Appears in postproceedings of an NII Shonan Workshop published by Springer in July 2020, Yamine Ait Ameur, Dominique Mery and Shin Nakajima eds., pp 259-279

The Indefeasibility Criterion for Assurance Cases

John Rushby

Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025 USA

Abstract. Ideally, assurance enables us to know that our system is safe, or possesses other attributes we care about. But full knowledge requires omniscience, and the best we humans can achieve is well-justified belief. So what justification should be considered adequate for a belief in safety? We adopt a criterion from epistemology and argue that assurance should be "indefeasible," meaning that we must be so sure that all doubts and objections have been attended to that there is no (or, more realistically, we cannot imagine any) new information that would cause us to change our evaluation.
We explore application of this criterion to the interpretation and evaluation of assurance cases and derive a strict but practical characterization for a sound assurance case.

1 Introduction

One widely quoted definition for a safety case comes from the UK Ministry of Defence [1]:

> "A safety case is a structured argument, supported by a body of evidence that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given operating environment."

An *assurance case* is simply the generalization of a safety case to properties other than safety (e.g., security) so, *mutatis mutandis*, we can accept this definition as a basis for further consideration.

Key concepts that we can extract from the definition are that an assurance case uses a *structured argument* to derive a *claim* or goal (e.g., "safe for a given application in a given operating environment") from a body of *evidence*. The central requirement is for the overall case to be "compelling, comprehensible and valid"; here, "compelling" and "comprehensible" seem to be subjective judgments, so I will focus on the notion of a "valid" case and, for reasons I will explain later, I prefer to use the term *sound* as the overall criterion.

There are two ways one might seek a definition of "sound" that is appropriate to assurance cases: one would be to fix the notion of "structured argument" (e.g., as classical deduction, or as defeasible reasoning, or as Toulmin-style argumentation) and adopt or adapt its notion of soundness; the other is to look for a larger context in which a suitable form of soundness can be defined that is independent of the style of argument employed. I will pursue the second course and, in Section 3, I will argue for the *indefeasibility criterion* from epistemology. I will apply this to assurance case arguments in Section 4 and argue for its feasibility in Section 5. Then, in Section 6, I will consider how the indefeasibility criterion applies in the evaluation of assurance case arguments.

Surrounding these sections on assurance are sections that relate assurance to system behavior and to certification. The top-level claim of an assurance case will generally state that the system satisfies some critical property such as safety or security. Section 2 relates confidence in the case, interpreted as a subjective probabilistic assessment that its claim is true, to the likelihood that critical system failures will be suitably rare—which is the basis for certification. Section 7 considers probabilistic assessments of an assurance case in support of this process. Section 8 presents brief conclusions and speculates about the future.

2 Assurance, and Confidence in Freedom from Failure

A sound assurance case should surely allow—or even persuade—us to accept that its claim is true. There are many different words that could be used to describe the resulting mental state: we could come to *know* that the claim is true, or we could *believe* it, or have *confidence* in it. I will use the term "belief" for this mental state and will use "confidence" to refer to the strength of that belief.

So an assurance case gives us confidence in the belief that its claim is true. For a system-level assurance case, the top-level claim is generally some critical property such as safety (i.e., a statement that nothing really bad will happen), but we may also have "functional" claims that the system does what is intended (so it is useful as well as safe). A system-level assurance case will often be decomposed into subsidiary cases for its subsystems, and the functional and critical claims will likewise be decomposed. At some point in the subsystem decomposition, we reach "widgets" where the claims are no longer decomposed and we simply demand that the subsystem satisfies its claims.

Software assurance cases are generally like this: software is regarded as a widget and its local claim is correctness with respect to functional requirements, which then ensure the critical requirements of its parent system; of course there is a separate assurance task to ensure that the functional requirements really do ensure the critical requirements and hence the top-level claim. This division of responsibility is seen most clearly and explicitly in the guidelines for commercial aircraft certification, where DO-178C [2] focuses on correctness of the software and ARP 4754A [3] provides safety assurance for its requirements. If we assume the requirements are good and focus strictly on software assurance, any departure from correctness constitutes a *fault*, so a software assurance case gives us confidence that the software is fault-free. Confidence can be expressed numerically as a subjective probability so, in principle, a software assurance case should allow us to assess a probability p_{nf} that represents our degree of confidence that the software is free of faults (or $\underline{n}$on$\underline{f}$aulty).

What we really care about is not freedom from faults but absence of failure. However, software can fail only if it encounters a fault, so software that is, with high probability, free of faults will also be free of failures, with high probability. More particularly, the probability of surviving n independent demands without failure, denoted $p_{srv}(n)$, is given by

$$p_{srv}(n) = p_{nf} + (1 - p_{nf}) \times (1 - p_{F|f})^n, \tag{1}$$

where $p_{F|f}$ is the probability that the software $\underline{\text{F}}$ails, if $\underline{\text{f}}$aulty.[1] A suitably large n can represent the system-level assurance goal. For example, "catastrophic failure conditions" in commercial aircraft ("those which would prevent continued safe flight and landing") must be "so unlikely that they are not anticipated to occur during the entire operational life of all airplanes of one type" [5]. If we regard a complete flight as a demand, then "the entire operational life of all airplanes of one type" can be satisfied with n in the range 10^8 to 10^9.

The first term of (1) establishes a lower bound for $p_{srv}(n)$ that is independent of n. Thus, if assurance gives us the confidence to assess, say, $p_{nf} \geq 0.9$ (or whatever threshold is meant by "not anticipated to occur") then it seems we have sufficient confidence to certify the aircraft software. However, we also need to consider the case where the software does have faults.[2] We need confidence that the system will not suffer a critical failure despite those faults, and this means we need to be sure that the second term in (1) will be well above zero even though it decays exponentially.

This confidence could come from prior failure-free operation. Calculating the overall $p_{srv}(n)$ can then be posed as a problem in Bayesian inference: we have assessed a value for p_{nf}, 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 $p_{F|f}$, which may be difficult to obtain and difficult to justify. However, Strigini and Povyakalo [4] show there is a distribution that delivers *provably worst-case* predictions; using this, we can make predictions that are guaranteed to be conservative, given only p_{nf}, r, and n. For values of p_{nf} above 0.9, their results show that $p_{srv}(n)$ is well above the floor given by p_{nf}, provided $r > \frac{n}{10}$.

Thus, in combination with prior failure-free experience (which is gained incrementally, initially from tests and test flights, and later from regular operation), an assessment $p_{nf} > 0.9$ provides adequate assurance for extremely low rates of critical failure, and hence for certification. I have presented this analysis in terms of software (where the top claim is correctness) but, with appropriate adjustments to terminology and probabilities, it applies to assurance of systems and properties in general, even autonomous systems. (It also applies to subsystems; one way to mitigate faults and failures in low-assurance subsystems is to

[1] I am omitting many details here, such as the interpretation of subjective probabilities, and the difference between aleatoric and epistemic uncertainty. The model and analysis described here are due to Strigini and Povyakalo [4], who give a comprehensive account.

