Towards Making Safety Case Arguments Explicit, Precise, and Well Founded

Valentín Cassano, Thomas S. E. Maibaum, and Silviya Grigorova

Abstract The introduction of safety cases into the practice of safety assurance has revolutionized safety engineering. Via a 'safety argument', a safety case aims to explicate, and to provide some structure for, the kind of reasoning involved in demonstrating that a system is safe. To date, there are several notations for writing down safety arguments. These notations suffer from not having a well-founded semantics, making them deficient w.r.t. the requirements of a serious approach to engineering. We consider that a well-founded semantics for safety arguments ought to be based on logical principles in the form of a logical calculus. Logic is the basis for reasoning in mathematics, philosophy, and science, and the same should be true for safety reasoning. With this goal in mind, we take some steps towards constructing a logical calculus for safety arguments by exploring some of the features of this calculus. Moreover, we look into the essential role that evidence plays in safety arguments. Evidence sets apart safety arguments from their traditional logical counterpart, as assumptions in safety arguments must be grounded on (i.e., justified by) data from the empirical world. We present our thoughts on these matters, and illustrate them by means of examples. We consider that our work establishes a framework for discussing safety arguments in a more rigorous manner.

1 Introduction

The introduction of safety cases into the practice of safety assurance aims to make explicit and to organize the justification for a claim that some engineered artefact is safe. (What 'safe' means is, however, a totally different issue, which we choose to

V. Cassano · T. S. E. Maibaum (✉) · S. Grigorova
McMaster Centre for Software Certification, McMaster University, Hamilton, Canada
e-mail: tom@maibaum.org

V. Cassano
e-mail: cassanv@mcmaster.ca

S. Grigorova
e-mail: grigorsb@mcmaster.ca

Y. Ait-Ameur et al. (eds.), *Implicit and Explicit Semantics Integration in Proof-Based Developments of Discrete Systems*,
https://doi.org/10.1007/978-981-15-5054-6_11

set aside here.) A safety case is defined as: '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' (see [35]). This widely accepted definition of a safety case as a structured argument is a big step towards a more precise definition. Moreover, it sets a standard against which competing definitions of a safety case can be assessed [16]. See also Sect. 2.

A significant amount of useful work has been accomplished in turning the idea that a safety case is a structured argument into practical notations to support their development (see [1, 33]). However, to date, existing notations for safety cases have no semantics. This entails that, when presented with a safety case in one of these notations, we have no means for deciding whether the structured argument it formulates is syntactically well formed, never mind whether the reasoning it purports to represent is sound. In other words, though safety cases have been developed and used with some success, they seem to be largely supported by intuition and experience. This is in stark contrast to other disciplines of engineering where mathematical rigour is the norm. The situation is worrisome. Can we reasonably expect to deal with the increasing complexity of systems such as cyber-physical systems or autonomous cars largely based on intuition? It is well known that, in the end, intuition always fails us when confronted by complexity. Would we have entrusted the lives of astronauts to outer space missions had space shuttles been engineered based on intuition and not science? Of course not.[1] So why do we not hold the development of safety cases to the same high standards? It does certainly seem to be appropriate. Moreover, how do we teach new safety engineers the necessary rigour required in their field without a proper scientific basis? Do we appeal to intuition and experience? Intuition only takes us so far, and certainly not far enough to justify the safety of complex systems. The moral of the story is that history clearly demonstrates that notations lacking a well-founded semantics are deficient w.r.t. the requirements of a serious approach to engineering. This state of affairs in safety assurance has persisted for too long. We consider that it is time to bring this issue to the fore.

We do not think that the use of the term 'structured argument' is incidental in the definition of a safety case.[2] For this reason, our view is that a well-founded semantics for safety cases should be based on logical principles in the form of a logical calculus. We view the development of this logical calculus in the light of Logic Engineering (see [36]). Logic Engineering addresses the development of logical frameworks for specific purposes. In our case, the specific purpose is safety reasoning. As logic engineers, we then need to identify among the available logical calculi if there is one that is adequate for safety reasoning, and in case there is none, to construct one (possibly by combining or borrowing elements from those that exist).

[1] This does not mean that we expect engineering to be perfect. Engineers do make mistakes. However, engineers learn by experience and codify that knowledge in mathematical analyses and engineering methods, on which they can rely to build systems that are reliable.

[2] Tim Kelly, the developer of one of the most commonly used notations for safety cases, the Goal Structured Notation, see [33], and his PhD. supervisor John McDermid in [25] directly linked the notation to the argument language developed by Toulmin in [34]. Moreover, the UK MoD standard definition of a safety case, see [35], also links safety cases to arguments.

As a disclaimer, we do not propose to reduce safety reasoning to an existing logical calculus, nor will we fully develop a logical calculus for safety reasoning; we do not yet know enough to do so. Instead, we take some steps towards a logical calculus for safety reasoning by presenting and discussing some of its main elements in the form of *working definitions*. More precisely, we provide a precise definition of the notion of a structured argument in a safety case and discuss some of the elements of safety reasoning in relation to it. An important part of this reasoning is the essential role that evidence plays in safety cases. Evidence sets apart the kind of arguments involved in safety cases from their traditional logical counterpart, as assumptions in safety cases must be grounded on (i.e., justified by) data from the empirical world. An important source for ideas in this regard is the work of epistemologists such as Carnap, Hempel, and Popper, amongst others, see [5, 18, 29]. In particular, we have taken inspiration from Carnap's Two Level Language of Science. This language is a logical formalism that has a limited logic for observational reasoning, i.e., about evidence, which is included in another language, the so-called theoretical language, that is used for reasoning about universal generalizations. The current observations about safety cases and the distinctions between reasoning about evidence versus inferential reasoning of a more general nature are directly rooted in the ideas outlined above.

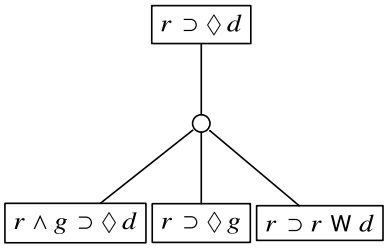
Our work has two main outcomes. First, we set up a framework for discussing the kind of reasoning involved in safety cases. Second, we set up a standard against which progress can be measured by providing some working definitions. Working definitions are the basis of science and engineering and are an essential tool against which to measure scientific progress. Working definitions allow us to make further progress in transforming safety cases into a properly grounded engineering tool, enabling a systematic and rigorous construction and analysis. But, of course, our working definitions should not be seen as defining a dogmatic position; we will happily make changes as we learn more and are able to justify their necessity.

Structure: In Sect. 2, we begin with some preliminary observations on safety cases. In Sects. 3 and 4, we discuss some of the main elements of safety reasoning and offer some working definitions. In Sect. 5, we put these elements together in an attempt to present a coherent picture. In Sect. 6, we offer some conclusions and comment on next steps.

2 Preliminary Observations

The first observation regarding safety cases concerns notations for their presentation/development (e.g., [1, 33]). To save us from having to continually refer to all of them, we use the diagrams in the Goal Structured Notation (GSN) as our witness. In our view, GSN diagrams do not present arguments in the usual logical sense of the term. Instead, they present decomposition structures for safety goals, i.e., a strategy S related to goals $\{G_i\}_{i\in n}$, G expresses a decomposition of G into $\{G_i\}_{i\in n}$. This is reminiscent of problem solving by decomposition, a well-known technique for coping

Fig. 1 Goal decomposition



with the complexity of large problems (see [28]), where solutions to sub-problems are combined in a prescribed way to solve the original problem. More closely, this is reminiscent of goal structured requirements approaches such as KAOS (see [23]), which apply problem decomposition ideas to requirements definition. (In fact, GSN researchers often refer to the notation as supporting safety goal decomposition.) But decomposition structures for safety goals and structured arguments in safety cases are different: while a goal decomposition structure breaks down a complex goal G into more manageable goals $\{G_i\}_{i \in n}$, as stated in some underlying logical calculus, a structured argument substantiates that G follows from the set $\{G_i\}_{i \in n}$. This difference is immediate if we draw an analogy with goal decomposition structures in KAOS. The following example, adapted from one presented in [9], clarifies this point. Figure 1 illustrates the decomposition of a goal G stating that 'if the train is *r*eady to depart, then, it eventually *d*eparts' into goals G_1, G_2, and G_3 stating that 'if the train is *r*eady to depart and the go signal turns green (*g*o), then, it eventually *d*eparts'; 'if the train is *r*eady to depart, then, the go signal turns green (*g*o) eventually'; and 'if the train is *r*eady to depart, then, it remains in that state at least until it *d*eparts'; respectively. In Fig. 1, goals G and $G_1 - G_3$ are formulated in ($\mathcal{TK}$; see [24]) as $r \supset \Box \Diamond d$, $r \wedge g \supset \Diamond d$, $r \supset \Diamond g$, and $r \supset r \mathsf{W} d$, respectively. Figure 2 illustrates what we consider to be a structured argument in the context of this goal decomposition. In Fig. 2, single lines correspond to inference steps of $\mathcal{TK}$, and double lines to the use of lemmas in $\mathcal{TK}$, i.e., combinations of inference steps (see [24]). Though related, it can be readily seen that decomposition structures for goals and structured arguments are not the same. GSN diagrams fall short at presenting structured arguments. So-called strategies do not involve structured arguments akin to that presented in Fig. 2, and what they do better at representing, a goal decomposition structure, while very important, is hindered, from a scientific and engineering point of view, by not having a properly defined semantics (as exists in KAOS).

Another popular notation for safety cases is CAE developed by Adelard (see [1]). The ideas and motivations are similar to those of GSN in many ways. In CAE claim decompositions can be viewed as logical conjunctions of the sub-claims. This links directly intended logical meaning of claim decomposition to reasoning about the claims. However, such a 'semantics' of claim decomposition is very limiting and simplistic. As noted above, in problem solving by decomposition, the way sub-solutions are put together to obtain a solution for the original problem must be well defined, but the composition method may be more complicated than conjunction.

Fig. 2 Structured argument

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{r \supset (r \mathsf{W} d)}}{r \supset ((r \mathsf{U} d) \vee \Box r)}}{r \supset (\Diamond d \vee \Box r)} \quad r \supset \Diamond g}{r \supset (\Diamond g \wedge (\Diamond d \vee \Box r))}}{r \supset ((\Diamond g \wedge \Diamond d) \vee (\Diamond g \wedge \Box r))} \quad \dfrac{r \wedge g \supset \Diamond d}{r \supset ((\Diamond g \wedge \Diamond d) \vee \Diamond(g \wedge r))}}{r \supset ((\Diamond g \wedge \Diamond d) \vee \Diamond \Diamond g)}}{r \supset \Diamond d}$$

Similarly, when we build proof procedures, another example of the principle of problem solving by decomposition, putting proofs together may be much more complex than simply 'build the conjunction' of the proofs. Breaking claims down to ones that can be put together by conjunction enormously, and unnecessarily, complicates the decomposition problem.[3]

There is also something to be said about attempts at reducing safety reasoning directly to First-Order Logic (FOL) and using automated deduction support and proof calculi for expressing a safety argument (see [31, 32]). These attempts try to provide a strong, well defined, foundation for eliciting what is meant by a structured argument. But they must also face up to the fact that safety reasoning is not FOL reasoning (and, more generally, not that captured by classical deductive logical calculi). There are several reasons for this. We proceed to enumerate some:

(i) Safety reasoning contains textbook examples of fallacies in FOL (e.g., arguments from authority, such as expert opinions). Independently of how we express them, including a fallacy in a proof renders the proof a fallacy, and thus a no-proof.

