Must Assurance be Indefeasible?

John Rushby

Computer Science Laboratory
SRI International
Menlo Park, CA

Overview

- Probabilistic justification for assurance of conventional systems
- Justified belief and indefeasibility
- Assurance cases and their interpretation and evaluation
- New challenges: can we/should we retain indefeasibility?

Introduction

- Assurance provides confidence that our (software) system will
  1. Work OK
  2. Not do serious harm
- Hard part is to obtain confidence in ultra-low probability of serious failure
- The numbers are daunting, e.g., catastrophic failures in aircraft are "not anticipated to occur during the entire operational life of all airplanes of one type"
- Airbus A320 family (type) already has 62 million flight hours, so operational life will be some multiple of 10^8 hours
- "when using quantitative analyses. . . numerical probabilities. . . on the order of 10^{-9} per flight-hour may be used. . . as aids to engineering judgment. . . "

Assurance Works

- Current methods seem to work for traditional systems
- No plane crashes due to software: DO-178C, ARP 4754A,. . .
- But how does it work?
- Here's how
- Extreme scrutiny of development, artifacts, code provides confidence software is fault-free
  - Or quasi fault-free (remaining faults have minuscule pfd)
- Can express this confidence as a subjective probability that the software is fault-free or nonfaulty: p_{nf}
- For a frequentist interpretation: think of all the software that might have been developed by comparable engineering processes to solve the same design problem
  - And that has had the same degree of assurance
  - Then p_{nf} is the probability that any software randomly selected from this class is nonfaulty

This is How it Works: Step 1

- Define $p_{F|f}$ as the probability that it Fails, if faulty
- Then probability $p_{srv}(n)$ of surviving n independent demands (e.g., flight hours) without failure is given by

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

- A suitably large n can represent "entire operational life of all airplanes of one type"
- First term in (1) establishes a lower bound for $p_{srv}(n)$ that is independent of n
- If assurance gives us the confidence to assess, say, $p_{nf} > 0.9$
- Then it looks like we are there
- But suppose we do this for 10 airplane types
  - Can expect 1 of them to have faults
  - So the second term needs to be well above zero
  - But it decays exponentially

This is How it Works: Step 2

- We need confidence that the second term in (1) will be nonzero, despite exponential decay
- Confidence could come from prior failure-free operation
- Calculating overall $p_{srv}(n)$ is a problem in Bayesian inference
  - We have assessed a value for p_{nf}
  - Have observed some number r of failure-free demands
  - Want to predict prob. of $n - r$ future failure-free demands
- Need a prior distribution for $p_{F|f}$
  - Difficult to obtain, and difficult to justify for certification
  - However, there is a distribution that delivers provably worst-case predictions
    - ⋆ One where $p_{F|f}$ is a probability mass at some $q_n \in (0, 1]$
  - So can make predictions that are guaranteed conservative, given only p_{nf}, r, and n

This is How it Works: Step 3

- For values of p_{nf} above 0.9
- The second term in (1) is well above zero
- Provided $r > \frac{n}{10}$
- So it looks like we need to fly 10^7 hours to certify 10^8
- Maybe not!
- Entering service, we have only a few planes, need confidence for only, say, first six months of operation, so a small n
- Flight tests are enough for this
- Next six months, have more planes, but can base prediction on first six months (or ground the fleet, fix things, like 787)
- And bootstrap our way forward
- This is a rational reconstruction of how aircraft software certification works (due to Strigini and Povyakalo)
- It provides a model that is consistent with practice

Confidence in Absence of Faults

- We have a probabilistic model that works
- Foundation is strong confidence in absence of faults: $p_{nf} > 0.9$
- How do we achieve that?
- Assurance cases!
- But how to attach a probability to our confidence in a case?
- More fundamentally, how do we establish confidence in a case?
- Confidence is justified belief
- The limit is justified true belief
- That's knowledge! (Plato)
- We want to know there are no faults

Knowledge as Justified True Belief

- Russell, 1912:

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

- Alice has a justified belief
  - But the justification is not very good
  - It happens to be true, but by accident
- In 1963 Gettier published additional examples of poorly justified beliefs that are accidentally true
- The most widely cited modern work in epistemology
  - Over 3,000 citations, 3 pages, he wrote nothing else
- Much work in response attempts to adjust the definition of knowledge by replacing or augmenting justified true belief