[2] Imagine using this procedure to provide assurance for multiple aircraft types; if $p_{nf} = 0.9$ and we assure 10 types, then one of them may be expected to have faults.

locate them within a suitable architecture where they can be buttressed with high-assurance monitors or other mechanisms for fault tolerance; Littlewood and Rushby [6] analyze these cases.) This analysis is the only one I know that provides a credible scientific account for how assurance and certification actually work in practice. Those who reject probabilistic reasoning for critical properties need to provide a comparably credible account based on their preferred foundations.

Failures of the assurance process do not invalidate this analysis. For example, the Fukushima nuclear meltdown used inappropriate assessment of hazards, and the Boeing 737Max MCAS appears to have violated every principle and process of safety engineering and assurance. Sections 3 to 6 consider how to structure and evaluate an assurance case so that aberrations such as Fukushima and the 737Max MCAS are reliably detected and rejected. In the remainder of this section and in Section 7, I focus on how a probabilistic assessment such as $p_{nf} \geq 0.9$ can be derived from a successful assurance case.

One approach would be to give a probabilistic interpretation to the argument of the case. It is certainly reasonable to assess evidence (i.e., the leaves of the argument) probabilistically, and I will discuss this in Section 4. However, a fully probabilistic interpretation requires the interior of the argument to be treated this way, too, which will take us into probability logics or their alternatives such as fuzzy set "possibility theory" or the Dempster-Shafer "theory of evidence." Unfortunately, despite much research, there is no generally accepted interpretation for the combination of logic and probability. Furthermore, it is not clear that any proposed interpretations deliver reliable conclusions for assurance case arguments. Graydon and Holloway [7,8] examined 12 proposals for using probabilistic methods to quantify confidence in assurance case arguments: 5 based on Bayesian Belief Networks (BBNs), 5 based on Dempster-Shafer or similar forms of evidential reasoning, and 2 using other methods. By perturbing the original authors' own examples, they showed that all the proposed methods can deliver implausible results.

An alternative approach is to revert to the original idea that the overall case should be sound in some suitable sense and the probabilistic assessment is a measure of our confidence in that soundness. So now we need a suitable interpretation for the soundness of an assurance case. The intent is that a sound case should lead us, collectively, to believe its claim, and that claim should be true. The means by which the case induces belief is by providing *justification*, so it looks as if soundness should involve these three notions: belief, justification, and truth. As it happens, epistemology, the branch of philosophy concerned with knowledge, has traditionally (since Plato) combined these three terms to interpret *knowledge* as Justified True Belief (JTB), so we may be able to draw on epistemology for a suitable characterization of a sound assurance case. This idea is developed and explored in the following four sections; we then return, in Section 7, to consider probabilistic assessment of confidence in the resulting process.

3 Epistemology and the Indefeasibility Criterion

Few philosophers today accept the basic version of JTB due to what are called "Gettier cases"; these are named after Edmund Gettier who described two such cases in 1963 [9]. Gettier's is the most widely cited modern work in epistemology with over 3,000 citations, many of which introduce new or variant cases. However, these all follow the same pattern, which had previously been exemplified by the "stopped clock case" introduced by Bertrand Russell in 1912 [10, p. 170]:

> Alice sees a clock that reads two o'clock, and believes that the time is two o'clock. It is in fact two o'clock. However, unknown to Alice, the clock she is looking at stopped exactly twelve hours ago.

The general pattern in these cases is "bad luck" followed by "good luck"; in the stopped clock case, Alice believes that it is two o'clock and her belief is justified because she has looked at a clock. But the clock is stopped ("bad luck") so her belief could well be false; however, the clock stopped *exactly* twelve hours ago ("good luck") so her belief is in fact true. Thus, Alice has a belief that is justified and true—but the case does not seem to match our intuitive concept of knowledge, so there must be something lacking in the JTB criterion.

Those interested in assurance will likely diagnose the problem as weakness in Alice's justification: if this were an assurance case it would be criticized for not considering the possibility that the clock is wrong or faulty. Many epistemologists take the same view and seek to retain JTB as the definition of knowledge by tightening the notion of "justification." For example, Russell's student Ramsey proposed that the justification should employ a "reliable process" [11], but this just moves the problem on to the definition of reliable process. A more widely accepted adjustment of this kind is the *indefeasibility* criterion [12–14]. A justified belief is indefeasible if it has no defeaters, where a *defeater* is a claim which, if we were to believe it, would render our original belief unjustified. (Thus, a defeater to an argument is like a hazard to a system.)

There are difficulties even here, however. A standard example is the case of Tom Grabit [12]:

> We see someone who looks just like Tom Grabit stealing a book from the library, and on this basis believe that he stole a book. Unbeknownst to us, Tom's mother claims that he is away on a trip and has an identical twin who is in the library. But also unbeknownst to us, she has dementia: Tom is not away, has no brother, and did steal a book.

The problem is that the claim by Tom's mother is a defeater to the justification (we saw it with our own eyes) for our belief that Tom stole a book. But this defeater is itself defeated (because she has dementia). So the indefeasibility criterion needs to be amended so that there are no *undefeated* defeaters to our original belief, and this seems to invite an infinite regress. Some current work in epistemology attempts to repair, refute, or explore this and similar difficulties [15, 16], but at this point I prefer to part company with epistemology.

Epistemology seeks to understand knowledge, and one approach is to employ some form of justified true belief. But truth is known only to the omniscient; as humans, the best we can aspire to is "well justified" belief. Much of the inventiveness in Gettier examples is in setting up a poorly justified belief (which is defeated by the "bad luck" event) that is nonetheless true (due to the second, "good luck," event). For assurance, we are not interested in poorly justified beliefs that turn out to be true, and many of the fine distinctions made by epistemologists are irrelevant to us. We are interested in well justified beliefs (since that is our best approach to truth) and what we can take from epistemology is indefeasibility as a compelling criterion for adequately justified belief.[3]

Observe that there are two reasons why an assurance case might be flawed: one is that the evidence is *too weak* to support the claim (to the extent we require) and this is managed by our treatment of the *weight of evidence*, as will be discussed in Section 4.1; the other is that there is something logically *wrong* or *missing* in the case (e.g., we overlooked some defeater), and these are eliminated by the notion of indefeasible justification.

Hence, the combination of justification and indefeasibility is an appropriate criterion for soundness in assurance cases. To be explicit, I will say that an assurance case is *justified* when it is achieved by means of a valid argument (and I will explain validity in Section 4), and I will say that an assurance case is justified *indefeasibly* when there is no (or, more realistically, we cannot imagine any) new information that would cause us to retract our belief in the case (i.e., no defeaters). A *sound* case is one that is justified indefeasibly and whose weight of evidence crosses some threshold for credibility.

In addition to contributing to the definition of what it means for a case to be sound, another attractive attribute of indefeasible justification is that it suggests how reviewers can challenge an assurance case: search for defeaters (flaws in the valid argument providing justification are eliminated by checking its logic, which can be automated). I discuss this in more detail in Section 6.

