Confidence About Evidence and Its Role in an Argument

John Goodenough
May 3, 2015

Eliminative Arguments

An eliminative argument establishes the basis for confidence in a claim

- Shows the role of evidence in particular arguments
- Confidence: “absence of doubt”
  - Increases as doubts are eliminated
  - Claims are supported (indirectly) by eliminating doubts

An eliminative argument

- Identifies all sources of doubt for an argument
- Shows why certain doubts are eliminated

Types of Doubt (Defeaters)

Doubts about validity of a claim

- A reason the claim can't be true (rebutting defeater)
- E.g., presence of a hazard would contradict a claim of system safety

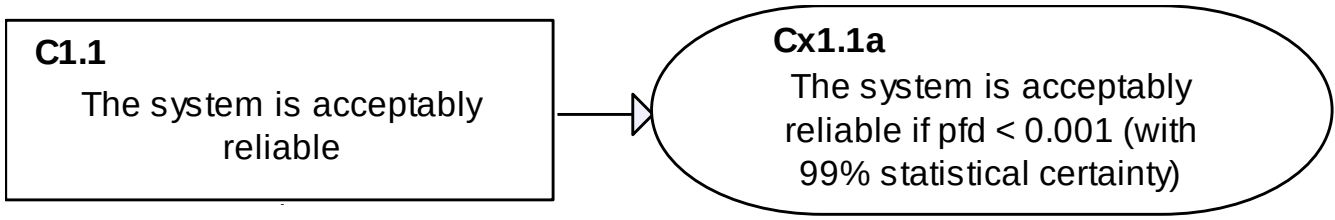
Doubts about validity of evidence

- Why an evidence assertion would be wrong (undermining defeater)
- E.g., less confidence that a set of tests is randomly selected from an operational profile if we are not satisfied with how tests were selected

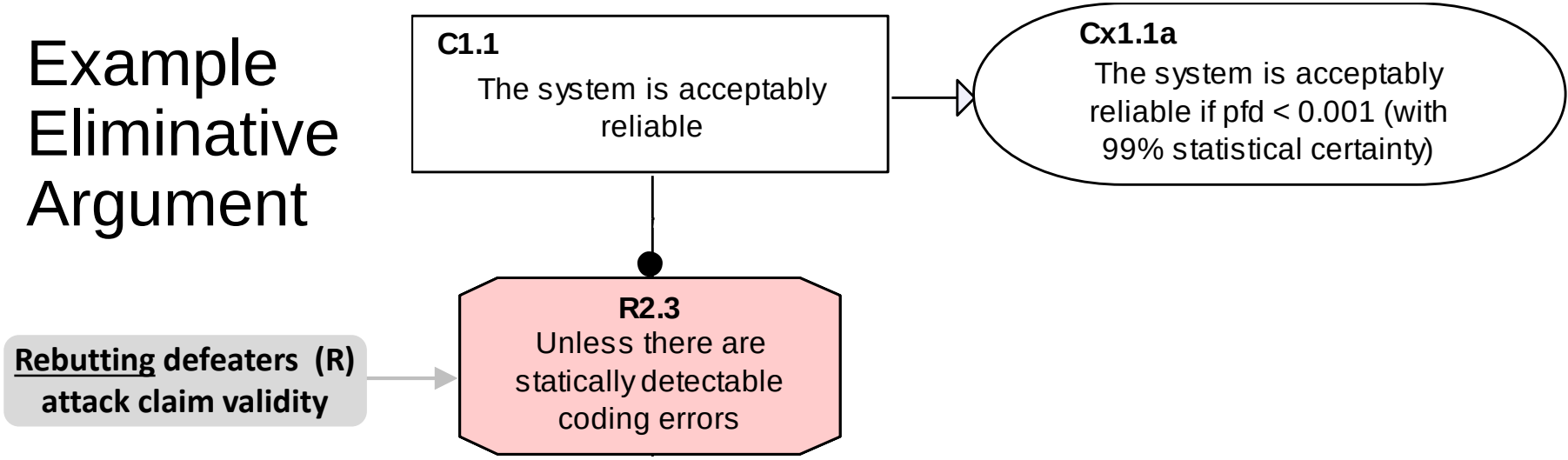
Doubts about validity of reasoning

- Conditions under which an inference does not necessarily hold (undercutting defeaters)
- E.g., **"test-success" -> "system-works"** unless tests missed important conditions

