and specifications. For example, its Section 6.3.2.d specifies the objective to "ensure that each low-level requirement can be verified."

We can introduce this idea into our model as the variable A in Figure 11 (with dependencies indicated in blue) and similarly in Figure 12. Here A assesses confidence that the specification Z is testable and takes values `high` and `low`; we suppose the probability that A is `high` is 95% when Z is `correct` and 20% when it is `incorrect`. There is no arc from A to S because A is not a general evaluation of the specification, just its testability. The probability distribution of O will now depend on both A and Z and we suppose it takes the following form.

Z	Correct Specification		Incorrect Specification	
A	High Conf	Low Conf	High Conf	Low Conf
O	99%	70%	2%	1%

If we require that A is `high` before we undertake testing, then we find that the probability of accepting an incorrect system is reduced from 33.07% to 13.33% while the probability of accepting a correct system increases from 98.53% to 99.45%. The probability the system is correct, given that testing succeeds, improves from 99.49% to 99.85% and, if testing fails, the probability the system is correct reduces from 59.21% to 36.33%.

Of course, we could informally apply similar reasoning to the argument of Figure 10 and derive the revised argument shown in Figure 13. As in the BBN analysis, we can observe that confidence in the test oracle is dragging the overall confidence down and we can add an evaluation of specification quality as shown (in blue) in Figure 13 where correctness of the oracle, which was previously an assumption, is now supported by a subargument (shown in blue) over testability of the specification and reviews of the oracle (recorded as the variable R). The node labels of Figure 13 include confidence measures, where those for the two decomposition blocks are evaluated using the averaging method (and those for their subclaims are simply assumed).

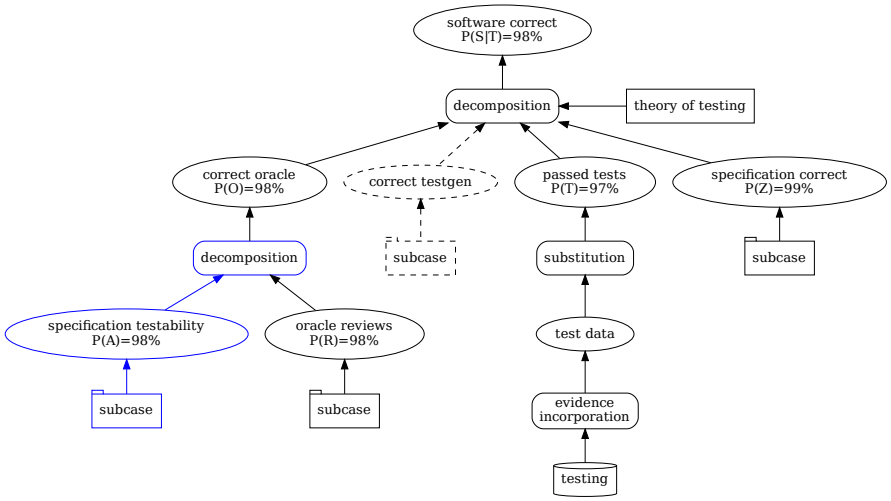

Figure 13: Revised Argument for Correctness by Testing

The primary difference between confidence propagation in the assurance argument and the BBN analysis is that the former treats the subclaims as if they are independent, while the latter explicitly models their dependencies. Both can reveal sources of weakness in an argument but a BBN allows this to be modeled more accurately and allows calculation of more probabilistic quantities than just propagation.

In this example, the probabilities and distributions used were "plucked from the air" and cannot be considered realistic. But in a real case, estimating probability values and tables forces serious consideration of the interdependencies among subclaims. However, it is not necessarily the actual values used and calculated that are important but the relationships among them. We believe that "what-if" explorations such as those performed here can develop understanding of these relationships and thereby help guide selection of subcases and evidence.

Because BBNs model relationships that are absent from assurance arguments, there is no automatic way to derive a BBN from an argument: it requires a creative act (compare Figures 10, 11, and 14 for example). Accordingly, we recommend that confidence propagation in assurance case arguments is used as the default method for quantified assessment, but that BBN analysis should substitute or augment basic propagation through those decomposition blocks whose subclaims have complex dependencies (simply drawing the corresponding BBN should reveal when this is so). Note that BBNs can also include the sideclaim, which for simplicity we have assumed here is treated separately.

Situations where several items of argument or evidence support a single claim are important. One approach assumes the separate subcases are independent and treats them as elements of a diversity argument; another explicitly models their dependencies in a BBN, as Littlewood and Wright do for the combination of testing and formal verification [32]. A third approach, which we advocate but will not otherwise discuss here, is to develop a separately justified theory for a specific combination of evidence: for example, MC/DC unit testing is known to detect certain fault classes [31] and we could imagine a theory that combines this with a method of static analysis that detects other fault classes.

5 Comparison With Other Methods

There is much work on quantified assessment of assurance cases. Graydon and Holloway [17, 18] provide a comprehensive description and evaluation of 12 different methods, of which six are based on Dempster-Shafer Theory (DST) or some other form of Evidential Reasoning, five are based on BBNs, and one uses weighted averages of "attributes."