(ii) Safety reasoning makes use of inductive generalizations (as in inductive reasoning, see [12]). An obvious example of this occurs when it is concluded from a test set extracted from an universe of data, where every test case is successful, that a corresponding property of the universe of data is the case. This kind of generalization requires a truly inductive reasoning step. FOL is not the logic for dealing with inductive generalizations.[4]

[3] When trying to define how components could be composed 'in parallel', researchers also proposed that the semantics was conjunction. This was found to be very limiting, failing to deal well with interaction and communication between components and was soon replaced by the use of categorical operations, such as co-limit, applied to diagrams of components and morphisms in an appropriate category.

[4] The position that inductive generalizations correspond to reasoning at the level of evidence, that, once this is sorted out, we can move to a more 'pure' form of reasoning, and that the non-evidential part of the reasoning in a safety case can be done in FOL is difficult to sustain. There is no clear distinction between when reasoning at the level of evidence stops and when we can move to 'pure' reasoning. In fact, the related literature categorically contradicts this position. When reasoning about statements that make assumptions about evidence, it seems implausible, at least, that FOL will do.

(iii) Safety reasoning includes elements of defeasible reasoning (as discussed in the field of non-monotonic logics; see [3]). Often safety reasoning makes inferences from incomplete information, i.e., neither are we certain that a property holds for an artefact, nor that it does not, yet we still conclude something about the artefact. Moreover, safety reasoning makes use of defeasible inferences. These inferences are defeasible because further investigation may invalidate the conclusions drawn from them, forcing their revision, or withdrawal. Defeasible reasoning falls outside of the scope of FOL.

(iv) FOL is inadequate for reasoning about actions, modalities, and agency. This part addresses the idea that modal reasoning can be better dealt with in FOL. This does not mean that actions, modalities, and agency cannot be reasoned about in FOL. It simply means that they are better dealt with by logics which were developed with that particular purpose in mind (see [10, 27]). From the perspective of logic engineering, these logics provide a more suitable formalism for the task at hand.

(v) Safety case reasoning sometimes also uses a form of reasoning called eliminative induction (see [15]). Eliminative induction, first developed by Francis Bacon, and taken up by philosophers such as John Stuart Mill, John Maynard Keynes, Karl Popper, Jonathan Cohen, et al., works like this: Suppose that we conclude property A and that, at the same time, we identify that A may not be true in the presence of one or more properties $B_1, \ldots, B_n$. The set of P_is associates some uncertainty to P. If none of the P_is can be concluded, then, the uncertainty associated with P is reduced. This form of reasoning is in fact an example of a form of probabilistic reasoning that departs from the frequentist based reasoning of probability and is more related to confidence (as in confidence in a scientific theory). Confidence underlies reasoning about scientific theories, legal cases, and other domains, and some valuable lessons can be learned from those domains. For example, confidence is the basis on which semantics for statements in law like 'beyond a reasonable doubt' or 'on the balance of probabilities' can be defined. (Toulmin includes 'qualifiers' as elements in the logical statements he uses in his arguments. He would recognize the two examples we just presented as examples of qualifiers. Safety case examples are replete with qualified statements such as 'sufficiently' safe or acceptably safe.) In safety reasoning, confidence is absolutely necessary for it manifests scientifically the conventional wisdom that safety cannot be absolutely guaranteed, and, therefore, the degree of confidence becomes an essential aspect of reasoning. Again, confidence falls outside of the scope of FOL.

(vi) Safety reasoning has a global rather than a compositional, inductive, nature. Defeasible and probabilistic reasoning exhibit this particularity. In these forms of reasoning it is not generally possible to put consequences together in a soundness-preserving way (see [2]). This has grave consequences for the possibility of devising incremental safety approaches that support the well tried and understood concept of incremental design improvement (see [37]). Lack of compositionality is not a feature of FOL.

(i)–(vi) lead to the observation that FOL is not a suitable framework for safety reasoning. There might be a need to look elsewhere for a logic for safety reasoning.

To summarize, it is unsurprising that safety reasoning presents a challenging topic for research. The practical implications of this are plainly evident. Taking on this challenge, we take some steps towards establishing a logical calculus for safety reasoning.

3 Structured Arguments in Safety Reasoning

The concept of a safety case is a cornerstone of safety reasoning. But what do we exactly mean by a safety case? A safety case is commonly defined as: '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' (see [35]). The introduction of safety cases into safety reasoning is a step in the right direction. Safety cases make a serious attempt to explicate, and to provide some structure for, the inference licenses, a.k.a. rules of inference, used in guaranteeing that a system is safe. Nonetheless, a striking feature of the definition of a safety case just given is its logical vagueness. It is unclear what is to be taken as constituting a structured argument, as in, what are its defining characteristics, and how is such a structured argument to be assessed in terms of the soundness of the reasoning it involves. In this section we discuss these issues from the perspective of a logical calculus. This presentation extends and clarifies an earlier work of ours (see [6]).

3.1 Background

We begin by introducing some basic definitions and comments on Gentzen's Calculus of Natural Deduction for Classical First-Order Logic ($\mathcal{NK}$ for short; see [8, 14]). With this, we aim to provide a well-defined context for discussion. In the process, we will fix the terminology that we will use in what follows, and we will make this terminology precise.

As explained in [14, p. 291], with his $\mathcal{NK}$, Gentzen intended to provide: 'A formalism that reflects as accurately as possible the actual logical reasoning involved in mathematical proofs'. Gentzen offers as an example of this kind of reasoning:

> $(\exists x \forall y F x y) \supset (\forall y \exists x F x y)$. The argument runs as follows: Suppose there is an x such that for all y $F x y$ holds. Let a be such an x. Then for all y: $F a y$. Now let b be an arbitrary object. Then $F a b$ holds. Thus there is an x, viz., a, such that $F x b$ holds. Since b was arbitrary, our result therefore holds for all objects, i.e., for all y there is an x, such that $F x y$ holds. This yields our assertion. (See [14, p. 292].)

In essence, the program laid out by Gentzen in [14] consists of the integration of the kind of mathematical proofs carried out above in an exactly defined calculus, his

$\mathcal{NK}$. To this end, Gentzen provides precise definitions of so-called *symbols*, *expressions*, and *figures*. Symbols are the alphabet of Classical First-Order Logic (FOL for short). Expressions are the language of FOL, i.e., the set of all formulæ defined recursively over the alphabet of FOL. We will need to refer to arbitrary formulæ in the language of FOL. We indicate these arbitrary formulæ with uppercase boldface letters. Figures are *inference figures* or *proof figures*. Inference figures consist of a finite set of formulæ called upper formulæ and a single formula called a lower formula. Regarding inference figures Gentzen explains in [14, p. 291] that: 'We shall have inference figures and they will be stated for each calculus as they arise'. The permissible inference figures which make up the $\mathcal{NK}$ correspond to the well-known rules of introduction and elimination of the logical connectives of the alphabet of FOL and the law of the excluded middle (see [14, pp. 292–295]). Gentzen states these permissible inference figures via a set of inference figure schemata. An inference figure schema is to be understood as: The permissible inference figure obtains from the inference figure schema by instantiating the syntactical variables for formulæ by corresponding formulæ. Figure 3 illustrates an inference figure schema (corresponding to the introduction of material implication). Observe that in this inference figure schema $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{A} \supset \mathbf{B}$ are not sentences, they are variables or templates for sentences. Figure 4 illustrates an instance of the inference figure schema in Fig. 3. In this inference figure, $\{(\exists x \forall y F x y), (\forall y \exists x F x y)\}$ is the set of upper formulæ, instantiating $\mathbf{A}$ and $\mathbf{B}$, respectively, and $(\exists x \forall y F x y) \supset (\forall y \exists x F x y)$ is the lower formula, instantiating $\mathbf{A} \supset \mathbf{B}$.

Proof figures, also called *derivations*, combine a number of formulæ to form inference figures such that: 'Each formula is a lower formula of at most one inference figure; each formula (with the exception of exactly one: the *endformula*) is an upper formula of at least one inference figure; and the system of inference figures is non-circular, i.e., there is in the derivation no cycle [...] of formulæ of which each upper formula of an inference figure has the lower formula as the next one in the series' (see [14, p. 291]). Figure 5 illustrates the result of incorporating the mathematical proof given above in Gentzen's $\mathcal{NK}$. (Numbering annotations in Fig. 5 identify where formulæ are *discharged* and they are solely used for bookkeeping purposes.)

Introducing some further terminology that we will use later on, Gentzen calls the formulæ of a derivation that are not lower formulæ of an inference figure *initial*; the formulæ of a derivation *D-formulæ*; the inference figures of a derivation *D-inferences*; and a series of D-formulæ in a derivation, whose first formula is an initial one and whose last formula is the endformula, and of which each formula but the last is an

Fig. 3 Inference figure schema

$$\frac{\begin{array}{c}[\mathbf{A}]\\ \mathbf{B}\end{array}}{\mathbf{A} \supset \mathbf{B}} \supset\!-I$$

Fig. 4 Inference figure

$$\frac{\begin{array}{c}[\exists x \forall y F x y]\\ \forall y \exists x F x y\end{array}}{(\exists x \forall y F x y) \supset (\forall y \exists x F x y)} \supset\!-I$$

Fig. 5 Proof figure (a.k.a. derivation)

$$\dfrac{\dfrac{\dfrac{\dfrac{[\forall y F a y]^1}{F a b}\,\forall\text{-}E}{\exists x F x b}\,\exists\text{-}I}{\forall y \exists x F x y}\,\forall\text{-}I}{(\exists x \forall y F x y) \supset (\forall y \exists x F x y)}\,\supset\text{-}I_2 \qquad \dfrac{[\exists x \forall y F x y]^2}{\forall y \exists x F x y}\,\exists\text{-}E_1$$

upper formula of a D-inference figure whose lower formula is next in the series, a *branch*. Note that, in Gentzen's formulation of the $\mathcal{NK}$, it is possible for some of the initial formulæ of a derivation not to be discharged. We call such initial formulæ *premisses*. At times, we need to refer to derivations without making their structure explicit. For this purpose, we use symbol $\vdash_{\mathcal{NK}}$. We understand this symbol as a relation between sets of formulæ and formulæ. The source of $\vdash_{\mathcal{NK}}$ is the set of undischarged formulæ in the derivation, the target of $\vdash_{\mathcal{NK}}$ is the endformula of the derivation. For example, we indicate the derivation in Fig. 5 as $\{\} \vdash_{\mathcal{NK}} (\exists x \forall y F x y) \supset (\forall y \exists x F x y)$.

3.2 Some Concepts

We make some observations about Gentzen's $\mathcal{NK}$ as a prelude to what follows. First, via the integration of mathematical proofs into $\mathcal{NK}$, Gentzen provides a precise definition of what is a mathematical proof, enabling an analysis of its scope and limits. For us, the importance of this cannot be underestimated, in particular, because, to a certain extent, the notion of a mathematical proof stands in analogy with that of a structured argument in a safety case, or a safety argument for short: while a mathematical proof aims at capturing the kind of reasoning involved in mathematics, a safety argument aims at capturing the kind of reasoning involved in safety reasoning. In that respect, we consider that safety arguments should be given a definition akin to the one that Gentzen provides for mathematical proofs. Without such a definition it is impossible to judge whether a proposed safety argument is indeed such. If logic, logical methods, and their history have taught us anything at all, it is that only through the provision of precise definitions and their analyses can we avoid fallacious reasoning steps. Two of the most important results about Gentzen's definition of a derivation are the Soundness and Completeness Theorems (see [8]); having, at the very least, a soundness theorem for a logical calculus for safety reasoning would greatly improve the state of the art in this domain of knowledge.

