SRI International

CSL Technical Report SRI-CSL-2022-02 • June 9, 2022

Assessing Confidence with Assurance 2.0

Robin Bloomfield (Adelard and City, Univ. of London) and John Rushby (SRI)

As Members of the CLARISSA Team
Honeywell, Adelard, UT Dallas, and SRI

Also issued as a CLARISSA Technical Report under the title
Assessing Confidence in Assurance Cases with CLARISSA

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

Abstract

An assurance case is intended to provide justifiable confidence in the truth of its top claim, which typically concerns safety or security. A natural question is then "how much" confidence does the case provide?

In this report, we explore issues in assessing confidence for assurance cases developed using the rigorous approach we call Assurance 2.0. We argue that confidence cannot be reduced to a single attribute or measurement. Instead, we suggest it should be based on attributes that draw on three different perspectives: positive, negative, and residual doubts.

Positive Perspectives consider the extent to which the evidence and overall argument of the case combine to make a positive statement justifying belief in its claims. We set a high bar for justification, requiring it to be *indefeasible*. The primary positive measure for this is *soundness*, which interprets the argument as a logical proof and delivers a yes/no measurement. The interior steps of an Assurance 2.0 case can be evaluated as logical axioms, but the evidential steps at the leaves derive logical claims epistemically—from observations or measurements about the system and its environment—and must be treated as premises. Confidence in these can be expressed probabilistically and we use *confirmation measures* to ensure that the probabilistic "weight" of evidence crosses some threshold.

In addition, probabilities can be aggregated from evidence through the steps of the argument using probability logics to yield what we call *probabilistic valuations* for the claims (in contrast to soundness, which is a logical valuation). The aggregated probability attached to the top claim can be interpreted as a numerical measure of confidence. We apply probabilistic valuations only to sound cases, and this avoids some of the difficulties that attend probabilistic methods that stand alone. The primary uses for probabilistic valuations are with less critical systems, where we trade assurance effort against confidence, and in assessing residual doubts.

Negative Perspectives record doubts and challenges to the case, typically expressed as *defeaters*, and their exploration and resolution. Assurance developers must guard against confirmation bias and should vigorously explore potential defeaters as they develop the case, and should record them and their resolution to avoid rework and to aid reviewers.

Residual Doubts: the world is uncertain so not all potential defeaters can be resolved. For example, we may design a system to tolerate two faults and have good reasons and evidence to suppose that is sufficient to cover the exposure on any expected mission. But doubts remain: what if more than two faults do arrive? Here we can explore consequences and likelihoods and thereby assess risk (their product). Some of these residual risks may be unacceptable and thereby prompt a review, but others may be considered acceptable or unavoidable. It is crucial however that these judgments are conscious ones and that they are recorded in the assurance case.

This report examines each of these three perspectives in detail and indicates how Clarissa, our prototype toolset for Assurance 2.0, assists in their evaluation.

Contents

List of Figures

1 Introduction

Assurance is the process of collecting evidence about a system and its environment and developing claims and an argument that use these to justify (or reject) deployment of the system; an *assurance case* is a way of organizing and presenting this information, together with other relevant facts, knowledge, models, and theories in a manner that facilitates overall comprehension and assessment. The purpose of such a case is to support the socio-technical process of making, justifying, and communicating the decision to deploy a system or service in a given context. In particular, we propose that evaluation of an assurance case should conclude with an explicit "sentencing statement": that is, a description of the evaluators' understanding of the system and its issues, their assessment of the case, and their decision on deployment of the system.

Assurance cases have evolved and changed over the past 30 or more years (see [95, Chapter 2] for a brief history); we advocate a further step in their development that we call Assurance 2.0 [22] whose slogan, which might seem paradoxical at first, is "simplicity through rigor." In keeping with this, our fundamental requirement is that a case should provide *indefeasible justification* for the decision to deploy the system or service concerned; this means that the justification must be so well supported, and all reasonable doubts and objections must have been so thoroughly considered and countered, that we are confident no credible doubts remain that could change the decision. (See [94] for an introduction to the epistemological notion of indefeasibility in the context of assurance.)

The world is uncertain and our understanding imperfect, so this assessment is very demanding. Thus it is often necessary to examine and evaluate an assurance case from several diverse perspectives that are combined to yield an overall assessment of its indefeasibility. For example, some perspectives will focus on "positive" aspects of the case, such as the evidence and argument in support of its claims, while others will consider the "negative" aspects (i.e., its potential defeasibility), such as doubts and objections that have been considered and refuted, and any that remain as residual risks.

Notice that we may wish to perform some assessments and measures differently during construction of a case than at its end, when the case will be used in the judgment whether to deploy the system. During construction, we may wish to evaluate the part of the case we have constructed and ignore the parts that are missing (or assume they are good), whereas the existence of missing parts would be catastrophic in a final evaluation. Similarly we may tolerate uninvestigated doubts during construction but will want them resolved for final evaluation. Figure 8 illustrates a possible process for assessment of the perspective we call "soundness" at various stages of development and review. Other perspectives will follow similar processes, and these may be interleaved with each other.

This report is about different perspectives on the assessment of an Assurance 2.0 case and the different measures that can support and quantify those perspectives. We also outline capabilities required of tools that can manage and assist these activities; in particular we are part of a project led by Honeywell that is developing prototype tool support for Assurance 2.0 (based on Adelard's ASCE tool [3]) that we call CLARISSA ("Consistent Logical Automated Reasoning for Integrated System Software Assurance"). These capabilities were explored and developed during construction of the CLARISSA/ASCE prototype. As stated earlier, we expect evaluation of an Assurance 2.0 case to conclude with a "sentencing statement" (see Section 7). This is a description by the evaluators' of their understanding and assessment of the case and their decision on deployment of the system. The assessments and measures described here will be used in support of the sentencing statement.

Before we proceed to describe these assessments and measures, we need to outline some ways in which an Assurance 2.0 case may differ from traditional interpretations of assurance cases as described, for example, in the tutorial by Holloway [67].

Assurance 2.0 cases follow an approach that builds on the earlier "Claims, Arguments, Evidence" or CAE [2], where the main component of an assurance case is a *structured argument* represented as a tree of *claims* linked by *argument steps*, and grounded on *evidence*. As we will explain in Section 1.3, argument steps in Assurance 2.0 are restricted to just five basic forms and these are subject to strong conditions that permit a rigorous, logical interpretation of the overall case. We contend that these restrictions simplify construction of an assurance case by reducing the "bewilderment of choice" and clarifying what must be achieved.

Another way in which we simplify construction and evaluation of assurance cases is by limiting the content and therefore the size of their arguments. Traditionally, assurance cases were largely identified with their structured argument, but we maintain that many parts of the case are best presented outside the argument. We concur with the traditional view that the purpose of an assurance case is to collect and integrate evidence and knowledge of the diverse topics that contribute to assurance of a complex system. But much of the knowledge pre-exists and the contribution of the case is to select it, not develop it. Similarly, evidence may be distilled from data or produced by specialized analyses (e.g., formal verification) and applied to specific representations of the system. These should be constructed as theories, models, and "evidential assemblies" that use their own established methods of validation and presentation, and are referenced and integrated by the assurance argument, not developed within it. For example, the argument may have a subcase concerning hazard analysis and will cite evidence discharging side-claims that this was performed by a competent team using accepted practice, and will use the list of hazards found, but it will not itself present the details of hazard analysis.

Anything that can be considered a single topic, with its own methods, knowledge, notations and documentation is a candidate for an external theory, model, or

evidential assembly. These are external to the argument but part of the larger assurance case: the role of the argument is to gather and integrate them. The assurance case argument will include justification that these external elements were selected or developed and applied appropriately, but will not include the internal details or justifications for these elements as part of the argument itself, although it will provide links to them. In justifying and reviewing the overall case, it is important that the assurance argument is consistent with this wider body of knowledge and evidence, and the ability to conduct deep dives into it is part of the evaluative probing of the case. The following sections develop these ideas in a little more detail.

1.1 Models and Theories in Support of an Assurance Case

We noted above that an assurance case will reference models and theories in addition to evidence. It is essential to understand the relationships of these to each other, and how they are used in an assurance case. At the bottom of an assurance case, we have evidence about the system in its operating context (henceforth we will speak of simply the system and take this to include its context): that is say, concrete observations, measurements, and analyses about the system in operation or under test or about its design and construction. This evidence is used to justify logical claims about the system. The claims are relative to various descriptions of the system: for example, if we say "the tests show the system performs correctly" we must have a notion of "correct" that is described somewhere. These descriptions concern the behavior or attributes of the system from some point of view (e.g., timing, power consumption, functional behavior) and at some level of abstraction; we refer to all these as *models*.

The properties that can directly be stated and observed about the system and its low-level models (e.g., percentage of objects correctly identified by a vision system under test) will generally be far removed from the properties about which we seek assurance: those will generally concern emergent properties stated about highly abstract models (e.g., safety of an autonomous car). It follows that a central task of an assurance case is to connect properties of the system and its low level models to those of high level models, and this is accomplished by argument steps that iterate through a series of intermediate models that generally align with steps in the design and development of the system, as in the classical "V" Diagram. A typical step of this kind will seek to justify that a property A of some model P ensures property B of a next higher model Q. In Assurance 2.0 this is precisely the purpose of a *substitution block*, one of the five kinds of step that may appear in an assurance case argument. The justification may be fairly large, intricate, and possibly mathematical: it is what we call a *theory*. The theory may depend on the models P, Q or properties A, B satisfying some constraints, and these will appear in the argument as side-claims on the substitution and must be justified by their own

arguments and evidence. Notice that this evidence may concern the model (e.g., is it "well formed") and not the actual system.

An example theory is that underlying Modified Condition/Decision Coverage (MC/DC) for requirements-based tests [29, 30, 65]: such a theory must explain this method of testing and coverage evaluation, how is it performed, why is it useful, what issues need to be considered, and what claims it can support. The theory could then be used, for example, to explain how MC/DC coverage of executable code can justify a claim that the code contains no unintended functions. When using a theory, the argument must provide justification that it is suitable and credible, and that it is applied appropriately but it does not present the theory as part of the argument: it merely references it.

We do this to control the size of the argument, and to allow compositional reasoning. The purpose of the structured argument in an assurance case is to collect, organize, and present evidence and information relevant to assurance of the system. This is already a formidable task and an assurance case argument is therefore generally large and difficult to comprehend in its entirety. This task should not be further complicated by presenting and justifying technical means of assurance or analysis within the argument. Such analyses are better developed, analyzed, justified, and evaluated separately as self-contained models and theories managed by experts and presented and assessed by the scientific and engineering methods traditional to their fields. These theories may conclude with suggested fragments for the structured argument of an assurance case referencing the theory, but those argument fragments emerge as part of the theory rather than vice versa.

Theories and models are part of the assurance case and must be assessed as such, but they are developed separately and not within the argument. The argument of an assurance case will integrate subcases that use several different theories but, because the theories are referenced rather than developed therein, the argument can be fairly systematic: at each point, we choose a suitable kind of argument step (see the "helping hand" in Figure 2) and the theory or other "warrant" that will justify it; these determine the evidence or subclaims required, and any necessary side-claims.

An example due to Holloway [67], illustrates these points by taking a contrary course (Holloway was using the example for illustration, not advocating its method). The example concerns a teenager Jon who asks his dad if his friend Tim may drive him to the game. The dad would like to see an assurance case to justify the claim that Tim is a safe driver. In Holloway's presentation, four topics are identified (and later elaborated) and provide the main substructure of the argument for the case:

1. Tim has satisfied all legal requirements for driving.
2. Tim has not been in an accident.
3. Tim has a reputation for driving safely.

4. Nothing is going on in Tim's life that might cause him to drive less safely than usual.

These seem reasonable, but doubts remain: are they the most important topics, and are they sufficient? Surely we would like to see some extended discussion of driving safety by young men, together with historical data, statistics, and risk factors. Holloway does provide some of this but it is represented explicitly in the argument of the case and would overwhelm it if included in full detail. Furthermore, we cannot expect the developers and evaluators of an assurance cases to be experts in every topic that might be relevant to a case: for example, this case is about driving safety, whereas another might concern the safety of plans for radiation therapy, while yet another is about mission risk due to pyrotechnic bolts in a spacecraft.

Instead, we advocate that such material is developed as a separate theory; here this would be an extended "theory of driving safety by young men" that is constructed and evaluated by those with specialist knowledge of the topic. The assurance case argument references this theory, and is structured accordingly, but the theory is not developed within the argument.[1]

Typically, the theory will justify a substitution step in which property A of model P ensures property B of model Q; most often, either the properties or the models are the same (i.e., either $A = B$ or $P = Q$). Here P and Q are the same "model" of young men's driving, and A is a collection of observable or measurable properties of young men whose conjunction supports the claimed property B that such men are safe drivers. It might well be that the observable properties identified by the theory are the same four identified by Holloway, but the structure and justification of the assurance case will be different. In Holloway's case, a detailed justification for the four properties must be embedded in the argument, whereas the alternative case justifies them by reference to the theory, supported by side-claims attesting to the credibility of the theory (authors, reviewers, accepted practice, previous applications etc.) and its application to the system under consideration. The theory is independent of the case and is developed and evaluated by experts and may evolve over time, whereas the embedded justification is developed and evaluated as part of the case by persons who presumably must also consider other topics requiring specialized expertise such as the safety of Tim's car, and any hazards of the route to be taken—topics that we would recommend as additional candidates for external theories or evidence.

[1] An alternative point of view is that these topics are the hazards posed by young male drivers and should emerge as part of hazard analysis. Hazard analysis is a form of evidence assembly (see Section 1.2), so in this approach the topics would be delivered to the argument by an evidence incorporation step rather than a substitution step, but the principle remains the same: evidence assembly (and hazard analysis in particular) is an important part of the assurance case, but it is external to the argument, just like a theory.

1.2 Evidence and its Assembly

An assurance case is based on evidence, and evidence can take many forms. For example, DO-178C [92] identifies 71 "objectives," which roughly correspond to evidentially supported claims in an assurance case retrofitted to DO-178C. Some of the evidence supporting these claims we call *monolithic* because it is a single judgment or observation: for example, "software load control is established" [92, Section 7.1.h]. Others, we call *aggregated* because they comprise judgments or observations iterated over some set of items: for example "test coverage of high level requirements is achieved" [92, Section 6.4.4.a], which iterates over the high level requirements to deliver aggregated evidence that coverage is achieved. In fact, monolithic evidence often proves to be aggregated on closer inspection. For example, "software load control is established," mentioned above, is defined as follows [92, Section 7.1.h].

> "Software load control ensures that the Executable Object Code and Parameter Data Item Files, if any, are loaded into the system or equipment with appropriate safeguards."

This suggests that the evidence should reference a theory explaining what load control means and what safeguards are required and how these are achieved, and an aggregated evaluation of its application iterated over each file that supplies "Executable Object Code" or "Parameter Data Items." We therefore take aggregated evidence as the standard case.

Within a structured argument, the purpose of evidence is to support a claim; without this association, we do not have evidence, merely data. In Assurance 2.0, this association is performed by the *evidence incorporation* block of a structured argument, but it must be justified by some process or theory that links the evidence to the claim. Furthermore, there must be some activity that performs the observations or judgments that provide the underlying data and, if necessary, aggregates it into evidence. In addition, evidence must generally be bound together with other information about its generation and provenance. We refer to this entire collection of tasks as *evidence assembly*; currently this is seldom recognized as a coherent activity and its tasks are distributed somewhat arbitrarily between the assurance case argument and its supporting data collection and this can have unfortunate consequences.

Consider, for example, what happens when one or a few items of aggregated evidence are changed (for example, some tests are rerun). Conceptually, evidence assembly updates its aggregated results and makes these available to the argument of the assurance case. We must then consider where is evidence assembly performed? The natural location, based on the preceding description, is that it is external to the tool that manages the assurance case argument. But an alternative design, which seems to be assumed in some discussions of tools for assurance cases, is that it is part

of that tool. Both approaches seem viable, and one could also imagine mixed forms, where some kinds of aggregated evidence are assembled locally, by the argument management tool, and others are assembled externally.

These choices may influence the structure of the assurance case that is developed. If evidence assembly is performed in the argument management tool, then the tool potentially has access to unaggregated or "fine-grained" evidence (e.g., individual test results) and developers of the assurance case may choose to structure some part of the argument at this level of detail. We do not favor this option, as some assessments of aggregated evidence (e.g., "does every requirement have a test?") may move from evidence assembly, where we believe it belongs, into the assurance case argument itself. As we have noted before, assurance case arguments can easily become overwhelmingly large and complex and correspondingly difficult to evaluate; we therefore suggest that it is important to identify components of the case that can be delegated to separate analysis and review: this includes models of the system and its artifacts, theories concerning both design (e.g., fault tolerance) and review (e.g., various kinds of testing), and the aggregation and assessment of evidence.

Observations and judgments may be the central focus of evidence, but they do not stand alone. Several other items that we refer to generically as "provenance" are usually needed as well. For example, test results might be the primary evidence provided by some activity, but we also need information on the test oracle, the test suite, the method of test generation and measurement of coverage, the execution or experimental platform employed, and the versions of the requirements and software employed. It is possible that some of these could be supplied as separate items of evidence, but the advantage of supplying it all in a single package is that it can be reviewed and assessed as part of evidence assembly, and then bound together (e.g., with a cryptographic checksum) so that its integrity and coherence are established and preserved. Evidence assembly must manage all these concerns, which will differ across different kinds of evidence, and for this reason we favor an architecture where evidence assembly is performed by tools that are specific to the kind of evidence concerned and that are external to, but closely allied with, the tool that manages the assurance case argument.

1.3 Structure of an Assurance 2.0 Case

Following the discussion of the previous subsections, we see that an assurance case argument is but part of a "full" assurance case that also includes supporting models and theories, together with evidence assembly and its underlying data. A tool that supports assurance cases does not need to manage the development and assessment of models, theories, nor the data underlying evidence, but it does need to be able to reference these. And it is a design choice whether it manages evidence assembly or delegates this to external tools but, however it is done, evidence assembly is separate

from the assurance case argument, whose management is the prime function of the assurance case tool.

Assurance case arguments are typically presented graphically, as a tree-like structure composed of *nodes* and *links* (see Figure 1). Assurance 2.0 follows this approach and the CLARISSA/ASCE tool provides a graphical user interface for the construction of graphical arguments.[2] There are three basic nodes: *claims*, *arguments*, and *evidence* and these combine to produce *argument steps* in which (sub)claims or evidence nodes link to an argument node that links to a parent claim. The argument node provides justification that the subclaims or evidence logically entail the parent claim; the justification may require additional (side) claims to ensure attributes such as indefeasibility. Other kinds of node may appear in an Assurance 2.0 argument, such as comment, defeater, and subcase, but these serve dialectical rather than logical purposes; see Section 5.

An assurance case argument, as we have specified it, has rather limited scope and can therefore take a rather restricted form: it is not concerned with constructing intricate chains of logical inferences, but just a few basic steps for structuring and organizing references to external models, theories and evidence. Thus, Assurance 2.0 uses only five different kinds of argument steps: these are called (building) *blocks* [21] and they comprise *decomposition*, *substitution*, *concretion*, *calculation*, and *evidence incorporation* and are described in more detail in Sections 2.1 and 2.2 and illustrated in Figure 1.

In the typical structure of an Assurance 2.0 case, general claims at the upper level are refined into more precise claims using concretion steps, then substitution steps are used to elaborate these claims about high level models into claims about low level models and their implementations, and these lowest level claims are discharged by evidence. Application of evidence is generally accomplished in two steps: the lowest step performs evidence incorporation to justify a claim about "something measured" (e.g., "we did requirements-based testing and achieved MC/DC coverage") and this supports a second step (typically a substitution based on application of an external theory) that connects this to a claim about "something useful" (e.g., "we have no unreachable code"); see Sections 2.2 and 3.1. At any stage, the argument may divide into subcases using decomposition or calculation steps that enumerate a claim over some structure (e.g., over components, requirements, hazards, etc.) or that split the conjuncts of a compound claim. This structure may recurse within subcases.

We require that argument steps are *deductive* whenever possible. This means that the claim justified by a step should be logically implied by the conjunction of evidence or subclaims supporting it. This is in contrast to *inductive* argument steps where the evidence and subclaims merely "suggest" their parent claim with various degrees of force. Assurance 2.0 blocks generally have *side-claims* whose purpose

[2] We are considering other representations.

is to enforce deductiveness: for example, a step that decomposes over hazards will have a side-claim that requires justification that all hazards have been identified and that all the individual hazards *and their combinations* are considered in the decomposition.

The reasons for advocating deductivism are that a) every inductive step harbors an anonymous *doubt* (otherwise it would be deductive) and we prefer to eliminate these if possible, or at least to identify them explicitly so that they can be analyzed and recorded, and b) we wish to interpret completed arguments as logical proofs using *Natural Language Deductivism* (NLD) [59]. In NLD, the evidence incorporation steps that constitute the leaves of the argument tree are interpreted as *premises* and the interior steps as *axioms* having the form of definite clauses: that is, conjunctions of subclaims that deductively imply their claim.

NLD is an informal counterpart to deductive proof in formal mathematics and logic but differs in that its premises and axioms are "reasonable or plausible" rather than certain, and hence its conclusions are likewise reasonable or plausible rather than certain [60, Section 4.2]. NLD differs significantly from standard interpretations of informal argumentation, where weaker or different forms of inference may be used [19]; the very term "natural language deductivism" was introduced by Govier [53] as a pejorative to stress that this style of argument does not adequately represent "informal argument." However, our focus is not informal arguments in general, but the structured arguments of assurance cases, where deductive validity is a natural counterpart to the requirement for indefeasibility, and so we depart from the association of assurance cases with informal argument and adopt the label NLD with pride.