There are two immediate objections to the indefeasibility criterion. The first is that to establish indefeasibility we must consider all potential defeaters, and that could be costly as we might spend a lot of resources checking potential defeaters that are subsequently discarded (either because they are shown not to defeat the argument or because they are themselves defeated). However, I believe

[3] When I said "truth is known only to the omniscient" I was implicitly employing the *correspondence* criterion for truth, which is the (commonsense) idea that truth is that which accords with reality. There are other criteria for truth, among which Peirce's *limit* concept is particularly interesting: "truth is that concordance of a . . . statement with the ideal limit towards which endless investigation would tend to bring . . . belief" [17, Vol 5, para 565]. Others paraphrase it as that which is "indefeasible—that which would not be defeated by inquiry and deliberation, no matter how far and how fruitfully we were to investigate the matter in question" [18]. Russell criticized Peirce's limit concept on the grounds that it mixes truth with epistemology, but I think it is interesting for precisely this reason: independent inquiries, performed 50 years apart, converge on indefeasibility as the fundamental basis for justification, knowledge, and truth.

that if a case is truly indefeasible, then potential defeaters can either be quickly discarded (because they are not defeaters, for reasons that were already considered and recorded in justifying the original case), or themselves quickly defeated (for similar reasons). The second objection is that indefeasibility is unrealistic: how can we know that we have thought of all the "unknown unknowns"? I address this objection in Section 5, but note here that the demanding character of indefeasibility is precisely what makes it valuable: it raises the bar and requires us to make the case that we have, indeed, thought of everything.

A variant on both these objections is the concern that indefeasibility can provoke overreaction that leads to prolix arguments, full of material included "just in case" or in anticipation of implausible defeaters. A related concern is that indefeasibility gives reviewers license to raise numerous imagined defeaters. The first of these must be excluded by good engineering management: proposed defeaters, or proposed counterevidence for acknowledged defeaters, must first be scrutinized for relevance, effectiveness, and parsimony. For the second, note that rather than inviting "nuisance" defeaters during development or review, indefeasibility is a tool for their exclusion. An indefeasible case anticipates, refutes, and records all credible objections that might be raised by its reviewers. So as a case approaches completion and we become more confident that all defeaters have been recognized, so it becomes easier to discard proffered "nuisance" defeaters—because either they are not new or not defeaters, for reasons that have already been considered, or because they can themselves be defeated (for similar reasons).

4 Interpretation and Application of Indefeasibility

An assurance case justifies its claim by means of a *structured argument*, which is a hierarchical collection of individual *argument steps*, each of which justifies a *local claim* on the basis of *evidence* and/or lower-level local *subclaims*. A trivial example is shown on the left in Figure 1, where a top claim $\mathtt{C}$ is justified by an argument step $\mathtt{AS}_1$ on the basis of evidence $\mathtt{E}_3$ and subclaim $\mathtt{SC}_1$, which itself is justified by argument step $\mathtt{AS}_2$ on the basis of evidence $\mathtt{E}_1$ and $\mathtt{E}_2$.

Assurance cases often are portrayed graphically, as in the figure, and two such graphical notations are in common use: Claims-Argument-Evidence, or CAE [19], and Goal Structuring Notation, or GSN [20] (the notation in Figure 1 is generic, although its element shapes are those of GSN). In a real assurance case, the boxes in the figure will contain, or reference, descriptions of the artifacts concerned: for evidence (circles) this may be substantial, including results of tests, formal verifications, etc.; for claims and subclaims (rectangles) it will be a careful (natural language or formal) statement of the property claimed; and for argument steps (parallelograms) it will be a detailed justification or "warrant" why the cited subclaims and evidence are sufficient to justify the local parent claim.

It is important to note that this interpretation of assurance case arguments applies to CAE, for example, but that GSN, although it appears similar, uses a very different interpretation. What I call argument steps (pictured as paral-

Here, **C** indicates a claim, **SC** a subclaim, and **E** evidence; **AS** indicates a generic argument step, **RS** a reasoning step, and **ES** an evidential step.

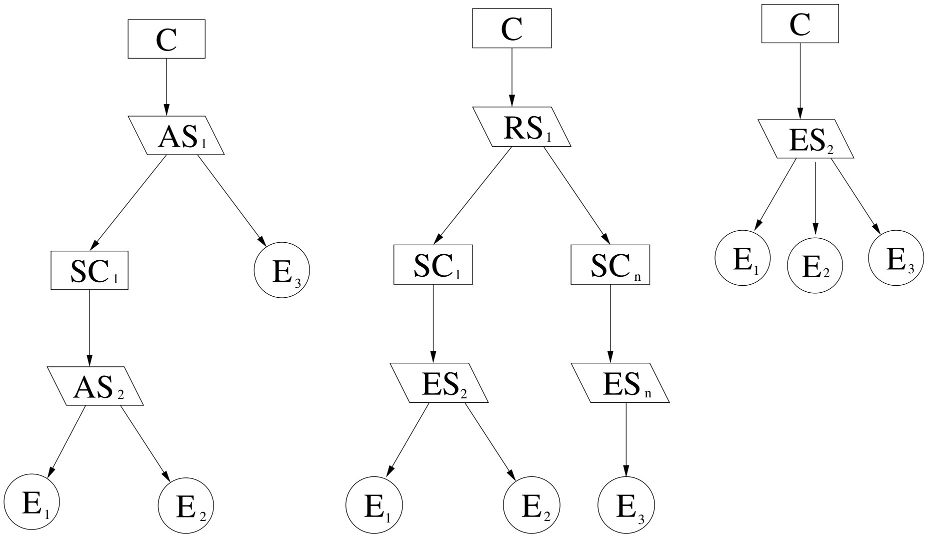

Fig. 1. A Structured Argument in Free (left) and Simple Form (center) and Refactored (right)

lelograms) are called "strategies" in GSN and their purpose is to describe how the argument is being made (e.g., as an enumeration over components or over hazards), rather than to state an inference from subclaims to claim. In fact, GSN strategies are often omitted and sets of "subgoals" (i.e., subclaims) are connected directly to a "goal" (i.e., claim), and the implicit argument is taken to be some obvious decomposition. I do not attempt to provide an interpretation for GSN strategies. In the interpretation described here, and in CAE "blocks" [21], an argument step that employs a decomposition must provide a narrative justification (i.e., warrant) and possibly some supporting evidence for the decomposition employed (e.g., why it is necessary and sufficient to enumerate over just *these* hazards, or why the claim distributes over the components).

As a concrete example of our interpretation, let us suppose that the left side of Figure 1 is a (trivialized) software assurance case, where the claim $\mathtt{C}$ concerns software correctness. Evidence $\mathtt{E}_1$ might then be test results, and $\mathtt{E}_2$ a description of how the tests were selected and the adequacy of their coverage, so that $\mathtt{SC}_1$ is a subclaim that the software is adequately tested and argument step $\mathtt{AS}_2$ provides a warrant or justification for this. In addition, we need to be sure that the deployed software is the same as that tested, so $\mathtt{E}_3$ might be version management data to confirm this and argument step $\mathtt{AS}_1$ provides a warrant that the claim of software correctness follows if the software is adequately tested, and the tested software is the deployed software. Of course, a real assurance case will concern more than testing and even testing will require additional items of supporting evidence (e.g., the trustworthiness of the test oracle), so real assurance cases are large. On the other hand, evidence must support a specific claim, and claims must contribute to an explicit argument, so there is hope that assurance cases can be more focused and therefore more succinct than current processes driven by guidelines such as DO-178C that require large quantities of evidence with no explicit rationale.