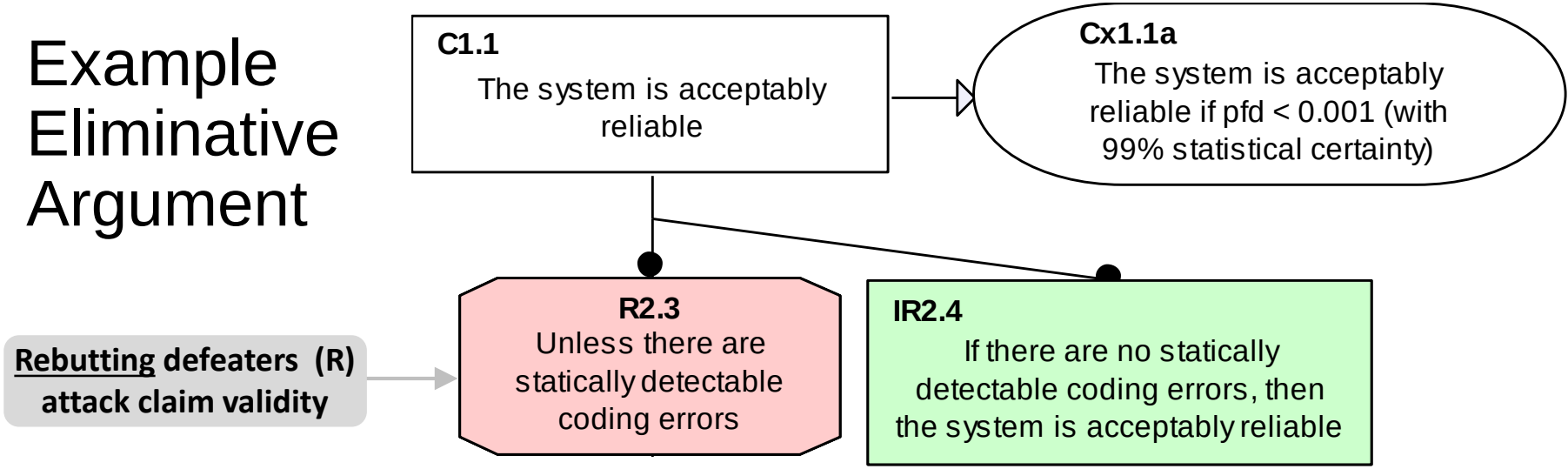
Example Eliminative Argument



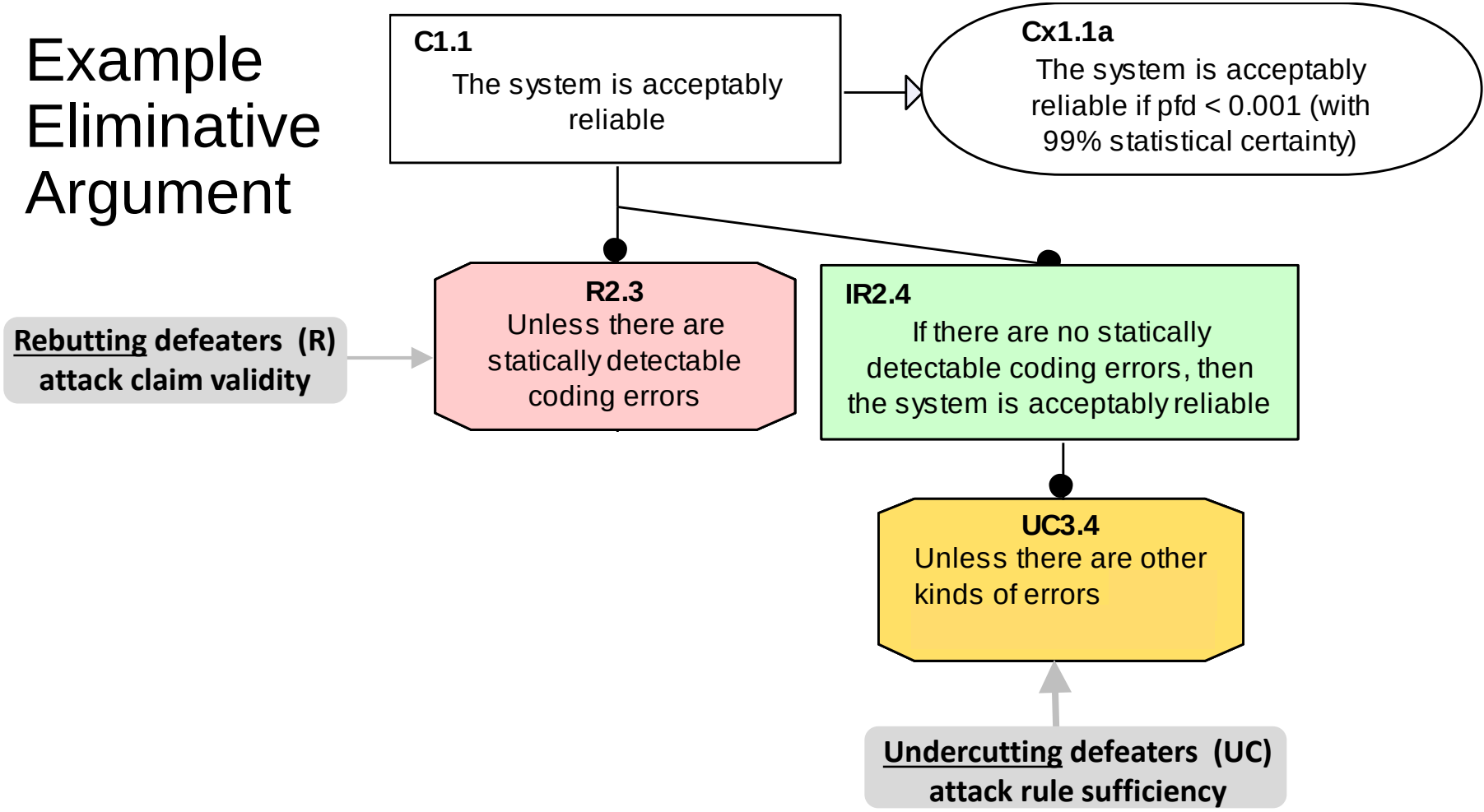
Example Eliminative Argument



Example Eliminative Argument



Example Eliminative Argument



Inference Deficiencies (Defeaters)

Inference: Absence of statically detected errors implies system contains no errors (and so is acceptably reliable)