Because our treatment is close to that of formal logic, we adopt its terminology and say that an argument is *valid* if its reasoning steps are logically so (i.e., true in all interpretations) and that it is *sound* if, in addition, all its steps are so well justified that they can be accepted as true (i.e., we set a high bar for "reasonable or plausible").[3]

As the logic used in assurance cases is elementary (basically, propositional calculus), validity simply requires that all the argument steps "fit together" correctly. That is, if a step has a subclaim C, then the subargument that justifies this must deliver precisely C, and not some C'. The prototype tools that construct Assurance 2.0 arguments do so in a way that largely guarantees logical validity. For example, at the top of Figure 1, we have a decomposition block delivering a "more precise claim" that is used by a concretion block. It would be a logical validity fault if

[3] We explain later, in Section 2.1, that we prefer (but do not require) completed arguments to be sound and "fully valid" meaning that, in addition to being logically valid, they are deductive and have no unaddressed doubts or defeaters. During development, we allow arguments to be incomplete and lacking full validity; we evaluate these as best we can and tackle their deficiencies in an iterative manner.

the two instances of the claim (i.e., the one delivered and the one used) were different, but this is excluded because there is just one instance of the claim and it is referenced by both blocks. Nonetheless, it is possible that the justification for the lower block is inadequate, so that it really delivers only some C' and not C. Note, however, that this is not logical invalidity but unsoundness, which is discussed next.

A logically valid argument is *sound* when all its evidence incorporation steps cross some threshold for credibility, and all its interior or reasoning steps have indefeasible justifications. As we will explain in Section 2.2, we use *confirmation measures* to assess the "weight" of evidence in support of a claim and thereby judge soundness for evidence incorporation steps. For interior reasoning steps, we rely on human judgment to assess the justification supplied, but we expect most of these steps to be the application of some well-accepted theory, so the judgment builds on a reliable foundation.

Observe that the requirements for soundness (indefeasibility, deductiveness, confirmation measures), although rigorous, are straightforward: there is no doubt what is required in both construction and evaluation. This is what we mean by "simplicity through rigor" and it may be contrasted with the complex criteria proposed for less constrained forms of assurance case [31].

Soundness is the most fundamental property we desire of an assurance case, but there are other useful properties and in the next subsection we identify different perspectives and measures that can be used to develop confidence in a case.

1.4 Confidence in an Assurance Case

As we stated in the beginning, the purpose of an assurance case is to assemble and deliver indefeasible justification for its top claim. After assessing the case, we will have some degree of belief that this purpose has been achieved and we call this degree of belief the *confidence* in the case. Confidence is a human judgment that should be based on evaluable and measurable attributes of the case but we do not think it can be reduced to a single attribute or measurement. Instead, we think that confidence should be based on three related criteria.

1. Does the assurance case argument justify validity of its claims, and the top claim in particular?

2. Are there gaps, discrepancies, or weaknesses in the claims, evidence, argument steps or justifications?

3. Are the identified gaps and weaknesses within a tolerable threshold of risk?

We refer to these criteria as positive and negative perspectives on the assurance argument and its residual risks, respectively, and now describe them in more detail.

Positive Perspectives: These consider the extent to which the evidence and overall argument make a positive case to justify belief in its claims. We have already encountered one positive measure: namely, *soundness*, which interprets the argument as a logical proof by NLD and delivers a yes/no measurement.

In assessing whether the evidence provided in an evidence incorporation step justifies its claim with sufficient "weight" to be accepted as a premise in the NLD interpretation, we make use of the measures of *confirmation theory*, which are based on more basic assessments expressed as (possibly qualitative) subjective probabilities indicating, for example, our confidence in the claim, given the evidence. These valuations are discussed in Section 2.2.

Probabilities can be aggregated from evidence through the steps of the argument using various kinds of probability logic to yield what we call *probabilistic valuations* for the claims (in contrast to soundness, which is a *logical valuation*), including the top claim. Advocates of several alternative interpretations of assurance cases treat their preferred probabilistic valuations as *the* assessment of confidence in a case; Graydon and Holloway show these treatments are problematic [55, 56]. We apply probabilistic valuations only to cases that have already been judged sound and, furthermore, our assurance cases are limited to steps comprised of just the five building blocks; together, these restrictions eliminate many of the sources of difficulty exposed by Graydon and Holloway.

An alternative to probabilistic valuations constructed by external examination of the case is to include probabilistic measures among the claims. This is particularly appropriate for claims concerning properties such as reliability and for those based on probabilistic evidence such as statistically valid random testing [34, 88]. The claims will then be justified within the argument by reference to some accepted theory.

Negative Perspectives: These record doubts about the case, and their exploration and resolution. Reviewers will naturally have doubts and questions about an assurance case and it is sensible to anticipate these and respond to them within the case. In addition, the developers of the case need to guard against *confirmation bias* [61, 79]. To ensure that the positive perspective does not become an optimistic one, they should actively explore doubts and challenges as they develop the case—and should record these, both to avoid rework, and to aid reviewers.

We refer to any concern about a case as a *doubt*; as we explore the origin and nature of a doubt we will refine it to a *defeater*, that is, a specific (counter)claim or challenge that can be attached to a particular point in the argument. Investigation of a defeater may lead to correction or improvement of the assurance case, in which case the modifications become part of the positive case and the defeater that motivated them may be mentioned only in the text of some of its justifications. Alternatively, the defeater may, on investigation, be considered a "false alarm"; the investigation

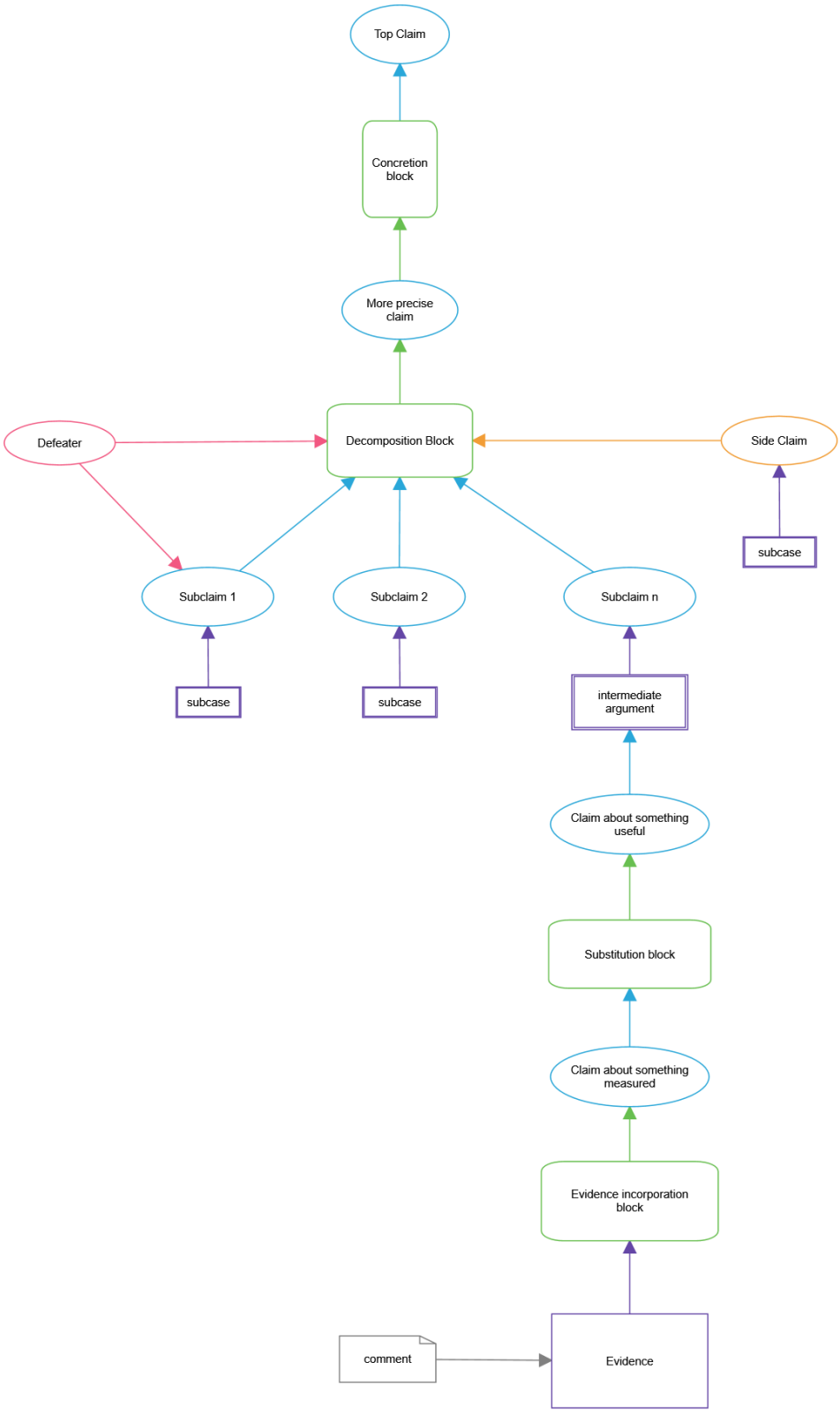

Figure 1: Example Assurance 2.0 Argument

that reveals this can be represented as a (sub)case in its own right, and may be recorded within the main case. CLARISSA/ASCE has functions for recording defeaters and for developing and recording the subcases that lead to their resolution (we refer to this as the defeaters' own defeat), and these can be revealed or hidden as desired.

The number of resolved defeaters might be considered a measure of the effectiveness or diligence with which the negative perspective has been explored [107]; this can be subject to gaming (e.g., using trivially different defeaters to raise the number of resolutions), but can be effective when undertaken with care as "eliminative argumentation" [36, 51].

A defeater to an assurance case is rather like a hazard to a system: that is a conjecture why things might go wrong. Hazard analysis is not judged by how many hazards are found, but by the historical record and rational justification of the method used, and the diligence of its performance. We think similar evaluations should be applied to the negative perspectives of an assurance case and so we advocate use of systematic (and potentially automated) methods for discovery of defeaters. These include methods based on Answer Set Programming [47] and application of "knowledge bases" concerning historical flaws in both systems [37] and assurance cases [57] (although the strict requirements of Assurance 2.0 cases are intended to exclude most fallacies of the latter kind "by construction").

Residual Doubts: The world is uncertain so not all doubts can be resolved. For example, we may design a system to tolerate any two faults and have good reasons and evidence to suppose that is sufficient to cover the exposure on any expected mission. But doubts remain: what if more than two faults do arrive? Or, for a security perspective, what if advances elsewhere render our cryptographic key length insufficient? Here we can explore consequences and likelihoods and thereby assess *risk* (their product). Some of these *residual risks* may be unacceptable and thereby prompt a review, but others may be considered acceptable or unavoidable. It is crucial however that these judgments are conscious ones and that they are recorded in the assurance case.

We now develop valuations and measurements for each of the perspectives introduced above.

2 Positive Perspectives: Logical Valuation (Soundness)

Recall that our fundamental requirement is that an assurance case should provide *indefeasible justification* for the decision to deploy the system or service concerned; this means that the justification must be so well supported, and all plausible doubts and objections must be so thoroughly considered and countered, that we are confident there are no credible doubts remaining that could change the decision.

Confidence in the investigation and resolution of doubts and objections is discussed in Section 5, and confidence that any remaining as residual doubts pose insignificant or manageable risks is discussed in Section 6. During development, we will probably attempt to identify and resolve doubts before proceeding to assessment from a positive perspective. Nonetheless, we present the positive perspective first, because it aligns with the most basic goal of an assurance case argument.

So, in this section and the next we are concerned with positive aspects of the case: namely that its evidence and argument justify confidence in its claims. In the next section, we consider these positive aspects from a probabilistic point of view, while in this section we consider a logical point of view, which we call *soundness*.

Soundness was introduced in Section 1.3 and is discussed in the paper on Assurance 2.0 [22] and its more theoretical precursors [93, 94] that together provide the basis for Assurance 2.0. Briefly, soundness interprets the argument of an assurance case as an informal proof based on *Natural Language Deductivism* (NLD). This means that the leaf steps of an assurance case argument are interpreted as *premises* in which evidence establishes a claim, and the interior steps are interpreted as *axioms* in which a conjunction of (sub)claims implies a parent claim (see Figure 1 for an example). The full argument is then a tree of claims (possibly with cross links if a (sub)claim is used in support of multiple claims) whose top claim is the conclusion of the assurance case.

We need to be sure that evidence really does establish its claim, and that subclaims really do imply their parent, so each argument step is supplied with a *justification* why this is so. The justification will be in natural language but it may reference some external theory, calculation, proof, or mechanized analysis etc.

The benchmark for an interior step is that the conjunction of its subclaims deductively implies or entails its parent claim (so that each interior step is what logicians call a "definite clause"). Doubts about the deductivism of a step—that is, a suspicion there is "something missing" among the subclaims, so that their conjunction merely "suggests" rather than entails the parent claim—causes a step to be considered "inductive" rather than deductive.[4]

We now consider how confidence in deductivism can be established.

[4]There is a stronger form of doubt, where the argument step is considered wrong rather than merely incomplete. This case is discussed in Section 6.

2.1 Confidence in Deductiveness (Interior Steps)

Free-form arguments often challenge their developers with the "bewilderment of choice"; in contrast, Assurance 2.0 arguments are built from a limited repertoire of just five different argument blocks, of which four apply to interior steps. Selection of an appropriate block is assisted by the "helping hand" pictorial memory aid shown in Figure 2.

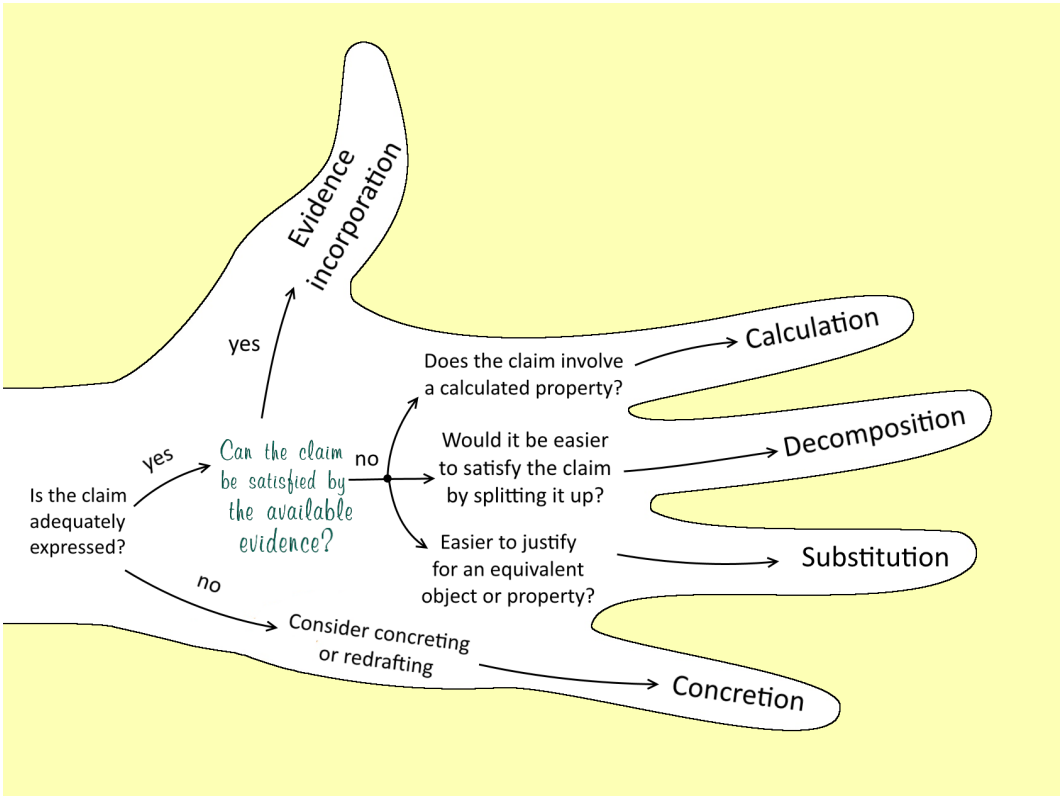

Figure 2: The Helping Hand Memory Aid

As illustrated in Figure 3, Assurance 2.0 blocks generally have a *side-claim*, which is a (conceptually distinct) subclaim whose purpose is to ensure that the other subclaims are well formed and really do deductively entail the parent claim. The relationship is then

$$\text{side-claim} \supset (\text{conjunction of other subclaims} \supset \text{parent claim}).^{5}$$

[5] We use $\supset$ for material implication, $\wedge$ for conjunction, $\vee$ for disjunction, and $\neg$ for negation.

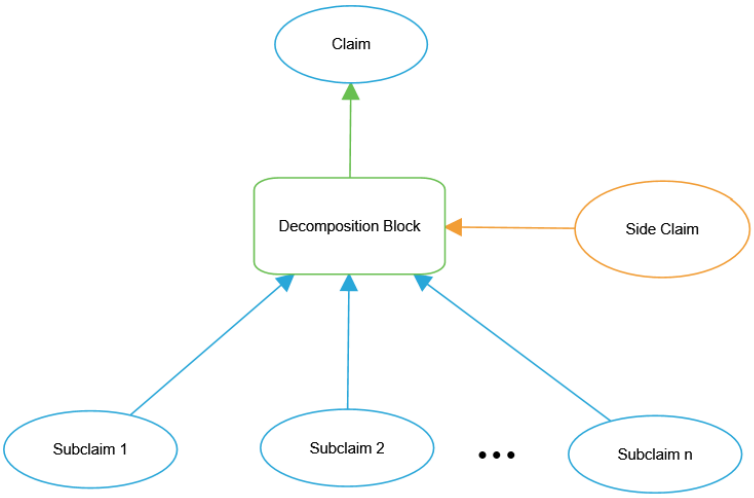

Figure 3: Generic Decomposition Building Block

This is logically equivalent to

$$(\text{side-claim} \wedge \text{conjunction of other subclaims}) \supset \text{parent claim}$$

and so we see that although the side-claim has a conceptually distinct status, it is logically no different from the other subclaims.

Every branch in a complete assurance case argument must terminate in evidence (or, exceptionally, a calculation). Assurance case arguments or subarguments that have unsupported claims as in Figure 3 are incomplete and unfinished. Some assurance case notations have a symbol that explicitly indicates unfinished parts of the case; Assurance 2.0 can use subcase nodes for this purpose, as illustrated in Figure 1, but does not require it, regarding unsupported claims as sufficient indication of an unfinished branch.[6]

The argument step of Figure 3 illustrates a decomposition block, whose purpose is to divide a claim into subclaims over some explicit enumeration, such as components of the system, or time (e.g., past, present, future), or hazards, and so on. In each case, the side-claim must establish that the decomposition is complete and satisfies any other properties that may be needed, such as that the claim distributes over components, or that some theory justifying the decomposition is properly applied. For example, if the decomposition is over hazards, then the side-claim will require that all hazards have been identified and that the decomposition considers them all, both individually and in combination; such a side-claim might

[6]Unsupported claims can also be interpreted as *assumptions*; in this case it would be useful to distinguish those unsupported claims that are assumptions from those that indicate incompleteness and we may revise these aspects as we gain experience with the CLARISSA prototype.

be discharged by evidence that attests to use of a well-accepted method of hazard analysis, performed diligently.

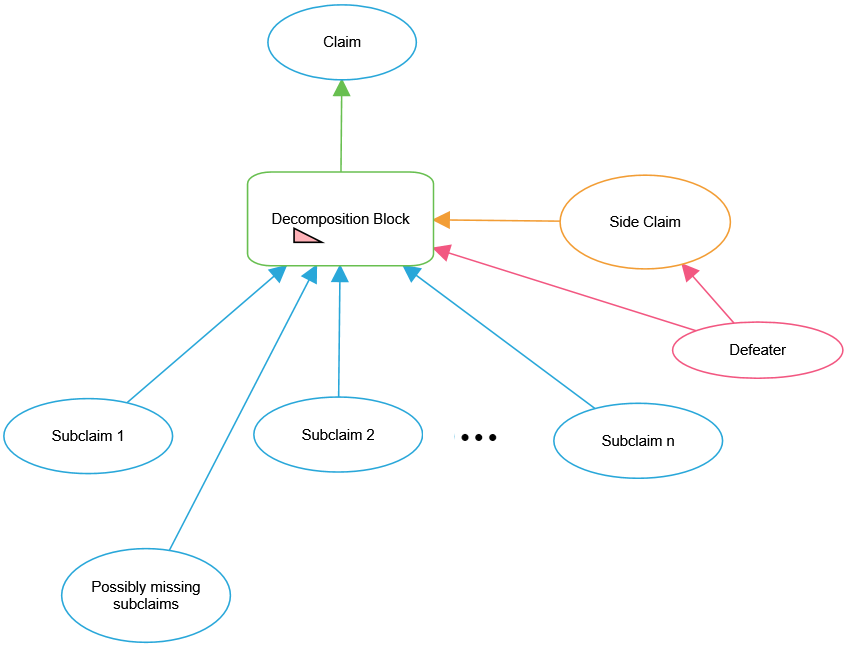

Figure 4: Three Ways to Indicate Doubt in an Argument Step

Now it may be that we have doubts about the decomposition (and therefore also about justification for its side-claim) and wish to indicate this in the developing assurance case so that we can return to it later. One approach would be to annotate the argument step as inductive. This can be indicated informally by adding a marker to the argument step as illustrated by the triangle in Figure 4; CLARISSA/ASCE allows this, but does not provide an interpretation for it at present. An alternative is to use an explicit *defeater* node; the defeater can point to the argument or to a specific subclaim to localize the perceived source of doubt, or to both as illustrated on the right of Figure 4. A third alternative is to add an unsupported subclaim (equivalently, an uninterpreted assumption) acknowledging something is "possibly missing" as shown at the bottom of Figure 4. The three options illustrated in this figure are alternatives: we should select one.