In light of the previous paragraph, we offer some clarifications to avoid any subsequent confusion. We are not saying that mathematical reasoning and safety reasoning are one and the same. There are most definitely some points of departure between the two, some of which we have already mentioned in Sect. 2, some of which we will make clear below. Neither are we saying that without a definition of a safety argument that stands on grounds analogous to Gentzen's definition of a derivation,

safety reasoning is vacuous. Though with some reservations, even in the absence of such a definition of safety argument, we see no major reason preempting logical progress in safety reasoning. (After all, it is not as if mathematical reasoning could not be carried out before Gentzen's definition of a derivation.) Lastly, we are not saying that the aforementioned definition of a safety argument can or shall be given from the outset. This would be a clear impossibility given the current state of the art of safety reasoning. Instead, our remarks are oriented towards the formulation of a working definition of a safety argument that is (i) suitable for capturing as accurately as possible the actual logical reasoning involved in safety assurance, and (ii) amenable for the logical analyses that are needed to establish the well-formedness and the soundness of the inference licenses to be used in safety assurance. It is relative to (i) and (ii) that a Logical Engineering approach proves its worth. We hope that by discussing and refining such a working definition we can establish a strong logical foundation on which to improve safety reasoning and ultimately develop a logical calculus for safety reasoning.

One final discussion may be of importance in clarifying what we are trying to do. It has been widely recognized that safety reasoning includes at least two forms of reasoning: so-called evidential reasoning, to incorporate experimental observations that might be relevant to our conclusions about the safety of a system, and so-called inferential reasoning, to enable us to manipulate statements that are not directly about experimental data. There seems to be a consideration that inferential reasoning is logical, as in FOL, while evidential reasoning is not logical, but so-called epistemological, based on conventional probability notions (see [31, 32]). Now, though there are good reasons to distinguish the two kinds of reasoning, there seems to be no good reason to demote evidential reasoning from the realm of logic. There is a century-old history of trying to do exactly the opposite. Carnap's Two Level Language of Science was an attempt at characterizing the logic behind scientific reasoning (see [5]). As in safety cases, Carnap had to deal with the incorporation of observation in science with the more general forms of reasoning that any mathematician, scientist, or logician would recognize. He divided his logical language into two parts: one that has to do with observations, the other that has to do with general, universal reasoning, e.g., about universal laws. The observational language was of limited expressive power; it included observations as ground atomic formulae (e.g., 'this glass is blue', 'the output of this program run in this test harness, when the input is a, is b', etc); the observational logic had the usual connectives, but limited inferential power (e.g., universal generalization is not allowed), only so-called empirical generalizations (e.g., 'all the swans we have observed are white'). The so-called Theoretical Level of discourse, on the other hand, was more like FOL and allowed universal generalization. This latter logic incorporated the former. Thus, reasoning about evidence (observations) and inferential reasoning are integrated into a single, coherent whole. When we refer below to a logic of safety cases, we have in mind a logic analogous to Carnap's. It incorporates elements for evidential reasoning as well as general (inferential) reasoning. It seems to us that making evidential reasoning not logical just leaves us with the non-trivial problem of integrating the two parts.

Fig. 6 s-inference figure schema

$$\frac{\mathbf{A}_1 \ldots \mathbf{A}_n}{\mathbf{B}} \ \langle \mathbf{R} \rangle$$

Fig. 7 s-derivation

$$\dfrac{\dfrac{r \wedge g \supset \Diamond d}{r \supset (\Diamond g \supset \Diamond \Diamond d)} \qquad \dfrac{\dfrac{\dfrac{(s \ \mathsf{U} \ g) \wedge (g \ \mathsf{U} \ s)}{(s \ \mathsf{U} \ g)}}{\Diamond g} \ \langle \neg v \rangle}{r \supset \Diamond g} \qquad \dfrac{r \supset (r \ \mathsf{W} \ d)}{r \supset (\Box r \vee \Diamond d)}}{\dfrac{r \supset ((\Diamond g \wedge \Diamond d) \vee \Diamond (g \wedge r))}{r \supset ((\Diamond g \supset \Diamond \Diamond d) \wedge ((\Diamond g \wedge \Diamond d) \vee \Diamond (g \wedge r)))}}{r \supset \Diamond d}$$

Following from these preliminary observations, similarly to Gentzen's aim of incorporating mathematical proofs into a logical calculus, Gentzen's $\mathcal{NK}$, what we have in mind is also the integration of safety arguments into a logical calculus, which we refer to as $\mathcal{SK}$. This integration provides the sought after definition of a safety argument. In working towards this end goal, we make precise first the concept of a s-derivation. Reminiscent of Gentzen's derivations, s-derivations consist of a number of s-formulæ which are combined to form s-inference figures. For each s-derivation, each s-formula is a lower s-formula of at most one s-inference figure; each s-formula (with the exception of exactly one, the s-endformula) is an upper s-formula of at least one s-inference figure; and the system of s-inference figures is non-circular. We consider a s-inference figure to be an instance of the s-inference figure schema in Fig. 6.[5] In the s-inference figure schema in Fig. 6, $\mathbf{A}_1, \ldots, \mathbf{A}_n$, $\mathbf{B}$, $\mathbf{R}$ are variables for s-formulæ; the part corresponding to $\langle \mathbf{R} \rangle$ is optional. We will have to consider a particular s-inference figure schemata in the definition of our sought after $\mathcal{SK}$, and these we will have to state precisely, but we are not in a position to do so yet. Following Gentzen's terminology, for a s-inference figure schema such as the one given above, we call the instances of $\mathbf{A}_1, \ldots, \mathbf{A}_n$ upper s-formulæ and the instances of $\mathbf{B}$ lower s-formulæ. We call the instances of $\mathbf{R}$ s-rebuttals. We will return to them shortly for they occupy a special place in s-derivations. We call the s-formulæ participating in a s-derivation s-formulæ and the s-inference figures participating in a s-derivation s-inference figures. Moreover, we call the s-formulæ of a s-derivation that are not lower formulæ of a s-inference figure initial s-formulæ. The initial s-formulæ of a s-derivation can be discharged (in the sense given by the introduction of appropriate conditionals, modalities, or quantifiers) or not. We call those initial assumptions of a s-derivation that are not discharged s-premisses. At times, we need to refer to s-derivations but not their structure. For this purpose, we use symbol $\vdash_{\mathcal{SK}}$. We understand this symbol as a relation between sets of s-formulæ and s-formulæ. The source of $\vdash_{\mathcal{SK}}$ is the set of undischarged s-formulæ in the s-derivation, the target of $\vdash_{\mathcal{SK}}$ is the s-endformula of the s-derivation. We label $\vdash_{\mathcal{SK}}$ with the s-rebuttals of the s-inference figures in the s-derivation.

[5] Technically, the s-inference figure schema in Fig. 6 is a s-inference figure schemata. We obtain a s-inference figure schema for each value of n.

Fig. 8 Goal decomposition

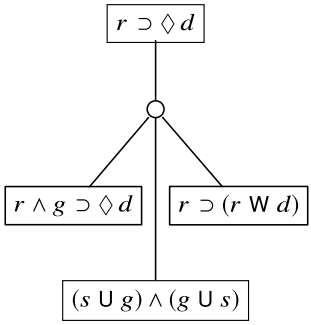


Fig. 9 s-inference Figure

$$\frac{\begin{array}{c}[r]\\ \Diamond g\end{array}}{r \supset \Diamond g}\ \langle \neg v \rangle$$

We illustrate what a s-derivation may look like in Fig. 7.[6] In this s-derivation, $\{r \wedge g \supset \Diamond d, (s \mathsf{U} g) \wedge (g \mathsf{U} s), r \supset (r \mathsf{W} d)\}$ is its set of s-premisses, and $r \supset \Diamond d$ is its s-endformula. With the exception of the s-inference figure in Fig. 9, to which we will return, the s-inference figures used in this s-derivation are obvious; they are inference figures of Temporal Logic (see [24]). We indicate the s-derivation in Fig. 7 as $\{r \wedge g \supset \Diamond d, (s \mathsf{U} g) \wedge (g \mathsf{U} s), r \supset (r \mathsf{W} d)\} \vdash_{\mathcal{SK}}^{\{\neg v\}} r \supset \Diamond d$.

We can understand the s-derivation in Fig. 7 in light of the goal decomposition in Fig. 8. In this goal decomposition, we broke the top goal $r \supset \Diamond d$ into goals $r \wedge g \supset \Diamond d$, $r \supset (r \mathsf{W} d)$, and $(s \mathsf{U} g) \wedge (g \mathsf{U} s)$. We borrowed $r \supset \Diamond d$, $r \wedge g \supset \Diamond d$, and $r \supset (r \mathsf{W} d)$ from the goal decomposition example in Sect. 2 and they have the same intuitive meaning. With $(s \mathsf{U} g) \wedge (g \mathsf{U} s)$ we capture the idea that 'the go signal turns from green (go) to red (stop) and from red (stop) to green (go)'. With $\neg v$ we capture the idea that 'the go signal is not visible to the operator of the train'. Relative to the goal decomposition in Fig. 8, the s-derivation in Fig. 7 identifies clearly and definitely a structured argument substantiating that the top goal follows from the (sub)goals it has been broken into.

The definition of $\mathcal{SK}$ concludes with the definition of the language of s-formulæ, and with the formulation of the permitted s-inference figures via s-inference figure schemata. We envision the language of s-formulæ as the $\mathcal{SK}$ counterpart of the claims involved in safety arguments, safety claims for short, and the permitted s-inference figures as the $\mathcal{SK}$ counterpart of the inference licenses used in the formulation of safety arguments. Their precise formulation is, however, an open research question and part of what makes the definition of a safety argument, via its integration into an exactly defined calculus, a working definition.

[6] In the s-derivation in Fig. 7, we assume that the language of Temporal Logic is part of the language of s-formulæ and that the inference figures of Temporal Logic are permissible (see [24]).

3.3 Some Comments on the Logic of Safety Arguments

A significant and non-trivial part of our $\mathcal{SK}$ needs to be completed. We need to: (i) provide a formal definition of s-formulæ; (ii) formulate the s-inference figure schemata for the permissible s-inference figures of $\mathcal{SK}$; and, more importantly, (iii) integrate a basic stock of examples into $\mathcal{SK}$. However, even at this early stage, the definition of a s-derivation allows us to discuss technically certain important issues regarding safety reasoning.

3.3.1 Regarding s-Formulæ

The following observation made by Gentzen in [14] provides some context for discussion: 'To the concept of "object", "function", "predicate", "theorem", "axiom", "proof", "inference", etc., in logic and mathematics there correspond, in the formalization of these disciplines, certain symbols or combinations of symbols'. What Gentzen implicitly assumes is the translation of some ordinary language of mathematics into the formal language FOL. Arguing about the faithfulness of the translation of statements in the ordinary language of mathematics into that of FOL is a moot point, first, because, to a large extent, the language of FOL has been designed having in mind the ordinary language of mathematics, and second, because statements in mathematics are rigorously precise and unambiguous. After all, no one will doubt that the ordinary statement of mathematics 'there is no natural number whose successor is zero' is expressed by the formula $\neg\exists n(S(n) = 0)$.