Here, we sketch some of the methods examined by Graydon and Holloway, and highlight our differences in approach and calculation. In the section after this we outline the shortcomings identified in these methods by Graydon and Holloway and consider whether they apply to our method.

Most of the other methods start from a structured assurance argument similar to us, though expressed in different notations, and we will consider them first (the remaining methods start from a BBN). Their methods of quantification are based on probabilistic modeling and seem to agree (among themselves and with us) in the way they deal with assumptions and with argument nodes having a single subcase. Where they differ is in their treatment of evidence nodes and argument nodes with multiple subclaims. The

differences can be in basic analysis of these nodes (e.g., which circumstances are recognized and treated specially) and/or the probabilistic modeling used to represent them (e.g., different applications of DST)

We note that none of the other methods have residual concerns or sideclaims in their arguments, nor do they see argument steps as applications of a theory (where sideclaims assess whether the theory is suitable and is used correctly). And several other methods use notations such as GSN (Goal Structuring Notation) [2] that look superficially similar to Assurance 2.0 but have different semantics. Furthermore, the other methods do not separate logical and numerical/probabilistic assessments: their numerical assessments have to determine basic soundness of the case as well as purposes specifically served by quantification.

We begin our comparison with an outline of DST and some methods that use it.

5.1 Methods Based on Dempster-Shafer Theory and Other Forms of Evidential Reasoning

Dempster-Shafer Theory was primarily developed as a theory of evidence where we have epistemic uncertainty on the interpretation of our data. For example, we may wish to determine whether a signal light is red, yellow, or green, but the best we can do with our noisy sensors is assess a belief *mass* (subjective probability) of so much for red, so much for yellow, and so much for green, and also so much for the weaker judgments that it is *either* red or yellow, red or green, and yellow or green, and also so much that it could be *any* of the three colors. In other words, we assign belief masses to the powerset of the set of basic judgments. The *belief* in a judgment is then the sum of the mass of its subsets, so the belief for "red or yellow" is the mass for that combination, plus those for red and yellow individually. In addition to belief in a judgment, we have its *plausibility*, which is 1 minus belief in its negation, so that the plausibility of red is 1 minus belief in "yellow or green,"

If we have evidence from several sources, then we will wish to combine their masses and associated functions into a consensus. Dempster's *rule of combination* assumes the separate sources are independent and the combined belief in a judgment (e.g., light is green) is the sum of the individual (positive) beliefs for that judgment from the different sources multiplied by a *correction factor* related to conflicting beliefs (i.e., its plausibility for each source). Many different combination rules have been proposed since

the original DST [39] and a variety of these are used by the methods that Graydon and Holloway consider.

Ayoub and colleagues [3] divide the different argument steps in a similar way to us. For evidence they use a "basic probability assignment" similar to our $P(C \mid E)$, but do not have the separation into measured and useful claims, nor do they have sideclaims. Nor do they use anything comparable to confirmation measures to assess the epistemic significance and relevance of evidence.

Their approach to reasoning nodes that have a single subclaim is similar to ours, except they do not apply a function `f`, nor do they have sideclaims. Where Ayoub and colleagues differ most from us is in probabilistic modeling of decomposition steps, where they use DST. They divide cases in a similar way to us: what we call diversity, they call an "alternative" argument, and what we call partitioning, they call a "disjoint" argument. We will use our terminology. For partitioning arguments, they favor a "weighted mixing" combination rule which evaluates similarly to our partitioning calculation, but for diversity arguments they revert to a standard DST combination rule and obtain very different results to us. They also have containment arguments and other cases that they reduce to diversity and partitioned arguments in ways similar to our Sections 3.2.5 and 3.2.7. However, their numerical results usually differ from ours because their treatment of the underlying diversity cases is different to ours,

The method of Zeng, Lu, and Zhong [48] is similar to that just considered except that the subclaims to all nodes are assigned a weight as well as a confidence (like our partitioned cases) and are combined using an "improved Dempster's Rule."

Cyra and Górski [11] present a method for "Trust-IT" arguments that are rather different to the arguments of Assurance 2.0, so detailed comparison is difficult. In particular, they can have counterevidence or counterarguments present in the final case, whereas we regard these as defeaters that may be used to challenge the case during development and evaluation, but must be resolved in the final case, which is entirely positive.

They use Jøsang's Opinion Triangle [30] to evaluate evidence and other leaf nodes. The Opinion Triangle considers belief, disbelief, and uncertainty and has similar motivation, but different calculations, to the confirmation measures that we use. The resulting assessments are converted into DST belief and plausibility measures and these are propagated up the argument, step by step, using DST with a qualitative probability scale that combines "decision" and "confidence" to yield a 24-point scale.

They divide argument steps into four kinds and use different DST combination rules for each. Their "complementary" arguments are like our partitioned cases but may have a "gap" that is not covered by any subcase (which we would absorb into the sideclaim). Quantification for this case uses a modified DST combination rule rather similar to that of Ayoub and colleagues. Their "alternative" arguments correspond to our diversity cases and are quantified using Yager's [39] modified DST combination rule. Their "Necessary and Sufficient Condition List" arguments resemble our logical assessment of decomposition blocks, while their "Sufficient Condition List" arguments are similar except that not all the subclaims are necessary (so we might combine the "overlapping" subclaims into a diversity subcase). Both of these are quantified using novel DST combination rules.