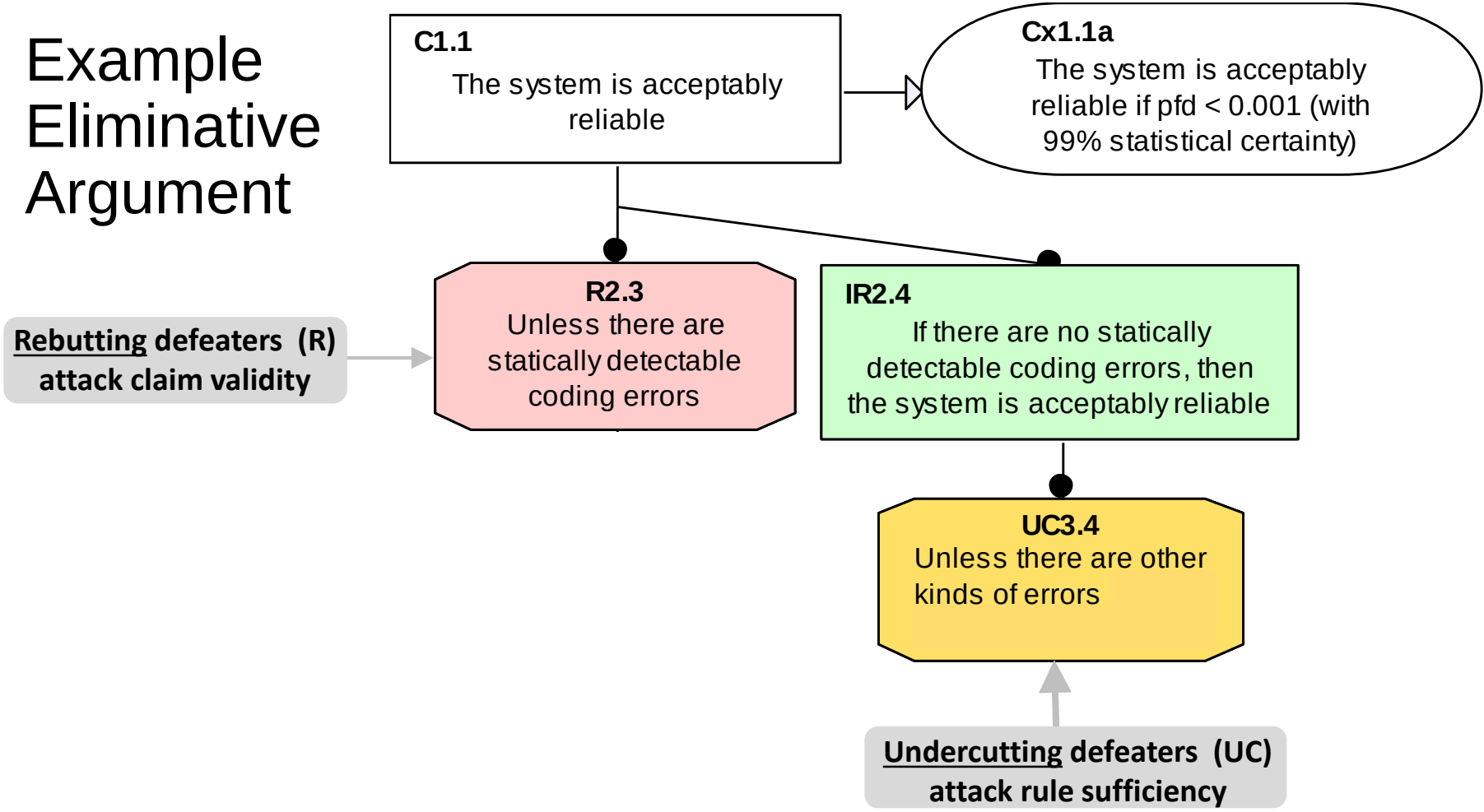
Defeaters for inference validity

- Various kinds of errors are not detectable with static code analysis, e.g.,
  - Timing errors
  - Design errors (e.g., error-prone user interface)