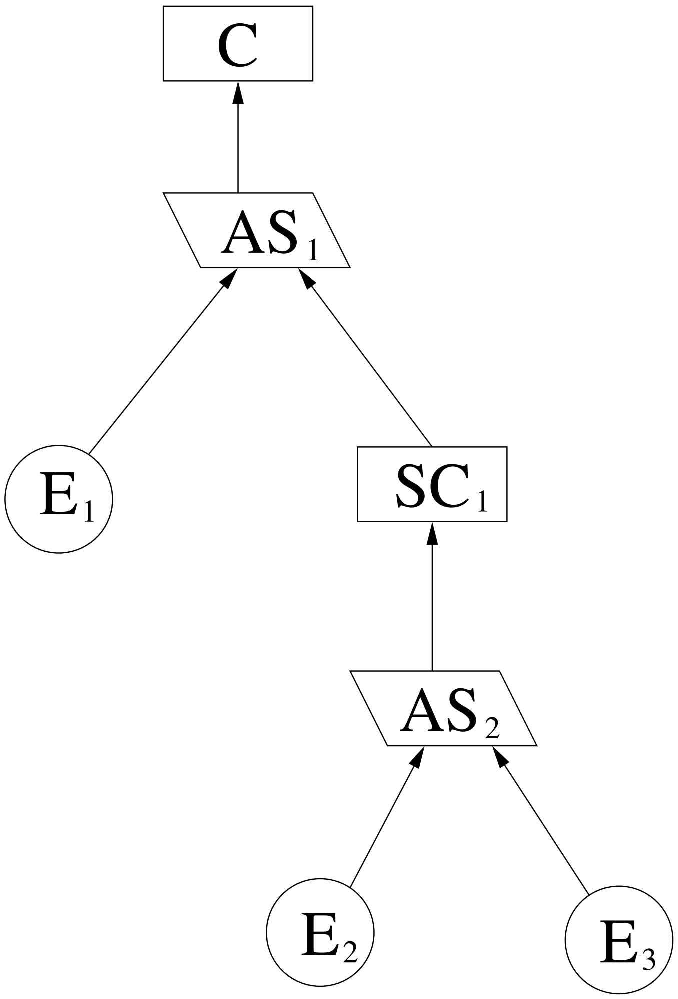
The Indefeasibility Criterion

- Want a good criterion for justified
  - One that excludes Alice's justification
  - She did not consider possibility of faulty clock
  - Should have sought evidence about this
- Recent work in epistemology proposes indefeasibility
  - For a belief to be justified indefeasibly, we must be so sure that all contingencies have been identified and considered that there is no (or, more realistically, we cannot imagine any) new evidence that would change our belief
- Truth is known only to the omniscient
- So in assurance we do not seek justified true belief
- But adequately justified belief
- Take indefeasibility as our criterion
  - If you have an indefeasibly justified belief, then

    what you don't know can't hurt you! (Barker)

Assurance Cases

We use a structured argument to justify the assurance claim



A hierarchical arrangement of argument steps, each of which justifies a claim or subclaim on the basis of further subclaims or evidence

C: Claim

AS: Argument Step

SC: Subclaim

E: Evidence

For Example

- The claim $\mathtt{C}$ could be system correctness
  - $\mathtt{E}_2$ could be test results
  - $\mathtt{E}_3$ could then be a description of how the tests were selected and the adequacy of their coverage

  So $\mathtt{SC}_1$ is a claim that the system is adequately tested

- And $\mathtt{E}_1$ might be version management data to confirm it is the deployed software that was tested