Identified defeaters must be investigated and resolved before a case can be considered complete, but it is not required to perform this resolution before considering other parts and other aspects of the case. If the doubt motivating a defeater is found to be unwarranted, then it is said to be a "defeated defeater" and the justification and possible subargument for this assessment will be recorded as part of the case and selectively revealed or hidden as will be described in Section 5. If, on the

other hand, the defeater identifies a legitimate doubt, then the missing or incorrect claim(s) must be discovered and the argument step (and the subarguments that justify it) suitably amended.

Missing and suspected missing subclaims are easily added to decomposition steps as shown by the "possibly missing" subclaim at the bottom of Figure 4, which will either be removed or be replaced by the truly missing subclaim(s) as the doubt is resolved. This is not so straightforward when the doubt concerns other kinds of argument steps because these take only a single subclaim. An example is illustrated in Figure 5 where a "not deductive" defeater is aimed at a generic subcase. Suppose we determine that the defeater is legitimate and that an additional subclaim is needed. There are two approaches: we can add this subclaim below the original argument, as shown on the left of Figure 6, or above it, as shown on the right of that figure.

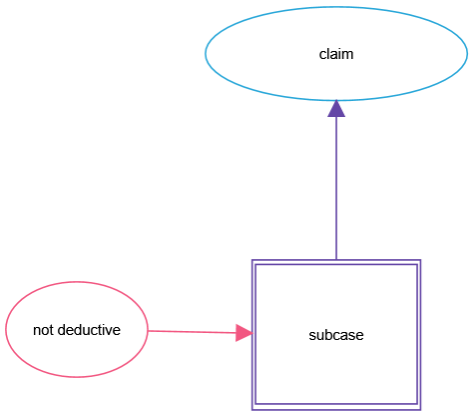

Figure 5: Defeater to a Generic Subcase

Notice that when the additional claim is added above the existing argument, we must recognize that only a weakened form of the original claim is supported by the existing subcase. Apart from this adjustment, however, the original case remains intact (i.e., we do not need to introduce a weakened claim) and this may be considered an advantage for the "added above" choice.

2.1.1 Embedded Links

The argument of an assurance case is required to satisfy certain structural properties. For example, claims cannot link directly to claims (there must be an intermediate argument step), and the pattern of links must be non circular. These requirements are easily checked and are also visibly obvious when all links are explicit, as we have assumed until now. However, the nodes of an Assurance 2.0 argument must each be supplied with narrative descriptions of their interpretation or justification, and in Clarissa/asce these may contain embedded links to other nodes.

These links serve a different, narrative, purpose than conventional links, whose purpose is logical. Consequently, they do not participate in logical (i.e., soundness) or probabilistic valuations of the argument. For example, in an assurance case for an autonomous drone, we may have a primary case that argues it is functionally safe in the style of "Overarching Properties" (OPs) [42, 68], and a secondary case that

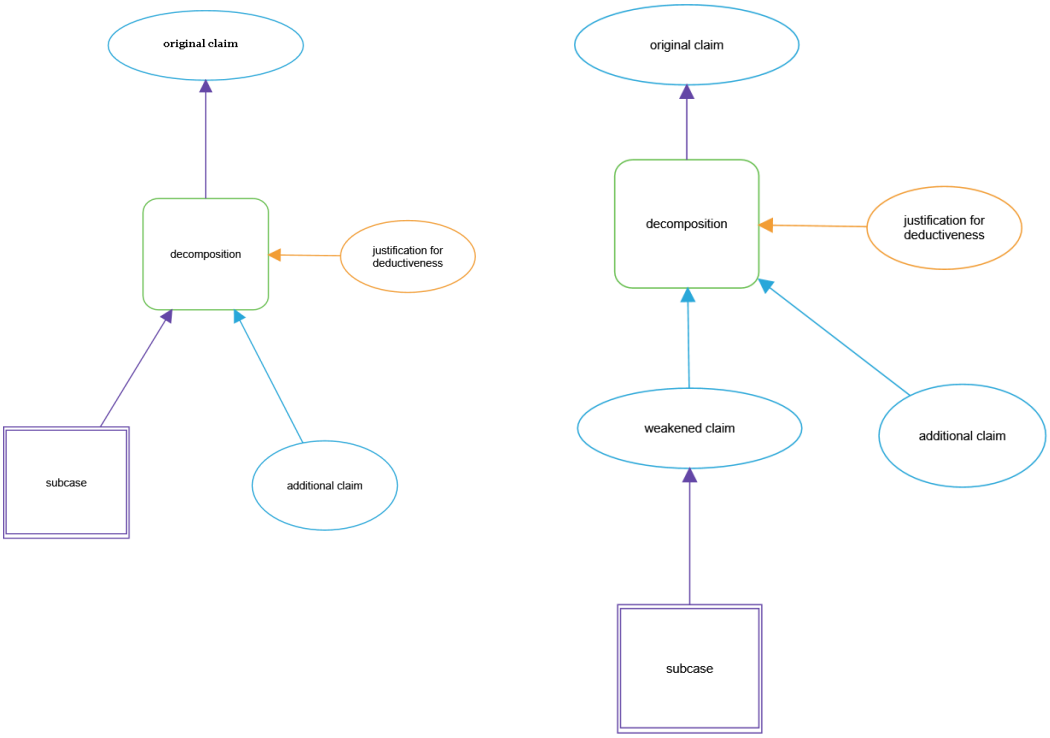

Figure 6: Additional Claim Added Below, or Above Original Argument

argues it complies with a relevant standard called F3269-17 [5]. The narrative for the node making the latter claim may contain a table of all the objectives required by the standard and, for each of them, an embedded link to an existing node in the case where that objective has been satisfied as part of the OP subcase. CLARISSA/ASCE is able selectively to display such links or not; they are shown displayed (in gray) in Figure 7. Note that the arrows are reversed compared with conventional logical links as an indication of their different purpose.

In this example, the secondary case serves as a source of potential defeaters. In particular, those objectives from the F3269-17 standard that are not satisfied by the OP subcase can be added to the argument as defeaters as shown on the right of Figure 7. Notice that here we use defeater nodes for the missing objectives, as opposed to the ordinary claim node used in Figure 4. As we will see in Section 5, resolution of a legitimate defeater begins by replacing it with (or interpreting it as) a claim, so Figure 4 can be considered to have skipped a step compared to Figure 7. If (or when) all the F3269-17 objectives are satisfied, the secondary case can be used to increase confidence in the primary case.

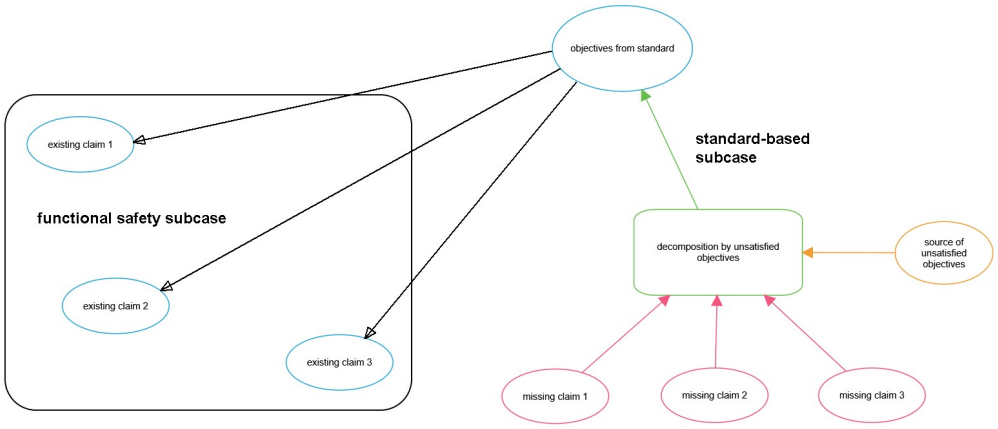

Figure 7: Embedded Links

Another use for embedded links is in referencing the application of a theory and its template to an argument subcase, as described in the following subsection.

2.1.2 Theories and Templates

We introduced the notion of *theories* early in this report, in Section 1.1; the idea is that a theory provides justification for an argument step or series of steps, or for a complete subcase. Considering the general case, each of the argument steps in a theory-supported subcase will cite some aspect of the theory as its justification, and the side-claims (and possibly their support) will likewise reference aspects of the theory, as will the evidence supplied.

The linkage just described between theory and subcase is informal and each theory-supported subcase is constructed afresh. We might be able to provide more automated assistance if the theory explicitly provides a generic subcase that can be instantiated to yield the specific subcase required in the context concerned. Such generic subcases would have much in common with what some other tools for assurance cases call "templates," and we will also use this term. However, unlike other methodologies and tools, we see templates as representations of the syntax for a theory, rather than independent entities, and we regard the theory as the primary item of interest, and the focus for assurance, because it provides semantics and is the source of justification.

When theories provide templates, we can envisage tool support that uses templates a) to check manually constructed theory-supported subcases, b) to construct those subcases automatically under human direction, and c) to synthesize automatically parts of an assurance case by searching for applicable theories and templates.

At present, Clarissa/asce provides the first of these and will soon provide the second; the third is a topic of ongoing research. To indicate the place where a theory/template should be applied, we use an embedded link (recall the previous subsection) to connect the theory/template concerned to the claim that it will justify. The terms appearing in that claim are used to instantiate the generic template, which is then used to check or construct the subcase supporting the claim concerned. By instantiate, we mean that the generic template will contain placeholders that behave like formal parameters in a programming language: for example, a theory to "establish correctness of CLEAR requirements using CLEAR tools" may have a generic top claim "requirements for x are correct," where X indicates a parameter. When this is applied to a claim "requirements for ArduCopter AFS are correct," the parameter will be instantiated as "ArduCopter AFS," with similar substitutions elsewhere in the template.[7]

When a theory/template is applied, we will eventually need to discharge the instantiations of its side-claims. If we are unable to do so, this may indicate that the selected theory/template was an inappropriate choice. We would like to learn this early, before effort has been wasted, so some side-claims in the template may be marked as *preconditions*; these must be discharged before the theory/template may be applied. For example, it would make no sense to apply the CLEAR theory for correctness assurance to requirements that are not written in CLEAR, so "requirements for X are written in CLEAR" is a precondition for this example.

2.2 Confidence in Evidential Support for Claims (Leaf Steps)

The argument of an assurance case is logically valid when all its leaf claims are supported by evidence and the claims supporting and delivered by interior steps "match up" to provide a coherent graph structure. We say the argument is *fully valid* when, in addition, all its steps are (judged to be) deductive, and there are no unresolved defeaters other than those marked as residual doubts. It is not important in what order validity, deductiveness, and absence of defeaters are tackled and assessed for full validity. A fully valid argument is sound when human evaluators attest that the residual doubts are negligible (see Section 6), that the justification for each interior step is satisfactory, and that the weight of each evidential step is sufficient to justify its claim, as we now describe.

In the logical valuation of an assurance case by NLD, we interpret evidence incorporation steps as premises. For soundness we therefore require strong justification that the evidence supplied to the step really does support its claim. Unlike other assurance steps, which inhabit the world of logic, evidence incorporation steps are

[7] CLEAR is a tool-supported requirements notation from Honeywell [13] and AFS stands for "Advanced Failsafe Monitor," which is a safety monitor for an autonomous autonomous quadcopter ("ArduCopter") being developed as part of the ARCOS program by the DesCert team [14, 98].

bridges to that world from the world of observation and measurement. Therefore we cannot assess evidence incorporation steps by the methods of logic, we need the methods of epistemology. Epistemology is about justified belief (as an approximation to truth) and it is natural to express the strength of our confidence in a belief as a number. We will expect those numbers to obey certain rules (the Kolmogorov Axioms) and so they function as (subjective) probabilities [70].

A natural measure of confidence in a claim C given the evidence E is the subjective posterior probability $P(C \mid E)$, which may be assessed numerically or qualitatively (e.g., "low," "medium," or "high"). However, confidence in the claim is not the same as confidence that it is justified by the evidence. It is possible that the reason for a high valuation of $P(C \mid E)$ is that our prior estimate $P(C)$ was already high, and the evidence E did not contribute much. So it seems that to measure justification we ought to consider the difference from the prior $P(C)$ to the posterior $P(C \mid E)$ as an indication of the "weight" of the evidence E. Difference can be measured as a ratio, or as arithmetic difference, thereby producing the following two *confirmation measures*, due to Keynes in 1921 and Eells in 1982, respectively.[8]

$$\begin{aligned} \text{Keynes}(C, E) &= \log \frac{P(C \mid E)}{P(C)}, \\ \text{Eells}(C, E) &= P(C \mid E) - P(C). \end{aligned}$$

There are many other confirmation measures proposed in the literature [102]. For example, some prefer to use the likelihood $P(E \mid C)$ rather than the posterior $P(C \mid E)$, because it is generally easier to estimate the probability of concrete observations (i.e., evidence), given a claim about the world, than vice-versa, thereby giving us the likelihood variants of Keynes' and Eells' measures:[9]

$$\begin{aligned} \text{L-Keynes}(C, E) &= \log \frac{P(E \mid C)}{P(E)}, \\ \text{L-Eells}(C, E) &= P(E \mid C) - P(E). \end{aligned}$$

We can see that these will tend toward their maxima when $P(E)$ is small, meaning that E should be unlikely in general. This suggests that we should favor evidence whose occurrence (in the absence of C) would be a *surprise*.

Similarly, if we have accepted evidence E_1 and seek additional evidence, we should look for E_2 that is (or remains) surprising in the presence of E_1. Thus, for

[8] The logarithm (which may use any positive base) in Keynes' and other ratio measures serves a normalizing purpose so that, as with arithmetic difference, positive and negative confirmations correspond to numerically positive and negative measures, respectively, and irrelevance corresponds to a numerical measure of zero.

[9] Notice that $\text{L-Keynes}(C, E) = \text{Keynes}(C, E)$ and $\text{L-Eells}(C, E) = \text{Eels}(C, E) \times \frac{P(E)}{P(C)}$, since $P(C \mid E) \times P(E) = P(C \wedge E) = P(E \wedge C) = P(E \mid C) \times P(C)$.

example, if E_1 is evidence of successful tests, it will not be surprising if additional tests are successful; instead we should seek evidence E_2 that is "diverse" from $E_!$, such as static analysis. More formally, we have, by the chain rule

$$\begin{aligned} P(C \wedge (E_2 \wedge E_1)) &= P(C \,|\, E_2 \wedge E_1) \times P(E_2 \,|\, E_1) \times P(E_1), \text{ and} \\ P(E_2 \wedge (C \wedge E_1)) &= P(E_2 \,|\, C \wedge E_1) \times P(C \,|\, E_1) \times P(E_1). \end{aligned}$$

The left (and hence right) hand sides are equal, and so

$$\frac{P(C \,|\, E_2 \wedge E_1)}{P(C \,|\, E_1)} = \frac{P(E_2 \,|\, C \wedge E_1)}{P(E_2 \,|\, E_1)}. \tag{1}$$

Thus, E_2 delivers the largest "boost" to Keynes' measure for the justification provided by E_1 (i.e., the left hand side of (1)) when E_2 would be surprising given only E_1, but not when given C as well, which confirms that E_2 should be *diverse* from E_1. These observations about "surprising" and "diverse" evidence are intuitively plausible, but it is satisfying to see them put on a rigorous footing.

An additional consideration when evaluating evidence is that it is not enough for the evidence to support a given claim C, it should also discriminate between that claim and others, and the negation, or "counterclaim" $\neg C$ in particular. Again, discrimination or distance can be measured as a ratio or as arithmetic difference, leading to the following two measures; the first is due to Good (and Turing) from codebreaking activities during World War 2:

$$\text{Good}(C, E) = \log \frac{P(E \,|\, C)}{P(E \,|\, \neg C)},$$

and the second is due to Kemeny and Oppenheim in 1952:

$$\text{KO}(C, E) = \frac{P(E \,|\, C) - P(E \,|\, \neg C)}{P(E \,|\, C) + P(E \,|\, \neg C)}.$$

We will refer to these as "Type 2" confirmation measures, and the previous examples as "Type 1." Likelihoods are related to posteriors by Bayes' Rule, and appearances of $P(\neg x)$ in Type 2 measures can be replaced by $1 - P(x)$, so

$$\begin{aligned} \text{Good}(C, E) &= \log \frac{P(C \,|\, E) \times P(E)/P(C)}{P(\neg C \,|\, E) \times P(E)/P(\neg C)} \\ &= \log \frac{P(C \,|\, E) \times P(\neg C)}{P(\neg C \,|\, E) \times P(C)} \\ &= \log \frac{P(C \,|\, E) \times (1 - P(C))}{(1 - P(C \,|\, E)) \times P(C)} \\ &= \log \frac{O(C \,|\, E)}{O(C)} \end{aligned}$$

where O denotes *odds* (i.e., $O(x) = P(x)/(1-P(x))$) and Good's measure is therefore sometimes referred to as the "log odds" or "log odds-ratio" measure for weight of evidence [48].

Similar manipulations can be performed on other Type 2 measures, so that appearances of $\neg C$ revert to just C and the distinction between Type 2 and Type 1 measures disappears. Furthermore, notice that appearances of $P(E)$ cancel out in the second line of the derivation above; manipulations of other measures exhibit the same behavior and we find that they then generally satisfy the following conditions:

1. They can be expressed as functions of $P(C \mid E)$ and $P(C)$ only,
2. They are increasing functions of $P(C \mid E)$, and
3. They are decreasing functions of $P(C)$.

Not all confirmation measures satisfy 1 above. For example, the following measure due to Carnap in 1962

$$\text{Carnap}(C, E) = P(C \wedge E) - P(C) \times P(E)$$

depends nontrivially on $P(E)$.[10] However, such measures can be manipulated by irrelevant evidence [6, section 2], so we prefer measures that do satisfy condition 1.

All confirmation measures indicate the extent to which evidence justifies a claim, but they are not ordinally equivalent. That is to say, a given confirmation measure may rank one scenario (i.e., combination of $P(C \mid E)$ and $P(C)$) higher than another, but a different measure may do the reverse. This is acceptable because, although all confirmation measures evaluate the degree to which evidence E justifies claim C, they do so in different ways and we may prefer one measure to the other (or prefer different measures for different purposes) [62].

Nonetheless, it is possible to add a fourth condition to the list above and all measures satisfying this enhanced set of conditions are ordinally equivalent. This condition is the following [6, 99].

4. Let C_1 and C_2 be claims, unconditionally independent and also conditionally independent on evidence E. If both C_1 and C_2 have measures of confirmation greater (resp. less) than t then their conjunction must also have measure greater (resp. less) than t.

Measures that satisfy all four conditions are called *justification measures* [99]. An example is Shogenji's measure

$$\text{Shogenji}(C, E) = 1 - \frac{\log P(C \mid E)}{\log P(C)}.$$

[10] Notice that $\text{Carnap}(C, E) = \text{L-Eells}(C, E) \times P(C)$, so the L-Eells measure does not satisfy condition 1 either.

Shogenji argues that only justification measures serve to increase true beliefs (e.g., claims) while reducing false ones [99], asserting that simple confidence measures (e.g., $P(C \mid E)$) fail to do this and that traditional confirmation measures may rank beliefs inconsistently. In assurance, however, we are usually seeking *strong* indications that evidence justifies a claim and the measures are likely to concur on this, so we are generally content to use confirmation rather than justification measures.

CLARISSA/ASCE can attach confirmation measures to evidence incorporation steps and it allows these steps to be marked as "accepted" when the weight of evidence (e.g., as indicated by an attached confirmation measure) is judged to exceed some threshold. Since the discussion above concludes that all confirmation measures will deliver similar conclusions, it may seem that we could have selected one and bypassed the discussion. However, although the conclusions may be similar, they are based on elicitation of different judgments and we believe there can be value in asking assessors to consider the different points of view underlying these judgments. For example, Keynes(C, E) elicits judgments $P(C \mid E)$ and $P(C)$, while L-Keynes(C, E) elicits $P(E \mid C)$ and $P(E)$. Furthermore, the two measures should yield the same value; we can therefore provide feedback to assessors if their judgments are inconsistent. Similarly, the original formulation of Good(C, E) elicits judgment of $P(E \mid \neg C)$, which requires consideration of a contrary point of view.

Some will be skeptical that human developers and evaluators are able to assess and manipulate probabilistic measures correctly, even qualitatively, and will also contend that confirmation measures are beyond everyday experience. They may point to alleged flaws in human evaluation of probabilities. Here is a standard illustration [103].

Evidence E: Linda is 31 years old, single, outspoken and very bright. She majored in philosophy. As a student, she was deeply concerned with issues of discrimination and social justice, and also participated in anti-nuclear demonstrations.

The challenge is to assess which of the following two claims is best supported by the evidence.

Claim C_1: Linda is a bank teller,

Claim C_2: Linda is a bank teller and active in the feminist movement.

When human subjects are exposed to this and similar examples, they overwhelmingly favor C_2. Psychologists label this the "conjunction fallacy" because C_2 is the conjunction of C_1 with another clause and a conjunction must always be *less* probable than either of its components; they then cite it as evidence for the assertion that intuitive human reasoning is poor at probabilities [74, 103]. However, a more

recent interpretation is that humans evolved to weigh evidence and actually base their judgments on mental measures more akin to confirmation than simple probabilities [33, 72, 99].[11] To see this, we observe that the evidence E seems to add nothing to our prior belief in C_1 but it does seem to support the second clause of claim C_2 (i.e., "is active in the feminist movement") and so by item 2 of the list of properties for confirmation measures, we conclude that the evidence indeed confirms C_2 more than C_1, thereby refuting the "fallacy" charge.