More generally, a faithful translation of an ordinary language, such as English, into a formal language, such as that of FOL, brings with it a number of non-trivial issues to address. In particular: Is there then a formal language in which to provide a precise definition of s-formulæ that caters for a faithful translation of safety claims formulated, say, in plain English? The answer to this question is, however, non-trivial.

It is not at all clear how to faithfully translate logical connectives in an ordinary language such as English into a formal language. For instance, we have chosen to translate the English claim 'if the train is *r*eady to depart, then, it eventually *d*eparts' into the formula $r \supset \Diamond d$. The problem with this is that, if we assume that the inference figure schemata ruling the introduction and elimination of $\supset$ in the $\mathcal{SK}$ are similar to those in Gentzen's $\mathcal{NK}$ for $\supset$, i.e., if $\supset$ is like the material conditional, then, we can establish $r \supset \Diamond d$ from $r \vee b \supset \Diamond d$. But there is something counter-intuitive in this situation; in particular, if we understand the formula $r \vee b \supset \Diamond d$ as a faithful translation of the English claim 'if the train is *r*eady to depart or it is *b*roken, then, it eventually *d*eparts'; for, clearly, we would not want a broken train to depart. The problems of conditional statements in ordinary English and material implication discussed in [7] offer some further food for thought on this issue.

To further complicate matters, a quick perusal of some safety claims reveals a heavy use of vaguely defined modal logical connectives, e.g., 'acceptably', 'sufficiently', 'adequately', in combination with quantifiers of a restricted nature, e.g.,

'All identified hazards'. It is well known in classical logical studies that these are not easily dealt with, and adding modal logical connectives intertwined with logical quantifiers to the mix does not simplify matters.

In addition to the above, there are also issues related to reasoning about actions, and about qualifiers on actions, that pose some challenges in their own right.

At this point, some may wonder: Why should we even bother in developing and proposing a formal language of s-formulæ if it is so devilishly complicated? First, because formal languages are unambiguous, easier to provide a clear semantics for, and, ultimately, more amenable to analyses and tool support. Second, because the unrestricted use of ordinary languages, e.g., English, is known to be prone to paradoxes, e.g., 'This sentence has five words', or the heinous 'This sentence is false'.

We are of the opinion that a version of a paradox of language is already present in safety reasoning. To explain this observation, we draw an analogy between reasoning about safety, and reasoning about correctness in Hoare's Calculus ($\mathcal{HK}$; see [20]).

Hoare's $\mathcal{HK}$ is a formalism enabling us to reason deductively about programs. Its lore involves claims such as: 'The program $\mathtt{S}$ is correct w.r.t. its precondition P and its postcondition Q'. But we note here an important point: There is no formula in the formal language of Hoare's $\mathcal{HK}$ capturing such claims about correctness. The formal language of Hoare's $\mathcal{HK}$ consists of triples $\{P\}\ \mathtt{S}\ \{Q\}$. These triples are the formal counterpart of claims of the form: 'If (the precondition) P is true before the initiation of (the program) $\mathtt{S}$, then, (the postcondition) Q will be true upon the completion of $\mathtt{S}$'. Claims such as 'The program $\mathtt{S}$ is correct w.r.t its precondition P and its postcondition Q' are formulated outside of Hoare's $\mathcal{HK}$ and refer, from outside the calculus, to the existence of a derivation, inside the calculus, which has the triple $\{P\}\ \mathtt{S}\ \{Q\}$ as an endformula. In other words, 'The program $\mathtt{S}$ is correct w.r.t its precondition P and its postcondition Q' is defined to be 'There is, in the $\mathcal{HK}$, a derivation which has the triple $\{P\}\ \mathtt{S}\ \{Q\}$ as an endformula'. Figure 10 illustrates what a derivation in Hoare's $\mathcal{HK}$ looks like. (In this figure, single lines correspond to inference steps of Hoare's $\mathcal{HK}$, and double lines to the use of lemmas in the calculus, i.e., combinations of inference steps).

The point is that the claim 'The program $\mathtt{x} := \mathtt{x} + \mathtt{y}; \mathtt{y} := \mathtt{x} - \mathtt{y}; \mathtt{x} := \mathtt{x} - \mathtt{y}$ is correct w.r.t. its precondition $\mathtt{x} = X \wedge \mathtt{y} = Y$ and its postcondition $\mathtt{x} = Y \wedge \mathtt{y} = X$' refers to the derivation in Fig. 10. But the claim of correctness does not belong to the language of Hoare's $\mathcal{HK}$. Including a formula that can refer to correctness inside of Hoare's $\mathcal{HK}$ yields a calculus which can refer to its own notion of derivation, giving rise to all sorts of logical problems, not to mention fallacies.

Our observation is that, though programs and systems are distinct entities, and so is reasoning about correctness and safety, we consider that in the same way that the correctness of a program in Hoare's $\mathcal{HK}$ refers to the existence of a derivation in the calculus, a statement about the safety of a system, whether acceptably, sufficiently, adequately, etc., refers to the existence of a s-derivation and, as such, it is not part of the s-derivation itself. In other words, a goal such as 'The system is acceptably/sufficiently/adequately safe' can never be the top-level goal of any safety argument. Instead, this top-level goal should correspond to a property akin to precon-

$$\dfrac{\dfrac{}{\{\mathtt{x} = X \wedge \mathtt{y} = Y\}\ \mathtt{x} := \mathtt{x} + \mathtt{y}\ \{\mathtt{x} = X + Y \wedge \mathtt{y} = Y\}} \qquad \dfrac{\dfrac{}{\{\mathtt{x} = X + Y \wedge \mathtt{y} = Y\}\ \mathtt{y} := \mathtt{x} - \mathtt{y}\ \{\mathtt{x} = X + Y \wedge \mathtt{y} = X\}} \quad \dfrac{}{\{\mathtt{x} = X + Y \wedge \mathtt{y} = X\}\ \mathtt{x} := \mathtt{x} - \mathtt{y}\ \{\mathtt{x} = Y \wedge \mathtt{y} = X\}}}{\{\mathtt{x} = X + Y \wedge \mathtt{y} = Y\}\ \mathtt{y} := \mathtt{x} - \mathtt{y}; \mathtt{x} := \mathtt{x} - \mathtt{y}\ \{\mathtt{x} = Y \wedge \mathtt{y} = X\}}}{\{\mathtt{x} = X \wedge \mathtt{y} = Y\}\ \mathtt{x} := \mathtt{x} + \mathtt{y}; \mathtt{y} := \mathtt{x} - \mathtt{y}; \mathtt{x} := \mathtt{x} - \mathtt{y}\ \{\mathtt{x} = Y \wedge \mathtt{y} = X\}}$$

Fig. 10 A derivation in Hoare's $\mathcal{HK}$

ditions/postconditions in Hoare's $\mathcal{HK}$. After all, fallacious reasoning begins with the use of formulæ that are, from the point of view of the candidate calculus, already logically problematic.

The moral of the story is that we should exercise great care in the formulation of safety claims, and what they are about, to avoid the kind of problems mentioned above, or others of a similarly problematic logical nature. As a first step, we may choose to restrict the formulation of safety claims to fragments of ordinary languages, such as English, that are expressive enough to capture the safety claims that we need, but that maintain a reasonable degree of logical tractability. This would provide us with a basis on which to engineer a formal language for s-formulæ, and a corresponding formal semantics, which caters for a faithful translation of safety claims, catering for a thorough and systematic understanding of the sort of claims involved in safety reasoning and how to reason about them.

3.3.2 Some Thoughts on Toulmin

The definition of a s-inference figure given in Fig. 6 provides a necessary level of technicality for putting in context and discussing the appeal to Toulmin's argument patterns (see [34]) in the formulation of a safety argument, an important topic at the heart of notations for safety arguments.

Let us recall some of the basics of Toulmin's notion of an argument pattern. In [34], Toulmin asks himself: How should we lay an argument out, if we want to show the sources of its validity? In answering this question, Toulmin identifies the following elements: claim (C), data (D), warrant (W), qualifier (Q), rebuttal (R), and backing (B). Resorting to this basic stock of concepts, Toulmin lays out his famous notion of an argument pattern as in Fig. 11.

A very simple explanation of these concepts is offered in [19]. There, it is explained that Toulmin articulates his argument patterns in the context of justifying an assertion in response to a challenge. The challenge starts with the assertion of a claim (C), of which we may be asked: What have we got to go on? To which we would offer the data (D). Upon offering the data (D), we may be asked: How do you get there? (How

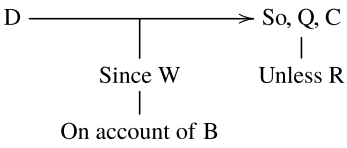


Fig. 11 Toulmin's argument pattern

do we get from D to C?) To which we would present the warrant (W). The warrant is, thus, what allows us to infer the claim from the data. Warrants may be qualified by modalities (Q), e.g., 'probably', 'generally', 'necessarily, or 'presumably'. If the warrant is defeasible, i.e., open to revision or annulment, then, we would state the conditions of rebuttal (R). Lastly, we may also be asked for a justification of the warrant itself, to which we would present the backing (B).

Setting aside Toulmin's notion of a backing (B), it is not difficult to see that, though with some restrictions, our formulation of a s-inference figure schema in Fig. 6 borrows elements from Toulmin's argument patterns and articulates them in Gentzen's terminology. More precisely, incorporating the modalities (Q) into the logical connectives of the language of s-formulæ, the set of $\mathbf{A}_i$s, **B**, and **R** in Fig. 6 can be viewed as standing in analogy with Toulmin's triple of data (D), claim (C), and rebuttal (R) in the obvious way, i.e., D relates to the conjunction of the $\mathbf{A}_i$s, C relates to **B**, and R relates to **R** (this is the reason why we named the instances of **R** rebuttals). Toulmin's notion of a warrant can be viewed as standing in analogy with the s-inference figure schema in Fig. 6.

The restrictions that we referred to above are linguistic and logical constraints on the kind of rebuttals allowed. According to Toulmin, rebuttals indicate circumstances in which the general authority of the warrant would have to be set aside (see [34, p. 94]). There are, at least, two possible ways in which Toulmin's view of a rebuttal can be understood. First, (i) as indicating a set of circumstances in which the *claim* licensed by the warrant would have to be set aside. Second, (ii) as indicating a set of circumstances in which the *warrant itself* would have to be set aside. The analogy between a warrant and a s-inference figure schema allows for the following clarification: (i) Implies that an instance of the s-inference schema cannot be used in a particular s-derivation; (ii) Implies that the s-inference schema cannot be part of the s-inference figure schemata defining the $\mathcal{SK}$. When understood in this sense, (i) speaks to the defeasible aspect of s-derivations, whereas (ii) results in a denial of the proposed calculus (intuitionistic reasoning, arising as a result of rejecting the principle of the excluded middle, see [8]; or the various systems of Deontic logic, arising in view of the so-called paradoxes of obligations and contrary-to-duties, see [26], are examples of the second kind of rebuttals). In defining rebuttals as s-formulæ, and under the proviso that the language for s-formulæ cannot refer to properties of the $\mathcal{SK}$, we preempt the formulation of rebuttals of the second kind. Including rebuttals of the second kind makes room for paradoxes of language, as they can refer to s-derivations. Paradoxes of language are something that we clearly wish to steer

away from. We consider that this restriction presents a firmer basis on which to start building the $\mathcal{SK}$.[7]

The relation between Toulmin's argument patterns and s-inference figures places the work of Toulmin in the context of safety reasoning: Toulmin's argument patterns present a framework in which to formulate what s-inference figures, or s-figure schemata, may look like. However, Toulmin's argument patterns are not s-inference figures, nor s-figure schemata. This means that Toulmin's argument patterns do not define, at least not obviously, a calculus for safety reasoning, our sought after $\mathcal{SK}$. Such a calculus, which we view as a fundamental tool for analysing the logical well-formedness of safety arguments, is only defined by the provision and justification of a sensible set of s-inference figures via a set of s-inference figure schemata. In other words, the appeal to Toumin's argument patterns in the context of safety reasoning is rather limited; it serves as a way of showing the sources of validity of a safety argument, but it does not propose a way of assessing the validity of said sources.