Applying the Indefeasibility Criterion

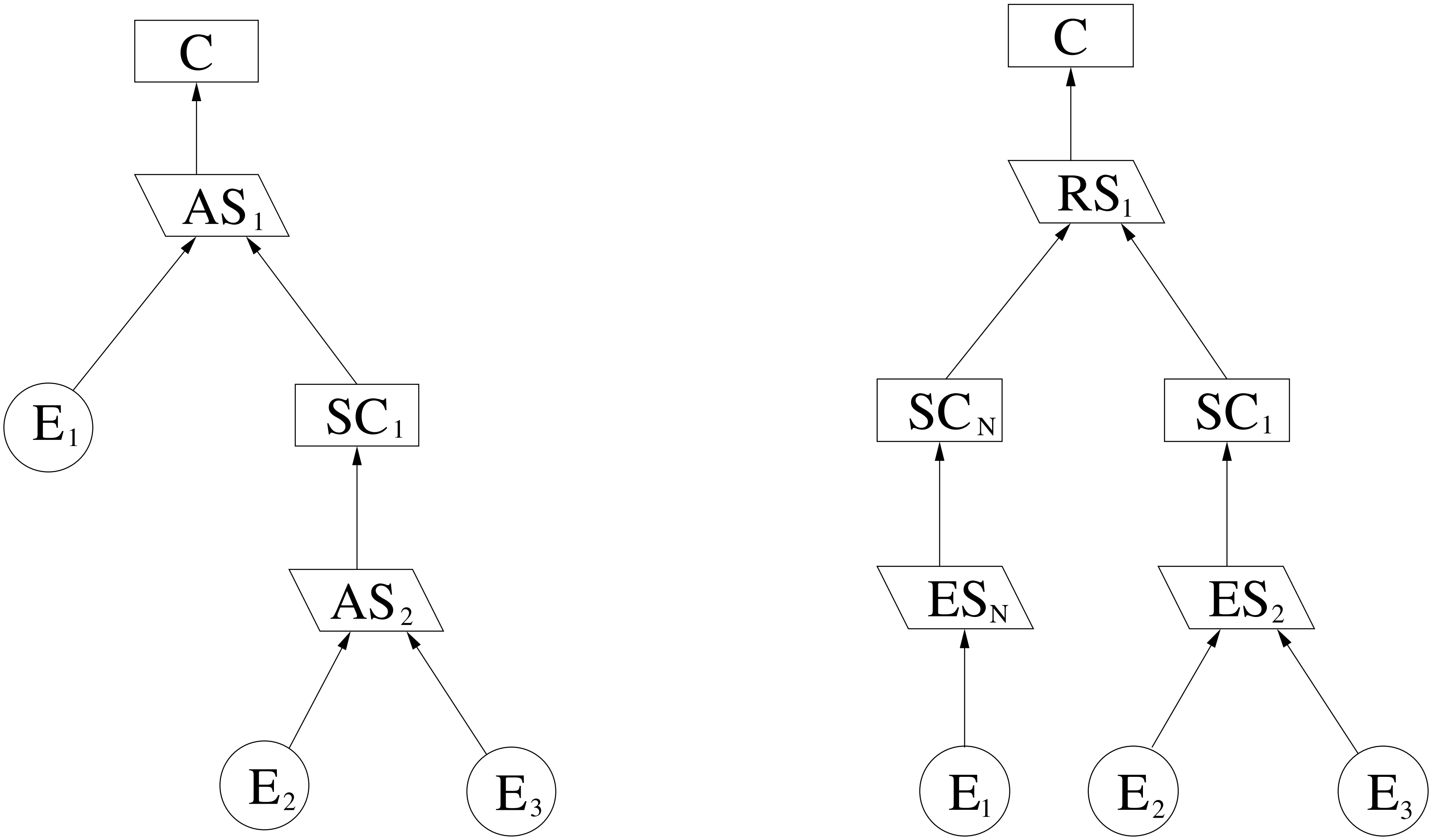
There are two ways in which the justification for an assurance case could be inadequate

1. Evidence is weak
   - e.g., not many tests, verified weak properties
   - Affects confidence, not “validity”
   - Can be measured/managed probabilistically
2. Evidence/subargument is missing
   - Failed to address some hazard or defeater
   - e.g., test oracle could be flawed, verifier unsound
   - Hazard is a reason the system could fail; defeater is a reason the argument could be “invalid”
   - Presence of either causes confidence to collapse
   - Indefeasibility requires these are excluded

Is Indefeasibility Realistic?

- Defeasible cases have gaps of unknown size
- Indefeasible cases have no gaps
- But can it be done?
- e.g., how do we know we have found all hazards?
- We do hazard analysis
  - Provides evidence we found them all
    - ⋆ Evidence describes method of hazard analysis employed, diligence of its performance, historical effectiveness, standards applied, and so on
- This transforms a gap into evidence there is no gap
  - And we can weigh that evidence
- No, it is not a trick
- Now, some details

Normalizing an Argument to Simple Form



RS: reasoning step; **ES**: evidential step

Why Focus on Simple Form?