Like Cyra and Górski, Duan and colleagues [16] use Jøsang's Opinion Triangle, but they apply it to the entire argument, not just the evidence nodes. In other words, they use Jøsang's method for evidential reasoning in place of DST. The method represents degrees of belief, disbelief, and uncertainty according to beta distributions and has formulas for combining multiple assessments similar to the rules of combination in DST. Like DST, we consider this method ill-suited to quantifying confidence in Assurance 2.0, where our goal is to assess the strength of a logically sound argument, not to assess whether it is sound.

Guiochet, Hoang, and Kaâniche [19] introduce a method based on DST that uses modified rules of combination so that its treatment for diversity and partitioned cases is arithmetically similar to ours except that it has some additional parameters. However, when an argument node has a context node attached (they assume GSN arguments), the analysis instead uses a BBN model similar to Figure 14. They treat context nodes rather like our sideclaims, and this is criticized by Graydon and Holloway who remark that in GSN "context is not a proposition" [17, page 59].

Nair and colleagues [33] use a method for evidential reasoning due to Yang and Xu [47] that is different to DST, but related to it. They do not apply evidential reasoning to the assurance case itself, but to a separate assessment of confidence in the case (a "confidence argument" [21]). Their constructions are somewhat specific to GSN and at variance with Assurance 2.0. In particular, probabilistic assessment in Assurance 2.0 is applied only to assurance arguments that are judged to be indefeasibly sound, whereas these assessments are combined in the method of Nair and colleagues.

At the beginning of this section we stated that evidential reasoning, whether based on DST, Jøsang's Triangle, or other methods, is an uncomfortable fit for Assurance 2.0 and we believe this is confirmed by the methods

sketched above. As the term "evidential reasoning" suggests, these methods are plausible for the evaluation of evidence with respect to claims (where we use confirmation theory), although Dempster's rule of combination has been called into question [14, 43] and the plethora of alternative rules indicates the details of its application have not been settled satisfactorily. But once we have incorporated evidence into claims, the interior part of an assurance case argument is about logical reasoning and this is poorly, not to say inaccurately, represented by DST and similar methods. We believe this is because these methods are attempting to perform three different assessments simultaneously: logical soundness, dialectical examination, and evaluation of argument strength. In assurance 2.0, we perform these assessments separately. Dialectical examination is performed with defeaters, counterarguments and counterevidence. Only once these have been resolved do we proceed to logical soundness, where we have only positive arguments and can employ standard logical reasoning. And only once soundness is confirmed do we assess the strength of the argument, using elementary probabilistic modeling.

In conclusion, we do not consider DST and other forms of evidential reasoning to be applicable or to add value to Assurance 2.0 and so we next turn to methods based on BBNs.

5.2 Methods Based on Bayesian Belief Networks

In Section 4 we described the use of BBNs to examine the consequences of dependencies among the subclaims of a decomposition block within an assurance argument. The BBN was constructed by human evaluators and served to augment and refine the standard confidence assessments for an assurance argument.

Among the BBN-based methods considered by Graydon and Holloway, those of Guo [20], Hobbs and Lloyd [23], and the SERENE Partners [40] use BBNs of a similar form to those in Section 4 but construct them directly, without first developing an explicit assurance argument: in effect, the BBNs *are* their representation of the argument. Because they do not relate their methods to assurance arguments as we understand them in Assurance 2.0, we find their methods remote from our concerns and do not consider them further.

The method of Denney, Pai, and Habli [13] does base its BBN on an assurance argument, but does so in an automated fashion, As we explained in Section 4, the dependencies among subclaims that are modeled and analyzed by BBNs are not represented in assurance arguments and so a BBN of this

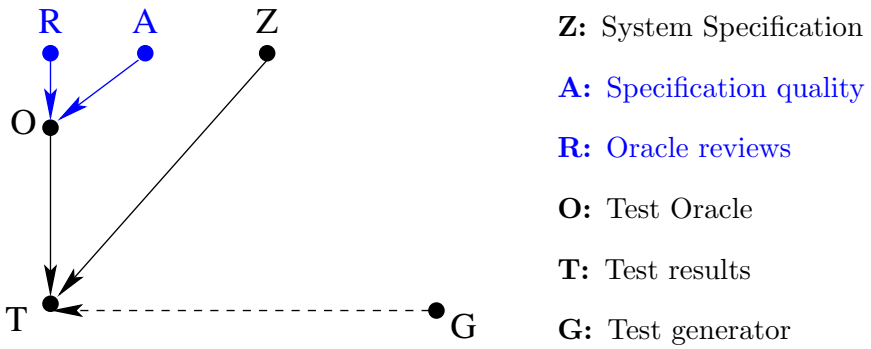

Figure 14: BBN Derived from Assurance Argument for Testing Evidence

form requires human insight and cannot be derived automatically from the argument. Consequently, the BBNs of Denney, Pai, and Habli have an impoverished form as illustrated in Figure 14, which is derived from Figure 13. This BBN has an arc from each (variable associated with a) claim to its parent, and is therefore isomorphic to the argument structure (with its arrows reversed, and drawn upside-down). Denney, Pai, and Habli do add a random variable "assurance deficits acceptable," supported by variables "argument sufficient" and "context appropriate" to the BBN, but these are unrelated to the explicit argument structure. The variables of the BBN are discretized probabilities with normal distributions assumed (presumably to simplify the calculations). It seems to us that this method introduces the complexity of BBNs while lacking their chief merit: the ability to model dependencies among subclaims to decomposition nodes.