Observe that the argument step $\texttt{AS}_1$ on the left of Figure 1 uses both evidence $\texttt{E}_3$ and a subclaim $\texttt{SC}_1$. Later, in Section 5, I will sketch how to interpret such "mixed" argument steps, but it is easier to understand the basic approach in their absence. By introducing additional subclaims where necessary, it is straightforward to convert arguments into *simple form* where each argument step is supported either by subclaims (boxes) or by evidence (circles), but not by a combination of the two. The mixed or free form argument on the left of Figure 1 is converted to simple form in the center by introducing a new subclaim $\texttt{SC}_n$ and a new argument step $\texttt{ES}_n$ above $\texttt{E}_3$.

The benefit of simple form is that argument steps are now of two kinds: those supported by subclaims are called *reasoning steps* (in the example, argument step $\texttt{AS}_1$ is relabeled as reasoning step $\texttt{RS}_1$), while those supported by evidence are called *evidential steps* (in the example, these are the relabeled step $\texttt{ES}_2$ and the new step $\texttt{ES}_n$) and the key to our approach is that the two kinds of argument step are interpreted differently.

Specifically, evidential steps are interpreted "epistemically" while reasoning steps are interpreted "logically." The idea is that evidential steps whose "weight of evidence" (as described below) crosses some threshold are treated as premises in a conventional logical argument in which the reasoning steps are treated as axioms. This is a systematic version of "Natural Language Deductivism" (NLD) [22], which interprets informal arguments as attempts to create deductively valid arguments. NLD differs from deductive proof in formal mathematics and logic in that its premises are "reasonable or plausible" rather than certain, and hence its conclusions are likewise reasonable or plausible rather than certain [23, Section 4.2]. Our requirement that the weight of each evidential step must cross some threshold systematizes what it means for the premises to be reasonable or plausible or, as we often say, credible. (Hence, there is no conceptual problem with evidence based on expert opinion, or incomplete testing, provided these are buttressed by warrants, and possibly additional evidence, for their credibility.)

Our treatment of reasoning steps shares with NLD the requirement that these should be deductively valid (i.e., the subclaims must imply or entail the parent claim); this differs from other interpretations of informal argumentation, which adopt criteria that are weaker (e.g., the subclaims need only "strongly suggest" the parent claim) [24], or different (e.g., the Toulmin style of argument) [25]. Weaker (or different) criteria may be appropriate in other argumentation contexts: indeed, the very term "natural language deductivism" was introduced by Govier [26] 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 can adopt the label NLD with pride. We consider the case of those who assert the contrary in Subsection 4.2.

Because our treatment is close to that of formal logic, we adopt its terminology and say that an argument is *valid* if it its reasoning steps are logically so (i.e., true in all interpretations) and that it is *sound* if, in addition, its evidential steps all cross their thresholds for credibility.[4] Thus, our requirement for a sound assurance case is that its argument is sound in the sense just described (which we also refer to as a *justified* argument), and indefeasible.

We now consider the two kinds of argument steps in more detail.

4.1 Evidential Steps

My recommended approach for evidential steps is described in a related paper [27]; here, I provide a summary and connect it to the indefeasibility criterion.

When we have an evidential step with some collection of evidence E, our task is to decide if this is sufficient to accept its local claim C as a premise. We cannot expect E to prove C because the relation between evidence and claims is not one of logic but of epistemology (i.e., it concerns knowledge and belief). Thus, when an evidential step uses two or more items of evidence to support a subclaim (as, for example, at the lower left of the arguments in Figure 1), the interpretation is not that the conjunction of the evidence logically supports the subclaim, but that each supports it to some degree and together they support it to a greater degree. The reason we have several items of evidence supporting a single claim is that there are rather few claims that are directly observable. Claims like "correctness" can only be inferred from indirect and partial observations, such as testing and reviews. Because these observations provide indirect and incomplete evidence, we combine several of them, in the belief that, together, their different views provide an accurate evaluation of that which cannot be observed directly. Furthermore, an observation may provide valid evidence only in the presence of other evidence: for example, testing is credible only if we have a trustworthy way of assessing test results (i.e., an oracle), so an evidential step concerning testing must also include evidence for the quality of the oracle employed.

Thus, as previously noted, the assessment of evidential steps is not a problem in logic (i.e., we are not *deducing* the claim from the evidence) but in epistemology: we need to assess the extent to which the evidence allows us to *believe* or *know* the truth of the subclaim. Subjective probabilities provide a basis for assessing and reporting confidence in the various beliefs involved and we need to combine these in some way to yield a measure for the "weight" of the totality of evidence E in support of claim C. This topic has been studied in the field of Bayesian Confirmation Theory [28] where suitable *confirmation measures* have been proposed. The crucial idea is that E should not only support C but should

[4] It is because these usages are standard in logic that we prefer *sound* to *valid* in [1].

discriminate between C and other claims, and the negation $\neg C$ in particular. This suggests that suitable measures will concern the difference or ratio of the conditional probabilities $P(E \mid C)$ and $P(E \mid \neg C)$.[5] There are several such measures but among the most recommended is that of Kemeny and Oppenheim [29]

$$\frac{P(E \mid C) - P(E \mid \neg C)}{P(E \mid C) + P(E \mid \neg C)};$$

this measure is positive for strong evidence, near zero for weak evidence, and negative for counterevidence.

When an evidential step employs multiple items of evidence $E_1, \ldots, E_i$, which may not be independent of one another, we need to estimate conditional probabilities for the individual items of evidence and combine them to calculate the overall quantities $P(E_1, \ldots, E_i \mid C)$ and $P(E_1, \ldots, E_i \mid \neg C)$ used in the chosen confirmation measure; Bayesian Belief Nets (BBNs) and their tools provide ways to do this ([27] gives an example).

This probabilistic model, supported by suitable BBN tools, can be used to calculate a confirmation measure that represents the weight of evidence in support of an evidential claim, and a suitable threshold on that weight (which may differ from one claim to another) can be used to decide whether to accept the claim as a premise in the reasoning steps of the argument. I concede that it is difficult to assign credible probabilities to the estimations involved, so in practice the determination that evidence is sufficient to justify a claim will generally be made by (skilled) human judgment, unassisted by explicit probabilistic calculations. However, I believe that judgment can be improved and honed by undertaking numerical examples and "what if" experiments using the probabilistic model described here. And I suggest that assurance templates that may be widely applied should be subjected to quantitative examination of this kind. The example in [27] provides an elementary prototype for this kind of examination.

The probabilistic model helps us understand how the various items of evidence in an evidential step combine to lend weight to belief in its claim. Applying the model to a specific evidential step, whether this is done formally with BBNs or informally by human judgment, involves determination that the collection of evidence is "valid" (e.g., does not contain contradictory items) and credible (i.e., its weight crosses our threshold for acceptance). The indefeasibility criterion comes into play when we ask whether the evidence supplied is also "complete." Specifically, indefeasibility requires us to consider whether any *defeaters* might exist for the evidence supplied. For example, testing evidence is defeated if it is not for exactly the same software as that under consideration, and formal verification evidence is defeated if its theorem prover might be unsound.

