Design Verification
COMS30026

(COMS30066 with coursework)

WEEKLY LIVE SESSION – W8

Kerstin Eder

Trustworthy Systems Lab

University of BRISTOL

Department of COMPUTER SCIENCE

Topics W7

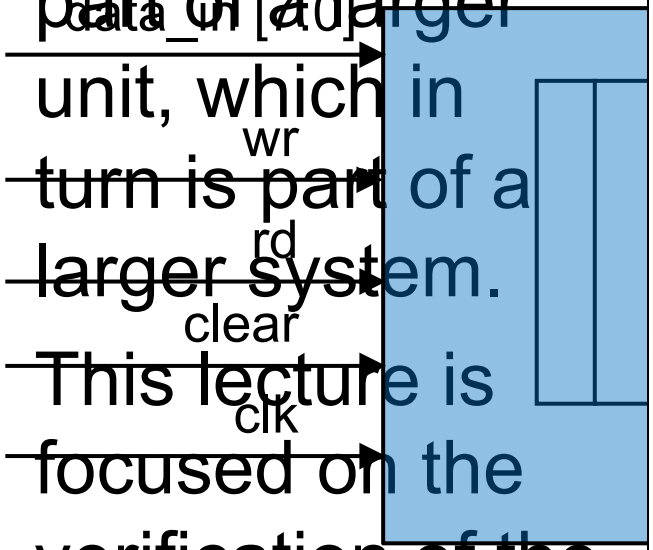
- ✓ Assertion-based Verification – Part I
  - ✓ Introduction and OVL
- ✓ Assertion-based Verification – Part II
  - ✓Property formalisation and SVA
- ✓ Functional Formal Verification
  - ✓ **Hands-on Demo**