- The two kinds of argument step are interpreted differently
- Evidential steps
  - These are about epistemology: knowledge of the world
  - Bridge from the real world to the world of our concepts
  - Multiple items of evidence are “weighed” not conjoined
- Reasoning Steps
  - These are about logic/reasoning
  - Conjunction of subclaims leads us to conclude the claim
- Combine these to yield complete arguments
  - Those evidential steps whose weight crosses some threshold of confidence are treated as premises in a classical deductive interpretation of the reasoning steps
- Can be seen as systematic treatment of the style of informal argumentation known as “natural language deductivism”
  - I feel like Molière’s character: speaking prose all his life

Weighing Evidential Steps

- We measure and observe what we can
  - e.g., test results
- To infer a subclaim that is not directly observable
  - e.g., correctness
- Different observations provide different views
  - Some more significant than others
  - And not all independent, so cannot just conjoin them
- Need to "weigh" all these in some way
- Probabilities provide a convenient metric
- And Bayesian methods and BBNs provide tools
- "Confidence" items can be observations that vouch for others
  - Or provide independent backup
  - Example in a few slides time

The Weight of Evidence

- What measure should we use for the weight of evidence?
- Plausible to suppose that we should accept claim C given collection of evidence E when $P(C \mid E)$ exceeds some threshold
- These are subjective probabilities expressing human judgement
- Experts find $P(C \mid E)$ hard to assess
- And it is influenced by prior $P(C)$, which may reflect ignorance. . . or prejudice
- Instead, factor problem into alternative quantities that are easier to assess and of separate significance
- So look instead at $P(E \mid C)$
  - ○ Related to $P(C \mid E)$ by Bayes' Rule
  - ○ But easier to assess likelihood of observations given a claim about the world than vice versa

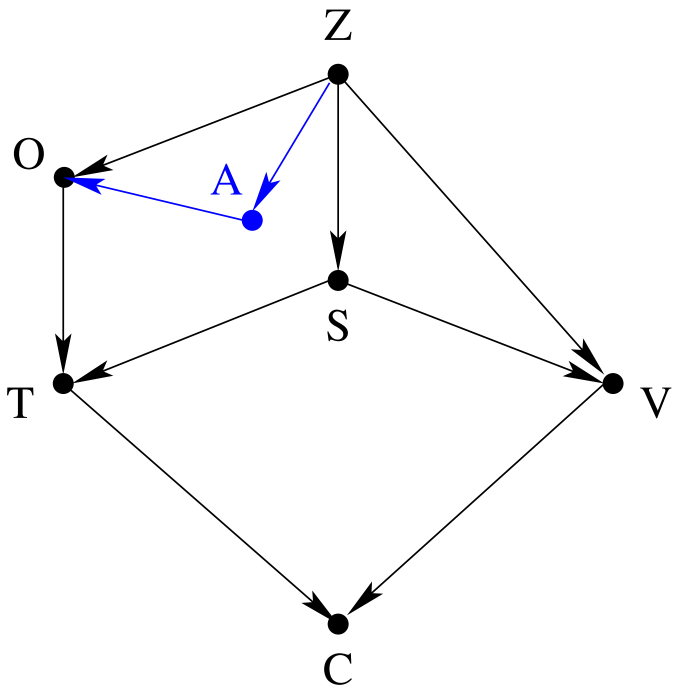
Confirmation Measures

- We really are interested in the extent to which E supports C rather than its negation $\neg C$
  - ○ Also want $P(E\,|\,C)$ is not vacuous (e.g., E is a tautology)
- So focus on the ratio or difference of $P(E\,|\,C)$ and $P(E\,|\,\neg C)$, . . . or logarithms of these
- These are called confirmation measures
- They weigh C and $\neg\,C$ "in the balance" provided by E
- Good's measure: $\log \dfrac{P(E\,|\,C)}{P(E\,|\,\neg\,C)}$
- Kemeny and Oppenheim's measure: $\dfrac{P(E\,|\,C) - P(E\,|\,\neg\,C)}{P(E\,|\,C) + P(E\,|\,\neg\,C)}$
- Much discussion on merits of these and other measures
- Suggested that these are what criminal juries should be instructed to assess (Gardner-Medwin)

Application of Confirmation Measures

- I do not think the specific measures are important
- Nor is quantification necessary for individual arguments
  - Informal evaluation and narrative description can be OK
- Rather, use BBNs and confirmation measures for what-if investigations to develop insight and sharpen judgement
  - Can help guide selection of evidence for evidential steps
  - e.g., refine what objectives DO-178C should require
  - Example (next slides) explores use of "artifact quality" objectives as confidence items in DO-178C
    - ⋆ e.g., "Ensure that each High Level Requirement (HLR) is accurate, unambiguous, and sufficiently detailed, and the requirements do not conflict with each other" [§ 6.3.1.b]