Claim C_2 entails the further claim that "Linda works outside the home" (since she is a bank teller), but the evidence provides no direct support for this and it could easily be false. Thus, we have evidence that soundly supports a claim that logically entails a further claim, yet that second claim could be false. For a more extreme example, the evidence that a card drawn from a deck is an Ace supports the claim that the card is the Ace of Hearts, and this entails the further claim that the card is red. But the card used in evidence could have been the Ace of Clubs, which refutes the derived claim that it is red. A pragmatic resolution to this apparent paradox is that the standard for assurance should be more demanding than basic confirmation: if an evidentially supported claim is a conjunction, then we need indefeasible support for all elements of the conjunction, and so, in the context of assurance, we should not accept that the evidence about Linda is sufficient to establish claim C_2.

Another approach would be to propagate probabilities and likelihoods from evidence through the directly supported claims to these "second-level" claims and to evaluate confirmation there. In the Linda and Ace of Hearts examples, we see that the evidence provides no support for the second-level claims (i.e., "works outside the home" and "card is red"). These "two level" steps for exploiting evidence are common in assurance cases: the bottom step, using an evidence incorporation block, connects the evidence to a claim about "something measured" (e.g., "we did requirements-based testing and achieved MC/DC coverage") while the second step (typically a substitution block based on application of an external theory) connects it to a claim about "something useful" (e.g., "we have no unreachable code"). We recommend that confirmation measures are evaluated against the "something useful" claims.

Sometimes a mismatch between the claims about "something measured" and "something useful" leads to the realization that one or the other is misstated. Here is an example: during World War 2, the US Army Air Force came to its Statistical Research Group in New York seeking advice on where best to add armor to improve the survival of their airplanes. Many damaged planes returning from engagements had been examined and this produced the following evidence.

[11] There are larger claims, widely accepted in areas of psychology and cognitive science, that much human perception and unconscious decision making are based on processes akin to Bayesian analysis over models [32, 44, 45, 58, 76, 91].

Section of plane	Bullet holes per sq. ft.
Engine	1.11
Fuselage	1.73
Fuel system	1.55
Rest of plane	1.8

The fuselage seems the most heavily damaged of the identified components, so the evidence seems to support the claim "the place where armor will best improve survival of the plane is the fuselage." This is actually a second-level ("something useful") claim; we should begin by using the evidence to justify a first-level ("something measured") claim. A plausible candidate for this is "the fuselage is the part of the plane with heaviest damage." An important difference in these two claims is that the first level speaks of "damage" while the second level speaks of "survival." Thus we need either some inference from damage to survival, or the first level claim should also speak of survival. This leads to a key insight: the evidence comes exclusively from planes that survived. Hence the first level claim should be changed to "the fuselage is the part of the plane that can survive heaviest damage." From there it is a short step to deduce that planes with heavy damage to the engines did not survive and hence the celebrated recommendation by Abraham Wald that the best place to apply armor is where the bullet holes are not [40].

Another area where human reasoning about probabilities is claimed to be notoriously unreliable concerns the "base-rate fallacy" [11]. The standard examples involve diagnostic tests for disease. Suppose we have a test that is perfectly accurate at diagnosing the disease when it is present (i.e., no false negatives), but also has 10% false positives. A person tests positive in a population where 1% has the disease. Human subjects are asked which of the following probabilities is closest to the true probability the person has the disease: a) 90%, b) 10%, c) 50%, d) 89%.

The correct answer is b) but human subjects overwhelmingly choose one of the other answers, with an average of 85% [10, page 44]. The psychologists' explanation is that humans overlook the very low base rate of the disease, which means that false positives (10%) overwhelm true positives (1%). (Others would say it is because they do not know or do not apply Bayes' Rule.) An alternative explanation again involves confirmation: the evidence (positive test) increases the probability of the claim (having the disease), so by item 2 of the list of properties for confirmation measures, we obtain positive confirmation; human subjects opt for a large number as a way of expressing this, despite being asked about probabilities, not confirmation.[12] Although intuitive human reasoning is again exonerated by supposing it employs (informally) confirmation rather than probability, we would hope that developers and evaluators of assurance cases are explicit in any choice between

[12]Few subjects will have technical knowledge of probabilities or confirmation; the point is that their intuitive reasoning is sound for many purposes, and uses confirmation rather than probabilities.

probability and confirmation, and also apply Bayes' Rule in circumstances like these (on representative numbers if only qualitative estimates are being used).

A different example where population probabilities may confound elementary reasoning is the "Paradox of the Ravens" [66]. Here, we seek evidence for the claim "all ravens are black"; the equally valid contrapositive of this claim is "all non-black objects are non-ravens" for which a white shoe is produced in evidence, allowing the triumphant declaration "that proves it—all ravens *are* black!"

A plausible escape from this "paradox" is *Nicod's criterion* [85] that only observations of ravens should affect our judgment whether all ravens are black. More generally, claims about some class of objects can be confirmed or refuted only by evidence about those objects. Under this criterion, we expect that observations of black ravens would tend to confirm our claim, while a non-black raven definitely refutes it. Good, in a cleverly titled one-page paper [49], rebuts this expectation with an example where observation of a black raven disconfirms the claim "all ravens are black."

> Suppose that we know we are in one or other of two worlds, and the claim under consideration is that all the ravens in our world are black. We know in advance that in one world there are a hundred black ravens, no non-black ravens, and a million other birds; in the other world there are a thousand black ravens, one white raven, and a million other birds. A bird is selected equiprobably at random from all the birds in our world and turns out to be a black raven. This is strong evidence that we are in the second world, wherein not all ravens are black.

Examples such as this are challenging to philosophers seeking to explain and justify the methods of science, but for assurance the salient points are that we need to be skeptical about evidence (hence consideration of alternative claims and counterclaims) and may need to collect additional evidence to rule out alternative explanations. (In Good's example, observations of additional birds would allow us to determine if we are in the world with a hundred ravens, or the one with a thousand.) Confirmation measures provide an attractive framework in which to probe these issues and, far from being difficult for human evaluators, they correspond to inbuilt human faculties for the weighing of evidence.

2.3 Overall Assessment of Soundness

As noted in Section 2.1, the argument of an assurance case is *fully valid* when, in addition to being logically valid, all its claims are supported by evidence, all its steps are deductive, and there are no undefeated defeaters. A fully valid argument is *sound* when human evaluators attest that the justification for each interior step is satisfactory, and that the weight of each evidential step is sufficient to justify its

"something useful" claim, as explained in Section 2.2. CLARISSA can check these requirements and highlight violations. In particular, CLARISSA/ASCE allows argument steps to be marked when they are fully valid and again when their justification is considered sound, and can propagate these assessments and indicate them by coloring nodes so that developers can readily perceive how much of the argument is considered fully valid and sound.

The requirements for full validity are strong: in practice, it may not be possible (or credible) to make some steps deductive or, equivalently, they may be made so only by using an unjustified claim (i.e., assumption). Therefore, exceptions can be tolerated in the assessment of full validity (and thus soundness), but the goal should be "as deductive as possible and inductive only as strictly necessary." In particular, the residual risks due to these exceptions must be adjudged small, as will be described in Section 6. We prefer assumptions to inductive steps as these identify the source of doubt more precisely. For the same reason, we also suggest that no undefeated defeaters should be present when a case is submitted for assessment: the source of concern should have been eliminated or mitigated and thereby reduced to a residual risk, or localized to an assumption.

Thus, in evaluating an assurance case, we recommend that the presence of inductive steps, defeaters, and unsupported claims or assumptions should cause the case to be labeled "inductive." During development, these deficiencies should be examined and progressively eliminated, or else reduced to residual risks that will be reported separately and evaluated as described in Section 6.

These steps in the process of logical assessment for an assurance case argument are portrayed in Figure 8. If the case is incomplete, the assessment proceeds in a relaxed mode; if minor errors (e.g., misspelled claims) are present, these should be corrected but we do not flag them as invalidities or defeaters. Logically invalid arguments should be corrected; those with nondeductive steps may either be revised or continue their assessment with the caveat "inductive" attached. Similarly, defeaters may either be addressed or the assessment allowed to proceed in their presence. The order in which validity, deductiveness, and defeaters are assessed is not important. To the extent possible, we also assess soundness for incomplete and inductive cases.

A completed case should have a fully valid argument and is considered sound when all its steps are assessed to justify their claims, based on the subclaims or evidence provided. Confirmation measures can assist this judgment for evidential steps but we do not recommend developers or evaluators of assurance cases should fix on specific confirmation measures and make their numerical assessments a rigid criterion for judging the evidential justification for a claim. Rather, we expect them to apply the ideas presented in Section 2.2 "qualitatively" and to consider prior beliefs and counterclaims when assessing the extent to which evidence justifies a claim. These qualitative judgments can be developed and honed by "what if" experiments using numerical representations for the probabilities concerned. When

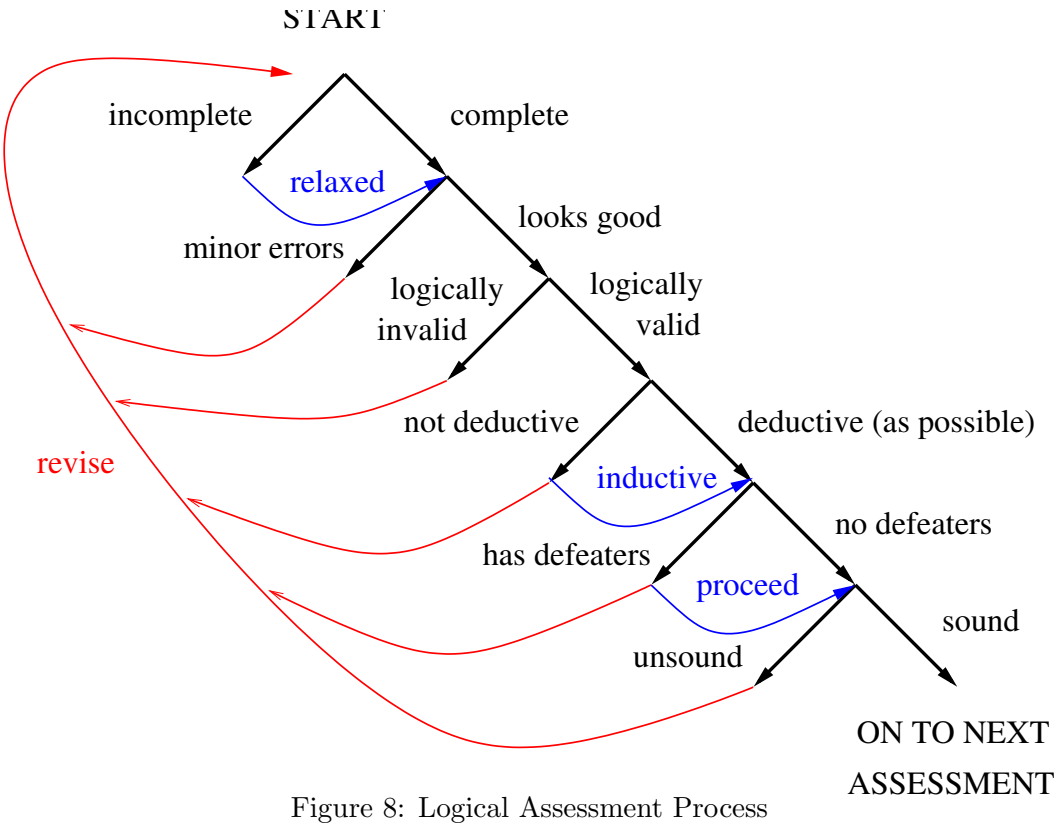

Figure 8: Logical Assessment Process

evidence is composed of multiple items that are assessed separately but are not conditionally independent, then it may be necessary to use advanced methods and tools such as Bayesian Belief Nets (BBNs) [43, 69, 83] to calculate the probabilities that will be used in the confirmation measure. Again, we recommend numerical experiments using BBN tools to develop understanding of the interactions involved and development of appropriately simplified rules for qualitative judgments.

Summarizing the discussion from the previous section, we suggest that selection and assessment of evidence can proceed as follows. First, it is sensible to seek evidence that would be surprising (i.e., not expected) if the claim is untrue and, if multiple forms of evidence are available, to seek those that are diverse, that is, not associated with each other unless the claim is true. The claim should be at the "something useful" level and there should be a suitable theory available that connects such claims to the measurements and observations delivered as evidence. We should follow Nicod's criterion and select evidence that supports the claim directly rather than by delicate inference and, if the claim is a conjunction, we should ensure that each conjunct is supported by some part of the evidence. Next, a confirmation measure can be evaluated and assessed as an indication of the weight of that evidence. Keynes' measure is attractive as its original and likelihood forms are the same. Numerical experiments can be performed to establish a suitable threshold for the measure. Finally, counterclaims should be considered and the discriminating

power of the evidence assessed with the aid of measures such as Good's. Again, numerical experiments can be used to establish thresholds.

We continue our examination of assurance cases from a positive perspective by proceeding from their logical to their probabilistic evaluation before turning to negative perspectives and the important role of defeaters.

3 Positive Perspectives: Probabilistic Valuation

Soundness is one aspect in the positive assessment of confidence for an assurance case, but it lacks gradation. Suppose, for example, we have a sound case, then reduce its threshold for weight of evidence and reduce the quantity or quality of evidence accordingly; the case remains sound, but we are surely less confident in the verity of its top claim. A different "weakening" is seen in DO-178C [92], where Design Assurance Levels (DALs) A to C require both High and Low Level Requirements (HLR and LLR), whereas (the lower) Level D requires only HLR. Intuitively, the idea is that we are less confident of the large "leap" in reasoning from implementation directly to HLR than of the combination of steps from implementation to LLR and then to HLR.

The motivation for these "weakened" cases is that they should be cheaper to produce, yet might still be adequate for less critical systems, or for less critical claims. Dually, we would like some basis for believing that the additional cost of the original "strong" cases does deliver greater confidence in their claims. What we seek, therefore, is a way to augment soundness with a graduated measure that indicates the strength of our confidence in the case.

The strength of confidence in an assurance case is naturally expressed as a probability. We could assess this as a subjective evaluation of the entire case, but a more principled method is to calculate it as the composition of assessments for the basic elements of the case, such as evidence, and individual argument steps. This will involve some combination of logic and probability, which is a notoriously difficult topic, because the two fields have different semantic foundations [1, 46, 86].

Nonetheless, there are numerous proposals for calculating probabilistic confidence in assurance cases by methods of this kind (e.g., [8, 35]), but a study by Graydon and Holloway [56] cast doubt on many of them. Graydon and Holloway examined 12 proposals that use probabilistic methods to quantify confidence in assurance case arguments: five based on Bayesian Belief Networks [43], five based on Dempster-Shafer [97] or similar forms of evidential reasoning such as Jøsang's opinion triangle and subjective logic [73], and two using other methods. By perturbing the original authors' own examples, they showed that all the proposed methods can deliver implausible results.

We suspect that the reason for this disappointing behavior is that the methods concerned are attempting a double duty: they aim to evaluate confidence in the case, but also (implicitly) its soundness. Probabilistic methods are poorly suited to the latter task, which is more naturally cast in terms of logic. In Assurance 2.0 we separate these evaluations and assess soundness as a logical property, as described in the previous section, and only for cases assessed to be sound do we proceed to assess probabilistic confidence. Nonetheless, we intend to explore Graydon and Holloway's examples when our tools are fully developed. Our methods for probabilistic valua-

tion are compositional over the five basic building blocks of Assurance 2.0 cases, as described in the following subsections.

3.1 Evidence Incorporation Blocks

A generic evidence incorporation block is shown in Figure 9. The basic idea is that observations (e.g., measurements, tests, or analyses etc.) on the system under consideration yield evidence E that is asserted to support claim C; the assertion of support is justified by a narrative attached to the evidence incorporation argument block, which may be further supported by a side-claim or warrant W. As indicated in the figure, the claim C will generally be about "something measured" and the side-claim W will generally concern provenance of the evidence and will be supported by a subcase establishing this. For example, if E is evidence from testing, then C will be a statement about the test outcomes, such as "no failures in n tests" or "MC/DC coverage achieved"; it will not be an interpretation or "something useful" statement about the tests, such as "achieved reliability 0.99" or "no unreachable code present" since these are "something useful" inferences derived from the test measurements via a suitable theory of testing that will be applied in a higher-level substitution block. The provenance side-claim W and its supporting subcase must establish that the tested artifact is the real thing, that the test oracle and any measurement harness are trustworthy, and that the test procedure is sound and was performed correctly, and so on.

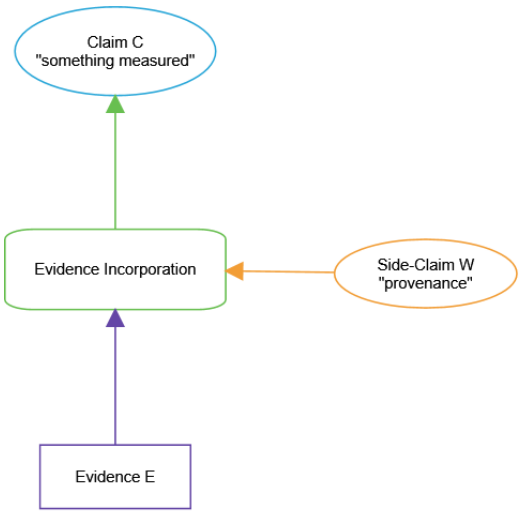

Figure 9: Evidence Incorporation Block

As we explained in Section 2.2, the subjective posterior probability $P(C \mid E)$ is a natural expression of confidence in the claim C, given the evidence E. However, when assessing soundness we use a confirmation measure rather than the posterior probability because we wish to evaluate the discriminating power, or "weight," of the evidence, and confirmation measures do this. But once we have assessed soundness, it is reasonable to use the posterior (or a qualitative approximation to this) as our measure of probabilistic confidence in the claim C and it is this that will be propagated through the probabilistic valuation for the rest of the case. (Note that if the evidence incorporation step uses multiple items of evidence then their individual

contributions $P(C \mid E_1), P(C \mid E_2), \ldots, P(C \mid E_n)$ will be combined into the overall $P(C \mid E)$ using methods such as BBNs.)

Now $P(C \mid E)$ is an epistemic judgment and it can be performed in several ways. One way would consider the totality of the information about E, including its provenance and other items represented in the side-claim W. In this case, probabilistic confidence in C, which we write as $P_{conf}(C)$, will be simply the "holistic" $P(C \mid E)$. Another way might assess $P(C \mid E)$ using only the direct contribution of E to C, with the side-claim W assessed separately. In this case, probabilistic confidence in C will be some combination of $P(C \mid E)$ and probabilistic confidence in the side-claim, $P_{conf}(W)$, which will be accumulated over the subcase supporting W. There are several plausible forms for the combination including arithmetic product (corresponding to logical conjunction)

$$P^P_{conf}(C) = P(C \mid E) \times P_{conf}(W)$$

and "sum of (probabilistic) doubts"

$$P^D_{conf}(C) = P(C \mid E) + P_{conf}(W) - 1.$$

We describe these in more detail in the following section.

3.2 Substitution and Concretion Blocks

A generic substitution or concretion block is shown in Figure 10. As described earlier, in Section 1.1, a substitution block has a subclaim S expressing some property A of a model P that is used to justify property B of model Q as the claim C. Special cases arise when the properties or models are the same. If the properties A and B are the same (so that, for example, we are justifying correctness of the HLR on the basis of correctness of the LLR), then the method of justification is generally to show that the models P and Q are "equivalent" which may be achieved informally by traceability analysis, or formally by verification of a homomorphism. If the models P and Q are the same (so that, for example, we are justifying absence of unreachable code on the basis of MC/DC coverage by requirements-based tests) then the method of justification is generally to appeal to some theory that addresses the topic.

Concretion blocks are somewhat similar: they justify a typically abstract property and model (e.g., "the system shall be correct") on the basis of a more concrete property and model (e.g., "the system shall satisfy its HLR").

The significant feature of substitution and concretion blocks from the perspective of probabilistic confidence propagation is that the claim C is derived from just a single subclaim S, subject to a side-claim or warrant W. For soundness, we require that the parent claim is deductively entailed by the subclaim, subject to the side-claim. That is, $W \supset (S \supset C)$, which is equivalent to $W \wedge S \supset C$. A narrative in the argument block must justify this relationship indefeasibly.

When we consider probabilistic confidence, we apply a probabilistic interpretation to the implication above, so that

$$P_{conf}(C) \approx P_{conf}(W \wedge S).$$

The probabilistic confidence $P_{conf}(W \wedge S)$ is given by $P_{conf}(W) \times P_{conf}(S \,|\, W)$ but we expect the lower steps of the argument (i.e., the subcases supporting W and S) to supply $P_{conf}(W)$ and $P_{conf}(S)$. Since $P_{conf}(S)$ is not the same as $P_{conf}(S \,|\, W)$ (unless S and W are independent), this is not quite what is required. However, the structure of a sound assurance case is such that all the claims and subclaims appearing in its argument must be true—so when we evaluated the subclaims and evidence contributing to S, we implicitly did so in a context where W is true. Hence, our assessment of probabilistic confidence in the subclaim S is really confidence *given* the rest of the case, and so the confidence we labeled $P_{conf}(S)$ is "really" $P_{conf}(S \,|\, W)$ and probabilistic confidence in $P_{conf}(W \wedge S)$ can indeed be taken as the product of probabilistic confidence in its two subclaims. Thus

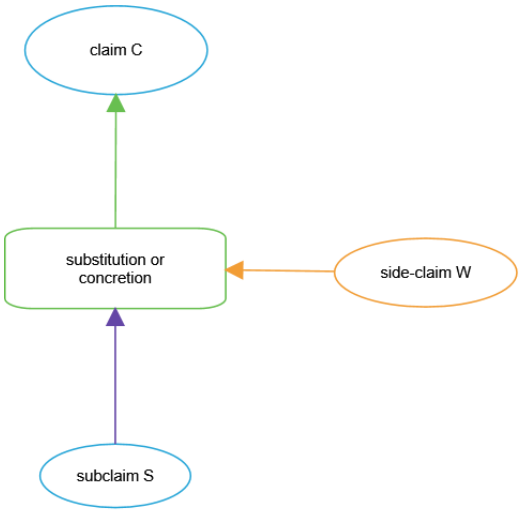

Figure 10: Substitution/Concretion Block

$$P^{P}_{conf}(C) = P_{conf}(S) \times P_{conf}(W)$$

where we use the superscript P in $P^{P}_{conf}(C)$ to indicate this is the "product" calculation.