It might seem that since testing merely samples a space, it must always be incomplete and therefore vulnerable to defeat. This is true, but I maintain that this kind of "graduated" defeat is different in kind and significance to true "noetic"

[5] It might seem that we should be considering $P(C \mid E)$ and its variants rather than $P(E \mid C)$; these are related by Bayes' rule but it is easier to estimate the likelihood of concrete observations, given a claim about the world, than vice-versa.

defeat. Almost all evidence is imperfect and partial; that is why evidential steps are evaluated epistemically and why we use probabilities (either formally or intuitively) to record our confidence. Testing is no different than other forms of evidence in this regard. Furthermore, we can choose how partial is our testing: depending on the claim, we can target higher levels of "coverage" for unit tests, or higher levels of statistical validity for random system tests. Some other kinds of evidence share this "graduated" character: for example, we can choose how much effort to devote to human reviews. Thus, the potential for defeat in graduated forms of evidence is acknowledged and managed. It is managed through the "intensity" of the evidence (e.g., effort applied, as indicated by hours of human review, or coverage measures for testing) and probabilistic assessment of its resulting "weight." If that weight is judged insufficient, then evidence that is vulnerable to graduated defeat might be buttressed by additional evidence that is strong on the graduated axis, but possibly weaker on others. Thus testing, which considers interesting properties but for only a limited set of executions could, for suitable claims, be buttressed by static analysis, which considers *all* executions, but only for limited properties.

"Noetic" defeat is quite different to graduated defeat: it signifies something is wrong or missing and undermines the whole basis for given evidence. For example, if our test oracle (the means by which we decide whether or not tests are successful) could be faulty, or if the tested components might not be the same as those in the actual system, then our tests have no evidential value.

The indefeasibility criterion requires us to eliminate noetic defeaters and to manage graduated ones. Consideration of potential noetic defeaters may lead us to develop additional evidence or to restrict the claim. According to the dependencies involved, additional evidence can be combined in the same evidential step as the original evidence or it can be used in dedicated evidential steps to support separate subclaims that are combined in higher-level reasoning steps. For example, in the center of Figure 1, evidence $\mathtt{E}_3$ might concern version management (to counter the noetic defeater that the software tested is not the same as that deployed) and it supports a separate claim that is combined with the testing subclaim higher up in the argument. On the other hand, if this were evidence for quality of the oracle (the means by which test results are judged) it would be better added directly to the evidential step $\mathtt{ES}_2$ since it is not independent of the other evidence in that step, leading to the refactored argument on the right of Figure 1.

We now turn from evidential steps to reasoning steps.

4.2 Reasoning Steps

Evidential steps are the bridge between epistemology and logic: they establish that the evidence is sufficient, in its context, to treat their subclaims as premises in a logical interpretation of the reasoning steps. That logical interpretation is a "deductive" one, meaning that the conjunction of subclaims in a reasoning step must imply or entail its claim. This interpretation is not the usual one: most other treatments of assurance case arguments require only that the collection of subclaims should "strongly suggest" the claim, a style of reasoning generally called "inductive" (this is a somewhat unfortunate choice as the same term is used with several other meanings in mathematics and logic). The deductive interpretation is a consequence of our requirement for indefeasibility: if a reasoning step is merely inductive, we are admitting a "gap" in our reasoning that can be filled by a defeater.

Some authors assert that assurance case arguments cannot be deductive due to complexity and uncertainty [30, 31]. I emphatically reject this assertion: the whole point of an assurance case is to manage complexity and uncertainty. In the interpretation advocated here, all uncertainty is confined to the evaluation of evidential steps, where (formal or informal) probabilistic reasoning may be used to represent and estimate uncertainty in a scientific manner. In the inductive interpretation, there is no distinction between evidential and reasoning steps so uncertainty can lie anywhere, and there is no requirement for indefeasibility so the argument can be incomplete as well as unsound.

Nonetheless, the requirement for indefeasibility, and hence for deductive reasoning steps, strikes some as an unrealizable ideal—a counsel of perfection—so in the following section I consider its feasibility and practicality.

5 Feasibility of Indefeasibility

One objection to the indefeasibility criterion for assurance cases is that it sets too high a bar and is infeasible and unrealistic in practice. How can we ever be sure, an objector might ask, that we have thought of all the "unknown unknowns" and truly dealt with all possible defeaters? My response is that there are systematic ways to develop deductive reasoning steps, and techniques that shift the doubt into evidential steps where it can be managed appropriately.

Many reasoning steps represent a decomposition in some dimension and assert that if we establish some claim for each component of the decomposition then we can conclude a related claim for the whole. For example, we may have a system `X` that is composed of subsystems `X1`, `X2`, ..., `Xn` and we argue that `X` satisfies claim `C`, which we denote `C(X)`, by showing that each of its subsystems also satisfies `C`: that is, we use subclaims `C(X1)`, `C(X2)`, ..., `C(Xn)`. We might use this reasoning step to claim that a software system will generate no runtime exceptions by showing it to be true for each of its software components. However, this type of argument is not always deductively valid—for example, we cannot argue that an airplane is safe by arguing that its wheels are safe, its rudder is safe, ... and its wings are safe. Deductive validity is contingent on the property `C`, the nature of the system `X`, and the way in which the subsystems `X1`, `X2`, ..., `Xn` are composed to form `X`. Furthermore, claim `C(X)` may not follow simply from the same claim applied to the subsystems, but from different subclaims applied to each: `C1(X1)`, `C2(X2)`, ..., `Cn(Xn)`. For example, a system may satisfy a timing constraint of 10ms. if its first subsystem satisfies a constraint of 3ms., its second satisfies 4ms. and its third and last satisfies 2ms. (together with some assumptions about the timing properties of the mechanism that binds these subsystems together).

I assert that we can be confident in the deductive character of systematically constructed reasoning steps of this kind by explicitly stating suitable assumptions or side conditions (which are simply additional subclaims of the step) to ensure that the conjunction of component subclaims truly implies the claim. In cases where the subclaims and claim concern the same property `C`, this generally follows if `C` distributes over the components and the mechanism of decomposition, and this would be an assumption of the template for this kind of reasoning step. In more complex cases, formal modeling can be used to establish deductive validity of the decomposition under its assumptions. Bloomfield and Netkachova [21] provide several examples of templates for reasoning steps of this kind, which they call "decomposition blocks."

Deductiveness in these steps derives from the fact that we have a definitive enumeration of the components to the decomposition and have established suitable assumptions. A different kind of decomposition is one over hazards or threats. Here, we do not have a definitive enumeration of the components to the decomposition: it is possible that a hazard might be overlooked. In cases such as this, we transform concerns about deductiveness of the reasoning step into assessment of evidence for the decomposition performed. For example, we may have a general principle or template that a system is safe if all its hazards are eliminated or adequately mitigated. Then we perform hazard analysis to identify the hazards—and that means *all* the hazards—and use a reasoning step that instantiates the general principle as a decomposition over the specific hazards that were identified and attach the evidence for hazard analysis as a side condition. Thus our doubts about deductiveness of the reasoning step that enumerates over hazards are transformed into assessment of the credibility of the evidence for the completeness of hazard analysis (e.g., the method employed, the diligence of its performance, historical effectiveness, and so on).