Weighing Evidential Steps With BBNs



Z: System Specification

O: Test Oracle

S: System's true quality

T: Test results

V: Verification outcome

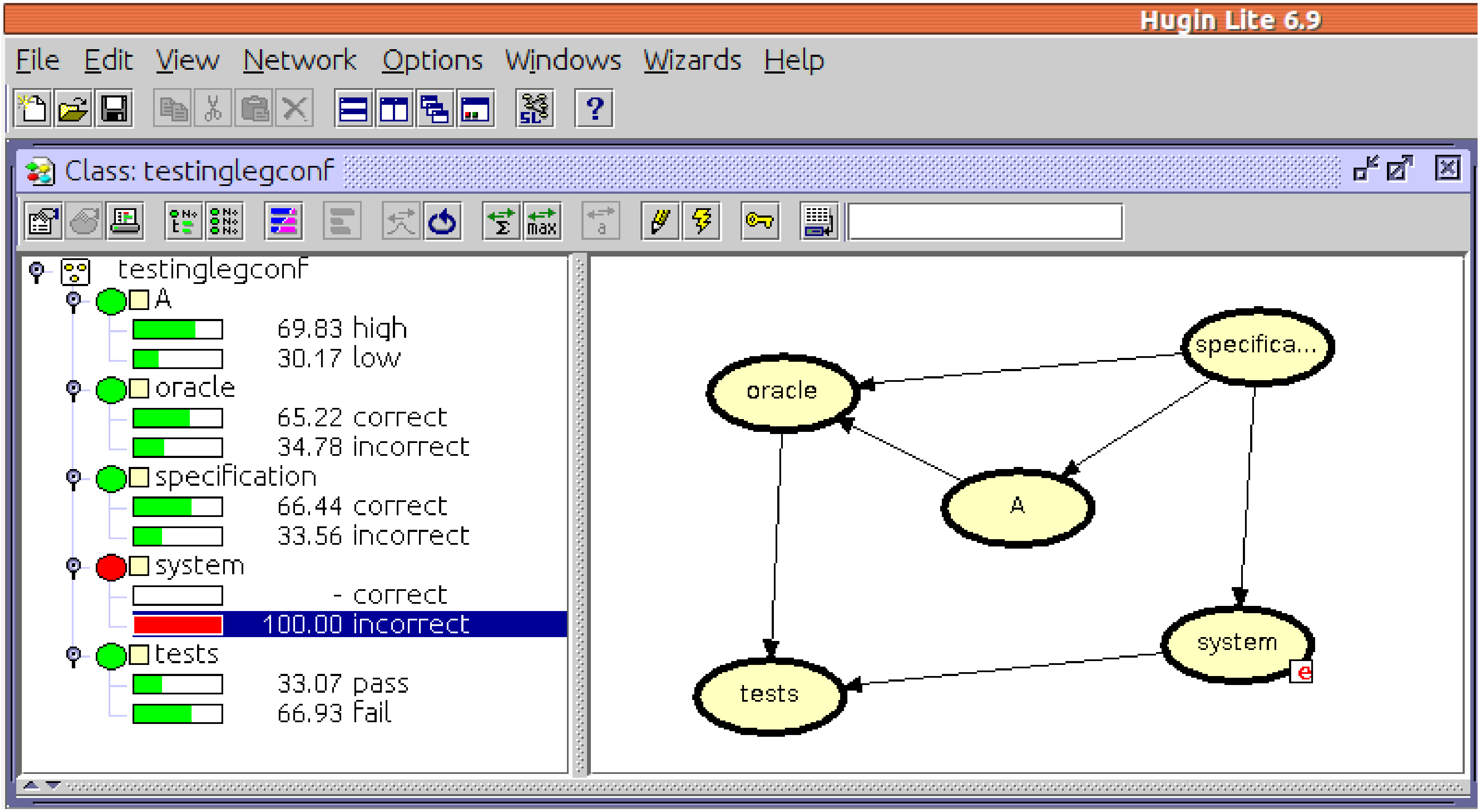
A: Specification "quality"

C: Conclusion

Example joint probability table: successful test outcome

Correct System		Incorrect System	
Correct Oracle	Bad Oracle	Correct Oracle	Bad Oracle
100%	50%	5%	30%

Example Represented in Hugin BBN Tool



```
testinglegconf
  A
      69.83 high
      30.17 low
  oracle
      65.22 correct
      34.78 incorrect
  specification
      66.44 correct
      33.56 incorrect
  system
          - correct
     100.00 incorrect
  tests
      33.07 pass
      66.93 fail
```

www.hugin.com

Interpretation of Reasoning Steps

- Evidential steps are weighed probabilistically

- When all evidential steps cross confidence threshold, use them as premises in a logical interpretation of reasoning steps