  - Specification errors
  - Requirements errors

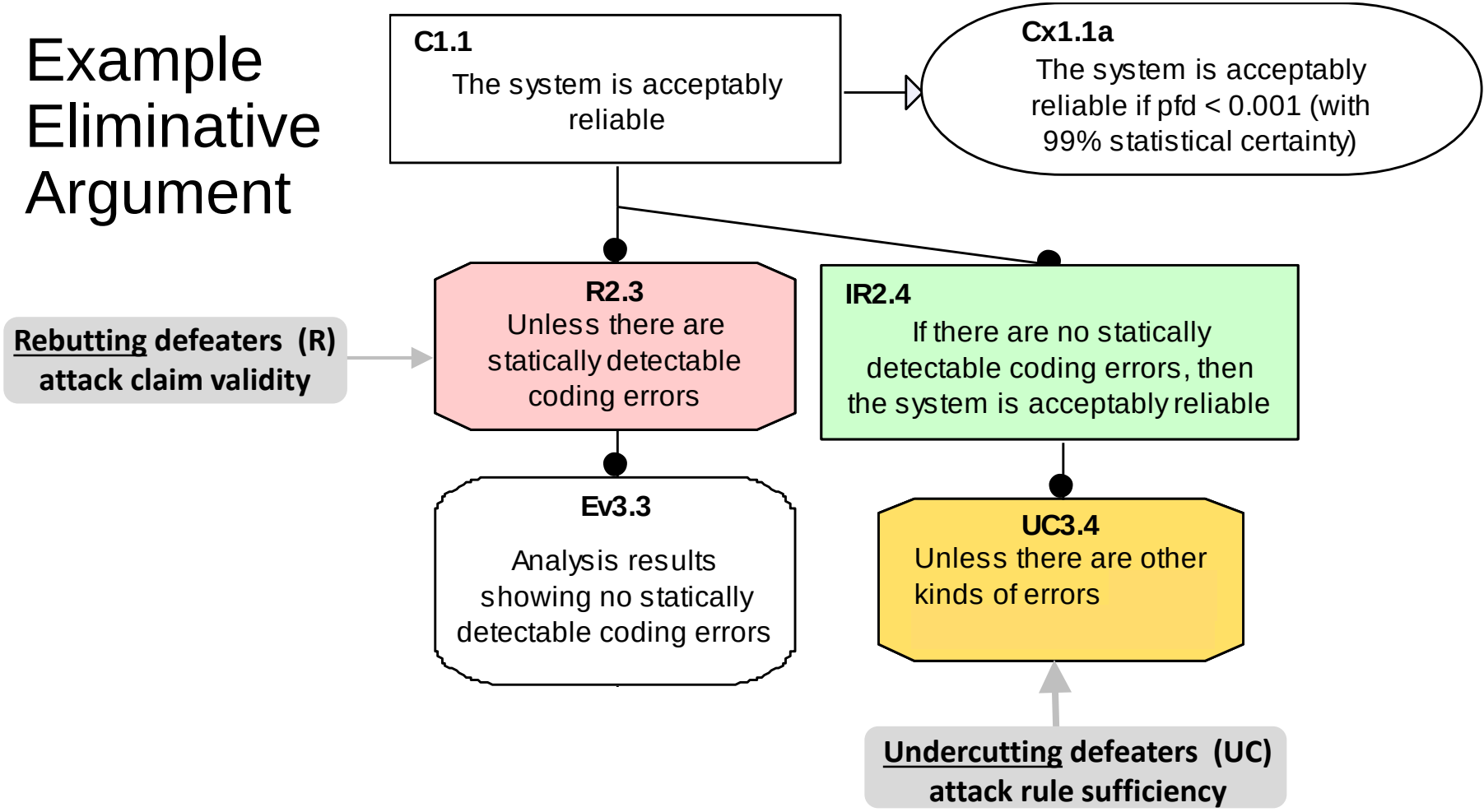
Even with valid evidence, the conclusion can be uncertain under some conditions