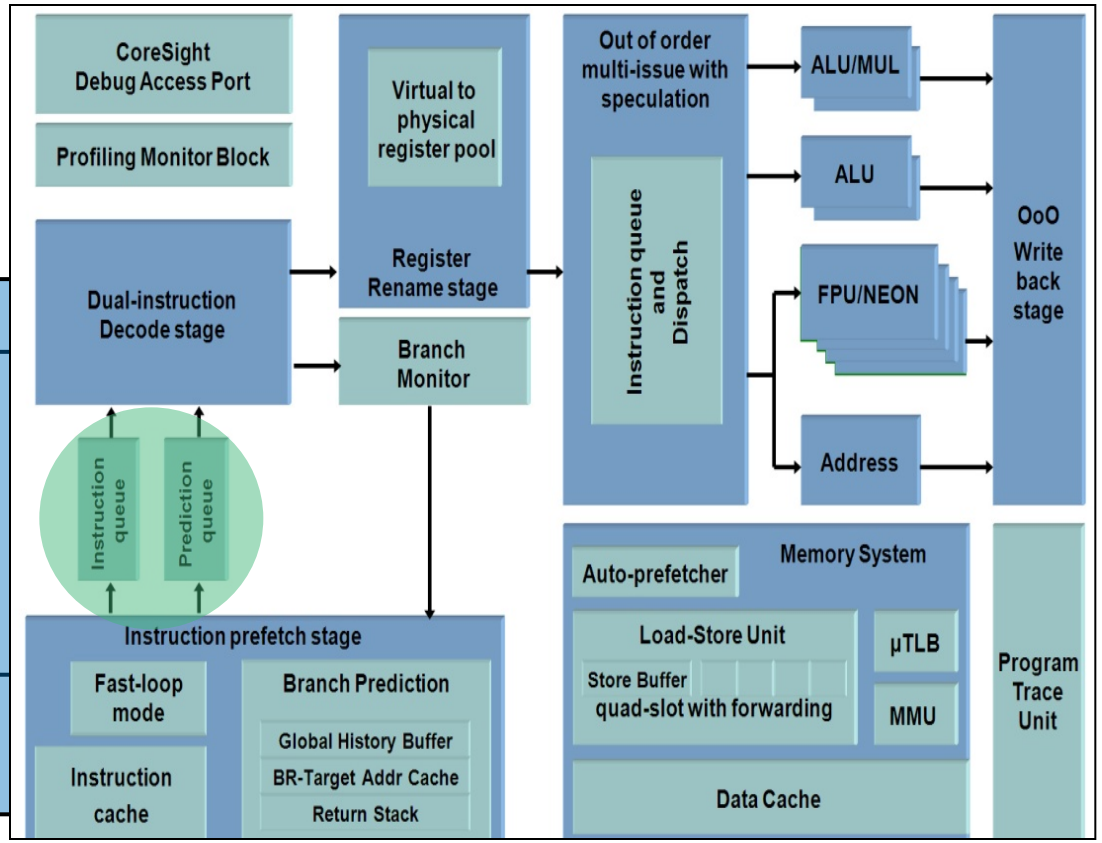
SPECIFICATION

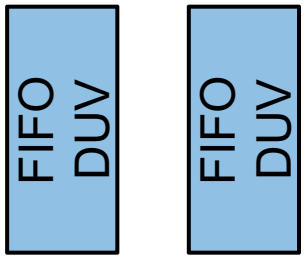
FIFO DUV AS FOR ABV SESSION

FIFO DUV

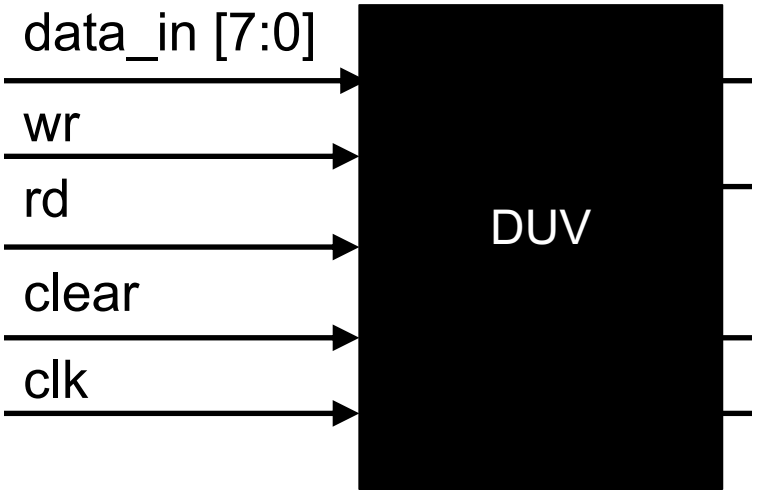
- Remember that the FIFO DUV is part of a larger unit, which in turn is part of a larger system.
- This lecture is focused on the verification of the FIFO at block level only.





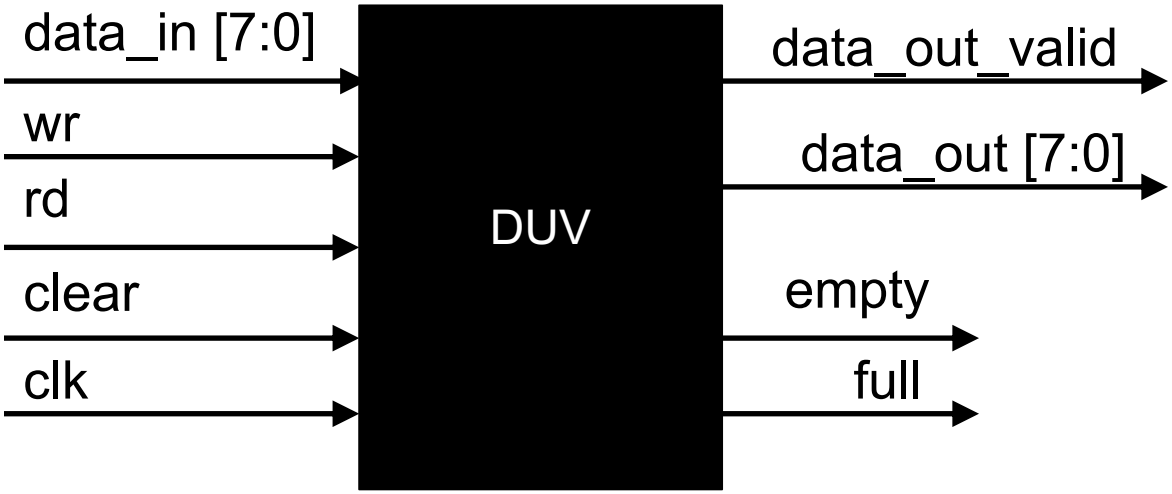


FIFO DUV Specification - Inputs



- Inputs:
  - wr indicates valid data is driven on the data_in bus
  - data_in is the data to be pushed into the DUV
  - rd pops the next data item from the DUV in the next cycle
  - clear resets the DUV