Zhao and colleagues [49] convert an assurance case argument into a sequence of Toulmin arguments [44] and then build a BBN for each. Their BBNs employ standard patterns and thereby fail to model dependencies among subclaims. We consider the excursion into Toulmin arguments and their method of BBN construction to be sufficiently different in philosophy and technique to ours that we do not find their techniques applicable to our concerns.

5.3 Other Methods

Yamamoto [46] describes a method for evaluating "attributes" of a GSN assurance argument. Attributes are not probabilities nor estimates of confidence; they are judgments about the truth of claims, represented as integers

on a five-point scale, where -2 corresponds to "strongly unsatisfied," -1 to "unsatisfied," 0 to "unknown," 1 to "satisfied" and 2 to "strongly satisfied."

He uses a calculation for GSN strategies that resembles our partitioned case, but provides no justification. His description is terse and difficult to understand. We reproduce it here in full [46, Section IV, A]:

> 2) Sub claim to parent claim propagation through strategy
>
> "Let a parent claim is decomposed by k sub claims through a strategy. And let $\ll P \gg$ be the attributes of the parent. Let $\ll Q_1, \ldots, Q_k \gg$ be the attribute of the strategy. Let $\ll R_1 \gg$ $, \ldots, \ll R_k \gg$ be the attributes of k sub claims, respectively.
>
> "Then the attribute value P of the parent claim is calculated by the following equation.
>
> $P = (\Sigma_{i=1,k}\ Q_i \times R_i \times W_i)$, where $\Sigma_{i=1,k}\ W_i = 1$"

Earlier, he explains that the W_i are weights:

> "$W_1, \ldots, W_k$, where k is the number of sub claims of the corresponding strategy. W_i is the weight for the i'th sub claim."

It is not clear how to interpret the Q_i and they are not used in the case study [46, Section V], so we will ignore them and the formula is then

$$P = \Sigma_{i=1,k}\ R_i \times W_i,$$

which is the same as our formula for partitioned decomposition blocks (6) stated in Section 3.2.4.

However, although Yamamoto's arithmetic calculation is the same as ours, he applies it to different quantities ("attributes" rather than probabilistic confidence) and provides no justification. Furthermore, he does not stipulate that it applies only to partitioned decompositions and does not consider the diversity case for decompositions. Consequently, we regard our method as distinct from his.

6 Graydon and Holloway's Critique

For many of the 12 methods they examine, Graydon and Holloway reproduce the authors' examples and then construct variants that generate implausible results. Many of the variants are based on similar ideas, so we next consider those. We reference the technical report [17] as its appendices give more details than the paper [18], and we use paragraph headings taken from the report.

Many Subclaims Examples [17, pp. 36–39]. Graydon and Holloway develop this group of examples for the method of Ayoub and colleagues [3] but provide similar constructions for several other methods. Abstracting from the details, the idea is that we have a decomposition step with many subcases, say 20. Of these, 19 have high confidence and one is very low. Most methods, including those introduced in Section 3, are not particularly sensitive to the degree of confidence in one subclaim among many, so confidence in the parent claim will be dominated by the majority and will therefore be high in this example. If acceptability of the argument is based on numerically high confidence, then this argument will be accepted despite having one very weak element, and Graydon and Holloway rightly regard this as a cause for concern.

But this is not how Assurance 2.0 works, and the example is no threat to our methods. In Assurance 2.0, we do not perform probabilistic assessment until we have determined that the argument is logically sound, which means each parent claim must be deductively entailed by its subclaims. The subclaim with low confidence would surely not have been adjudged `true` in logical assessment and so its parent claim and all those above it (including the top claim) will be `false` or `unsupported` and this assessment must pause while the case is examined and corrected.

On the other hand, logical assessment will not highlight the case where subclaims are plausibly judged `true` but with less certainty than others: it does not even provide a way to talk about this. Probabilistic assessment in Assurance 2.0 is a lens that gives a different view of the case and allows us to speak rationally of confidence, so that weakly `true` subclaims that might otherwise have gone unremarked will be identified and can then lead to further investigation.

Graydon and Holloway consider variants on this scenario where a subclaim provides undermining or counter-evidence but is overwhelmed by other, positive, subclaims so that the overall assessment is positive. These circumstances can arise in Yamamoto's method and also in those using DST and other evidential reasoning, where quantification extends to disbelief as well as belief. But in Assurance 2.0 we examine disbelief using defeaters and do this—and adjust matters to rectify serious sources of concern—prior to probabilistic assessment, where the case is entirely positive and the proposed scenarios cannot arise.