- Deficiencies relevant to the type of evidence and conclusion

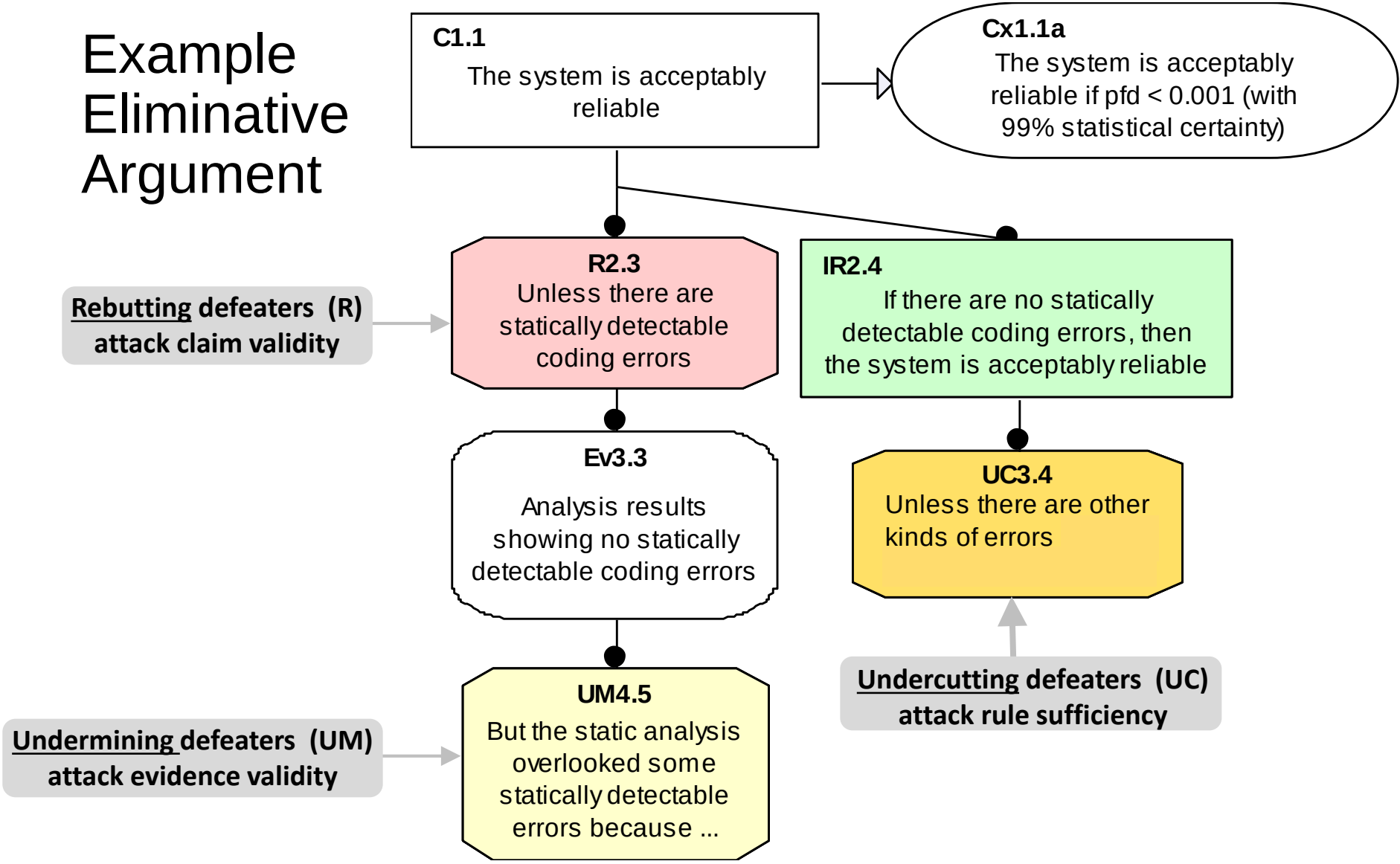
Example Eliminative Argument



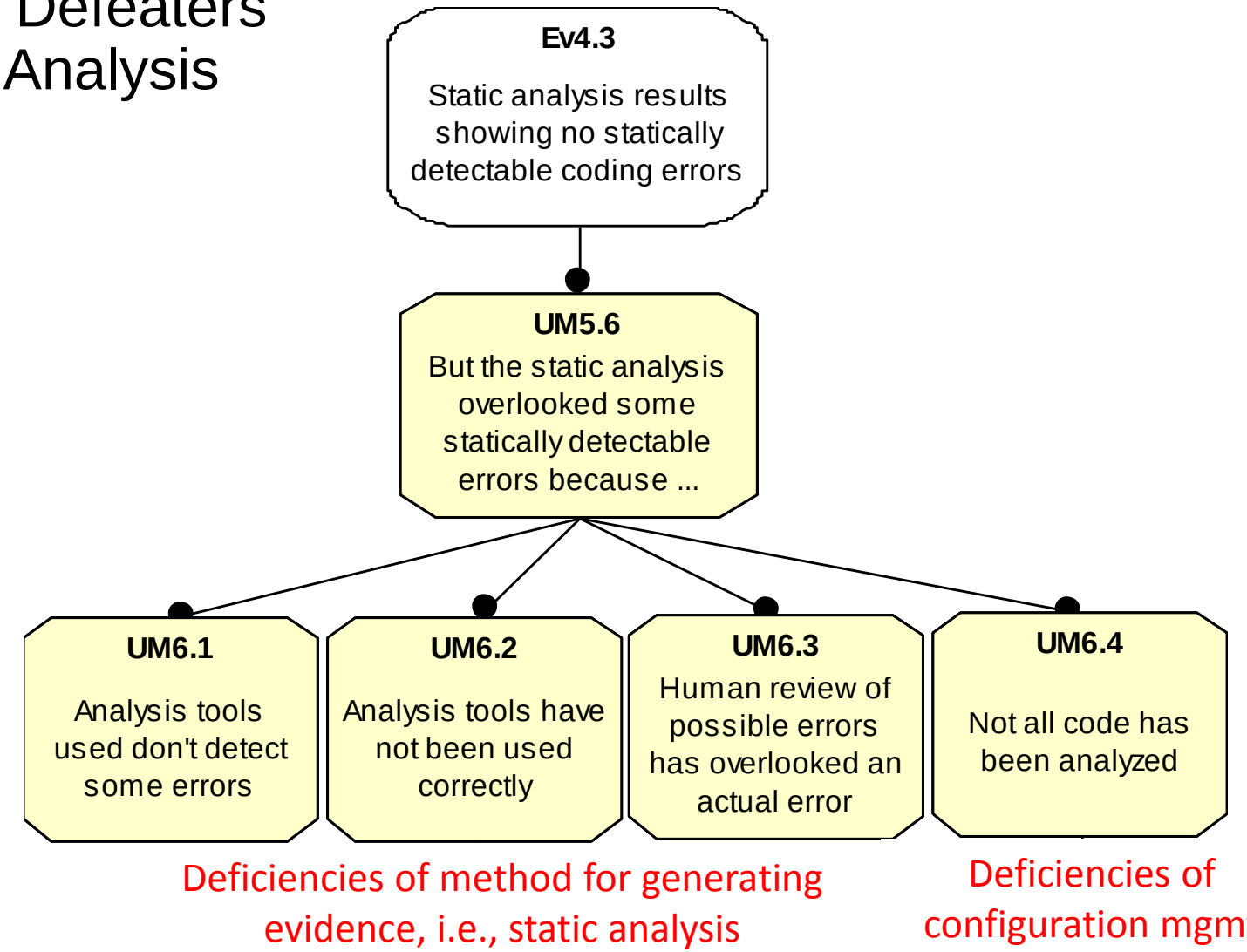
Example Eliminative Argument



Example Eliminative Argument



Evidence Defeaters for Static Analysis



Evidence Adjectives

Relevance: inferential force, i.e., extent to which inference is valid

Trustworthiness: validity

Strength

- Sometimes about confidence in the validity of the evidence, e.g.
  - Confidence that all statically detectable errors have been found
  - Eyewitness vs video testimony
  - Weak evidence has more (unresolved) reasons to doubt its validity
- Sometimes about inferential force, e.g.,
  - Lack of statically detectable errors is weak evidence of system reliability
  - DNA testing as basis for identification is strong evidence

Types, Instance Trust, Instance Capability [Hawkins, Sun]

Types of evidence: generic classes of evidence

- Examples: Results of testing, code reviews, static code analysis, proofs
- Instances of each type have validity defeaters in common

Instance trust: is the expected capability delivered?