3.3.3 Regarding s-Derivations

Two immediate questions may be asked about the $\mathcal{SK}$: (i) Are s-derivations suitable as the formal counterpart of safety arguments? (ii) Are s-derivations suitable for supporting the logical analyses needed to establish the well-formedness and the soundness of the inference licenses used in safety arguments?

Our answer to question (i) is, at this point, an expression of desire. Evidently, we do consider that s-derivations present a suitable framework for incorporating safety arguments. This view is partly justified by the intent of notations such as the GSN or the CAE. Whether this view is fully justified is debatable. So far, we have been unable to produce an example of the incorporation of a safety argument as a s-derivation. This is, partly, due to our own limitations, to the lack of a language for s-formulæ, and to the logical rigour that we intend to put in place in the integration of a safety argument into a s-derivation (something that we hope to improve on); but this is also due to the logical havoc reigning over the handful of examples of safety arguments that we have inspected in detail (something that we expect to shed some light on).

Our answer to question (ii) is, even at this point, more satisfactory. In particular, the concept of an s-derivation enables us to discuss some basic notion of well-formedness. Let π_1 and π_2 be two s-derivations; if the s-premisses of π_2 are a subset of the s-premisses of π_1 and the s-endformula of π_2 belongs to the set of rebuttals of the s-inference figures in π_1, then, we call π_2 a rebutting derivation for π_1. We call a s-derivation internally coherent in the absence of a rebutting s-derivation for it.[8,9] The

[7] If we really wish to revise a logic supporting safety reasoning by revising its inference rules, then this is a logic engineering job and we have not thought about what this may entail.

[8] We are calling a s-derivation internally coherent in the absence of a rebutting s-derivation for it, not in the presence of a proof that such a rebutting s-derivation does not exist; the latter is far more difficult to establish.

[9] Internally coherent s-derivations also make precise the role of rebuttals. They are not negated premisses, nor premisses of any kind; they are a source of defeasibility. To consider rebuttals as

second footnote is in response to the comment that rebuttals are negated assumptions. Obviously, s-derivations that are internally incoherent are logically ill-formed.

The concept of internal coherence makes precise in which sense a s-derivation is defeasible, i.e., open to revision or annulment. To illustrate this point, let π be the s-derivation in Fig. 7. As it stands, π is internally coherent. It remains so even if we extend its set of s-premisses. However, if we are, from the extended set of s-premisses, able to construct a s-derivation with $\neg v$ as its s-endformula, then, π is no longer internally coherent. Losing this status is a direct result of the s-inference figure in Fig. 9. Intuitively, this s-inference figure may be read as: If $\Diamond g$ has been established by means of s-premiss r, we have $r \supset \Diamond g$, now without the s-premiss r. The rebuttal $\neg v$ states the conditions under which this inference license is locally inapplicable, i.e., in those situations in which there is also a derivation of $\neg v$, i.e., in those situations in which it is possible to establish that 'the go signal is not $\underline{v}$isible'. The s-inference figure in Fig. 9 is 'locally inapplicable' for there are situations in which its use is perfectly permissible (e.g., in Fig. 7).

To be noted, the discovery of a rebutting s-derivation π_1 for a s-derivation π calls for a revision of π as a *whole*, and *possibly* establishes its annulment; as a 'whole', because s-inference figures in s-derivations are not, in general, localized to parts of the s-derivation; and, 'possibly', and not 'necessarily', because even in the presence of a rebutting s-derivation, we may still be able to 'repair' the original s-derivation, e.g., by resorting to s-inference figures not affected by the rebutting s-derivation.

The discussion on internal coherence is important for two main reasons. First, because it sets apart safety reasoning from mathematical reasoning; the former is defeasible while the latter is not. Second, because it has a bearing on compositional safety reasoning. Let us explain this with an example. Suppose that the s-derivation (i) $\{r \wedge g \supset \Diamond d, (s \mathsf{U} g) \wedge (g \mathsf{U} s), r \supset (r \mathsf{W} d)\} \mathrel{\sim}^{\{\neg v\}}_{\mathcal{SK}} r \supset \Diamond d$ in Fig. 7 results from composing two s-derivations, (ii) $\{(s \mathsf{U} g) \wedge (g \mathsf{U} s)\} \mathrel{\sim}^{\{\neg v\}}_{\mathcal{SK}} r \supset \Diamond g$, and (iii) $\{r \wedge g \supset \Diamond d, r \supset \Diamond g, r \supset (r \mathsf{W} d)\} \vdash_{\mathcal{SK}} r \supset \Diamond d$, in the obvious way, i.e., by gluing (ii) and (iii) together at $r \supset \Diamond g$. This composition is reminiscent of Gentzen's Cut Rule.[10] In this case, two internally coherent s-derivations, i.e., (ii) and (iii), are composed into an internally coherent s-derivation, i.e., (i). But this is not the general case, as there is no guarantee that we will not be able to obtain the rebuttal of one of the s-inference figures in one of the composing s-derivations from the joint

premisses has some drawbacks. Suppose that π is an s-derivation with rebuttal $\mathbf{R}$. Let π be internally coherent, i.e., we do not have an s-derivation π' of $\mathbf{R}$ from the premisses of π. Adding $\mathbf{R}$ to the premisses of π implies that we need to discharge it in some conditional; otherwise, we are changing the set of premisses from which we argue. Not discharging an added rebuttal has the unwanted consequence that it makes any s-derivation incoherent in the presence of the rule of reflexivity, i.e., we can always conclude what is in the premisses. Adding the $\neg\mathbf{R}$ also has unwanted effects. If it so happens that from the premisses of π we can prove $\mathbf{R}$, but we have not done so yet, e.g., because we have not found such a s-derivation, adding $\neg\mathbf{R}$ to the premiss set of π means that we now have to deal with a premiss set that involves a glaring contradiction, i.e., $\mathbf{R}$ and $\neg\mathbf{R}$. The moral of the story is: rebuttals occupy a special place in s-derivations as sources of defeasibility; considering them as part of the premisses of a s-derivation needs to be done with extreme care.

[10] Gentzen's Cut Rule: If $\{\mathbf{A}_i\} \vdash_{\mathcal{NK}} \mathbf{B}$ and $\{\mathbf{B}_j\} \cup \{\mathbf{B}\} \vdash_{\mathcal{NK}} \mathbf{C}$, then, $\{\mathbf{A}_i\} \cup \{\mathbf{B}_j\} \vdash_{\mathcal{NK}} \mathbf{C}$ (see [14]).

set of premisses of the composed s-derivations. In other words, the composition of internally coherent s-derivations to form a larger s-derivation may result in the larger s-derivation being internally incoherent; and in order to establish whether the larger s-derivation is internally coherent or not, we may have to revise this s-derivation as a whole. In any case, compositionality is lost.

The preceding discussion on internal coherence also allows us to discuss technically the use of a form of *eliminative induction* in safety reasoning. As we mentioned in Sect. 2, eliminative induction was first developed by Bacon, and taken up by philosophers such as Mill, Keynes, Popper, et al. A reference to eliminative induction in safety reasoning is [15]. Briefly, eliminative induction works like this: Let us suppose that we conclude a property P and that, at the same time, we identify that this may not be the case in the presence of one or more properties P_i. The set of P_is associates some uncertainty to P. If none of the P_is can be concluded, then, the uncertainty associated with P is reduced. In the context of s-derivations and internal coherence, eliminative induction takes the following form: For a s-derivation π, the property P corresponds to the s-endformula of π. Each of the P_is corresponds to a rebuttal of π. Let us now suppose that π is internally coherent, i.e., that we have not found a rebutting s-derivation for it. Internal coherence associates some uncertainty to π, i.e., that expressed by its rebuttals. Since internal coherence alone does not establish that there are no rebutting s-derivations for π, simply that we have not found them, and since establishing that there are no rebutting s-derivations for π is non-trivial, instead, as a form of eliminative induction, we can attempt to construct s-derivations whose s-enformulæ are the negations of the rebuttals of π. The latter s-derivations enable us to reduce the uncertainty associated with π, and thus with P. This form of eliminative induction involves the presentation of a set of s-derivations. In this set, one s-derivation is designated as a main s-derivation. The assumption is that there is some uncertainty associated with this main s-derivation, as indicated by its rebuttals. The remaining s-derivations in the set are intended to reduce this uncertainty. This form of reasoning is an example of a form of probabilistic reasoning related to confidence, a topic that we discuss in more detail in Sect. 4.2.3. Let us illustrate this view of eliminative induction with a simple example. As it stands, the s-derivation $\{r \wedge g \supset \Diamond d, (s \mathsf{U} g) \wedge (g \mathsf{U} s), r \supset (r \mathsf{W} d)\} \vdash_{SK}^{\{\neg v\}} r \supset \Diamond d$ in Fig. 7 is internally coherent. There is, associated to this s-derivation, some uncertainty, namely, that indicated by $\neg v$. Recall that this s-derivation corresponds to an argument which concludes that 'if the train is *r*eady, then, it eventually *d*eparts', that one of the s-inference figures is contingent on the go signal being visible, and that the s-formula $\neg v$ corresponds to the property 'the go signal is not *v*isible'. To reduce this uncertainty, we can focus on constructing a s-derivation having v as its s-endformula, i.e., an argument which concludes that 'the go signal is *v*isible'.

3.4 Some Final Remarks on Safety Arguments

There are some final remarks about the difference between s-derivations and GSN diagrams that we can only elucidate at this point.

First, GSN strategies have no concept analogous to that of discharging an initial s-formula. This limitation severely restricts most forms of conditional reasoning. Is conditional reasoning forbidden in safety cases? How are we to reason conditionally without suitable mechanisms for introducing and discharging conditionals?

Let us digress for a moment to the issue of an initial formula being discharged to explain its ramifications in some detail. We begin by discussing what is the case in Gentzen's $\mathcal{NK}$. In Gentzen's $\mathcal{NK}$, discharging an initial formula means: (i) incorporating said formula into the lower formula of some inference figure in the derivation and (ii) eliminating said formula from the set of premisses of the derivation. Though (ii) is not necessary, keeping initial formulæ that have been discharged as part of the premisses of a derivation is superfluous; and this is something that we wish to avoid (see [8]). In fact, what Gentzen is after with his $\mathcal{NK}$ is a derivation that is *logistic*, i.e., one in which all initial formulæ in a derivation are discharged (see [14, p. 295]). To achieve this, Gentzen proposes to convert any non-logistic derivation π_1 into a logistic derivation π_2 whose endformula is an instance of $\mathbf{A} \supset \mathbf{B}$; in this instance of $\mathbf{A} \supset \mathbf{B}$, the instance of $\mathbf{B}$ is the endformula of π_1, and the instance of $\mathbf{A}$ is the conjunction of the formulæ in the set of premisses of π_1. In a more general setting, Gentzen's proposal requires the use of the Compactness and Deduction Theorems for the logical calculus in which derivations are formulated (see [8]). It is not clear to us whether or not such (meta) theorems hold for safety reasoning, i.e., whether they hold in our sought after $\mathcal{SK}$ (and we are inclined to believe that they do not). In other words, it seems that in safety reasoning we are required to be able to deal with genuine premisses, i.e., premisses that cannot be discharged. To have at hand suitable mechanisms for dealing with such premisses is of utmost importance. In addition, it is well known that different discharge policies give rise to different conditionals. For example, in the s-inference figure in Fig. 7 we allowed for the s-premiss r to be discharged vacuously (as is usually done in the introduction of the material conditional). If we forbid this, then, we obtain a form of a relevant conditional. Alternatively, if we allow for r to be discharged only once, then, we obtain a form of linear conditional (see [30]). It is clear to us that safety reasoning involves different kinds of conditionals. Discussing what discharge policies are allowed in safety reasoning may shed some light on which conditional we are referring to. How are we supposed to do this without proper mechanisms for tracking which initial s-formulæ correspond to which conditional? These issues are not at all properly dealt with in GSN diagrams.