The examples with many subclaims also raise the question of arbitrary scope: why do we have these subclaims and not some other number and selection? Graydon and Holloway consider methods that perform calculations similar to our partitioning case (Section 3.2.4) and suppose we have two sub-

claims with confidence 95% and 50%, respectively. The averaging variant of our partitioning case will calculate confidence in the parent claim as 72.5%. Now suppose we replace the first subclaim by 10 smaller subclaims, all with the same 95% confidence. The averaging method now calculates the parent confidence as 90.9%.

Graydon and Holloway criticize this example's sensitivity to an arbitrary choice in how to divide subclaims, but we consider it contrived. The idea of a partitioning case is that the argument naturally divides into separate subclaims, so the choice is unlikely to be arbitrary. And where there is some flexibility of choice, confidence in the subclaims will vary appropriately and the parent confidence should not be distorted. As a trivial example, we could replace evidence from a test campaign with evidence from the same campaign divided into 10 parts—but confidence in each smaller campaign will be less than that in the undivided whole, so the parent confidence will not change wildly but appropriately.

Imperfect Examples [17, page 45]. Graydon and Holloway provide several examples that challenge the method of Cyra and Górski. Essentially, one of Cyra and Górski's formulas calculates confidence in a parent claim as something similar to the product of confidence in its subclaims. Consequently one subclaim of slightly lower confidence than the others will sharply lower confidence in the parent claim; confidence in the parent claim is also very sensitive to the number of subclaims.

Their method is similar to our older "product" calculation of Section 2 and observations such as these are the reason we now deprecate its use except in special circumstances (e.g., recall Section 3.2.6, where it is exactly the right method). However, we note that calculations that are sensitive to subclaims of lower confidence and to numbers of subclaims, such as product and sum of doubts, can be useful indicators of circumstances where further investigation may be warranted, so although we no longer recommend these methods as the primary means of probabilistic assessment, we can recommend them as special-purpose supplements to those of Section 3.

Optimistic and Pessimistic Counterexamples [17, pp. 87–90]. Graydon and Holloway consider the trivial example from Zhao and colleagues [49, Figure 7] that we translate from GSN to Assurance 2.0 and reproduce in Figure 15. They show that when confidence in the sideclaim (that establishes completeness of the hazard list) is changed from the original 85% to 99.9% and then to 0.01%, confidence in the top claim of system

safety changes from 89.460% to 89.996% and then to 86.404%. They rightly note that "it is not plausible that extreme changes in confidence in hazard analysis would produce as small a change in confidence in safety."

In Assurance 2.0, the version with confidence 0.01% would be rejected as logically unsound, as a sideclaim with such low confidence could not possibly be adjudged `true`. For the other cases, our methods of Section 3 calculate confidence in a parent claim as the product of confidence in its combined subclaims (independently of how that is calculated) and its sideclaim. Zhao and colleagues assume confidence in the combined subclaims is 90%, so our methods deliver 76.5% confidence in the top claim when sideclaim confidence is 85% and 89.9% when it is 99.9% (and would deliver 0.009% if we persisted with sideclaim confidence of 0.01%). We consider these to be appropriate values.

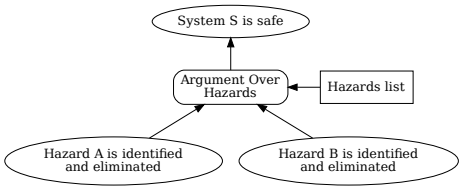

Figure 15: Example from [49, Fig. 7]

The method of Zhao and colleagues uses BBNs, which we also endorse in suitable circumstances but, as we noted in Section 4, BBNs need to be constructed to reflect the specific circumstances of the argument under consideration. We suggest that this example of Graydon and Holloway demonstrates the unsuitability of generic "one size fits all" BBN constructions. Similar examples and observations apply to other BBN-based methods and examples examined by Graydon and Holloway.

7 Summary and Conclusion

An Assurance 2.0 argument may contain probabilistic claims, including claims about confidence, and can employ different kinds of probabilistic reasoning within the argument. We distinguish four kinds and a single assurance case may employ elements of all four.

1. At one extreme there are no probabilistic claims, only unconditional claims such as "implementation is correct wrt. specification" supported by similarly unconditional reasoning.

2. Next, we may have qualitative claims about stochastic properties such as "the new system will be no less reliable than the old" with limited evidence and reasoning about relative reliability.

3. Then, we may have quantitative claims about stochastic properties such as "reliability is better than x" supported by evidence, theories, and reasoning about these properties.

4. Finally, we may have claims explicitly about confidence, such as "95% confidence that reliability is better than x" supported by suitable probabilistic theories and reasoning.

The last three of these include various amounts and degrees of numerical and probabilistic reasoning that is internal to (i.e., part of) the argument. In addition, the first three can be supported by probabilistic reasoning that is external to (i.e., separate from) the reasoning in the argument (so can the fourth, but it seems redundant). This external reasoning assesses *confidence* in each claim: that is, a subjective probabilistic assessment of its truth taking aleatoric and epistemic uncertainties into account. This is different to the logical assessment of truth, which assesses validity and soundness of the argument (given that all evidence and all reasoning steps are justified indefeasibly).