- Extent to which the evidence is valid, e.g.,
  - Is the static analysis tool powerful enough?
  - Has all code been analyzed?

Instance capability: can the instance (of a type) support a claim

- Extent to which the inference, ***instance -> conclusion***, is valid, e.g.,
  - Given no statically detectable coding errors and conclusion: *pfd* ≤ 10^{-3} (with 99% confidence) …
    - Lack of statically detectable errors is not sufficient to support the conclusion, i.e., the instance capability is low

Hawkins and Kelly 2010; Sun 2012

Role of Counterevidence

Evidence that contradicts a claim

- (Could also be evidence that shows an inference is insufficient or that challenges the validity of other evidence)

Potential response

- Eliminate the counterevidence
  - Fix the system (e.g., prior to deployment)
  - Restrict the claim (e.g., don't use the system under certain conditions)
  - Undermine the counterevidence (the system is actually working; the reported error is not an error)
- Accept the counterevidence
  - Live with reduced confidence by modifying the argument
    - Add a new defeater and/or inference rule
    - "Uneliminate" an existing defeater (decrease its probability of elim.)
  - Don't change the argument
    - The counterevidence is consistent with an uneliminated defeater

Discussions of evidence

[Hawkins 2010]
Hawkins, R. & Kelly, T. “A Structured Approach to Selecting and Justifying Software Evidence,” 1-6. In *5th IET International Conference on System Safety System Safety 2010.* Manchester, Oct. 2010. IEEE Computer Society, 2010.

[Sun 2013]
Sun, L. & Kelly, T. “Elaborating the Concept of Evidence in Safety Cases,” 111–126. *Assuring the Safety of Systems: Proceedings of the Twenty-first Safety-Critical Systems Symposium*. Bristol, U.K., Feb. 2013. Edited by C. Dale & T. Anderson. Safety Critical Systems Club, 2013.

[OMG]
Object Management Group. *Structured Assurance Case Metamodel (SACM).* Object Management Group, 2013. http://www.omg.org/spec/SACM

[Goodenough 2015]
Goodenough, John; Weinstock, Charles; & Klein, Ari. *Eliminative Argumentation: A Basis for Arguing Confidence in System Properties* (CMU/SEI-2015-TR-005). Software Engineering Institute, Carnegie Mellon University, 2015.