Some may feel that this calculation is too aggressive and would prefer a more conservative approach. One such is the "sum of doubts" approach: our doubt in the parent claim is no worse than the sum of doubts for its subclaims and side-claims [1].

The "doubts" referred to here are *probabilistic doubts* as opposed to the other use of the term to mean general disquiet or concern. Probabilistic doubt in a claim C is probabilistic confidence in its negation $P_{conf}(\neg C)$, so

$$\begin{aligned} P_{conf}(\neg C) &\approx P_{conf}(\neg(S \wedge W)) \\ &= P_{conf}(\neg S \vee \neg W) \\ &= P_{conf}(\neg S) + P_{conf}(\neg W) - P_{conf}(\neg S \wedge \neg W) \\ &\leq P_{conf}(\neg S) + P_{conf}(\neg W). \end{aligned}$$

Then, since $P_{conf}(\neg C) = 1 - P_{conf}(C)$, and similarly for $\neg S$ and $\neg W$, we have

$$P^{D}_{conf}(C) \geq P_{conf}(S) + P_{conf}(W) - 1$$

where the superscript D indicates this is the "sum of (probabilistic) doubts" calculation.

These derivations of $P^P_{conf}(C)$ and $P^D_{conf}(C)$ also justify the similar formulas presented in the previous subsection for evidence incorporation steps.

Choosing a method for propagation of probabilistic confidence is a matter for judgment by the developers and evaluators of an assurance case. We have proposed two candidates for $P_{conf}(C)$: one, $P^P_{conf}(C)$ is aggressive but makes strong assumptions; the other, $P^D_{conf}(C)$ is conservative but requires only weak assumptions. There is no reason to think that either candidate is "correct" and developers may use either of these or an alternative method (with a satisfactory explanation) of their own devising. We do, however, consider it reasonable that any good estimate will lie between those given by $P^P_{conf}(C)$ and $P^D_{conf}(C)$.

The indefeasibility criterion of Assurance 2.0 requires that the conjunction of subclaim and side-claim should deductively entail the parent claim; hence we identify confidence in the parent claim with confidence in this conjunction. However, it is possible that, although we are persuaded of the logical entailment, our probabilistic confidence in the parent claim differs from that suggested by the calculation above. For example, we noted earlier that DO-178C requires both High and Low Level Requirements at DALs A–C, but only HLR at DAL D. Intuitively, the idea is that the more costly combination of substitution steps from implementation to LLR and then to HLR engenders more confidence than the single large step from implementation directly to HLR, even though all the steps are assessed as deductive. This can be accommodated by defining a factor f such that confidence in the parent claim C is given by

$$P_{conf}(C) = f \times P_{conf}(S \wedge W)^{13} \tag{2}$$

(or a qualitative approximation thereto), where $P_{conf}(S \wedge W)$ represents the chosen method for combining $P_{conf}(S)$ and $P_{conf}(W)$.

Exceptionally, we allow the *inductive* justification of an argument step, where the conjunction of the subclaims and side-claims "strongly suggest" but do not imply the parent claim. This means there must be some missing element[14] M that would make the relationship deductive:

$$W \wedge M \wedge S \supset C.$$

Presumably M is unknown (otherwise we would have included it), but the fact that we have labeled the argument step inductive means that we recognize its (possible) existence. The reason we deprecate inductive steps is that the absent M represents

[13] Alternatively, f could be a function: $f(P_{conf}(S \wedge W))$.

[14] It is possible that W or S are *wrong* rather than weak, and therefore cannot be corrected by conjoining an M. See the discussion of residual interior doubts in Section 6.

a *defeater*, that is a condition that can invalidate the argument, and we prefer these to be identified more specifically.

Rather than label the step inductive, we could instead represent M explicitly as an unsupported claim labeled "something missing here." This may be conjoined either to W (i.e., the side-claim is too weak) or to S (i.e., the subclaim is too weak), or split between them. Alternatively, M could be represented by an explicit defeater node. We prefer these alternatives to inductive steps because they are more explicit about the existence and location of the potential defeater.

Recall, from the discussion around Figure 8 in Section 2.3, that in evaluating the soundness of an assurance case with inductive steps or with unsupported "something missing" claims or explicit defeaters, we ignore these elements, but label the whole case "inductive" and therefore in need of intense and skeptical scrutiny.

When evaluating probabilistic confidence in such a case, a strict approach would assess zero confidence for inductively justified claims, unsupported "something missing" claims, and the targets of defeater nodes. Such assessments will propagate upwards and deliver zero confidence in the top claim. However, we have already labeled the case "inductive" and are aware of its defeasible character, so we would like the confidence assessment to say something useful beyond this. CLARISSA/ASCE accommodates this by allowing manual adjustment to calculated values for probabilistic confidence: confidence in a defeated claim can be left unchanged, or set to zero, or reduced to some intermediate value. Nodes are color coded according to user-selected thresholds on their probabilistic confidence and adjustment to these settings can provide developers with visualizations that help focus their attention on weak areas of the case and to comprehend the scope of their impact.

3.3 Decomposition Blocks

Decomposition blocks are used when a claim can be decomposed into subclaims distributed over some set or structure, such as components, properties, configurations, hazards, or time, and so on. A side-claim ensures that the decomposition is valid: that is, all elements of the decomposition are considered, and the subcases are disjoint, etc. A generic example was shown earlier, in Figure 3.

As with substitution and concretion blocks, there are several plausible ways to estimate probabilistic confidence in the parent claim C from that in its subclaims $S_1, \ldots, S_n$ and side-claim W. These include the product calculation

$$P^P_{conf}(C) = P_{conf}(W) \times \prod_{i=1}^{n} P_{conf}(S_i)$$

and the sum of probabilistic doubts

$$P^D_{conf}(C) = P_{conf}(W) - n + \sum_{i=1}^{n} P_{conf}(S_i),$$

n	$P_{conf}(S_i)$	$P_{conf}(W)$	$P^P_{conf}(C)$	$P^D_{conf}(C)$
1	0.99	0.99	0.98	0.98
1	0.95	0.95	0.90	0.90
1	0.90	0.90	0.81	0.80
1	0.95	0.80	0.76	0.75
2	0.99	0.99	0.97	0.97
2	0.95	0.95	0.86	0.85
2	0.90	0.90	0.73	0.70
2	0.95	0.80	0.72	0.70
3	0.99	0.99	0.96	0.96
3	0.95	0.95	0.81	0.80
3	0.90	0.90	0.66	0.60
3	0.95	0.80	0.69	0.65
5	0.99	0.99	0.94	0.94
5	0.95	0.95	0.74	0.70
5	0.90	0.90	0.53	0.40
5	0.95	0.80	0.62	0.55

Figure 11: Confidence Calculations for Representative Decomposition Blocks

which can each be derived by generalizing the corresponding description in the previous subsection.

Also as with substitution and concretion blocks, developers and evaluators of assurance cases may choose one of these two methods or invent some other way to estimate propagation of confidence. The product calculation assumes that the subclaims to the decomposition are independent, which may not be so. For example, we might establish partitioning among tasks by decomposing this into time partitioning and space partitioning. These are logically disjoint, but both might be enforced by the operating system kernel, so they are hardly independent. Conversely, the sum of probabilistic doubts calculation is very sensitive to subclaims with low confidence, but analysts may consider those subclaims to apply only to unimportant (i.e., low risk) circumstances. Thus, analysts may adjust either of these estimates or use one of their own devising, perhaps employing BBNs or Dempster-Shafer's theory of evidence [97], but we do suppose that reasonable estimates will lie between $P^D_{conf}(C)$ and $P^P_{conf}(C)$.

Also as we have seen before, it may be that the method of decomposition raises or reduces confidence in the parent claim, and this can be represented by applying some factor or function f to the calculation of $P_{conf}(C)$.

We have often advocated numerical "what if" experiments to develop intuition on the behavior of probabilistic measures and in Figure 11 we present the results of one such experiment. We consider a decomposition block with claim C having n subclaims S_i and a side-claim W. The first three columns of the table list values for n, $P_{conf}(S_i)$ (assumed to be the same for all subclaims), and $P_{conf}(W)$,[15] while the two rightmost columns list corresponding values for $P^P_{conf}(C)$ and $P^D_{conf}(C)$, respectively. The first four rows consider the case $n = 1$, which covers substitution and concretion blocks (and, to a certain extent, evidence incorporation blocks as well).

We see that for most combinations, $P^P_{conf}(C)$ and $P^D_{conf}(C)$ are very close[16]; they diverge as confidence in the subclaims is reduced (corresponding to the third row in each block) and their number is increased. We also see that one low-confidence input (corresponding to the fourth row in each block) has a substantial impact and that this is most marked for the sum of doubts measure and for larger n.

The main conclusions seem to be that decomposition blocks with many subclaims require strong confidence in those subclaims, and that all subclaims (and side-claims) need similar levels of confidence.

3.4 Calculation Blocks

Calculation blocks are much like decomposition blocks except the claim and subclaims concern the values of some (usually numerical) quantities and the quantity in the parent claim is calculated from those in the subclaims using a formula justified (presumably by citing some theory) in the body of the calculation block, subject to constraints cited in the side-claim. The analysis of probabilistic confidence then follows that for decomposition blocks.

3.5 Overall Assessment of Probabilistic Confidence

If the goal is to make a strongly argued case for some probabilistic assessment of the system under consideration, such as its reliability, then it seems best to make this quantity a part of the top claim and to arrange the case to justify it explicitly, probably by reference to suitable theories for reliability estimation (see [22, Section 3.2] for an example). We call this an *internal* probabilistic assessment because it is constructed inside (i.e., as part of) the argument. A variant is an *indirect* probabilistic assessment where we use a substitution block to justify a probabilistic claim by adherence to a standard. For example, DO-178C [92] and governing regulations [39,41]

[15] Of course, the results would be unchanged if the same probabilities were distributed differently among the subclaims and side-claim.

[16] Note that the product calculation with n total side- and subclaims yields $(1 - d)^n$, where d is probabilistic doubt in each of these claims and, by the binomial expansion, this is $1 - n \times d +$ smaller terms. Similarly, the sum of doubts is $n \times d$, which yields confidence $1 - n \times d$. Thus the product and sum of doubts calculations deliver very similar values for confidence when d is small.

indicate DO-178C Level A is suitable for a probability of failure on demand (*pfd*) of 10^{-9} while Level D is sufficient for *pfd* of 10^{-3}. (The notion of *demand* is flexible; for example, it can be interpreted as a single iteration of a cyclic control system, or an hour of operation—the usual case in avionics, or a complete mission.)

In contrast to these approaches, an *external* probabilistic assessment is constructed outside the (otherwise independent) argument in the manner illustrated by the previous subsections, and will generally be much more approximate as it depends on generic analyses. Note that an external assessment will deliver *probabilistic confidence in a (logical) claim* (e.g., 99.9% confident in the absence for critical faults), not a *probabilistic claim* (e.g., reliability wrt. critical failures better than 99.9%). Additional work is required to connect these concepts, as described in Section 4.

Unlike other authors who develop theories for probabilistic confidence in assurance cases, we do not advocate any particular approach (although the chosen approach should be used consistently) nor consider any of them "correct." Because these analyses are generic, they are conservative to the point where their actual numerical estimates may be of little direct value. For example, if we take the $n = 3$ case in Figure 11, we see that when confidence in the subclaims or evidence supporting the block, and in the side-claim, is 0.99, then confidence in the parent claim is 0.96 (i.e., 0.99 to the fourth power, using the product calculation); if we iterate this (i.e., we use four such blocks to support the three subclaims and the side-claim for another such block), confidence in the next level will be 0.85 at best (i.e., 0.96 to the fourth power), then 0.52 at the third level, and a mere 0.07 at the fourth level. Using sum of doubts, the corresponding doubts are 0.04 (i.e., 0.01 times 4) at the first level, 0.16 at the second, then 0.64 at the third, and 1 (since doubts cannot exceed 1) at the fourth level, corresponding to a confidence of zero. These calculations seem to suggest that larger arguments, with larger quantities of evidence, tend to reduce total confidence. This might be true, all other things being equal, but they do not remain equal as we will now see.

First, we recognize that the basic product and sum of doubts calculations do indeed depend only on confidence in the evidence supplied and in any assumptions[17], but not on the shape or size of the argument tree above them. That is, for the product calculation, probabilistic confidence in the top claim is simply the product of probabilistic confidence in all evidence and assumptions (as can be proved by induction on the height of the argument tree). Similarly, for the sum of doubts calculation, the doubt in the top claim is simply the sum of probabilistic doubts over all evidence and assumptions.

However, although the basic calculation of confidence depends only on the evidence and assumptions and not on the argument, this does not imply that confidence is invariant under transformation to the argument—for different arguments may re-

[17]We treat unsupported claims as assumptions, and expect some probabilistic assessment of confidence to be assigned to them.

quire different evidence, and even the same evidence may deliver different confidence for different claims. Furthermore, the transformed argument must remain sound and this may lead to different side-claims that require different evidence or assumptions. More significantly, a transformed argument may have better justification. Recall that in Section 3.2 formula (2), we introduced a factor or function f that is used to adjust probabilistic confidence according to the strength of the justification provided, and this may change as the argument changes.

DO-178C illustrates these topics. Design Assurance Levels (DALs) D and E of DO-178C reason directly from code to HLR, whereas the higher DALs A, B, and C insert intermediate steps involving LLR. These extra steps involve additional evidence and it might seem this will reduce overall confidence. It certainly would do so if confidence in the additional steps were comparable to that in the originals but, as described earlier, confidence in the smaller steps from code to LLR and then to HLR can be much greater (due to stronger theories and methods for verification) than the single large leap from code to HLR. This will be reflected in stronger confidence in the evidence and in the justifications (and hence in the factors f), resulting in greater confidence overall.

As we noted earlier, the absolute numbers delivered by our external methods of probabilistic assessment must be used with caution. Because it is based on human assessment of confidence in evidence and assumptions, and because this may vary greatly with different assessors or circumstances [75], the very foundation of the calculation may be considered unstable. Furthermore, because the methods of propagation are generic and conservative, they likely underestimate the overall confidence that a "more exact" probabilistic calculation would assess, if such were available.

We are comfortable with these limitations: we consider soundness to be the critical property, with probabilistic confidence as a useful but inexact augmentation that assessors can use to keep track, in a rational manner, of the location and extent of weak and strong parts of an argument and that may be particularly valuable when exploring residual doubts (see Section 6) and graduated assurance. That is to say, although probabilistic confidence calculated for the top claim may be small, the reasons for this apply uniformly and so it can be used to compare different assessments of residual risk and different methods for graduating an argument by trading cost (typically, for gathering evidence) against confidence for less critical components or properties.

Accordingly, Clarissa/asce can calculate probabilistic confidence based on user-specified confidence in evidence and using a variety of propagation rules (e.g., the product or sum of doubts methods, or those defined by the user) and can color nodes (e.g., red, amber, green) according to user-selected thresholds to indicate the confidence calculated for them, as illustrated in Figure 12. In addition to valuations of probabilistic confidence calculated by propagation from evidence, Clarissa/asce

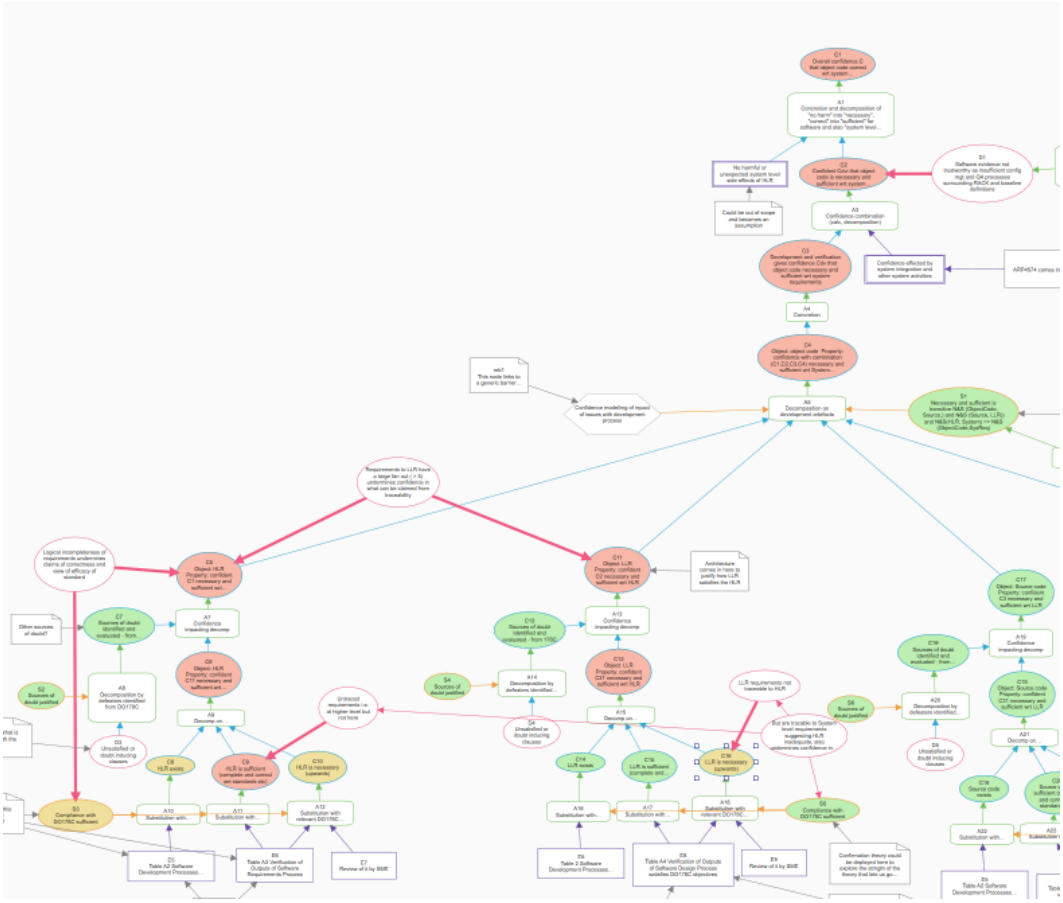

Figure 12: Portion of CLARISSA/ASCE Canvas Showing Probabilistic Valuations

allows manual adjustment to these values (as an informal version of the factors f), both to increase (e.g., due to justification by a strong theory) or reduce them (e.g., due to residual doubts).

We might wonder how it can be that the record shows modern aircraft, say, to be safe and reliable when the calculations discussed here deliver rather small estimates for probabilistic confidence in these properties? One response is that the conservative and generic calculations considered here severely underestimate the true values because they take no account of the particular kinds of system concerned. In general, these are highly redundant control systems. As control systems, they sample sensors and inputs and drive actuators and signals many times a second, and substantial state is stored in the physical plant itself. Thus, a "small" fault in a calculation (i.e., one encountered very infrequently) will affect only one or a few iterations of a controller (otherwise it would be a "big" fault and we are confident these will be found and eliminated) and the controller will correct the effects of these

in the next few cycles, just as it does for environmental disturbances.[18] In addition, the system will generally be heavily redundant with a lot of voting and averaging: these are present to tolerate major failures, but will often mask small defects, too. Truly accurate assessment of probabilistic confidence would need to consider these contextual and architectural factors.

A second response is that the confidence assessments we apply to individual argument steps *a priori* severely underestimate those that can be applied retrospectively based on *a posteriori* experience.

In the next section we present a method for deriving estimates for long run safety (i.e., reliability with respect to critical failures) on the basis of *a priori* probabilistic confidence in the absence of faults as delivered by an assurance case, plus limited *a posteriori* operational or test experience. This method employs Conservative Bayesian Inference (CBI) to derive posterior estimates for safety from prior confidence in the absence of faults (or, in a more detailed analysis, absence of "large" faults). To generate useful conclusions, these methods require the prior probabilistic confidence to exceed 0.9. We have seen that generic methods of confidence propagation may struggle to achieve this and so we expect the final determination that the assurance case is "good enough" to justify adequate confidence will be a holistic human assessment, based on a number of factors (such as are being developed throughout Sections 2–6), rather than a conservative calculation of probabilistic confidence using the methods described in this section.

[18] Of course, Murphy's Law ensures that some "small" defects can have major consequences: for example, the in-flight upset to an Airbus A330 [7].

4 From Confidence to Safety

The top claim of an assurance case often asserts that the system has some logical property (e.g., "is safe"); implicitly, this implies that the system has no faults that could jeopardize the property. We have explored methods for gaining and assessing confidence in such claims, but how does that relate to the properties we really care about, which are typically absence (or rarity) of critical failures? In other words, how do we get from confidence in absence of faults to rarity of failures?

In this section we present a theory called *Conservative Bayesian Inference* (CBI) that can accomplish this step. We present it as a separate theory, external to any assurance case argument, but we note that it could be incorporated by extending an argument above its existing top claim using a substitution block to take that (logical) top claim to a new higher-level top claim concerning the probability of critical failures over the life of the system.