Second, GSN diagrams have no concept analogous to rebuttals. In this sense, they are more limiting than goal decomposition structures in KAOS, which incorporate the notion of an *obstacle* to a goal (see [23]). Without rebuttals, the defeasible aspects of safety arguments are left implicit or are simply ignored.

In summary, the discussion that we have presented in this section is not a matter of logical pedantry. Instead, our discussion pinpoints some important issues to be

addressed if safety reasoning is meant to be grounded on logical principles, and it exposes the leading causes of fallacies and the challenges in safety reasoning by bringing them into the foreground with the use of appropriate logical machinery.

4 Evidence in Safety Reasoning

As we mentioned in Sect. 3, safety cases are a cornerstone of safety reasoning. In addition to structured arguments, a defining characteristic of safety cases is the use of evidence as a grounding mechanism for safety arguments. In this section, we pay close attention to the concept of evidence, to how it can be incorporated into our program for formalizing safety reasoning in the form of a logical calculus, and to some of the challenges that it brings with it.

4.1 Evidence in Safety Cases

To provide some context for discussion, let us recall some basic facts about the role of initial formulæ in Gentzen's $\mathcal{NK}$. Gentzen mentions in [14, p. 292] that a distinguishing feature of his $\mathcal{NK}$ is that derivations start from what he calls *assumptions*, to which logical deductions are then applied. Gentzen's assumptions are the initial formulæ of a derivation. As we have noted in Sect. 3.1, in Gentzen's formulation of the $\mathcal{NK}$, it is possible for the initial formulæ of a derivation to be discharged or not. We have called initial formulæ that are not discharged *premisses*. An important characteristic of the premisses of a derivation is that they are, in a sense, given *deus ex machina*. This is not the case in safety reasoning, where the safety claims from which a safety argument is built need to be provided with a rationale which justifies their postulation. In other words, the s-premisses in a s-derivation cannot be taken as being given *deus ex machina*. This is reminiscent of the notion of justified belief in studies in epistemology or scientific explanation. It is at this point that evidence makes an appearance.

The definition of a s-derivation given in Sect. 3.2 enables us to discuss the use of evidence in safety cases in technical terms. However, in order to do so, we need, first and foremost, to be (i) precise about what we mean by evidence and to be (ii) able to refer to evidence in the language of s-formulæ.

As to (i), the uses of 'evidence' that we have observed in safety arguments, in particular in those referred to as *solutions* in GSN diagrams, refer to results obtained via testing, simulation, model analyses, or other observation-based mechanisms, including past experiences. These uses regard 'evidence' as some kind of data. This view of evidence is problematic for data does not, and cannot, in and of itself, be used as a basis for constructing a safety argument. To explain this issue, we take an example presented in [38, p. 195]. In a court of law, a bloodied knife, i.e., a piece of data, can be used both by the prosecution or the defense in their respective cases.

The use of the bloodied knife in court, i.e., the use of this piece of data in court, may involve claims such as: 'the bloodied knife was found at the crime scene', 'the bloodied knife was used by the accused to stab the victim', 'the bloodied knife was planted at the crime scene', etc. The bloodied knife is a source of many such claims (some of which may be incompatible with others). What this example shows is that, in isolation, a piece of data is not a truth bearer, i.e., it cannot be assigned a truth value; a truth bearer is a claim about it. In other words, data becomes evidence, in the epistemological or scientific sense of the term, when it stands in a precisely defined testing relationship with some claims postulated about it. To avoid any confusion, we will refer to a piece of data as *evidence*, and to a claim about a piece of data as an *evidential claim*.

Let us now turn our attention to (ii). Immediately from the distinction between evidence and evidential claim, we would need *evidence terms* and *evidence formulæ* in the language of s-formulæ. Evidence terms would include, at least, constants for concrete pieces of evidence, and variables for arbitrary pieces of evidence. Evidence formulæ would include, at least, quantifiers binding variables in evidence terms. An evidence formula is *grounded* if it has no free variables (where 'free variable' has the usual meaning). An evidence formula is said to be *ground atomic* if it has no quantifiers and if its testing relationship with its evidence term is self-evident (intersubjectively agreed).

Evidence terms and evidence formulæ can be understood by drawing an analogy between terms and formulæ in the language of FOL. More precisely, in the language of FOL, terms denote objects; formulæ are the formal counterpart of claims about objects. For instance, in their standard interpretation, the terms $S(n)$, 0, denote the successor of a natural number, and the natural number zero, respectively. In these terms, the variable n is used to indicate an arbitrary natural number, and the constant 0 to indicate the number zero. In turn, the formula $\neg\exists n(S(n) = 0)$ is the formal counterpart of 'there is no natural number whose successor is zero'. In this formula, the existential quantifier binds the variable n. If we understand evidence terms and evidence formulæ in this way, the former serve as a way to denote pieces of evidence, while the latter are the formal counterpart of claims about evidence.

We are now in a position to make precise in which sense a safety argument is to be taken as being grounded on evidence. We do this in relation to s-derivations. Namely, we define a s-derivation as grounded on evidence if its s-premisses, i.e., its undischarged initial s-formulæ, are ground atomic evidence formulæ. In consequence, a safety argument is grounded on evidence if its incorporation into a s-derivation results in the latter being grounded on evidence.

4.2 Some Comments on Evidence in Safety Arguments

As with safety arguments, a significant part of the definitions of an evidence term and an evidential s-formula needs to be completed and fully worked out. Nevertheless, evidence terms and evidential s-formulæ allow us to discuss technically some aspects of the use of evidence in safety cases.

4.2.1 Regarding Ground Atomic Evidence Formulæ

We have defined a safety argument as being grounded on evidence if its incorporation into a s-derivation results in the latter being grounded on evidence. The first part of this definition corresponds to our program of making precise what is a safety argument via its incorporation into a logical calculus (our sought after $\mathcal{SK}$). The second part of this definition corresponds to our view of the use of evidence in safety cases and its logical characterization. The idea is that a ground atomic evidence formula plays a role similar to an *axiom* of a classical logical theory, i.e., a formula that is regarded as accepted or self-evident. This is precisely what a ground atomic evidence formula aims to capture. More elaborate evidence formulæ, e.g., those that are not ground atomic, must perforce involve some reasoning.

To illustrate the points above, let us suppose that the go signal example in Sects. 2 and 3, indicating whether the train can depart or not, consists, among other things, of a piece of software toggling the light from red to green, and from green to red. Let us suppose further that this piece of software is proven correct in Hoare's $\mathcal{HK}$, i.e., that there is, for this piece of software, a derivation π akin to that in Fig. 10. Technically speaking, π is not a proof that the piece of software itself is correct, but rather a proof of the correctness of a (syntactical) model of the piece of software in Hoare's $\mathcal{HK}$ in relation to some specification. But the piece of software itself and its (syntactical) model in Hoare's $\mathcal{HK}$ are different things. Now, let us suppose that we use the proof of correctness of the (syntactical) model of the piece of software in Hoare's $\mathcal{HK}$ to argue that the piece of software itself is dependable (in a more general sense than 'correct'). In this context, the former is a piece of evidence and the latter is an evidence claim. In the language of s-formulæ, we would then have an evidence term to denote the piece of evidence, i.e., π, and an evidence formula as the formal counterpart of the evidence claim, i.e., that the piece of software itself is dependable, respectively. A question that we could ask ourselves at this point is: Would this evidence formula meet the criterion of being ground atomic? The answer is *no*. The problem is that the testing relationship between the evidence term and the evidence formula is not self-evident, i.e., it already involves some reasoning, e.g., about the adequacy of the proof of correctness of the (syntactical) model of the piece of software in relation to a claim about the dependability of the piece of software itself. For this reason, the evidence formula cannot be used as a premiss in a s-derivation. What can be used as a ground atomic evidence formula is the formal counterpart of a claim along the lines of 'the (syntactical) model of the piece of software meets its specification'. It is the role of a safety argument to take us from this basic claim about evidence (possibly in conjunction with other basic claims about evidence), to the claim that the piece of software itself is dependable.

The preceding discussion shows that the burden is on finding ground atomic evidence formulæ, i.e., evidence terms and evidence formulæ whose testing relationship is self-evident. These ground atomic evidence formulæ serve as the basis on which we would construct the s-derivation that would take us to a s-endformula. The danger is that, without a proper formulation of a ground atomic evidence formula, or set thereof, a significant amount of effort needs to be devoted to eliciting in which

sense a piece of evidence relates to an evidence claim, something that is prone to error. An open question is whether the testing relationship between evidence terms and evidence formulæ is part of the $\mathcal{SK}$ or is external to it.

4.2.2 Regarding Multiple Atomically Grounded Evidence Formulæ

The discussion about ground atomic evidence formulæ raised to the surface the use of multiple pieces of evidence in relation to a single evidence claim.

To illustrate this phenomenon, let us take up again the go signal example in Sects. 2 and 3. Namely, let us suppose that the go signal, indicating whether the train can depart or not, consists, among other things, of a piece of software toggling the light from red to green, and from green to red. In addition, let us suppose that this piece of software is proven correct in Hoare's $\mathcal{HK}$, i.e., that there is, for this piece of software, a derivation π akin to that in Fig. 10. Let us suppose further that we use π to argue that the piece of software itself is dependable (in a more general sense than 'correct'). Repeating ourselves, in this context, the former is a piece of evidence and the latter is an evidence claim; which would cause us to have, in the language of s-formulæ, an evidence term to denote the piece of evidence, and an evidence formula as the formal counterpart of the evidence claim, respectively. In Sect. 4.2.1 we discussed that this evidence formula is not ground atomic for it already involves some reasoning, e.g., about the adequacy of π in relation to a claim about the dependability of the piece of software itself. Among other things, the adequacy of π in relation to a claim about dependability hinges on how faithful the model of the piece of software in Hoare's $\mathcal{HK}$ is to the piece of software itself, something which depends, in turn, on some assumptions on the piece of software itself, e.g., that arithmetic computations do not result in an overflow. The use of input/output testing data on the piece of software itself presents an interesting use of evidence to validate this kind of assumption. Moreover, this leads in a more or less natural way to the use of different input/output testing data, e.g., obtained from different testing methods, to validate the same assumption, e.g., because the different testing methods cover different aspects of the assumption. In technical terms, we are in a scenario in which a s-formula, i.e., the formal counterpart of one of the assumptions, is the s-endformula of various s-derivations, each of which has as its premisses ground atomic evidence formulæ whose evidence terms denote the different input/output testing data. To put all these different s-derivations together in one single s-derivation, we need to relax the definition of a s-derivation to allow for the upper s-formula of a s-inference figure to be the lower s-formula of more than one s-inference figure. Such a relaxation has no analogy in Gentzen-like derivations, for in traditional logical calculi, one derivation of an endformula is as good as any other. However, the situation is different in safety arguments due to the *confidence* value that we tend to associate with them. We have here an example of what has sometimes been referred to as a *multi-legged* argument (see [2]). The idea is that each leg is logically sufficient, but the legs taken together provide greater confidence in the logical result. We discuss this in Sect. 4.2.3, after we introduce some basics on *confidence measures*.