FIFO DUV Specification - Outputs



- Outputs:
  - data_out_valid indicates that valid data is driven on the data_out bus
  - data_out is the data item requested from the DUV
  - empty indicates that the DUV is empty
  - full indicates that the DUV is full

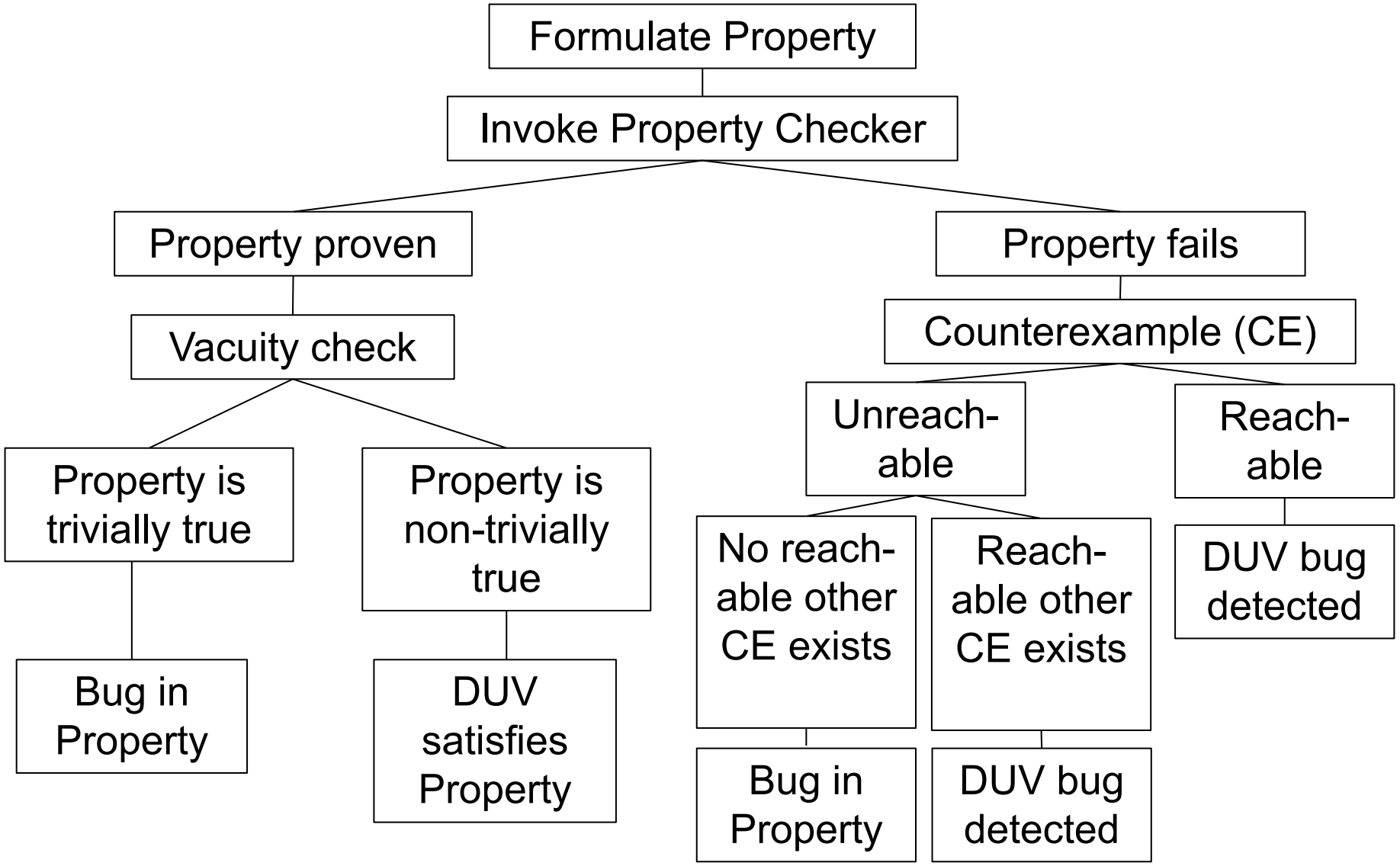
DUV Specification

- High-Level functional specification of DUV
  - The design is a FIFO.
  - Reading and writing can be done in the same cycle.
  - Data becomes valid for reading one cycle after it is written.
  - No data is returned for a read when the DUV is empty.
  - Clearing takes one cycle.
  - During clearing read and write are disabled.
  - Inputs arriving during a clear are ignored.
  - The FIFO is 8 entries deep.