Confidence in the case can be expressed as probabilistic confidence in its top claim $P_{conf}(C^T)$. The top claim C^T typically concerns absence of faults or defects that could lead to a critical failure. Let us abbreviate "suffers a critical failure" by simply "fails" and "has a fault that could lead to a critical failure" as "faulty" (with "nonfaulty" as its negation); these definitions are consistent with standard usage [78]. Then, by the formula for total probability

$$\begin{aligned} & P(\text{system fails [on a randomly selected demand]}) \\ = \;& P(\text{system fails} \,|\, \text{system nonfaulty}) \times P(\text{system nonfaulty}) \\ & + P(\text{system fails} \,|\, \text{system faulty}) \times P(\text{system faulty}). \end{aligned} \tag{3}$$

The first term in this sum is zero, because the system does not fail if it is nonfaulty (as we have defined those terms). We let P_{fif} be the probability that the system "fails if faulty," we have $P_{conf}(C^T)$ as the probability the system is nonfaulty (so that $P(\text{system faulty}) = 1 - P_{conf}(C^T)$), and therefore its probability of failure on demand, P_{fd} is given by

$$P_{fd} = P_{fif} \times (1 - P_{conf}(C^T)). \tag{4}$$

Different industries make different assessments about the parameters to (4). Early nuclear protection, for example, seemed to presume the system *is* faulty, so in effect it set $P_{conf}(C^T)$ to 0 and performed extensive random testing to substantiate (typically) $P_{fif} < 10^{-3}$. If those regulators had accepted that modest amounts of assurance could deliver $P_{conf}(C^T) \geq 0.9$, then by (4) the same probability of failure could be achieved[19] with the much less costly testing required to validate merely $P_{fif} < 10^{-2}$.

[19] We are cutting a lot of corners here: the full treatment must distinguish *aleatoric* from *epistemic* assessment and must justify that beliefs about the two parameters can be separated; [80–82, 110] give details.

Dually, FAA AC 25.1309 [41] and the corresponding European regulations [39] for aircraft certification indicate $P_{fd} \leq 10^{-9}$ for catastrophic failure conditions and seem to presume the system *will* fail if it is faulty, so in effect they set $P_{fif} = 1$. The whole burden for assurance then rests on the value assessed for $P_{conf}(C^T)$—so that we need $P_{conf}(C^T) \geq 1 - 10^{-9}$, which is completely implausible. In fact, there is no credible assignment of values to the parameters of (4) that delivers $P_{fd} \leq 10^{-9}$ per hour [25]; an alternative model is needed.

Rather than the figure of 10^{-9} per hour, which is intended only as "an aid to engineering judgment" [41], let us look at the fundamental requirement of these regulations: that a catastrophic failure condition is "not anticipated to occur during the entire operational life of all airplanes of one type." Extending (4), the probability of surviving n independent demands without failure, denoted $P_{srv}(n)$, is given by

$$P_{srv}(n) = P_{conf}(C^T) + (1 - P_{conf}(C^T)) \times (1 - P_{fif})^n. \tag{5}$$

Demands can be interpreted as hours of operation, or flights, or some other measure of exposure and, whichever is chosen, a suitably large n can represent "the entire operational life of all airplanes of one type." The notable feature of (5) is that the first term establishes a lower bound for $P_{srv}(n)$ that is independent of n. Thus, if assurance gives us the confidence to assess, say, $P_{conf}(C^T) \geq 0.9$ (or whatever threshold is used to interpret "not anticipated to occur") then it seems we have sufficient confidence to certify the aircraft as safe.

However, we can imagine using this procedure to provide assurance for multiple airplane types; if $P_{conf}(C^T) = 0.9$ and we assure 10 types, then we can expect that one of them will have faults. In this case, we need confidence that the system will not suffer a critical failure despite the presence of faults, and this means we need to be sure that the second term in (5) will be well above zero even though it decays exponentially. This confidence could come from prior failure-free operation (e.g., flight tests). Calculating the overall $P_{srv}(n)$ can then be posed as a problem in Bayesian inference: we have assessed a value for $P_{conf}(C^T)$, have observed some number r of failure-free demands, and want to predict the probability of seeing $n - r$ future failure-free demands. To do this, we need a prior distribution for P_{fif}, which may be difficult to obtain and difficult to justify for certification. However, Strigini and Povyakalo [100] show there is a distribution (specifically, one in which P_{fif} is concentrated in a probability mass at some $q_n \in (0, 1]$) that delivers *provably worst-case* predictions; hence, we can make predictions that are *guaranteed* to be conservative, given only $P_{conf}(C^T)$, r, and n. Using this approach, which is known as Conservative Bayesian Inference (CBI), Strigini and Povyakalo show that if $P_{conf}(C^T)$ is above 0.9, then $P_{srv}(n)$ is well above this floor, provided $r > \frac{n}{10}$.

If we regard a complete flight as a demand, then "the entire operational life of all airplanes of one type" might require n to be in the range 10^8 to 10^9 (e.g., as of 2019, the Airbus A320 series had performed over 150 million flights [23]).

Flight tests prior to certification might provide only $r = 10^3$, so it appears this is insufficient for certification by the criterion above. However, it can be argued that when an airplane type is certified we do not require (and in fact cannot feasibly obtain) sufficient evidence to predict failure-free operation over the entire lifetime of the type; instead, we initially require sufficient confidence only for, say, the first six months of operation and the small number of aircraft that will be manufactured and deployed in that period. This will be a much smaller value of n, and our $P_{conf}(C^T)$ (from assurance) and our r (from flight tests) will be sufficient for confidence in its failure-free operation. Then we will need confidence in the next six months of operation, with a larger fleet, (i.e., a larger n) but now have the experience of the prior six months failure-free operation (i.e., a larger r) and in this way we can "bootstrap" our way forward [17].

It remains to consider what happens if experience in operation does reveal a fault (by manifesting a failure, hopefully not catastrophic—indeed, there is an FAA requirement that no single fault may cause a catastrophic failure condition). Commercial airplanes operate in a legal and ethical framework where all incidents and accidents are promptly reported and dispassionately investigated. The FAA issues Airworthiness Directives mandating workarounds or corrections to detected faults; in extreme cases it may temporarily ground the fleet (as it did for Boeing 787 battery problems in January 2013 and 737 MCAS faults in 2019[20]). Bishop [15] constructs a statistical model for this scenario and shows that, under plausible assumptions, detection and repair of faults significantly increases long run safety, even if the fleet continues to operate after a fault has been discovered, and even if repairs may be imperfect.

In its totality, the analysis above (which is based on research at Adelard and City University in London [12,15–17,100]) provides—for the first time, we believe—a plausible statistical model that retrospectively explains the success of aircraft certification, and other certification regimes based on similar practices. At the base of this analysis is an assessed confidence (e.g., $P_{conf}(C^T) > 0.9$) that the system is nonfaulty or "fault-free" with respect to critical requirements.

Traditionally, software assurance cases have delivered a top claim that the software is nonfaulty with respect to critical requirements. Confidence in this claim was generally assessed separately (or left implicit) but we suggest that the case should now be expanded to include this attribute. The connection from confidence in nonfaultiness to reliability in operation was also assessed separately (or left implicit as

[20]The fatal crashes caused by design faults in the MCAS system of 737 Max aircraft may seem to repudiate the safety and certification arguments made here. However, it is clear that Boeing was not following either the spirit or the letter of established safety requirements and guidelines, and FAA oversight was weak and possibly captured. Thus, although this example does not repudiate the methods described here, it does illustrate that they cannot operate outside a genuine safety culture.

prior to CBI and other conservative approaches [18] there was no good theory to account for it) and we suggest that this, too, should now be made explicit and included in an expanded assurance case whose top claim would now become reliability with respect to critical failures.

There are other top claims, architectures, and methods of analysis that function similarly to that described above. A modification to the analysis above replaces strict fault-freeness by *quasi* fault-freeness, meaning the system is either nonfaulty or is faulty but with only a minuscule probability of failure [110]. This is a more robust model and yields attractive results [111], but the details are more complicated. Alternative properties to nonfaulty and failure-free include mission risk, and the claim that a new system is no worse than the old one. And an alternative to external calculation of probabilistic confidence in the case is internal justification of probabilistic quantities such as these within the case itself (i.e., the claims in the case make probabilistic statements; recall section 3.5). And as an alternative to single-threaded architectures, Littlewood and Rushby [82] provide a rigorous analysis of "monitored architectures" in which a highly trustworthy monitor checks the behavior of a less trusted primary system as advocated, for example, by the F3269-17 standard for unmanned aircraft [5]. Bishop and Bloomfield [16,18] develop an alternative worst-case analysis that predicts long term reliability from an estimate of the number of faults N at time of release (as opposed to confidence in their absence) and the operating time T. The predicted number of faults can be based on models of the software development process, such as the empirically validated "barrier" model [20].

This concludes our discussion on the positive view of assurance cases and we now turn from these to negative perspectives and the important role of doubts and defeaters.

5 Negative Perspectives: Doubts and Defeaters

In Assurance 2.0, the criterion for a completed assurance case is that it should be *indefeasible*, meaning that all identified doubts have been addressed, and we are confident no credible doubts remain that could change the decision supported by the assurance case [22, 94]. When we say that all doubts have been addressed, we do not require that they are eliminated; in appropriate circumstances, some doubts may be accepted as residual risks, as will be discussed in Section 6. What makes the case indefeasible is that we know about these doubts, have examined them, and made a conscious decision about them so that no credible information would make us change our decision. Indefeasibility is lost when there may be doubts that we do not know about, or doubts we do know about but have not consciously addressed.

Doubts are suspicions that some part of a case may be inadequate or wrong. On investigation, the location and nature of the doubt should become sharpened so that it can be expressed as a *defeater*: that is, a node in the assurance case argument that challenges or refutes the specific claims, arguments, or evidence represented by other nodes. There are several ways to address doubts and their associated defeaters. One way is to argue that the defeater is unjustified or incorrect, so there will be a subcase that *defeats the defeater*. If a doubt is legitimate, however, we must adjust either the argument, or the system (and then possibly the argument as well) to eliminate or mitigate its defeater(s). A defeater that requires adjustment to the system (as opposed to just its assurance case) is more properly considered a *hazard* than a defeater, but the issues we are developing here will apply to these as well.

Pollock [89] was the first to examine defeasible arguments of the kind that interest us, and he distinguished between *undercutting* and *rebutting* defeaters. Weinstock et al [51] later added *undermining* defeaters in an approach to assurance that they call *eliminative argumentation* (because one proceeds by eliminating doubts). Generally speaking, an undercutting defeater challenges an argument (i.e., its justification), a rebutting defeater challenges a claim, and an undermining defeater challenges evidence. However, when we first explore a doubt we may not know exactly how it should be manifested as a defeater. For example, we may have doubts whether a formal verification handles concurrency correctly, but until we investigate further we may not know if the problem is in the claim, the evidence, or the argument, or even in the mechanisms of the system itself. So the doubt may initially be represented in the case as a generic defeater expressing concern and aimed at the relevant claim and, as the investigation proceeds, it may become a specific defeater (i.e., one that makes its own claim) supported by its own subcase and targeted more precisely at one element of the case. Because their role evolves, Assurance 2.0 is somewhat relaxed about the representation of defeaters and the constraints applied to them (see the discussion around Figure 4). For example, we usually represent a defeater

by a node having the same shape as a claim, but we allow it to point to another claim with no intervening argument block. Dually, it may point to an argument block without being referenced in the justification for that block. In these ways, a defeater operates rather like a comment; on the other hand, it can also be interpreted as a (counter)claim and be justified by its own subcase as will be outlined in Section 5.3.

Investigation and resolution of doubts and defeaters serves two purposes: firstly, it is the primary means by which we avoid confirmation bias and drive the case toward soundness and indefeasibility; secondly, it helps reviewers comprehend the case as they find their own doubts have been anticipated and answered. For the first of these, it is enough just to deal with the doubts and move on, but for the second it is necessary to record the changes made and to support some kind of *dialectical* examination that allows reviewers to see how the case responded to previously considered doubts. This raises significant issues in the representation of an assurance case.

For example, one step in a case may decompose a claim into subclaims over some enumeration and we may doubt that this is done correctly, so we establish a defeater that attacks the decomposition step (e.g., by making the claim that the decomposition is incomplete). We may then develop a subcase that refutes this claim. But that subcase may include an element that is itself challenged by another defeater, and that is defeated by a further subcase, and so on. One issue is how to represent and evaluate assurance cases in the presence of defeaters and their subcases. We have the basic case that is attempting to substantiate some positive claim, then defeaters (negative claims) and subcases to substantiate them, and then further counter-defeaters and their subcases. Do we show all of these layers of defeat and counter-defeat? And do we have some way of showing which subcases are contributing to the basic case and which to a defeater or counter-defeater or counter-counter-defeater? We will refer to this as the problem of representing *defeasible arguments* and will briefly examine it in Sections 5.1.1 through 5.1.3.

When we have satisfied ourselves that a defeater is itself defeated, the investigation has served its purpose and we could eliminate everything associated with it because the original case was valid after all. But reviewers may have the same doubts, and so for them it is desirable to retain the defeaters and their subcases in a way that permits interactive replay and exploration. We will refer to this as the problem of representing *dialectical arguments* and will examine it in Section 5.1.4.

The problem of representing dialectical arguments becomes more complex when the original case, or system, or both, need to be adjusted in response to a defeater. Whereas the subcase for an unsuccessful defeater and the subcases for its counter-defeaters and so on can be seen as decorations to the basic, sound, case, a successful defeater requires a change to the original case and so we might need to retain the original case and its defeaters and then branch to the adjusted case.

An important special case is that of missing assumptions. Slavoj Žižek [112] identified the "unknown knowns," the "silent presuppositions we are not aware of," as a significant source of error in all human deliberation. In an assurance case, a defeater that identifies a missing assumption may be defeated by supplying the required assumption and, possibly, a subcase with evidence, to justify it. This amendment to the case is an addition rather than a change and its representation and management can be simpler than those for more general amendments in response to defeaters.

5.1 Defeaters in Reasoning, Argumentation, and Dialectics

Classical formal logic cannot tolerate contradictions among premises: these render the argument invalid. But in AI and in the study of human and commonsense reasoning it is reasonable to draw conclusions on the basis of incomplete and inconsistent information, and to revise these as new information becomes available. Consider, for example, an emergency room physician, updating her diagnosis and adjusting her treatment plan as new observations and test results become available.

In AI, the topic of drawing reasonable conclusions from incomplete, inconsistent, and changing information is referred to as "defeasible reasoning"; much the same topic is considered in formal logic as "nonmonotonic logic" ("nonmonotonic" because conclusions are not stable and may need to be withdrawn as more knowledge is acquired). A standard example has premises "birds can fly" and "Tweety is a bird" but then we learn that, contrary to the obvious conclusion, Tweety cannot fly. It turns out that Tweety is a penguin and the first premise needs to be modified to "most birds can fly." A different, though related, application arises in the study of "argumentation": here, different parties have different views and may advance premises that contradict each other.

We briefly examine defeasible argumentation seeking ideas that might prove useful for our application. We look first at methods for defeasible reasoning, followed by argumentation theory, eliminative argumentation, and dialectics.

5.1.1 Defeasible Reasoning

The crucial notion of *defeater* is from Pollock in 1987 [89] (although he published on these topics as early as 1967) where he proposed what is known as an epistemological approach to evaluation of defeasible arguments. This is a set of rules that defines how a cognitively ideal agent would arrive at warranted conclusions given a set of premises and their defeaters. Pollock's system has been criticized for lacking a "normative standard," being based on ad hoc intuitions about how a reasonable agent would respond to this or that cognitive situation, but Koons [77] observes that the same criticism can be lodged against several other theories of defeasible reasoning.

Theories for nonmonotonic logic do have more justified foundations, starting with McCarthy's *circumscription* [84] which, roughly speaking, prefers the most specific applicable premises. However, this approach gives intuitively incorrect answers in some cases, epitomized by the "Yale Shooting Problem" [63] and so, as with defeasible reasoning, there is a large literature of attempts to find more satisfactory treatments.

The goal of defeasible reasoning and of nonmonotonic logic is to work out what can be concluded when there are contradictory premises or when premises can change (e.g., in the Yale Shooting Problem, the gun is initially unloaded but later becomes loaded). However, these are not really relevant for assurance cases. In Assurance 2.0 we use Natural Language Deductivism (NLD) and expect completed assurance cases to be deductively sound, although we may tolerate some "inductive" argument steps if necessary. That is, we may accept some "gaps" in our knowledge, but not contradictions or changes.

While developing a case, and in supporting its review, we use defeaters (i.e., contradictions) to probe and challenge a case, but we do not expect to conclude anything from a case with unresolved defeaters: such a case is acknowledged as imperfect and incomplete. What we would like to know is: how much of the overall argument does a defeater cast into doubt, and how much is repaired by a counter-defeater? Unfortunately, these purposes are not served by defeasible reasoning and so we next turn to argumentation theory.

5.1.2 Argumentation Theory

Defeasible reasoning and nonmonotonic logic attempt to understand what conclusions can be drawn from a single argument when some of its premises are inconsistent. The field of argumentation on the other hand, considers multiple competing arguments and tries to deduce which ones emerge from the competition with their plausibility intact. Defeasible reasoning employs a defeater relation on premises, whereas argumentation has the relation of *attack* between arguments. The two ideas are related however. If we have an argument and then introduce a defeater, we can think of this as two arguments: the original, and a new one consisting of that plus the defeater, and in which the second attacks the first.

This is *abstract argumentation* theory, introduced by Dung [38] in a paper that has over 4,500 citations. Despite the term "argumentation" appearing in the name, the level of abstraction reduces it to an exercise on graphs: arguments are nodes in a graph that are connected by attack relations and we ask for rules to determine which sets of nodes are "accepted." Dung defined this in terms of "extensions" but "labeling" provides a more intuitive treatment [26]: each argument in the graph is labeled *in* (accepted), *out* (rejected) or *undecided*. The *reinstatement* rule on labeling stipulates: an argument is *in* iff all arguments that attack it are *out*; an

argument is *out* iff it is attacked by at least one argument that is *in*; arguments that are neither *in* nor *out* are *undecided*.

As with defeasible reasoning, the purposes served by abstract argumentation theory are sufficiently different from those in assurance cases that we do not find any direct application for its methods.

However, argumentation theory has a connection to logic programming that supports a computational interpretation for defeasible reasoning [28] and this may suggest methods for applying Answer Set Programming (ASP) [47] that are being explored in CLARISSA.

5.1.3 Eliminative Argumentation

"Eliminative Induction" is a method of reasoning that dates back to Francis Bacon who, in 1620 [9], proposed it as a way to establish a scientific theory by refuting all the reasons why it might be false (i.e., its defeaters). Modern treatments see falsifiability as the key characteristic of science [90] but the two can be related via Bayesian Epistemology [24], where Bayesian methods are seen as the best way to select among so-far unfalsified theories [64, 104]. These Bayesian methods relate to Confirmation Theory, discussed in Section 2.2.

Weinstock, Goodenough, and Klein [50] develop the idea of Eliminative Induction as a means of assurance they call *Eliminative Argumentation*. An argument is presented as a *confidence map*, which is rather like an assurance case with defeaters included and rules for accepting a claim only if all its defeaters have been eliminated. Colors are used to highlight the "positive" and "negative' parts of a case. Diemert and Joyce report successful application of eliminative argumentation in assurance of real systems [36]. We apply some of these ideas in Assurance 2.0, as described in Section 5.2.

5.1.4 Dialectics and Agreement Technologies

Dialectics refers to the back-and-forth nature of arguments employed in active debate. One of the goals of dialectical debate is to reach an agreed conclusion, so its methods are sometimes referred to as agreement technologies [87].

There are many approaches to dialectics and agreement, but an influential one that has been applied in several domains and is supported by tools is framed in terms of *argument schemes* [105,106]. These are outlines for many canonical kinds of argument: for example, argument from analogy, argument from expert opinion, and so on (there are about 30 in Walton's book [106]). These are supported by *critical questions*, which can be thought of as defeaters customized to each specific frame. For example, argument from expert opinion has six critical questions (e.g., "how credible is E as an expert source?"). Raising and responding to critical questions

gives rise to the dialectical element, just as it does in an assurance case challenged by defeaters.

Despite this apparent similarity, the compendium on agreement technologies [87] contains no reference to assurance or safety cases. Nor do more than 1,000 articles available online that were published in the journal *Argumentation* (Springer). Assurance and safety cases likewise make no reference to agreement technologies. An exception is the work of Yuan and Kelly, who have applied argument schemes and critical questions to assurance [108, 109].

Carneades [52] is a system that supports dialectical reasoning in a different way, allowing a subargument to be *pro* or *con* its conclusion and allowing weights to be attached to premises. A *proof standard* is calculated by "adding up" the *pro*s and *con*s supporting the conclusion and their attendant weights (rather like the labelings of argumentation theory). For example, a claim is "in" if it is not the target of a *con* that is itself "in" (unless it is also the target of an "in" *pro* ...); a conclusion is supported to the *preponderance of evidence* proof standard if it has at least one *pro* argument that is "in" and weighs more than any "in" *con* argument. The system, which is available at `http://carneades.github.io/`, provides several kinds of argument graphs for visualizing arguments. Recent work by Takai and Kido [101] builds on these ideas and is implemented in the commercial tool Astah GSN [4].