4.2.3 Evidence and Confidence

As previously mentioned, there is an inherent uncertainty associated with safety arguments. To begin with, the use of data in evidential reasoning naturally involves uncertainty. There is uncertainty in gathering data, in the processes of observation and measurement from which we obtain the data, in how the data is used, in the claims that we formulate about data, etc. For example, we often use some form of inductive reasoning to assert a universal conclusion from a finite number of observations, e.g., in testing of programs, and this inherently involves some lack of certainty in the universal conclusion, e.g., whether the test cases are sufficient to justify the conclusion. Second, multi-legged arguments are often used to reduce the uncertainty (i.e., increase confidence) in an argument, and this clearly means that we are not entirely sure of the conclusion of some arguments, no matter how stringent we might have been in developing the argument. Thirdly, the use of eliminative induction involves uncertainty of various kinds. For example, we cannot be certain that all the possibilities for issues to be examined have been discovered. We may also not be able to positively eliminate all possible cases, leaving some open as the risk involved is deemed too low to worry about. Modelling this uncertainty is key to evaluating the confidence we place in the safety argument, and in the claim that it establishes. In the previous section, we focused on rebuttals as a source of uncertainty. In this section, we focus on evidence.

The kind of uncertainty associated with evidence that we have in mind goes beyond uncertainty associated with statistical values (e.g., test cases returned the expected result 8 out of 10 times). It also includes the uncertainty associated with the way in which the evidence is obtained (e.g., the test cases are devised properly, they are executed in the right environment, the results are repeatable, etc). The latter kind of uncertainty associated with evidence is typically systematized by using acceptance criteria for the inclusion of certain data as evidence. The various confirmation measures for work products proposed in ISO 26262 provide an example of such acceptance criteria (see [21]).[11] In the cases where we have a pass/fail acceptance criterion, things are relatively simple, i.e., the data item either gets included in the safety argument, or not, i.e., the data is accepted as evidence or not. However, if there is a degree of acceptability, and a certain threshold that needs to be met in order for the data to be acceptable as evidence, we would like to have acceptability values at hand when evaluating the confidence we may place in the safety argument. For example, consider the following confirmation measure found in ISO 26262: 'The work products referenced in the safety case are available and sufficiently complete' (see [21, pt. 2, p. 21]). Checking whether the work products are *available* is not difficult, but measuring whether they are *sufficiently complete* is not trivial, as this needs to be precisely defined, so as not to be open to arbitrary interpretation. Unfortunately, this kind of precision is often missing.

[11] ISO 26262 is a safety standard developed to fit the needs of the automotive domain. The standard applies to electrical and/or electronic (E/E) systems within road vehicles (see [21]).

The above focuses our attention on one of the biggest issues that plague confidence modelling and evaluation. There is a lack of precise definitions, benchmarks, and evaluation techniques, all of which hinder the possibility of defining meaningful acceptance criteria, i.e., ones to which we can assign values. This is very similar to the situation in quality management, where vague definitions are common. Let us draw an analogy to elaborate on this point. In quality management, some concepts are associated with an abundance of definitions and quality measures. This has meant that something as simple as the efficiency of a computer program might mean completely different things to different stakeholders. In addition, since there are various models and methods for product quality assurance, e.g., various ways of measuring the efficiency of a computer program (memory usage and/or speed), any value associated with one of the quality characteristics of a product has to be accompanied by additional information in order for this value to be meaningful (see [17]). The same is clearly true for confidence modelling in the safety domain. To refer back to the example above, if an expert has stated that the safety case references work products that are sufficiently complete, this expert needs to provide a definition of *sufficiently complete* in measurable terms, and explain the measurement procedure used to arrive at this conclusion. If done in this way, the claim made by the expert can be reviewed and potentially compared on a more objective basis, an otherwise well-nigh impossible task. Of course, objective measures are difficult to come by, and sometimes relying on subjective measures is the only practical approach. But these subjective measures still ought to be given explicit definitions to enable results to be reproduced. This is crucial for the validation of confidence metrics and the measures associated with them.

In addition to providing a precise definition of the confidence metrics and the measures associated with them, we need rules for combining and decomposing those measures. One example of the latter in the safety-critical domain is the decomposition of *Automotive Safety Integrity Levels*, ASILs, in ISO 26262 (see [21, pt. 9]). In more detail, in ISO 26262, each safety requirement of the E/E system being considered has an ASIL associated with it. The stringency of the ASIL depends on the criticality of the safety requirement. Note that ISO 26262 allows for ASILs to be weakened during the decomposition of a safety requirement. In this respect, the obvious question that arises is: What is needed to guarantee that the weakened ASILs associated with the decomposed safety requirements guarantee the ASIL of the original safety requirement? The problem is that the decomposition of ASIL levels suggested in ISO 26262 has not been explicitly justified. Instead, it rests on domain experience, as does its validation. This is clearly a shortcoming. One way in which the combination of different measures is addressed in the field of quality management is through utility functions. Though the definition of such utility functions may be subjective, when explicitly defined, they enable us to reproduce results and to interpret these results in a repeatable and objective fashion.

In light of the above, and in order to make some progress, we need to start with widely agreed upon definitions of confidence and what its evaluation entails. Without this, the production of confidence measures becomes very vague and is largely based on the opinions and prejudices of experts. In [16], we find a survey of con-

fidence modelling approaches suggested for use in the safety-critical domains. All the various approaches are illustrated by means of examples. Borrowing from them, though we are not yet ready to provide a more complete working definition for this framework, we can outline some of its key elements. First, each evidence term has to be associated with a confidence value (or a tuple of values), produced as a reflection of its acceptance criteria as well as additional sources of uncertainty. The measures for these confidence values have to be precise and to meet the representation condition of measurement theory (see [11]), namely, that the mapping from the empirical domain of attributes to the formal domain of measures is a homomorphism (i.e., that the assignment of measures to attributes does not violate properties of attributes, e.g., that height does not make a baby's height bigger than a grown person's). In addition, the measurement statements must be meaningful, i.e., the truth value of the measurement statements must remain invariant under all admissible scale transformations (e.g., 'the temperature in Toronto is 20 C and is twice as much as in Buenos Aires, where it is 10 C' is not a meaningful statement as the transformation of Celsius to Fahrenheit, an admissible transformation, does not preserve the truthfulness of the statement). The measurement scales might exhibit different properties (being classified, being ordered, having quantified differences, etc.) based on the scale used. This would in turn depend on the property (subject to uncertainty) being modelled. To again make a parallel with quality management, we know that quality measures can take a number of forms (non-numeric, or quantitative, both of which are further subdivided and correspond to different scales) depending on the domain-specific content they model. In fact, due to the fact that the different types of measures sometimes cannot be meaningfully combined, it is possible that instead of a single confidence value, we end up with a tuple of confidence values. Transforming between different types of measures is not impossible, but it might not bring any added benefit, and might instead obscure some valuable information (e.g., through transforming a precise value into a range one). The difference between confidence modelling and quality management, and indeed the biggest issue, lies in the propagation of confidence measures associated with uncertainty. One possible way in which this issue may be addressed is through the use of Jøsang's Subjective Logic (see [22]). After introducing well-defined confidence measurement scales and procedures, and utility functions for combining the confidence values as well as a logic for propagating them, we should proceed to empirically validate our framework and make any necessary adjustments.

An additional challenge to modelling confidence is what some people call the three Ps: Process, Product, and People. For example, ISO 26262 states that we need to explicitly note the qualifications and level of independence of the people tasked with carrying out the confirmation measures (see [21, pt. 2, p. 12]). This may be construed as a form of multi-legged argument. Each leg of the argument corresponds to how an independent team arrives at a conclusion, the premisses of each leg would then correspond to items of evidence that have been independently obtained, or, alternatively, independently vetted. Let us illustrate this by extending our train example. Suppose that we recognized the visibility of the go sign as one of the sources of uncertainty. In other words, for whatever reason, we cannot be totally sure whether

the go sign is visible or not. This said, we do want to establish, with some level of confidence, that the go sign is visible. To reach this conclusion, we approach two experts in the field of vision inspection, who will conduct independent experiments. Each expert starts out with the same set of experiment participants (the train operators), and the same experimental environment (riding in the train alongside the train operators). Though both experts start with the same premisses, i.e., in the same setting, the individual experiments that they conduct might be devised in a different way, e.g., based on the different impediments to visibility that they might have thought of and decided to check against. For example, both experts might take into account the fact that one of the train operators uses allergy medication, reported to cause blurry vision as a side effect, but only one of them considers the use of sunglasses, and that their lens hue and tint density might negatively affect visibility in certain conditions (pink, blue, and green lenses can make red lights indistinguishable). Having the experts design their experiments independently guards against confirmation bias and leads to increased confidence in the final result, i.e., that the go signal is visible. The results, observations, claims, and the like, made by each of the experts would then be included in their own leg of the overall safety argument. Each leg by itself may not provide sufficient confidence, but when put together they reduce the uncertainty and increase the confidence in the claim that 'the go signal is visible'.

5 Discussion

In Sects. 3 and 4, we discussed some various bits and pieces of the puzzle of safety reasoning independently from each other. In this section, we put these bits and pieces together in an attempt to present a coherent picture.

We frame our discussion in the context of our running example: a train departing from a station. Our goal is to establish that this is done safely, for which we would like to build a safety case, i.e., a structured argument. This structured argument is our claim of safety. The structured argument itself corresponds to a s-derivation π in the $\mathcal{SK}$. As a first step in the construction of π, we need to determine its s-endformula. This s-endformula is the property which, if established, via a structured argument, assures safety. (It is not directly a claim of 'safety', for such a claim of 'safety' is outside the structured argument and the logical calculus in which we state it, and it is associated with our conception of what it means for the train to depart safely.) Let us suppose that this property is 'the train $\underline{d}$eparts iff it is $\underline{r}$eady'. To this property, there corresponds, in the language of s-formulæ, the s-formula $(r \supset \Diamond d) \wedge (\neg r \supset \neg \Diamond d)$. Given the structure of the s-endformula, we can think of proceeding with the construction of the s-derivation which establishes separately in two s-derivations, π_1 and π_2; π_1 would have $r \supset \Diamond d$ as its s-endformula; π_2 would have $\neg r \supset \neg \Diamond d$ as its s-endformula; π would obtain by combining π_1 and π_2. π_1 would correspond to a structured argument establishing that 'if the train is $\underline{r}$eady then it eventually $\underline{d}$eparts'. π_2 would correspond to a structured argument establishing that 'if the train is not $\underline{r}$eady then it does not eventually $\underline{d}$epart'. We have shown what π_1 may look like in Fig. 7. We have also

shown that there is some uncertainty associated with π_1, namely, that indicated by $\neg\upsilon$. The latter s-formula corresponds to the property 'the go signal is not $\underline{v}$isible'. As we have said, in the absence of a s-derivation which has $\neg\upsilon$ as its s-enformula, π_1 is internally coherent. Since finding that there are no such derivations, what we can do instead, is to construct a s-derivation π_3, which has υ as its s-endformula. This s-derivation corresponds to a structured argument making a case for 'the go signal is $\underline{v}$isible'. In this way, π_1 would be accompanied by a π_3, with π_3 being there to reduce the uncertainty associated with π_1. This is an application of eliminative induction. The case with π_2 would be similar. We are now in a situation in which we have two internally coherent s-derivations, π_1 and π_2, which we want to combine in a single internally coherent s-derivation, π. A priori, we could glue π_1 and π_2 together to form π by introducing the missing logical connective, $\wedge$. However, to guarantee that π is internally coherent, we would have to inspect π as a whole. This is a necessary step to eliminate the possibility of one of the rebuttals of π_1 being established from the combination of the s-premisses of π_1 in combination with those of π_2, and similarly with the rebuttals of π_2.