Probabilistic confidence assessment in Assurance 2.0 is performed only for arguments that have been judged logically sound. Concerns are explored by dialectical examination using defeaters and is a separate (prior) activity to logical and probabilistic assessment, which are applied only to strictly positive cases in which any residual concerns are explicitly noted and accepted. The value added by probabilistic confidence assessment is to supplement dialectical and logical assessments (by providing additional information or a different point of view), to enable one argument to be compared with another, to examine allocation of effort across an argument, and to support rational tradeoffs of effort against risk for graduated levels of assurance.

An example supplementation is "chain of confidence" reasoning where we model the impact of being wrong, and how wrong we might be [26, Annex I 1.5]. For example, we may estimate R_{ok} as a measure of risk posed by the system, given the assurance argument. But if P_{ok} is the probability our assurance argument is correct (i.e., our confidence in it), and $R_{\neg\text{ok}}$ is an estimate of risk absent this argument, then the chain rule gives

$$R = P_{\text{ok}} \times R_{\text{ok}} + (1 - P_{\text{ok}}) \times R_{\neg\text{ok}}$$

and we can use this to estimate a refined measure of overall risk R.

In this report, we have described a new method for calculating confidence in the claims of an Assurance 2.0 argument. The method proceeds one argument step after another, propagating confidence from leaf nodes such

as evidence, assumptions, and residual concerns, up to the top claim. At the leaves, confidence in evidence `E` is estimated as `P(C | E)` where `C` is its "something useful" claim. Confidence in assumptions is estimated by subject matter experts or other assessors. Confidence for residual concerns is an estimation of confidence that they are truly residual (i.e., pose only negligible risk).

At each argument step above the leaves, the method estimates the quantity of concerns in the parent claim that are eliminated by its subclaims. For decomposition steps, the assessment depends on the reasoning concerned: for example, do the subclaims partition the concerns, or do they eliminate concerns in diverse ways? Despite their differences, these assessments are all derived from the same concern-elimination model and are easy to understand, to justify, and to calculate. In some circumstances, where subclaims have complex interdependencies, we endorse use of BBNs, but these must be carefully crafted to represent the dependencies concerned and cannot use the generic forms proposed for some other methods. For most purposes, however, the complexity of an ad-hoc BBN is unnecessary: a specialized theory for the relevant dependencies (e.g., for testing used in combination with static analysis) will be preferable, while a conservative approximation using one of our basic constructions will often suffice.

At each argument step, calculated confidence in the parent claim is multiplied by confidence in its sideclaim, if there is one (because truth of the parent follows from the subclaims *given* the sideclaim), and ad-hoc adjustments, represented by a function `f` or a product `k`, may be applied if warranted.

Graydon and Holloway examine several methods for probabilistic assessment of assurance cases and show that all of them can generate implausible results [17,18]. Most of these arise because the methods do not assess logical soundness separately from their probabilistic calculations, which can therefore be applied to unsound arguments. Our method is not susceptible to these or other counterexamples developed by Graydon and Holloway.

Several experiments, surveys, and interviews have attempted to explore users opinions and experiences in evaluating assurance cases. Graydon and Holloway mention (and criticize) a couple and Diemert, Shortt, and Weber describe several more and present their own [15]. Few of the participants in these reviews show much enthusiasm for quantitative methods. Diemert and colleagues report skepticism that nuanced assessments can be "reduced to a number" and also doubts that quantitative methods produce trustworthy results. We hope that the role we propose for quantitative methods in Assurance 2.0, where they complement reviews and logical and dialecti-

cal assessments, and the intuitive and simple calculations that underlie our methods, can allay these doubts and concerns and will prove valuable when using Assurance 2.0.

One of the roles where we propose that our quantitative methods may be particularly useful is in evaluating tradeoffs in effort *vs.* confidence when developing graduated assurance. Daw, Beecher, and Holloway discuss these tradeoff issues [12] and it would be interesting to compare our quantitative methods with their informal approaches.

Acknowledgment and Disclaimer.

We thank Peter Bishop, Bev Littlewood, and Andrey Povyakalo of City St George's for their challenging critique of some of our description and derivations.

This material is based upon work supported by the United States Air Force and DARPA under Contract No. FA8750-23-C-0519 and is released under Distribution Statement "A" (Approved for Public Release, Distribution Unlimited). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force, DARPA, or the United States Government.

References

[1] Ernest Wilcox Adams. *A Primer of Probability Logic*. Center for the Study of Language and Information (CSLI), Stanford University, 1998.

[2] *Goal Structuring Notation Community Standard Version 3*. The Assurance Case Working Group, York, UK, May 2021.

[3] Anaheed Ayoub, Jian Chang, Oleg Sokolsky, and Insup Lee. Assessing the overall sufficiency of safety arguments. In Chris Dale and Tom Anderson, editors, *Assuring the Safety of Systems: Proceedings of the 21st Safety-Critical Systems Symposium*, pages 127–144, Safety-Critical Systems Club, Bristol, UK, February 2013.

[4] Stephen Barrett et al. Assessing confidence in frontier AI safety cases. arXiv:2502.05791, February 2025.

[5] Robin Bloomfield and Bev Littlewood. Multi-legged arguments: The impact of diversity upon confidence in dependability arguments. In *The*

International Conference on Dependable Systems and Networks, pages 25–34, IEEE Computer Society, San Francisco, CA, June 2003.