5.2 Approach Adopted in Clarissa

We have briefly reviewed how defeaters are used and represented in defeasible reasoning, argumentation theory, eliminative argumentation, and in dialectics and agreement technologies. The crucial notion of "defeater" comes from defeasible reasoning, but we did not find much else in that field that is relevant to our concerns, since we require inconsistencies (i.e., defeaters) to be resolved rather than to reason in their presence. Argumentation theory has the same defect, but provides the useful notion of an argument being *in* or *out* and argumentation schemes and dialectics apply this to subarguments and provide tool support in the Carneades framework; this idea is applied to assurance cases in Astah GSN. Eliminative argumentation uses several of these ideas and strongly advocates that the search for and elimination of defeaters should be a key element in assurance. We build on these latter elements in the Clarissa/asce platform.

As described earlier, there are three ways in which defeaters can affect the development of a case.

1. The defeater is itself defeated by a counterargument, and this can proceed to arbitrary depth. We represent this by a refutational subcase (see Section 5.3) with additional nodes that can be revealed or hidden and are color coded to represent *in* and *out* (i.e., active vs. defeated) status. The desired end state

is that defeaters pointing to elements of the main case are all *out* (and are hidden by default).

2. The defeater is sustained (possibly by a supporting subcase) and requires adjustment to the existing case, which can take two forms.

   (a) The defeater identifies a missing assumption (an "unknown known") and the response is to add this assumption to the argument as a new claim, which may have a supporting subcase. We represent this in a similar way as the first case: that is, the additional claim and subcase can be revealed or not, but here the desired end state is that the defeater is *out* and the subcase is revealed.

   (b) The defeater identifies a flaw in the argument and/or the system (in which case it is a hazard). The CLARISSA/ASCE canvas can record multiple snapshots of a case and we use this to capture the "before and after" versions of the case, with the optional display of defeaters and their subcases providing additional opportunities for dialectical examination. The default case is that defeaters are *in* and revealed in the "before" version, and *out* and hidden in the "after" version.

The desired conclusion to the development of an assurance case is that all defeaters are resolved (i.e., *out*), so that only a positive case remains, although this may contain residual doubts (see Section 6).

During development, active defeaters may be present and it can be useful to see how much of the case is thereby called into question. We believe there are two useful views. One is to ignore defeaters and evaluate soundness, full validity, and probabilistic confidence as if they were not present. This enables assessment of the state of the basic case, apart from defeaters. The other is to mark (e.g., by color coding) those defeaters that are *in* and the ordinary nodes that are *out* as a result. This allows the impact of active defeaters to be assessed. CLARISSA/ASCE mechanizes both views using manual overrides on the propagation of probabilistic confidence: confidence in a defeated claim can be left unchanged for the first view, and set to zero (or reduced from its prior value) for the second; nodes are color coded according to user-selected thresholds on probabilistic confidence. The purpose of these manipulations is not to assess justifiable confidence in a case, but to provide visualizations that help focus developers' attention on the weakest areas and the scope of their impacts (recall Figure 12).

In addition to defeaters, there are several other reasons why an assurance case may be considered imperfect or unfinished. These include minor errors (e.g., misspellings), non deductiveness, logical invalidity, unsoundness, and undeveloped subcases. Figure 8 suggests a process to examine and incrementally remove these defects. The CLARISSA/ASCE platform provides ways to assess these different flaws

and to annotate the case, some automatically and some by hand, but most of them can also be indicated by use of defeaters, as described earlier, and we prefer this method as it can be more informative and precise, and also allows more uniform assessment.

We consider its facilities for the dialectical examination of defeaters and other imperfections to be the main support for assessment of negative perspectives in the Clarissa/asce platform. In particular, the narrative associated with each defeater node can record the origins of the defeater and whether and how it has been addressed: for example, it might have been considered out of scope and no action taken, it might have led to a claim being modified to include an extra assumption, or it might have led to significant changes to the case with new nodes added to the case. Clarissa/asce also provides statistical summaries on its "Dashboard." These can include, for example, number of defeaters proposed and examined during development (as an indication of developer diligence) and during assessment (as an indication of assessor diligence).

5.3 Counterarguments and Refutations

The intended end state in development of an assurance case is a fully valid and sound argument whose defeaters have all themselves been defeated or mitigated. The defeaters, therefore, serve only a transient purpose and we could decide that there is no need to define an interpretation for arguments with defeaters as these will not be present in the finished case; in this view defeaters function rather like specialized comments and seldom need to be supported by their own subcase: the reasons for accepting or rejecting them can be recorded in the narrative associated with the defeater.

However, there are circumstances in which it is appropriate for a defeater to be supported or refuted by a subcase. An example is where different parties do not agree on the validity of a given defeater: e.g., the evaluators of a case propose a defeater and are not convinced by the developer's informal arguments against it, so each party develops a subcase to advance its point of view.

This introduces the notion of a *counterargument*, which can be useful in circumstances quite apart from contested defeaters. One such is where developers of an assurance case find it difficult to build a subcase to justify some claim A: they may be able to develop their understanding or to gain insight by attempting instead to justify the *counterclaim* $\neg A$, or by attempting to *refute* the claim A. Another is where the developers of a large case are unpersuaded by the subcase for a claim supplied by others; again, it may be useful to develop the case for a counterclaim or refutation to the given claim.

Counterarguments can take two forms: positive or refutational. A positive counterargument aims to establish a counterclaim $\neg A$ in the standard way, whereas a refutational counterargument aims to refute the original claim A.

Counterclaims and refutations are equivalent in classical logic: that is to say, verifying $\neg A$ is the same as refuting A. However, this may not be so straightforward in an assurance argument. One reason is philosophical: in an assurance case we generally prefer a positive argument to a refutational one: that is, we prefer to establish that the system is safe, rather than it is not unsafe (eliminative argumentation would take the contrary position). In logic, this preference for positive arguments corresponds to use of *intuitionistic* rather than classical reasoning: intuitionistic logic eschews the axiom for "excluded middle" (i.e., $A \vee \neg A$) so that all proofs must be of a positive, constructive nature. It might seem that this would be a good choice for assurance case arguments, but there difficulties in doing so. The interior steps of an assurance argument are axioms, and in an intuitionistic setting we must be careful that these do not accidentally introduce the excluded middle. For example, most assurance cases have essential steps that decompose over hazards and since we cannot *know* that all hazards have been identified (although we try very hard to do so), these steps have a somewhat non-intuitionistic character: instead of arguing that the system is safe in all circumstances, we are saying that we know of no circumstances where it is unsafe.

In general, it seems very onerous to insist and to check that all argument steps are intuitionistic (it is hard enough to insist that they are deductive) so we suggest it is best to conduct assurance arguments within classical logic, but with an informal preference for positive perspectives.

A second way in which counterclaims and refutations differ in assurance cases is that assurance arguments use a very limited logic: all argument steps reduce to a parent claim implied by a conjunction of subclaims or evidence (these structures are called definite clauses). Claims may involve negation, which is how we state a counterclaim, but there is no argument step that performs negation and so it is not possible to invoke a refutational setting.

To overcome this limitation, Clarissa/asce will allow a counterargument to be explicitly marked as refutational. The negative nature of refutational counterarguments will invert aspects of their interpretation. For example, in a positive counterargument, the conjunction of subclaims should entail the parent counterclaim but, in a refutational counterargument any one subclaim is sufficient to refute the parent, so a *disjunction* is the appropriate interpretation. Similarly, the *absence* of evidence can serve to refute a claim.

We are still exploring how tool support for positive and refutational counterarguments should be handled in Clarissa/asce.

6 Residual Doubts and Risks

A sound assurance case delivers confidence in its top claim. Typically, this is either an internal probabilistic assessment (recall Section 3.5) of some property related to critical failure (e.g., failure rate or time to failure), or a logical claim asserting that the system has no faults that could lead to a critical failure. In the latter case, the degree of confidence in the claim can support a conclusion concerning rate of critical failure (see Section 4). Thus, in either case, we derive strong confidence in the related claims that the system contains no critical faults and that it will suffer few critical failures.

However, the assurance case may contain residual doubts: these are potential defeaters that we are unable to eliminate or mitigate. They may be due to uncertainty in the environment: for example, the system is designed to withstand two faults and evidence indicates this is sufficient, but it is always possible for it to encounter more than that. Or they may be due to limitations of human review (e.g., human requirements tracing cannot be guaranteed to be free of error), or of automated analysis (e.g., automated static analysis may be unable to discharge some proof obligations, leading to alarms that may be false and must be reviewed by humans). If true, these defeaters may expose a hazard and hence a fault.

In assessing soundness and probabilistic confidence in an assurance case, we ignore residual doubts (recall Sections 2, 3 and 5.2): thus we achieve confidence in the absence of faults by ignoring the remaining doubts and defeaters that could possibly expose their existence! The justification for doing this is that we assess the likelihood of such faults, or more particularly the risk that they pose (i.e., the likelihood of activating them multiplied by the potential cost of the failure they may incur) to be insignificant.

Let us first consider the kinds and significance of residual doubts that may be present. Our concern is that these doubts may be sufficient to undermine the indefeasible justification of the claim C, so that we have to consider the possibility that C is false and $\neg C$ is true. The probability of this event can be conditioned on whether or not the argument supporting C is deductive or not. Thus,

$$\begin{aligned} P(\neg C) &= P(\neg C \,|\, \text{deductive}) \times P(\text{deductive}) \\ &\quad + P(\neg C \,|\, \neg\text{deductive}) \times P(\neg\text{deductive}). \end{aligned}$$

It is conservative to set any term in this equation to 1. Hence

$$\begin{aligned} P(\neg C) &\leq P(\neg C \,|\, \text{deductive}) \times 1 + 1 \times P(\neg\text{deductive}) \\ &\leq P(\neg C \,|\, \text{deductive}) + P(\neg\text{deductive}). \end{aligned} \tag{6}$$

If we have correctly identified all residual doubts to the argument, then $P(\neg\text{deductive})$ will be the cumulative probability of those doubts that concern de-

ductiveness (e.g., the unsupported "something missing here" claims), which we can write as $P(\text{deductiveness doubts})$.

That takes care of the second term in the right hand side of (6), so we now consider the first term. We can condition this on whether the deductive argument supporting C is valid or not.

$$\begin{aligned} P(\neg C \,|\, \text{deductive}) &= P(\neg C \,|\, \text{deductive} \wedge \text{valid}) \times P(\text{deductive} \wedge \text{valid}) \\ &\quad + P(\neg C \,|\, \text{deductive} \wedge \neg\text{valid}) \times P(\text{deductive} \wedge \neg\text{valid}). \end{aligned}$$

Again, It is conservative to set any term in the equation to 1. Furthermore, validity of the argument can be assured mechanically (in Assurance 2.0 it is assured by construction), so we assume any invalid argument has already been rejected; hence $P(\text{deductive} \wedge \neg\text{valid}) = 0$. Thus

$$\begin{aligned} P(\neg C \,|\, \text{deductive}) &\leq P(\neg C \,|\, \text{deductive} \wedge \text{valid}) \times 1 + 1 \times 0 \\ &\leq P(\neg C \,|\, \text{deductive} \wedge \text{valid}). \end{aligned} \tag{7}$$

Finally, we consider (7) and condition this on whether the deductively valid argument is sound or not.

$$\begin{aligned} &P(\neg C \,|\, \text{deductive} \wedge \text{valid}) = \\ &\quad P(\neg C \,|\, \text{deductive} \wedge \text{valid} \wedge \text{sound}) \times P(\text{deductive} \wedge \text{valid} \wedge \text{sound}) \\ &\qquad + P(\neg C \,|\, \text{deductive} \wedge \text{valid} \wedge \neg\text{sound}) \times P(\text{deductive} \wedge \text{valid} \wedge \neg\text{sound}). \end{aligned}$$

If the argument is deductive and valid and sound, then C is true and $\neg C$ is false. Hence, the first term on the right hand side is 0 and then, again setting some terms conservatively to 1, we have

$$\begin{aligned} P(\neg C \,|\, \text{deductive} \wedge \text{valid}) &\leq 0 \times 1 + 1 \times P(\text{deductive} \wedge \text{valid} \wedge \neg\text{sound}) \\ &\leq P(\text{deductive} \wedge \text{valid} \wedge \neg\text{sound}). \end{aligned} \tag{8}$$

Now, $P(\text{deductive} \wedge \text{valid} \wedge \neg\text{sound})$ will be the probability of all residual doubts concerning soundness of the argument, which we can partition into those concerning the evidence incorporation steps, which we write as $P(\text{evidential doubts})$, and those concerning the interior reasoning steps of the argument, which we denote $P(\text{interior doubts})$. Thus combining (6) to (8), we have

$$\begin{aligned} &P(\neg C) \leq \\ &\quad P(\text{deductiveness doubts}) + P(\text{evidential doubts}) + P(\text{interior doubts}). \end{aligned} \tag{9}$$

We have already described deductiveness doubts; an example would be the case that more than two faults afflict a system that is designed to withstand only two faults. Presumably the argument will contain a decomposition step on the number

and nature of faults, and this will not be deductive unless the "impossible" case of more than two faults is taken into account.

Evidential doubts are an interesting topic. We devote considerable attention to the assessment of evidence (recall Sections 2.2 and 3.1) so any doubts are surely already included in those assessments. However, it may be that we have systematic doubts about certain types of evidence: for example, static analysis may generate "false alarms" that must be rejected by human review. We could reduce our probabilistic assessment $P(C \mid E)$ for claims C supported by such evidence E due to doubts engendered by false alarms—but presumably this reduction will be minor, or we would do something about it. But then concern about false alarms risks being lost in the details and the cumulative impact of these doubts may not be recognized. Thus, we think there may be some merit in recording small systematic concerns about evidence as residual evidential doubts.

Interior doubts can arise for two reasons. One is that the justification for a reasoning step may be unconvincing; the other is that we may suspect that the step could be *wrong*. In the first case, we accept that the subclaims entail the parent claim but are dissatisfied with the justification provided; in the second, we have doubts that the subclaims really do entail the parent claim (and therefore distrust the justification also). The latter case may have serious consequences: if accepted, its correction may require a change to the argument step, which may propagate to additional changes in other nearby steps. Hence, we suggest that interior doubts of this kind cannot be considered merely "residual": they should be investigated and eliminated.

With these caveats, (9) provides a classification of the residual doubts that should be investigated in validation of an assurance case. We do not need precise assessments of their likelihood of occurrence, merely a sufficiently good estimate to determine if it is significant or not. We earlier, at the end of Section 3.2 and in Section 5.2, described how Clarissa/asce allows manual adjustment to the accumulated probabilistic confidence at selected nodes and color-coding of nodes according to user-selected thresholds on these values. We attach little significance to the actual values, but the visualization can help comprehend and assess the potential impact of residual doubts (again, recall Figure 12). Assessment of their significance might be refined by taking into account the potential consequences of failures that could arise if the doubt were to be realized: that is, the *risk* (product of likelihood and consequences) posed by each residual doubt. Initially, we suggest that these risks, and the threshold where they are considered significant, should be assessed and documented by best-efforts expert review. Later, we hope to develop ways of using historical experience and conservative probabilistic modeling to assist this process.

It may not be necessary to assess each residual risk individually, nor to strive for exactness: all we need to know is that the risk is well below some acceptable

threshold. We suggest it is useful to categorize residual risks into three (plus 1) levels.

Significant: an individual residual doubt 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 later eliminated or mitigated.

Minor: an individual residual doubt poses a risk that is below the threshold for concern, but it is possible that many such might cumulatively exceed the threshold. These risks may need to be managed explicitly.

Negligible: multiple residual doubts of a similar kind collectively pose a risk that is below the threshold for concern. This case may arise when the source of doubt occurs many times but is adjudged to be trivial. An example might be static analysis, where we use human review to evaluate proof obligations that the automation cannot decide.

CLARISSA/ASCE allows unmitigated defeaters to be annotated with their estimated severity, based on the scale above (4 = significant, 3 = minor, 2 = negligible, 1 = default—indicating severity not yet determined), and can report on the totals in each category. Those judged significant must be eliminated or mitigated. Those judged minor should be monitored and their cumulative impact should be assessed. If the number and cumulative severity of some category of minor risks can be kept below the threshold of concern, then we consider that category to be **Manageable**. At final assessment, the only residual doubts that remain should be those considered minor and manageable and those considered negligible.

Ideally, assessment of residual doubts should consider the faults they might precipitate, the failures those faults might cause, together with the frequency of their occurrence and severity of their outcomes (i.e., their risk), all in the worst case. This is feasible for some doubts, such as those concerning the maximum number of sensor failures that might occur. For others it seems less so. Suppose for example, that we have residual doubts about static analysis because it generates many proof obligations that cannot be discharged automatically and require human review. Here, the best we can do might be to collect statistics on human reliability for this task in an effort to constrain the potential frequency of failure, since the potential consequences seem very hard to ascertain. If we assume the worst case, that all human reviews are in error and these cases represent real bugs, then each bug might be encountered very rarely, but collectively they could arise unacceptably often. This would be an example where each residual risk is minor but their aggregate is not manageable, and action must therefore be taken to eliminate or further mitigate their cumulative

impact (for example, by using a better theorem prover or diverse means of analysis) so that it can be assigned to the manageable or negligible categories.[21]

[21] Observe that autonomous systems, such as self-driving cars, can exhibit this behavior, but for different reasons. For example, the vision system of the car may misinterpret some scenes; each reason for misinterpretation may apply very rarely, but in total they cause the vision system to fail quite often, suggesting a minor but unmanaged residual doubt. On the other hand, misinterpretations may have little consequence if they are localized, for the preceding and following frames will be interpreted correctly and the misinterpreted frame will be but a negligible "blip" [71].

7 Sentencing Statement

Assurance cases generally serve a single purpose: to support the decision whether a system may be deployed. However, they may be evaluated by reviewers occupying different roles within the processes leading up to that decision. For example, an auditor might be focused on process and regulatory compliance in the construction of the system and its assurance case, whereas a technical evaluator will want to gain a deep understanding of how the system works and how its safety is ensured. It is important that an assurance case and its supporting tools provide information and means for comprehending it that support these diverse viewpoints.

Graydon contends [54] that cases developed to support one perspective or "vision" may be misunderstood by those with different viewpoints. We are not so pessimistic but we believe that assurance cases and their tools must support communication, so that diverse reviewers can develop confidence and consensus in their understanding of the system and its case, and reasoning, so that they can test their understanding and can also challenge the case. We have described several mechanisms by which Assurance 2.0 and Clarissa/asce hope to achieve these goals. These include a limited number of basic building blocks so that arguments are readily interpreted, strict criteria on what constitutes a soundly reasoned argument step or evidence of adequate weight, tools for inspecting and navigating arguments and for evaluating them from both logical and probabilistic perspectives, and methods and tools for challenging them by means of defeaters.

The final assessment of an assurance case and the corresponding decision on system deployment are serious matters; assessors cannot merely sample the case with its claims, argument, and evidence and then "sign off" on the top level claim. We expect them, assisted by those in supporting roles, to avail themselves of the tools and intellectual structures mentioned above to actively explore both the positive and negative aspects of the case and to challenge their understanding by proposing defeaters and exploring other questions.

The evaluators' task should be informed by and conclude with a "sentencing statement" that indicates their diligent execution of these tasks and declares their understanding of the system and its context, the key points of its architecture and design, its hazards and their mitigations, the soundness and probabilistic confidence of the assurance argument with its supporting theories, models, and evidence assembly, and the relationship of the top claim to acceptance criteria for deployment.

Possible headings for a sentencing statement could be the following.

"*On the basis of this assurance case and an examination of other relevant documentation, I judge the proposed system to be adequately safe/unsafe...*" (or "*the case is insufficient to make a judgment*").

"*I believe my judgment of this case is sound and valid because...*

- *I understand the context and criticality of the decision...*
- *I understand the system...*
- *I find a clear thread of reasoning from evidence to claim...*
- *The evidence provided is sufficient/insufficient to support evidence-based decision making...*
- *I have actively explored doubts...*
- *I have also identified what evidence would be capable of disproving...*
- *I have considered and addressed biases and fallacies..."*

CLARISSA is experimenting with the decision support needed to assist evaluators make and substantiate these judgments, with explicit links from Assurance 2.0 concepts and CLARISSA/ASCE functions to the bullet points above.

8 Summary and Conclusion

We have explored methods for gaining and assessing confidence in assurance cases based on Assurance 2.0 and its automated assistance with Clarissa/asce. Here, we summarize these methods and provide brief conclusions. We do not provide references here: they can be found in the earlier sections specific to each topic.

Assurance 2.0 is more rigorous and demanding than earlier treatments of assurance cases, but we argue that this simplifies the development and assessment of cases because issues that were previously treated in an *ad hoc* manner and subject to contention and challenge are now made explicit and treated systematically.

In particular, we are explicit that the goal of Assurance 2.0 is indefeasible justification, meaning we must be confident there are no overlooked or unresolved doubts that could change evaluation of the case. A consequence of this is a strong preference in Assurance 2.0 for argument steps to be deductive, and for steps that are merely inductive to acknowledge this and to explicitly manage the doubts thereby admitted. Similarly, evidence in Assurance 2.0 is weighed very deliberately using confirmation measures and we distinguish carefully between facts established by the evidence (claims about "something measured") and inferences drawn from it (claims about "something useful").

These rigorous requirements and other supporting constraints enable straightforward evaluation of the positive criterion for assurance case arguments that we call soundness. Note that we say the evaluation is straightforward, meaning it is clear what must be done, not that it is easy: it requires sophisticated technical judgment, but this judgment can focus on technical issues without being distracted by unmanaged doubts and contested interpretations.

Another way in which Assurance 2.0 simplifies the assessment of assurance cases is by being clear what is developed within the assurance argument and what is referenced and integrated by it through application of external theories and models. The overall case will include all necessary theories, models, and evidence assemblies but the detail of these items is excluded from the argument, not because they are unimportant but because they each have specialized form and content and are therefore well suited to presentation and assessment by scientific and engineering methods traditional to their fields. The assurance case argument, on the other hand, must integrate these disparate items and its structured, logical form is tailored for that function and dedicated to it.