Thus far, nothing has been said about π being grounded on evidence, i.e., the s-premisses of π could be arbitrary s-formulæ. This is a situation that we would wish to remedy. For this, we would have to show how the s-premisses of π that are not ground atomic evidence s-formulæ can be obtained from ground atomic evidence s-formulæ. This involves an extension of π. This extension is also a s-derivation, let us call it π'. In contrast to π, the s-premisses in π' are ground atomic evidence s-formulæ. It would seem that π' contains a distinguished part that is 'purely evidential', i.e., obtained through evidential reasoning (this distinction is, although with a different flavour, also noted in [32]). It is open to debate where this 'purely evidential' part ends. Perhaps Carnap's distinction between the observable and the theoretical in the language of science (see [5, ch. 23]) provides a foundation on which to settle this debate. But this thesis needs further investigation. It should be noted that 'purely evidential' reasoning needs not appear solely when there is a need to make a s-derivation grounded on evidence. It may also appear while attempting to remove the uncertainty associated with a s-derivation. For example, in the example above, it is possible, perhaps even natural, for π_3, i.e., the s-derivation corresponding to a structured argument making a case for 'the go signal is $\underline{v}$isible', to be 'purely evidential'.

Emerging from our discussion in Sect. 4.2.3, we would associate with each s-derivation a confidence value. In order to assign a confidence value (or a tuple of values) to each evidence term, we shall start by reviewing the sources of uncertainty associated with it, including any acceptance criteria that have been specified. If the acceptance criteria have been properly defined, they would provide the scale of measurement and the measurement procedure to be used. However, for the sources of uncertainty that have not been explicitly considered we would need to add two steps. Firstly, based on the property being modelled (availability, objectivity, independence, etc.), we would select a scale for its measurement such that we can formulate useful and truth-preserving measurement statements. Then, we would select and describe a measurement procedure, which is practical and reliable (it should return the same

result under the same conditions). After defining the confidence measurement scales and procedures, and obtaining precise confidence values for the evidence terms, we would introduce utility functions for combining them (these might vary across products and companies), and a logic for propagating them. Lastly, the framework shall be empirically validated and adjusted so as to make sure that the representation condition still holds after the use of our chosen utility functions and logic. Resuming the example above, we would associate with π, i.e., the s-derivation corresponding to a structured argument making a case for 'the train *d*eparts iff it is *r*eady', some confidence value. This value will, in turn, be the value associated with our claim of safety.

6 Conclusions

The present practice of safety cases, recorded in some notation, is the result of over 25 years of work. However, to date, notations for safety cases have no semantics. This makes their understanding and assessment difficult, if not well nigh impossible, and prone to error, with the apparent negative consequences. In this work, we have started to travel the long road to providing a semantics for safety cases. Our work builds on the idea that the semantics for what is a structured argument should be based on a logical calculus. We have discussed the main ingredients of such a logical calculus, as well as the challenges that its development represents.

The situation with notations for safety cases is not new. Immediately coming to mind is the Unified Modelling Language (UML) (see [4]). This language underwent a similar historical development over a similar period of time. In both cases it has become clear that simply providing a loose syntax is not enough. Engineering disciplines rely on scientific theories and mathematics to enable precision in design and analyses to support sound engineering decisions. This was acknowledged by the OO community, who started to incorporate mathematical precision into its notations some years ago, not without its hurdles and sometimes against the protests of the notation's inventors! The safety case community is slowly awakening to this. The increasing complexity of safety-critical systems, and the recognition that relying on the informal understanding and intuition of individuals, regardless of their experience, is not only unscientific, but a historic invitation to disaster, have been the major forces pushing the need for proper engineering guarantees about safety; notations for safety cases are no exception.

We propose to develop a proper scientific and engineering basis for safety case understanding and construction on logical grounds. To this end, we have introduced a working definition of a safety case via its incorporation in a precisely defined calculus. In line with other researchers in the area (see [25]), we observe that assurance case reasoning is more akin to the argument based reasoning ideas of Toulmin than to the conventional deductive logic reasoning well known to mathematicians and software engineers (or computer scientists). This form of reasoning is already known in domains such as legal reasoning and scientific reasoning/explanation (from which

we have taken some of our ideas). The logical roots of our proposal are based on Gentzen's program for formalizing mathematical reasoning in terms of a logical language, inference rules to support reasoning steps, and proofs to capture the 'informal' notion of argument used by mathematicians. One can debate about the adequacy of Gentzen's formalization, but if one accepts it, and most mathematicians have, then one can make remarkable progress in analysing mathematical reasoning, including developing automated tools such as theorem provers and model checkers. Though safety reasoning is very different in character from mathematical reasoning, we can use an analogous approach to that of Gentzen. In particular, we can focus on the same ingredients, i.e., a formalized logical language for expressing safety claims, a well-defined notion of inference step (enlarged by incorporating some of the ideas of Toulmin's definition of an argument pattern), a well-defined notion of derivation (capturing what is a safety argument), and a new ingredient, grounded proofs, i.e., the idea that all initial formulæ in a derivation cannot be taken for granted but that they need to be justified by evidence. The latter enables a proper understanding of the notion of evidence and the role it plays in safety arguments. We hope to have taken some steps in the right direction.

References

1. Adelard, Claim, Argument, Evidence Notation. Adelard (2015), http://www.adelard.com/asce/choosing-asce/cae.html
2. R. Bloomfield, B. Littlewood, Multi-legged arguments: the impact of diversity upon confidence in dependability arguments, in *International Conference on Dependable Systems and Networks (DSN'03)* (IEEE, 2003), pp. 25–34
3. A. Bochman, Non-monotonic reasoning, in *Handbook of the History of Logic: The Many Valued and Nonmonotonic Turn in Logic*, vol. 8, ed. by D. Gabbay, J. Woods (North-Holland, Amsterdam, 2007), pp. 555–632
4. G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide*, 2nd edn. (Addison-Wesley Professional, Boston, 2005)
5. R. Carnap, *An Introduction to the Philosophy of Science*, 5th edn. (Dover, Mineola, 1966)
6. V. Cassano, T. Maibaum, S. Grigorova, A (proto) logical basis for the notion of a structured argument in a safety case. In: *18th International Conference on Formal Engineering Methods (ICFEM'16)*. LNCS, vol. 10009 (2016), pp. 1–17
7. W. Cooper, The propositional logic of ordinary discourse. Inquiry **11**(1–4), 295–320 (1968)
8. D. van Dalen, *Logic and Structure*, 5th edn. (Springer, Berlin, 2013)
9. R. Darimont, A. van Lamsweerde, Formal refinement patterns for goal-driven requirements elaboration, in *4th ACM SIGSOFT Symposium on Foundations of Software Engineering (SIGSOFT'96)* (ACM, 1996), pp. 179–190
10. J. van Eijck, M. Stokhof, The gamut of dynamic logics (2011), in [13], pp. 499–600
11. N.E. Fenton, Software measurement: a necessary scientific basis, in *Predictably Dependable Computing Systems* (Springer, Berlin, 1995), pp. 67–78
12. D. Gabbay, J. Woods (eds.), *Handbook of the History of Logic: Inductive Logic*, vol. 10 (North-Holland, Amsterdam, 2011)
13. D. Gabbay, J. Woods (eds.), *Handbook of the History of Logic: Logic and the Modalities in the Twentieth Century*, vol. 7 (North-Holland, Amsterdam, 2011)
14. G. Gentzen, Investigations into logical deduction. Am. Philos. Q. **1**(4), 288–306 (1964)

15. J. Goodenough, C. Weinstock, A. Klein, Eliminative induction: a basis for arguing system confidence, in *35th International Conference on Software Engineering (ICSE'13)* (2013), pp. 1161–1164
16. P. Graydon, M. Holloway, An investigation of proposed techniques for quantifying confidence in assurance arguments. Saf. Sci. **92**, 53–65 (2017)
17. S. Grigorova, The elusive quest: software product quality evaluation. Master's thesis, McMaster University, Canada, 2009
18. C. Hempel, *Aspects of Scientific Explanation: And Other Essays in the Philosophy of Science* (Free Press, New York, 1965)
19. D. Hitchcock, Toulmin's warrants, in *Anyone Who Has a View: Theoretical Contributions to the Study of Argumentation*, ed. by F. van Eemeren et al. (Springer, Berlin, 2003), pp. 69–82
20. C. Hoare, An axiomatic basis for computer programming. Commun. ACM **12**(10), 576–580 (1969)
21. International Organization for Standardization, ISO 2626: Road Vehicles – Functional Safety. Version 1 (2011)
22. A. Jøsang, *Subjective Logic: A Formalism for Reasoning Under Uncertainty* (Springer International Publishing, Berlin, 2016)
23. A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications* (Wiley, Hoboken, 2009)
24. Z. Manna, A. Pnueli, *The Temporal Logic of Reactive and Concurrent Systems: Specification* (Springer Science+Business Media, Berlin, 1991)
25. J. McDermid, Safety arguments, software and system reliability, in *2nd International Symposium on Software Reliability Engineering (ISSRE'91)* (IEEE, 1991), pp. 43–50
26. P. McNamara, Deontic logic (2011), in [13], pp. 197–288
27. P. Øhrstrøm, P. Hasle, Modern temporal logic: the philosophical background (2011), in [13], pp. 447–498
28. G. Pólya, *How to Solve It*, 2nd edn. (Princeton University Press, Princeton, 2004)
29. K. Popper, *An Introduction to the Philosophy of Science* (Routledge, Abingdon, 2002)
30. G. Restall, *Proof Theory and Philosophy*. Draft Book (2006), http://consequently.org/writing/ptp
31. J. Rushby, Logic and epistemology in safety cases, in *32nd International Conference on Computer Safety, Reliability, and Security (SAFECOMP'13)*. LNCS, vol. 8153 (2013), pp. 1–7
32. J. 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)*. LNCS, vol. 10091 (2015), pp. 331–347
33. The GSN Working Group, Goal Structuring Notation. The GSN Working Group (2011), http://www.goalstructuringnotation.info/
34. S. Toulmin, *The Uses of Argument* (Cambridge University Press, Cambridge, 2003)
35. UK Ministry of Defense, Defence standard 00-56 issue 4: safety management requirements for defence systems (2007)
36. S. Veloso, P. Veloso, R. de Freitas, An application of logic engineering. Log. J. IGPL **13**(1), 29–46 (2005)
37. W. Vincenti, *What Engineers Know and How They Know It: Analytical Studies from Aeronautical History* (Johns Hopkins University Press, Baltimore, 1993)
38. T. Williamson, *Knowledge and Its Limits* (Oxford University Press, Oxford, 2000)