This is not a trick; when reasoning steps are allowed to be inductive, there is no requirement nor criterion to justify how "close" to deductive (i.e., indefeasible) the steps really are. Under the indefeasibility criterion, we need to justify the deductiveness of each reasoning step, either by reference to physical or logical facts (e.g., decomposition over enumerable components or properties) or to properly assessed evidence, such as hazard analysis, and this is accomplished by the method described.

Both kinds of decomposition discussed above employ assumptions or side conditions (or as will be discussed below, "provisos") to ensure the decomposition is indefeasible. Assumptions (as we will call them here) are logically no different than other subclaims in an argument step. That is, an argument step

$$p_1 \ \texttt{AND}\ p_2\ \texttt{AND}\ \cdots\ \texttt{AND}\ p_n\ \texttt{IMPLIES}\ c,\ \texttt{ASSUMING}\ a.$$

is equivalent to

$$a\ \texttt{AND}\ p_1\ \texttt{AND}\ p_2\ \texttt{AND}\ \cdots\ \texttt{AND}\ p_n\ \texttt{IMPLIES}\ c. \tag{2}$$

If the original is an evidential step (i.e., $p_1, p_2, \ldots p_n$ are evidence) and a is a subclaim, then (2) is a mixed argument step involving both evidence and subclaims. In Figure 1 of Section 4, we explained how such arguments could be converted to simple form. By that method we might obtain

$$p_1 \texttt{ AND } p_2 \texttt{ AND } \cdots \texttt{ AND } p_n \texttt{ IMPLIES } c_1 \tag{3}$$

$$a \texttt{ AND } c_1 \texttt{ IMPLIES } c \tag{4}$$

and an apparent problem is that the required assumption has been lost from (3). However, this is not a problem at all. The structure of an assurance case argument (as we have defined it) is such that every subclaim must be true. Hence, it is sound to interpret (3) under the assumption a even though it is established elsewhere in the tree of subclaims. In the same way, evidence $\texttt{E}_3$ in the left or center of Figure 1 can be interpreted under the assumption of subclaim $\texttt{SC}_1$. This treatment can lead to circularity, and checks to detect it could be expensive. A sound and practical restriction is to stipulate that each subclaim or item of evidence is interpreted on the supposition that subclaims appearing earlier (i.e., to its left in a graphical presentation) are true. Thus, mixed argument steps like (2) are treated as reasoning steps subject to the evidentially supported assumptions represented by a and this interpretation can be applied either directly or via the conversion to simple form.

Beyond the objection, just dismissed, that the indefeasibility criterion is unrealistic or infeasible in practice, is the objection that it is the *wrong* criterion—because science itself does not support deductive theories.

This contention derives from a controversial topic in the philosophy of science concerning "provisos" (sometime spelled "provisoes") or *ceteris paribus* clauses (a Latin phrase usually translated as "other things being equal") in statements of scientific laws. For example, we might formulate the law of thermal expansion as follows: "the change in length of a metal bar is directly proportional to the change in temperature." But this is true only if the bar is not partially encased in some unyielding material, and only if no one is hammering the bar flat at one end, and.... This list of provisos is indefinite, so the simple statement of the law (or even a statement with some finite set of provisos) can only be inductively true. Hempel [32] asserts there is a real issue here concerning the way we understand scientific theories and, importantly, the way we attempt to confirm or refute them. Others disagree: in an otherwise sympathetic account of Hempel's work in this area, his student Suppe describes "where Hempel went wrong" [33, pp. 203, 204], and Earman and colleagues outright reject it [34].

Rendered in terms of assurance cases, the issue is the following. During development of an assurance case argument, we may employ a reasoning step asserting that its claim follows from some conjunction of subclaims. The assertion may not be true in general, so we restrict it with additional subclaims representing necessary assumptions (i.e., provisos) that are true (as other parts of the argument must show) in the context of this particular system. The "proviso problem" is then: how do we know that we have not overlooked some necessary assumption? I assert that this is just a variant on the problem exemplified by hazard enumeration that was discussed earlier, and is solved in the same way: we provide explicit claims and suitable evidence that the selected assumptions are sufficient.

Unlike inductive cases, where assumptions or provisos may be swept under the rug, in deductive cases we must identify them explicitly and provide evidentially supported justification for their correctness and completeness.

Some philosophers might say this is hubris, for we cannot be sure that we do identify all necessary assumptions or provisos. This is, of course, true in the abstract but, just as we prefer well-justified belief to the unattainable ideal of true knowledge, so we prefer well-justified assumptions to the limp veracity of inductive arguments. With an inductive reasoning step we are saying "this claim holds under these provisos, but there may be others," whereas for a deductive step we are saying "this claim holds under these assumptions, and this is where we make our stand." This alerts our reviewers and raises the stakes on our justification. The task of reviewers is the topic of the following section.

6 Challenges and Reviews

Although reasoning steps must ultimately be deductive for the indefeasible interpretation, I recommend that we approach this via the methods and tools of the inductive interpretation. The reason for this is that assurance cases are developed incrementally: at the beginning, we might miss some possible defeaters and will not be sure that our reasoning steps are deductive. As our grasp of the problem deepens, we may add and revise subclaims and argument steps and only at the end will we be confident that each reasoning step is deductive and the overall argument is indefeasible. Yet even in the intermediate stages, we will want to have some (mechanically supported) way to evaluate attributes of the case (e.g., to check that every subclaim is eventually justified), and an inductive interpretation can provide this, particularly if augmented to allow explicit mention of defeaters.

Furthermore, even when we are satisfied that the case is deductively sound, we need to support review by others. The main objection to assurance cases is that they are prone to "confirmation bias" [35]: this is the human tendency to seek information that will confirm a hypothesis, rather than refute it. The most effective counterbalance to this and other fallibilities of human judgment is to subject assurance cases to vigorous examination by multiple reviewers with different points of view. Such a "dialectical" process of review can be organized as a search for potential defeaters. That is, a reviewer asks "what if this happens," or "what if that is not true."

The general idea of a defeater to a proposition is that it is a claim which, if we were to believe it, would render our belief in the original proposition unjustified. Within argumentation, this general idea is refined into specific kinds of defeaters. Pollock [36, page 40] defines a *rebutting defeater* as one that (in our terminology) contradicts the claim to an argument step (i.e., asserts it is false), while an *undercutting defeater* merely doubts it (i.e., doubts that the claim really does follow from the proffered subclaims or evidence); others subsequently defined *undermining defeaters* as those that doubt some of the evidence or subclaims used in an argument step. This taxonomy of defeaters can be used to guide a systematic critical examination of an assurance case argument.

For an elementary example, we might justify the claim "Socrates is mortal" by a reasoning step derived from "all men are mortal" and an evidential step "Socrates is a man." A reviewer might propose a rebutting defeater to the reasoning step by saying "I have a CD at home called 'The Immortal James Brown,'[6] so not all men are mortal." The response to such challenges may be to adjust the case, or it may be to dispute the challenge (i.e., to defeat the defeater). Here, a proponent of the original argument might rebut the defeater by observing that James Brown is dead (citing Google) and therefore indubitably mortal. An undercutting defeater for the same reasoning step might assert that the claim cannot be accepted without evidence and an adjustment might be to interpret "mortal" as "lives no more than 200 years" and to supply historical evidence of human lifespan. An undermining defeater for the evidential step might challenge the assumption that Socrates was a historical figure (i.e., a "real" man).