Whereas traditional assurance case arguments seem almost arbitrary in their structure, so that reviewers do not know what to expect, arguments in Assurance 2.0 generally follow a systematic pattern where general claims at the upper level are refined into more precise claims using concretion steps, then substitution steps are used to elaborate these claims about high level models into claims about low level models and their implementations, and these lowest level claims are discharged by

evidence. Substitution steps relate a claim about one model to a possibly different claim about a possibly different model, although either the claim or the model is typically held constant. Along the way, the argument may divide into subcases using decomposition or calculation steps that enumerate a claim over some structure (e.g., over components, requirements, hazards, etc.) or split the conjuncts of a compound claim. This structure may recurse within subcases. For example, we may have a lower-level claim that software development conforms to standards, and will use a concretion step to refine this to a specific standard and then develop the rest of the subcase in the manner just described.

Their systematic structure allows argument steps in Assurance 2.0 to be limited to just the five basic forms mentioned above (concretion, substitution, evidence incorporation, decomposition, and calculation). These each have a precise and obvious purpose and it is generally straightforward to decide which to use at each argument step (see the "helping hand" visual mnemonic of Figure 2). Each type of argument step has side-claims that ensure it is used appropriately and with a sound (e.g., deductive) justification.

Soundness is the most fundamental valuation for an assurance case: it tells us that the argument and its evidence truly do support the top claim, but it does not tell us how strongly they do so. We therefore define a method for probabilistic valuation that does this and Clarissa/asce can color argument nodes accordingly to support visual comprehension of the weak and strong parts of an argument. We apply probabilistic valuation only to sound assurance case arguments, and this eliminates (although we need to confirm this claim) the vulnerabilities that have been found in other probabilistic forms of assessment. Our probabilistic methods are conservative and the numerical valuations are of limited absolute value, but they do serve to explore the risk of residual doubts and the relative strengths of different cases for the same system. This allows rational tradeoffs of effort and cost versus confidence, which is needed in developing graduated forms of assurance for different levels of risk, as exemplified by the SILs (Software Integrity Levels) of IEC 61508, the ASILs (Automotive SILs) of SAE 26262, and the DALs (Design Assurance Levels) of DO-178C.

While building a forceful positive case, the developers of an assurance case must guard against confirmation bias. This can be assisted by vigorous and active exploration of challenges to, and doubts about, the case during its construction. In Assurance 2.0, doubts are refined and recorded as defeaters, which are nodes in the graphical representation of the assurance case argument that explicitly challenge other nodes and that may have their own subcases to validate or refute them. Valid defeaters require revision to the assurance case and possibly the system itself.

Clarissa/asce can selectively reveal or hide defeaters and their subcases and can display the changes made in response to valid defeaters. In addition to guarding against confirmation bias, the record of doubts explored as defeaters assists

assessors of the case. Assessors begin their work by gaining an understanding of the case, perhaps by posing "what-if" questions, and then probing it more deeply for weak spots and oversights. When previously examined defeaters are recorded as part of the case, assessors may find that their own questions and doubts have been anticipated and answered, thereby streamlining their task and also enabling a constructive, dialectical examination of the case by "eliminative argumentation."

All identified defeaters must be examined and resolved. However, a conscious decision may be made to accept some as residual doubts. For example, a subcase that uses testing to justify absence of runtime exceptions may have residual doubt due to incompleteness of testing. The risks posed by such doubts (i.e., the likelihood that they may be falsified, and the potential impact and cost if they are so) must be assessed and only those considered tolerable (technically, those considered minor and manageable, and those considered negligible, see Section 6) can be allowed to remain as residual risks: others must be eliminated or mitigated by revisions to the argument or the system. The probabilistic valuation of CLARISSA/ASCE can be used to help visualize the potential impact of residual doubts on the overall argument.

In conclusion, Assurance 2.0 assesses confidence in an assurance case by considering both positive and negative aspects. The positive aspects are logical soundness and (optionally) a probabilistic valuation; the negative aspects are vigorous exploration of potential defeaters, and careful evaluation of all residual doubts. Assessors should not simply inspect and "sign off" on an assurance case; we expect them to actively explore and question both its positive and negative aspects and to conclude with a "sentencing statement" that declares their understanding of the system and its context, its hazards and their mitigations, the key points of its architecture and design, the soundness and probabilistic confidence of the assurance argument with its supporting theories, models, and evidence assemblies, their defeaters and residual doubts, and the relationship of the top claim to acceptance criteria. CLARISSA/ASCE provides assistance in these evaluations and together they provide a comprehensive and rigorous assessment for assurance cases that should be independent of the vagaries of individual assessors.

These ideas were developed and explored during construction of CLARISSA/ASCE using a variety of examples and represent a work in progress. We plan to develop worked examples to will support more detailed exploration and exposition that will be published in a companion report. In particular, we wish to explore what forms of guidance and automated support are most useful for developers and assessors.

Acknowledgments. The work described here was developed in partnership with other members of the CLARISSA project, notably Kevin Driscoll and Brendan Hall of Honeywell, Gopal Gupta of UT Dallas, and Kate Netkachova of Adelard. Separately, N. Shankar of SRI provided valuable criticism.

This material is based upon work supported by the Air Force Research Laboratory (AFRL) and DARPA under Contract No. FA8750-20-C-0512. 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 Air Force Research Laboratory (AFRL) and DARPA.

References

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

[2] *ASCAD: Adelard Safety Case Development Manual*. Adelard LLP, London, UK, 1998. Available from https://www.adelard.com/resources/ascad.html.

[3] ASCE. *ASCE home page*. https://www.adelard.com/asce/choosing-asce/index.

[4] Astah. *Astah GSN home page*. http://astah.net/editions/gsn.

[5] ASTM. *Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions*. ASTM (American Society for Testing and Materials), 2017. ASTM F3269-17.

[6] David Atkinson. Confirmation and justification. A commentary on Shogenji's measure. *Synthese*, 184(1):49–61, January 2012.

[7] ATSB. *In-Flight Upset 154 km West of Learmonth, WA, 7 October 2008, VH-QPA, Airbus A330-303*. Australian Transport Safety Bureau, December 2011. Aviation Occurrence Investigation AO-2008-070, Final.

[8] 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, SCSC, Bristol, UK, February 2013.

[9] Francis Bacon. *The Novum Organon: Or, A True Guide to the Interpretation of Nature*. Oxford University Press, 1855. English translation by G. W. Kitchin; the original Latin is from 1620).

[10] Prasanta S. Bandyopadhyay, Gordon G. Brittan, and Mark L. Taper. *Belief, Evidence, and Uncertainty: Problems of Epistemic Inference*. Springer, 2016.

[11] Maya Bar-Hillel. The base-rate fallacy in probability judgments. *Acta Psychologica*, 44(3):211–233, 1980.

[12] Antonia Bertolino and Lorenzo Strigini. Assessing the risk due to software faults: Estimates of failure rate vs. evidence of perfection. *Software Testing, Verification and Reliability*, 8(3):156–166, 1998.

[13] Devesh Bhatt, Anitha Murugesan, Brendan Hall, Hao Ren, and Yogananda Jeppu. The CLEAR way to transparent formal methods. In *4th Workshop on Formal Integrated Development Environment*, Oxford, UK, July 2018. Absent from published proceedings but available at http://47.52.94.58/Floc2018/FLoC2018-pages/volume43.html.

[14] Devesh Bhatt, Hao Ren, Anitha Murugesan, Jason Biatek, Srivatsan Varadarajan, and Natarajan Shankar. Requirements-driven model checking and test generation for comprehensive verification. In *NASA Formal Methods Symposium*, Volume 13260 of Springer-Verlag *Lecture Notes in Computer Science*, pages 576–596, Springer-Verlag, Pasadena, CA, May 2022.

[15] Peter Bishop. Does software have to be ultra reliable in safety critical systems? In SafeComp [96], pages 118–129.

[16] Peter Bishop and Robin Bloomfield. A conservative theory for long-term reliability-growth prediction. *IEEE Transactions on Reliability*, 45(4):550–560, 1996.

[17] Peter Bishop, Andrey Povyakalo, and Lorenzo Strigini. Bootstrapping confidence in future safety based on past safe operation. `arXiv:2110.10718`, October 2021.

[18] Peter G. Bishop and Robin E. Bloomfield. Worst case reliability prediction based on a prior estimate of residual defects. In *13th International Symposium on Software Reliability Engineering (ISSRE)*, pages 295–303, IEEE, Annapolis, MD, November 2002.

[19] J. Anthony Blair. What is informal logic? In Frans H. van Eemeren and Bart Garssen, editors, *Reflections on Theoretical Issues in Argumentation Theory*, volume 28 of *The Argumentation Library*, pages 27–42. Springer, 2015.

[20] Robin Bloomfield and Sofia Guerra. Process modelling to support dependability arguments. In *The International Conference on Dependable Systems and Networks*, pages 113–122, IEEE Computer Society, Bethesda, MD, June 2002.

[21] 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.

[22] 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. Final draft available as `arXiv:2004.10474`.

[23] Boeing. *Statistical Summary of Commercial Jet Aircraft Accidents, Worldwide Operations, 1959–20119*. Boeing Commercial Airplane Group, Seattle, WA, July 2020. Published annually by Boeing Airplane Safety Engineering, available at http://www.boeing.com/news/techissues/pdf/statsum.pdf.

[24] Luc Bovens and Stephan Hartmann. *Bayesian Epistemology*. Oxford University Press, 2003.

[25] Ricky W. Butler and George B. Finelli. The infeasibility of experimental quantification of life-critical software reliability. In *SIGSOFT '91: Software for Critical Systems*, pages 66–91, New Orleans, LA, December 1991. Published as ACM SIGSOFT Engineering Notes, Volume 16, Number 5.

[26] Martin Caminada. On the issue of reinstatement in argumentation. In *European Workshop on Logics in Artificial Intelligence*. pages 111–123, Springer, 2006.

[27] António Casimiro et al., editors. *Computer Safety, Reliability, and Security (SAFECOMP 2020)*, Volume 12234 of Springer *Lecture Notes in Computer Science*, Lisbon, Portugal, September 2020. Springer.

[28] Carlos Iván Chesñevar, Ana Gabriela Maguitman, and Ronald Prescott Loui. Logical models of argument. *ACM Computing Surveys*, 32(4):337–383, 2000.

[29] John Joseph Chilenski and Steven P. Miller. Applicability of modified condition/decision coverage to software testing. Issued for information under FAA memorandum ANM-106N:93-20, August 1993.

[30] John Joseph Chilenski and Steven P. Miller. Applicability of modified condition/decision coverage to software testing. *IEE/BCS Software Engineering Journal*, 9(5):193–200, September 1994.

[31] Thomas Chowdhury, Alan Wassyng, Richard F. Paige, and Mark Lawford. Systematic evaluation of (safety) assurance cases. In Casimiro et al. [27], pages 18–33.

[32] Andy Clark. Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3):181–204, 2013.

[33] Vincenzo Crupi, Branden Fitelson, and Katya Tentori. Probability, confirmation, and the conjunction fallacy. *Thinking & Reasoning*, 14(2):182–199, 2008.

[34] P. Allen Currit, Michael Dyer, and Harlan D. Mills. Certifying the reliability of software. *IEEE Transactions on Software Engineering*, SE-12(1):3–11, January 1986.

[35] Ewen Denney, Ganesh Pai, and Ibrahim Habli. Towards measurement of confidence in safety cases. In *Fifth International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 380–383, IEEE Computer Society, Banff, Canada, September 2011.

[36] Simon Diemert and Jeff Joyce. Eliminative argumentation for arguing system safety—a practitioner's experience. In *IEEE Systems Conference*, 2020.

[37] Kevin Driscoll. Real system failures. NASA Dashlink, November 2012. https://c3.nasa.gov/dashlink/resources/624/.

[38] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

[39] EASA. *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes, CS-25 and AMC-25*. The European Aviation Safety Agency (EASA), June 2016. Amendment 18; available at https://www.easa.europa.eu/document-library/certification-specifications.

[40] Jordan Ellenberg. *How Not to be Wrong: The Power of Mathematical Thinking*. Penguin, 2015.

[41] FAA. *System Design and Analysis*. Federal Aviation Administration, June 21, 1988. Advisory Circular 25.1309-1A.

[42] FAA. Understanding the overarching properties: First steps. Technical Report DOT/FAA/TC-xx/xx, FAA, September 2018.

[43] Norman Fenton and Martin Neil. *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press, 2012.

[44] Karl Friston. The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2):127, 2010.

[45] Karl Friston. The history of the future of the Bayesian brain. *NeuroImage*, 62(2):1230–1233, 2012.

[46] Brian R. Gaines. Fuzzy and probability uncertainty logics. *Information and Control*, 38(2):154–169, 1978.

[47] Michael Gelfond and Yulia Kahl. *Knowledge Representation, Reasoning, and The Design of Intelligent Agents: The Answer-Set Programming Approach.* Cambridge University Press, 2014.

[48] I. J. Good. Weight of evidence: A brief survey. In J.M Bernardo et al., editors, *Bayesian Statistics 2: Proceedings of the Second Valencia International Meeting*, pages 249–270, Valencia, Spain, September 1983.

[49] Irving John Good. The white shoe is a red herring. *The British Journal for the Philosophy of Science*, 17(4):322, 1967.

[50] John B. Goodenough, Charles B. Weinstock, and Ari Z. Klein. Eliminative induction: A basis for arguing system confidence. In *Proceedings International Conference on Software Engineering, New Ideas and Emerging Results*, pages 1161–1164, IEEE Computer Society, San Francisco, CA, May 2013.

[51] John B. Goodenough, Charles B. Weinstock, and Ari Z. Klein. Eliminative argumentation: A basis for arguing confidence in system properties. Technical Report CMU/SEI-2014-TR-013, Carnegie Mellon Software Engineering Institute, Pittsburgh, PA, 2014.

[52] Thomas F. Gordon, Henry Prakken, and Douglas Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10):875–896, 2007.

[53] Trudy Govier. *Problems in Argument Analysis and Evaluation*, volume 5 of *Studies of Argumentation in Pragmatics and Discourse Analysis.* De Gruyter, 1987.

[54] Patrick J. Graydon. The many conflicting visions of "safety case". In *47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 103–104, June 2017.

[55] 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.

[56] 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.

[57] William S. Greenwell, John C. Knight, C. Michael Holloway, and Jacob J. Pease. A taxonomy of fallacies in system safety arguments. In *Proceedings of the 24th International System Safety Conference*, Albuquerque, NM, 2006.

[58] Richard L. Gregory. *Eye and Brain: The Psychology of Seeing*. Princeton University Press, 5th edition, 1997.

[59] Leo Groarke. Deductivism within pragma-dialectics. *Argumentation*, 13(1):1–16, 1999.

[60] Leo Groarke. Informal logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2017 edition, 2017.

[61] Charles Haddon-Cave. The Nimrod Review: An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006. Report, The Stationery Office, London, UK, October 2009. Available at http://www.official-documents.gov.uk/document/hc0809/hc10/1025/1025.pdf.

[62] Allan Hájek and James M. Joyce. Confirmation. In Martin Curd and Stathis Psillos, editors, *The Routledge companion to philosophy of science*, chapter 14, pages 115–129. Routledge, 2008.

[63] Steve Hanks and Drew McDermott. Nonmonotonic logic and temporal projection. *Artificial intelligence*, 33(3):379–412, 1987.

[64] James Hawthorne. Bayesian induction IS eliminative induction. *Philosophical Topics*, 21(1):99–138, 1993.

[65] 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.

[66] Carl G. Hempel. Studies in the logic of confirmation. *Mind*, 54(213):1–26, 1945.

[67] C. Michael Holloway. Understanding assurance cases: An educational series in five parts. Informal report, NASA Langley Research Center, 2015.

[68] C. Michael Holloway. Understanding the overarching properties. Technical Memorandum NASA/TM-2019-220292, NASA Langley Research Center, Hampton VA, July 2019.

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

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

[71] Susmit Jha, John Rushby, and N. Shankar. Model-centered assurance for autonomous systems. In Casimiro et al. [27], pages 228–243.

[72] Martin L. Jönsson and Tomoji Shogenji. A unified account of the conjunction fallacy by coherence. *Synthese*, 196(1):221–237, 2019.

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

[74] Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[75] Daniel Kahneman, Olivier Sibony, and Cass R. Sunstein. *Noise: A Flaw in Human Judgment*. Little Brown, 2021.

[76] David C. Knill and Alexandre Pouget. The Bayesian brain: The role of uncertainty in neural coding and computation. *TRENDS in Neurosciences*, 27(12):712–719, 2004.

[77] Robert Koons. Defeasible reasoning. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2017 edition, 2017.

[78] J. C. Laprie. Dependable computing and fault tolerance: Concepts and terminology. In *Fault Tolerant Computing Symposium 15*, pages 2–11, IEEE Computer Society, Ann Arbor, MI, June 1985.

[79] Nancy Leveson. The use of safety cases in certification and regulation. *Journal of System Safety*, 47(6):1–5, 2011.

[80] Bev Littlewood and Andrey Povyakalo. Conservative bounds for the pfd of a 1-out-of-2 software-based system based on an assessor's subjective probability of "not worse than independence". *IEEE Transactions on Software Engineering*, 39(12):1641–1653, 2013.

[81] Bev Littlewood and Andrey Povyakalo. Conservative reasoning about the probability of failure on demand of a 1-out-of-2 software-based system in which one channel is "possibly perfect". *IEEE Transactions on Software Engineering*, 39(11):1521–1530, 2013.

[82] Bev Littlewood and John Rushby. Reasoning about the reliability of diverse two-channel systems in which one channel is "possibly perfect". *IEEE Transactions on Software Engineering*, 38(5):1178–1194, September/October 2012.

[83] 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.

[84] John McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1), 1980.

[85] Jean Nicod. *Foundations of Geometry and Induction*. The International Library of Philosophy. Routledge, 1930.

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

[87] Sascha Ossowski, editor. *Agreement Technologies*. Law, Governance and Technology Series, vol. 8. Springer, 2013.

[88] David L. Parnas, A. John van Schouwen, and Shu Po Kwan. Evaluation of safety-critical software. *Communications of the ACM*, 33(6):636–648, June 1990.

[89] John L. Pollock. Defeasible reasoning. *Cognitive Science*, 11:481–518, 1987.

[90] Karl Popper. *The Logic of Scientific Discovery*. Routledge, 2014. First published in German 1934, English 1959.

[91] Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999.

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

[93] 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.

[94] John Rushby. The indefeasibility criterion for assurance cases. In *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.

[95] John Rushby, Xidong Xu, Murali Rangarajan, and Thomas L. Weaver. Understanding and evaluating assurance cases. NASA Contractor Report NASA/CR-2015-218802, NASA Langley Research Center, July 2015.

[96] SafeComp *2013: Proceedings of the 32nd International Conference on Computer Safety, Reliability, and Security*, Volume 8153 of Springer-Verlag *Lecture Notes in Computer Science*, Toulouse, France, September 2013. Springer-Verlag.

[97] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[98] N. Shankar et al. Descert—design for certification, phase 1 report. `arXiv:2203.15178`, March 2022.

[99] Tomoji Shogenji. The degree of epistemic justification and the conjunction fallacy. *Synthese*, 184(1):29–48, January 2012.

[100] Lorenzo Strigini and Andrey Povyakalo. Software fault-freeness and reliability predictions. In SafeComp [96], pages 106–117.

[101] Toshinori Takai and Hiroyuki Kido. A supplemental notation of GSN to deal with changes of assurance cases. In *4th International Workshop on Open Systems Dependability (WOSD)*, pages 461–466, IEEE International Symposium on Software Reliability Engineering Workshops, Naples, Italy, November 2014.

[102] Katya Tentori, Vincenzo Crupi, Nicolao Bonini, and Daniel Osherson. Comparison of confirmation measures. *Cognition*, 103:107–119, 2007.

[103] Amos Tversky and Daniel Kahneman. Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological review*, 90(4):293–315, 1983.

[104] Susan Vineberg. Eliminative induction and Bayesian confirmation theory. *Canadian Journal of Philosophy*, 26(2):257–266, 1996.

[105] Douglas Walton, Christopher Reed, and Fabrizio Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

[106] Douglas N. Walton. *Argumentation Schemes for Presumptive Reasoning*. Psychology Press, 1996.

[107] Charles B. Weinstock, John B. Goodenough, and Ari Z. Klein. Measuring assurance case confidence using Baconian probabilities. In *1st International Workshop on Assurance Cases for Software-Intensive Systems (ASSURE)*, San Francisco, CA, May 2013.

[108] Tangming Yuan and Tim Kelly. Argument schemes in computer system safety engineering. *Informal Logic*, 31(2):89–109, 2011.

[109] Tangming Yuan, Tim Kelly, and Tianhua Xu. Computer-assisted safety argument review—a dialectics approach. *Argument & Computation*, 6(2):130–148, 2015.

[110] Xingyu Zhao, Bev Littlewood, Andrey Povyakalo, Lorenzo Strigini, and David Wright. Modeling the probability of failure on demand (pfd) of a 1-out-of-2 system in which one channel is "quasi-perfect". In *Reliability Engineering and System Safety*. pages 230–245, Elsevier Ltd, February 2017.

[111] Xingyu Zhao, Valentin Robu, David Flynn, Kizito Salako, and Lorenzo Strigini. Assessing the safety and reliability of autonomous vehicles from road testing. In *Twentieth International Symposium on Software Reliability Engineering, ISSRE '19*, IEEE Computer Society, Berlin, Germany, October 2019.

[112] Slavoj Žižek. Philosophy, the "unknown knowns," and the public use of reason. *Topoi*, 25(1-2):137–142, 2006.