[6] Robin Bloomfield, Kate Netkachova, and John Rushby. Defeaters and eliminative argumentation in Assurance 2.0. Technical Report SRI-CSL-2024-01, Computer Science Laboratory, SRI International, Menlo Park, CA, May 2024. Also arXiv:2405.15800.

[7] Robin Bloomfield and Kateryn Netkachova. Building blocks for assurance cases. In *ASSURE: Second International Workshop on Assurance Cases for Software-Intensive Systems*, pages 186–191, IEEE International Symposium on Software Reliability Engineering Workshops, Naples, Italy, November 2014.

[8] Robin Bloomfield and John Rushby. Assurance 2.0: A Manifesto. In Mike Parsons and Mark Nicholson, editors, *Systems and Covid-19: Proceedings of the 29th Safety-Critical Systems Symposium (SSS'21)*, pages 85–108, Safety-Critical Systems Club, York, UK, February 2021. Preprint available as arXiv:2004.10474.

[9] Robin Bloomfield and John Rushby. Confidence in Assurance 2.0. Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, May 2022. Updated May 2024. Also available as arXiv:2205.04522.

[10] Robin Bloomfield and John Rushby. Confidence in Assurance 2.0 Cases. In Ana Cavalcanti and James Baxter, editors, *The Practice of Formal Methods: Essays in Honour of Cliff Jones, Part I*, Volume 14780 of Springer-Verlag *Lecture Notes in Computer Science*, pages 1–23, Springer-Verlag, York, UK, September 2024. Expanded version available at arXiv:2409.10665.

[11] Lukasz Cyra and Janusz Górski. Support for argument structures review and assessment. *Reliability Engineering & System Safety*, 96(1):26–37, 2011. Special Issue on Safecomp 2008.

[12] Zamira Daw, Scott Beecher, and Michael Holloway. Leveling arguments: Easier said than done. In *42nd AIAA/IEEE Digital Avionics Systems Conference*, Barcelona, Spain, October 2023.

[13] Ewen Denney, Ganesh Pai, and Ibrahim Habli. Towards measurement of confidence in safety cases. In *5th ACM/IEEE International Sym-*

posium on Empirical Software Engineering and Measurement, pages 380–383, Banff, Canada, 2011.

[14] Jean Dezert, Pei Wang, and Albena Tchamova. On the validity of Dempster-Shafer theory. In *15th International Conference on Information Fusion*, pages 655–660, Singapore, July 2012.

[15] Simon Diemert, Caleb Shortt, and Jens H. Weber. How do practitioners gain confidence in assurance cases? *Information and Software Technology*, (185):107767, 2025.

[16] Lian Duan et al. Representing confidence in assurance case evidence. In SafeComp *Workshops, ASSURE, DECSoS. ISSE, ReSA4CI, and SASSUR*, Volume 9338 of Springer-Verlag *Lecture Notes in Computer Science*, pages 15–26, Springer-Verlag, Delft, The Netherlands, September 2015.

[17] Patrick J. Graydon and C. Michael Holloway. An investigation of proposed techniques for quantifying confidence in assurance arguments. Technical Memorandum NASA/TM-2016–219195, NASA Langley Research Center, Hampton VA, May 2016.

[18] Patrick J. Graydon and C. Michael Holloway. An investigation of proposed techniques for quantifying confidence in assurance arguments. *Safety Science*, 92:53–65, February 2017.

[19] Jérémie Guiochet, Quynh Anh Do Hoang, and Mohamed Kaaniche. A model for safety case confidence assessment. In *Computer Safety, Reliability, and Security* (SafeComp), Volume 9337 of Springer-Verlag *Lecture Notes in Computer Science*, pages 313–327, Springer-Verlag, Delft, The Netherlands, September 2015.

[20] Baofeng Guo. Knowledge representation and uncertainty management: Applying Bayesian Belief Networks to a safety assessment expert system. In *IEEE International Conference on Natural Language Processing and Knowledge Engineering*, pages 114–119, Beijing, China, 2003.

[21] Richard Hawkins, Tim Kelly, John Knight, and Patrick Graydon. A new approach to creating clear safety arguments. In Chris Dale and Tom Anderson, editors, *Advances in System Safety: Proceedings of the Nineteenth Safety-Critical Systems Symposium*, pages 3–23, Springer, Southampton, UK, February 2011.

[22] Kelly J. Hayhurst, Dan S. Veerhusen, John J. Chilenski, and Leanna K. Rierson. A practical tutorial on modified condition/decision coverage. NASA Technical Memorandum TM-2001-210876, NASA Langley Research Center, Hampton, VA, May 2001.

[23] Chris Hobbs and Martin Lloyd. The application of Bayesian Belief Networks to assurance case preparation. In *Achieving Systems Safety: Proceedings of the Twentieth Safety-Critical Systems Symposium*, pages 159–176, Springer, Bristol, UK, February 2012.

[24] HUGIN Expert. *Hugin home page*, Retrieved 2015. http://www.hugin.com/.

[25] Hidetomo Ichihashi and Hideo Tanaka. Jeffrey-like rules of conditioning for the Dempster-Shafer theory of evidence. *International Journal of Approximate Reasoning*, 3(2):143–156, 1989.