I think the record of such challenges and responses (and the narrative justification that accompanies them) should be preserved as part of the assurance case to assist further revisions and subsequent reviews. The fields of defeasible and dialectical reasoning provide techniques for recording and evaluating such "disputed" arguments. For example, *Carneades* [37] is a system that supports dialectical reasoning, allowing a subargument to be *pro* or *con* its conclusion: a claim is "in" if it is not the target of a *con* that is itself "in" unless . . . (the details are unimportant here). Weights can be attached to evidence and a *proof standard* is calculated by "adding up" the *pro*s and *con*s supporting the conclusion and their attendant weights. For assurance cases, we ultimately want the proof standard equivalent to a deductive argument, which means that no *con* may be "in" (i.e., every defeater must be defeated). Takai and Kido [38] build on these ideas to extend the *Astah GSN* assurance case toolset with support for dialectical reasoning [39].

7 Probabilistic Interpretation

In Section 2, we explained how confidence in an assurance case, plus failure-free experience, can provide assurance for extremely low rates of critical failure, and hence for certification. Sections 3 to 6 have described our approach to interpretation and evaluation of an assurance case, so we now need to put the two pieces together. In particular, we would like to use the determination that a case is sound (i.e., its argument is valid, all its evidential steps cross the threshold for credibility, it is indefeasible, and all these assessments have withstood dialectical challenge) to justify expressions of confidence such as $p_{nf} \geq 0.9$ in the absence of faults. This is a subjective probability, but one way to give it a frequentist interpretation is to suppose that if 10 systems were successfully evaluated in the same way, at most one of them would ever suffer a critical failure in operation.

This is obviously a demanding requirement and not one amenable to definitive demonstration. One possibility is to justify $p_{nf} \geq 0.9$ for this assurance case by a separate assurance case that is largely based on evidential steps that cite

[6] The CD in question is actually called "Immortal R&B Masters: James Brown."

historical experience with the same or similar methods (for example, no civil aircraft has ever suffered a catastrophic failure condition attributed to software assured to DO-178B/C Level A[7]). For this reason among others, I suggest that assurance for really critical systems should build on successful prior experience and that templates for their assurance cases should be derived from existing guidelines such as DO-178C [2] rather than novel "bespoke" arguments.

Different systems pose different risks and not all need assurance to the extreme level required for critical aircraft software. Indeed, aircraft software itself is "graduated" according to risk. So a sharpened way to pose our question is to ask how a given assurance case template can itself be graduated to deliver reduced assurance at correspondingly reduced cost or, dually, how our overall confidence in the case changes as the case is weakened. Eliminating or weakening subclaims within a given argument immediately renders it defeasible, so that is not a viable method of graduation. What remains is lowering the threshold on evidential steps, which may allow less costly evidence (e.g., fewer tests), or the elimination or replacement of some evidence (e.g., replace static analysis by manual review). When evidence is removed or changed, some defeaters may be eliminated too, and that can allow the removal of subclaims and their supporting evidence (e.g., if we eliminate static analysis we no longer need claims or evidence about its soundness).

It is difficult to relate weakened evidence to explicit reductions in the assessment of p_{nf}. Again, we could look to existing guidelines such as DO-178C, where 71 "objectives" (essentially items of evidence) are required for Level A software, 69 for Level B, 62 for Level C, and 26 for Level D. Alternatively, we could attempt to assess confidence in each evidential step (i.e., a numerical value for $P(C \mid E)$) and assess p_{nf} as some function of these (e.g., the minimum over all evidential steps). The experiments by Graydon and Holloway mentioned earlier [7,8] suggest caution here, but some conservative approaches are sound. For example, it follows from a theorem of probability logic [40] that *doubt* (i.e., 1 minus probabilistic confidence) in the claim of a reasoning step is no worse than the sum of the doubts of its supporting subclaims.

It has to be admitted that quantification of this kind rests on very subjective grounds and that the final determination to accept an assurance case is a purely human judgment. Nonetheless, the model of Section 2 and the interpretation suggested here do establish a probabilistic approach to that judgment, although there is clearly opportunity for further research.

8 Conclusion

I have reviewed the indefeasibility criterion from epistemology and argued that it is appropriate for assurance case arguments. I also proposed a systematic version of Natural Language Deductivism (NLD) as the basis for judging soundness of

[7] This remains true despite the 737Max MCAS crashes; as far as we know, the MCAS software satisfied its requirements; the flaws were in the requirements, whose assurance is the purview of ARP 4754A [3], which Boeing apparently failed to apply with any diligence.

assurance case arguments: the interior or reasoning steps of the argument should be deductively valid, while the leaf or evidential steps are evaluated epistemically using ideas from Bayesian confirmation theory and are treated as premises when their evaluation crosses some threshold of credibility. NLD ensures correctness or soundness of the argument, while indefeasibility ensures completeness. I derived requirements for the evidential and reasoning steps in such arguments and argued that they are feasible and practical, and that postulating defeaters provides a systematic way to challenge arguments during review.

I propose that assurance case templates satisfying these criteria and derived from successful existing assurance guidelines (e.g., DO-178C) can provide a flexible and trustworthy basis for assuring future systems.

The basis for assurance is systematic consideration of every possible contingency, which requires that the space of possibilities is knowable and enumerable. This is true at design time for conventional current systems such as commercial aircraft, where conservative choices may be made to ensure predictability. But more recent systems such as self-driving cars and "increasingly autonomous" (IA) aircraft pose challenges, as do systems that are assembled or integrated from other systems while in operation (e.g., multiple medical devices attached to a single patient). Here, we may have software whose internal structure is opaque (e.g., the result of machine learning), an imperfectly known environment (e.g., a busy freeway where other road users may exhibit unexpected behavior), and interaction with other systems (possibly due to unplanned stigmergy via the plant) whose properties are unknown. These challenge the predictability that is the basis of current assurance methods. I believe this basis can be maintained and the assurance case framework can be preserved by shifting some of the gathering and evaluation of evidence, and assembly of the final argument, to integration- or run-time [41–43], and that is an exciting topic for future research.

Acknowledgments. This work was partially funded by SRI International and builds on previous research that was funded by NASA under a contract to Boeing with a subcontract to SRI International.

I have benefited greatly from extensive discussions on these topics with Robin Bloomfield of Adelard and City University. Patrick Graydon of NASA provided very useful comments on a previous iteration of the paper, as did the editors and reviewers for this volume.