- Traditionally, two such interpretations
  - Deductive: p_1 `AND` p_2 `AND` $\cdots$ `AND` p_n `IMPLIES` c
  - Inductive: p_1 `AND` p_2 `AND` $\cdots$ `AND` p_n `SUGGESTS` c

- Note that inductive reasoning is not modular: must believe either the gap is insignificant (so deductive), or taken care of elsewhere (so not modular)

- Indefeasibility requires the deductive interpretation

Overall Confidence In A Case

- We could try to attach a probabilistic confidence measure to each evidential step
- Then take their product (recall, subclaims are independent)
- To get probabilistic confidence in top claim
- But difficult to assess and justify
- Remember, when we use confirmation measures to “weigh” evidential steps, the numbers are components of a model used to guide judgement, not solid estimates
- So I suggest we accept adequate confidence in top claim (i.e.,absence of faults) when all evidential steps cross their thresholds
- And we are confident of indefeasibility (coming up)
- But what about graduated assurance?

Graduated Assurance

- Not all (sub)systems need the same level of assurance

- What dials can we turn to adjust assurance (and costs) for different circumstances?

- Eliminate some subclaims?
  - No!
  - Would surely make the case defeasible (unless redundant)

- Reduce evidential thresholds?
  - OK
  - And that could allow elimination/substitution of evidence

    e.g., eliminate static analysis, or replace by more testing
  - And that in turn could allow elimination of subclaims

    e.g., soundness of static analyzer

Challenges and Indefeasibility

- Main concern with assurance cases is confirmation bias
- Cases must be subjected to serious dialectical challenge
- Can be organized as a search for defeaters
  - Reasons the argument might be defeasible/wrong
  - Cf. hazards to a system

  And construction of a rebuttal for each
- Defeaters and rebuttals should be recorded as part of the case
  - And likely organized as subarguments
- Although final case should be indefeasible/deductive
- Preliminary and intermediate stages could be inductive
- So could be value in tools that can support this
- Can maybe learn from field of Argumentation and its tools

  e.g., Astah GSN has Carneades-like capabilities

Present and Near Future

- I think this analysis explains the success of present methods of assurance and suggests modest improvements
- Treatment of assurance cases is both simple and strict
- My personal opinion is that bespoke assurance cases are likely to be unreliable
    - Insufficient dialectical challenge
- So best approach may be to reformulate standards and guidelines as assurance cases
    - I think that will make them better
    - And provide a basis for customization
- Alternative: build assurance cases from accepted patterns (GSN) or blocks (CAE)

Imminent
But What Of The ~~More Distant~~ Future?

- E.g., self-driving cars
- Existing model of assurance and certification depends on both the system and the environment being predictable, so that with enough work we gain near-omniscient (i.e., indefeasible) knowledge of all possible behaviors
- Not so here
  - ○ Internal operation of own software may be unpredictable
    - ⋆ e.g., machine learning in vision system
    - ⋆ It is opaque, too
  - ○ External environment is unpredictable
    - ⋆ e.g., behavior of other road users
    - ⋆ No good model
- On the other hand, we have lowered expectations
  - ○ No worse than human

The Imminent Future

- There seem to be two options
  1. Retain indefeasibility
     - But then how to cope with unpredictability?
     - Massively reduce thresholds on evidence?
  2. Abandon indefeasibility
     - But indefeasibility is what requires us to try to think of everything
     - Do we dare give this up?
     - And replace it by learning from experience (i.e., crashes)?
- Maybe there's a third way: monitoring and backups

Monitoring and Backups

- Weaker knowledge may suffice for weaker properties
- And monitoring may alert us to violated assumptions
  - There are imaginative ways of monitoring
    - e.g., checking for liveness of vision system (TTTech)
- Can then build Monitored Architectures
  - Handover on detected violation of assumptions
  - Similar to present, doesn't work: e.g., AF447
- And Simplex Architectures
  - Revert to weaker behavior on detected violation of assumptions
    - Last-ditch behavior may be unassurable
    - e.g., AF447, no air data, no safe option
    - But no worse than human

Conclusion

- We have a good story for current systems
- Breaks down for imminent future systems
- Are there ways to prevent the breakdown?
  - Monitoring?
  - More advanced engineering and verification for learning systems?
    - Promising work at Stanford, Oxford, Fortiss
  - A new approach?
- And can we/should we retain indefeasibility?
  - I think we should keep it: it is what creates the obligation to try to think of everything
- What do you think?