[26] *Dependability Assessment of Software for Safety Instrumentation and Control Systems at Nuclear Power Plants*. International Atomic Energy Agency, 2018. Nuclear Energy Series, NP-T-3.27.

[27] *IEC 61508—Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*. International Electrotechnical Commission, Geneva, Switzerland, March 2004. Seven volumes; see http://www.iec.ch/zone/fsafety/fsafety_entry.htm.

[28] *Road Vehicle—Functional Safety*. International Organization for Standardization, Geneva, Switzerland, 2011. ISO Standard 26262 (in 10 parts).

[29] Richard Jeffrey. *Subjective Probability: The Real Thing*. Cambridge University Press, 2004.

[30] Audun Jøsang. *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer, 2016.

[31] D. Richard Kuhn. Fault classes and error detection capability of specification-based testing. *ACM Transactions on Software Engineering and Methodology*, 8(4):411–424, 1999.

[32] Bev Littlewood and David Wright. The use of multi-legged arguments to increase confidence in safety claims for software-based systems: a study based on a BBN analysis of an idealised example. *IEEE Transactions on Software Engineering*, 33(5):347–365, May 2007.

[33] Sunil Nair et al. An evidential reasoning approach for assessing confidence in safety evidence. In *IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*, pages 541–552, Gaithersbury, MD, November 2015.

[34] Nils J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–87, 1986.

[35] RTCA. *DO-178C: Software Considerations in Airborne Systems and Equipment Certification*. Requirements and Technical Concepts for Aviation (RTCA), Washington, DC, December 2011.

[36] John Rushby. Mechanized support for assurance case argumentation. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2013 Workshops, LENLS, JURISIN, MiMI, AAA, and DDS, Revised Selected Papers*, Volume 8417 of Springer-Verlag *Lecture Notes in Artificial Intelligence*, pages 304–318, Springer-Verlag, Kanagawa, Japan, October 2013.

[37] John Rushby. On the interpretation of assurance case arguments. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2015 Workshops, LENLS, JURISIN, AAA, HAT-MASH, TSDAA, ASD-HR, and SKL, Revised Selected Papers*, Volume 10091 of Springer-Verlag *Lecture Notes in Artificial Intelligence*, pages 331–347, Springer-Verlag, Kanagawa, Japan, November 2015.

[38] John Rushby. The indefeasibility criterion for assurance cases. In Yamine Ait-Ameur, Shin Nakajima, and Dominique Méry, editors, *Implicit and Explicit Semantics Integration in Proof Based Developments of Discrete Systems*, Communications of NII Shonan Meetings, pages 259–279, Springer, Kanagawa, Japan, July 2020. Postproceedings of a workshop held in November 2016.

[39] Kari Sentz and Scott Ferson. Combination of evidence in Dempster-Shafer theory. Technical Report SAND 2002-0835, Sandia National Laboratories, Albuquerque, NM, April 2002.

[40] SERENE Partners. *The SERENE Method Manual*. SafEty and Risk Evaluation using bayesian NEts: SERENE, Queen Mary, University of London, UK, May 1999. Available at https://www.eecs.qmul.ac.uk/~norman/papers/serene.pdf.

[41] David Spiegelhalter. *The Art of Uncertainty: How to Navigate Chance, Ignorance, Risk and Luck*. Random House, 2024.

[42] Lorenzo Strigini and Andrey Povyakalo. Software fault-freeness and reliability predictions. In SafeComp *2013: Proceedings of the 32nd International Conference on Computer Safety, Reliability, and Security*, Volume 8153 of Springer-Verlag *Lecture Notes in Computer Science*, pages 106–117, Springer-Verlag, Toulouse, France, September 2013.

[43] Albena Tchamova and Jean Dezert. On the behavior of Dempster's rule of combination and the foundations of Dempster-Shafer theory. In *6th IEEE International Conference on Intelligent Systems*, pages 108–113, Sofia, Bulgaria, September 2012.

[44] Stephen Edelston Toulmin. *The Uses of Argument*. Cambridge University Press, 2003. Updated edition (the original is dated 1958).

[45] Wikipedia contributors. *Fréchet Inequalities*. https://en.wikipedia.org/wiki/Fréchet_inequalities.

[46] Shuichiro Yamamoto. Assuring security through attribute GSN. In *5th International Conference on IT Convergence and Security (ICITCS)*, pages 1–5, IEEE, Kuala Lumpur, Malaysia, August 2015.

[47] Jian-Bo Yang and Dong-Ling Xu. On the evidential reasoning algorithm for multiple attribute decision analysis under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 32(3):289–304, 2002.

[48] Fuping Zeng, Manyan Lu, and Deming Zhong. Using DS evidence theory to evaluation of confidence in safety case. *Journal of Theoretical & Applied Information Technology*, 47(1):184–189, 2013.

[49] Xingyu Zhao, Dajian Zhang, Minyan Lu, and Fuping Zeng. A new approach to assessment of confidence in assurance cases. In *Computer Safety, Reliability, and Security* (SafeComp) *Workshops: Sassur, ASCoMS, DESEC4LCCI, ERCIM/EWICS, IWDE*. Volume 7613 of Springer-Verlag *Lecture Notes in Computer Science*, pages 79–91, Springer-Verlag, 2012.