References

1. UK Ministry of Defence: Defence Standard 00-56, Issue 4: Safety Management Requirements for Defence Systems. Part 1: Requirements. (2007)
2. Requirements and Technical Concepts for Aviation (RTCA) Washington, DC: DO-178C: Software Considerations in Airborne Systems and Equipment Certification. (2011)
3. Society of Automotive Engineers: Aerospace Recommended Practice (ARP) 4754A: Certification Considerations for Highly-Integrated or Complex Aircraft Systems. (2010) Also issued as EUROCAE ED-79.
4. Strigini, L., Povyakalo, A.: Software fault-freeness and reliability predictions. In: SafeComp 2013: Proceedings of the 32nd International Conference on Computer

Safety, Reliability, and Security. Volume 8153 of Lecture Notes in Computer Science, Toulouse, France, Springer-Verlag (2013) 106–117

5. Federal Aviation Administration: System Design and Analysis. (1988) Advisory Circular 25.1309-1A.
6. Littlewood, B., Rushby, J.: Reasoning about the reliability of diverse two-channel systems in which one channel is "possibly perfect". IEEE Transactions on Software Engineering **38** (2012) 1178–1194
7. Graydon, P.J., Holloway, C.M.: An investigation of proposed techniques for quantifying confidence in assurance arguments. Safety Science **92** (2017) 53–65
8. Graydon, P.J., Holloway, C.M.: An investigation of proposed techniques for quantifying confidence in assurance arguments. Technical Memorandum NASA/TM-2016219195, NASA Langley Research Center, Hampton VA (2016)
9. Gettier, E.L.: Is justified true belief knowledge? Analysis **23** (1963) 121–123
10. Russell, B.: Human Knowledge: Its Scope and Limits. George Allen & Unwin, London, England (1948)
11. Ramsey, F.P.: Knowledge. In Mellor, D.H., ed.: Philosophical Papers of F. P. Ramsey. Cambridge University Press, Cambridge, UK (1990) 110–111 (original manuscript, 1929).
12. Lehrer, K., Paxson, T.: Knowledge: Undefeated justified true belief. The Journal of Philosophy **66** (1969) 225–237
13. Klein, P.D.: A proposed definition of propositional knowledge. The Journal of Philosophy **68** (1971) 471–482
14. Swain, M.: Epistemic defeasibility. American Philosophical Quarterly **11** (1974) 15–25
15. Turri, J.: Is knowledge justified true belief? Synthese **184** (2012) 247–259
16. Williams, J.N.: Not knowing you know: A new objection to the defeasibility theory of knowledge. Analysis **75** (2015) 213–217
17. Hartshorne, C., Weiss, P., Burks, A.W., eds.: Collected Papers of Charles Sanders Peirce. Volumes 1–8. Harvard University Press, Cambridge, MA (1931–1958)
18. Misak, C.: Review of "Democratic Hope: Pragmatism and the Politics of Truth" by Robert B. Westbrook. Transactions of the Charles S. Peirce Society **42** (2006) 279–282
19. Adelard LLP London, UK: ASCAD: Adelard Safety Case Development Manual. (1998) Available from https://www.adelard.com/resources/ascad.html.
20. Kelly, T.: Arguing Safety—A Systematic Approach to Safety Case Management. DPhil thesis, Department of Computer Science, University of York, UK (1998)
21. Bloomfield, R., Netkachova, K.: Building blocks for assurance cases. In: ASSURE: Second International Workshop on Assurance Cases for Software-Intensive Systems, Naples, Italy, IEEE International Symposium on Software Reliability Engineering Workshops (2014) 186–191
22. Groarke, L.: Deductivism within pragma-dialectics. Argumentation **13** (1999) 1–16
23. Groarke, L.: Informal logic. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. Spring 2017 edn. Metaphysics Research Lab, Stanford University (2017)
24. Blair, J.A.: What is informal logic? In van Eemeren, F.H., Garssen, B., eds.: Reflections on Theoretical Issues in Argumentation Theory. Volume 28 of The Argumentation Library. Springer (2015) 27–42
25. Toulmin, S.E.: The Uses of Argument. Cambridge University Press (2003) Updated edition (the original is dated 1958).
26. Govier, T.: Problems in Argument Analysis and Evaluation. Volume 5 of Studies of Argumentation in Pragmatics and Discourse Analysis. De Gruyter (1987)

27. Rushby, J.: 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 Lecture Notes in Artificial Intelligence, Kanagawa, Japan, Springer-Verlag (2015) 331–347
28. Earman, J.: Bayes or Bust? A Critical Examination of Bayesian Confirmation Theory. MIT Press (1992)
29. Tentori, K., Crupi, V., Bonini, N., Osherson, D.: Comparison of confirmation measures. Cognition **103** (2007) 107–119
30. Cassano, V., Maibaum, T.S., Grigorova, S.: Towards making safety case arguments explicit, precise, and well founded. (In: This volume)
31. Chechik, M., Salay, R., Viger, T., Kokaly, S., Rahimi, M.: Software assurance in an uncertain world. In: International Conference on Fundamental Approaches to Software Engineering (FASE). Volume 11424 of Lecture Notes in Computer Science, Prague, Czech Republic, Springer-Verlag (2019) 3–21
32. Hempel, C.G.: Provisoes: A problem concerning the inferential function of scientific theories. Erkenntnis **28** (1988) 147–164. Also in conference proceedings "The Limits of Deductivism," edited by Adolf Grünbaum and W. Salmon, University of California Press, 1988.
33. Suppe, F.: Hempel and the problem of provisos. In Fetzer, J.H., ed.: Science, Explanation, and Rationality: Aspects of the Philosophy of Carl G. Hempel. Oxford University Press (2000) 186–213
34. Earman, J., Roberts, J., Smith, S.: *Ceteris Paribus* lost. Erkenntnis **57** (2002) 281–301
35. Leveson, N.: The use of safety cases in certification and regulation. Journal of System Safety **47** (2011) 1–5
36. Pollock, J.L.: Cognitive Carpentry: A Blueprint for How to Build a Person. MIT Press (1995)
37. Gordon, T.F., Prakken, H., Walton, D.: The Carneades model of argument and burden of proof. Artificial Intelligence **171** (2007) 875–896
38. Takai, T., Kido, H.: A supplemental notation of GSN to deal with changes of assurance cases. In: 4th International Workshop on Open Systems Dependability (WOSD), Naples, Italy, IEEE International Symposium on Software Reliability Engineering Workshops (2014) 461–466
39. Astah: (Astah GSN home page) http://astah.net/editions/gsn.
40. Adams, E.W.: A Primer of Probability Logic. Center for the Study of Language and Information (CSLI), Stanford University (1998)
41. Rushby, J.: Trustworthy self-integrating systems. In Bjørner, N., Prasad, S., Parida, L., eds.: 12th International Conference on Distributed Computing and Internet Technology, ICDCIT 2016. Volume 9581 of Lecture Notes in Computer Science, Bhubaneswar, India, Springer-Verlag (2016) 19–29
42. Rushby, J.: Automated integration of potentially hazardous open systems. In Tokoro, M., Bloomfield, R., Kinoshita, Y., eds.: Sixth Workshop on Open Systems Dependability (WOSD), Keio University, Tokyo, Japan, DEOS Association and IPA (2017) 10–12
43. Rushby, J.: Assurance and assurance cases. In Pretschner, A., Peled, D., Hutzelmann, T., eds.: Dependable Software Systems Engineering (Marktoberdorf Summer School Lectures, 2016). Volume 50 of NATO Science for Peace and Security Series D. IOS Press (2017) 207–236