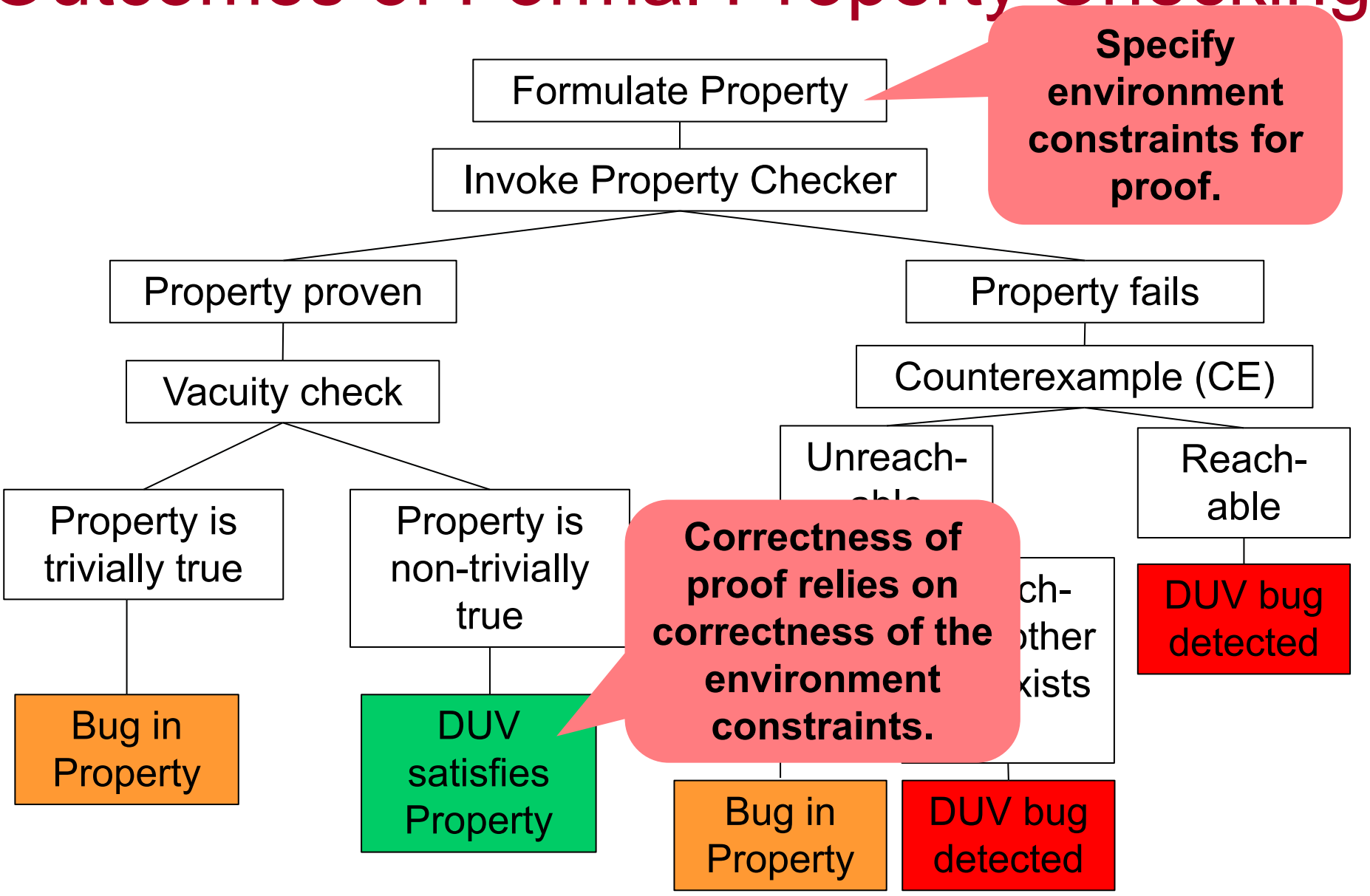
Revision

FORMAL VERIFICATION OUTCOMES

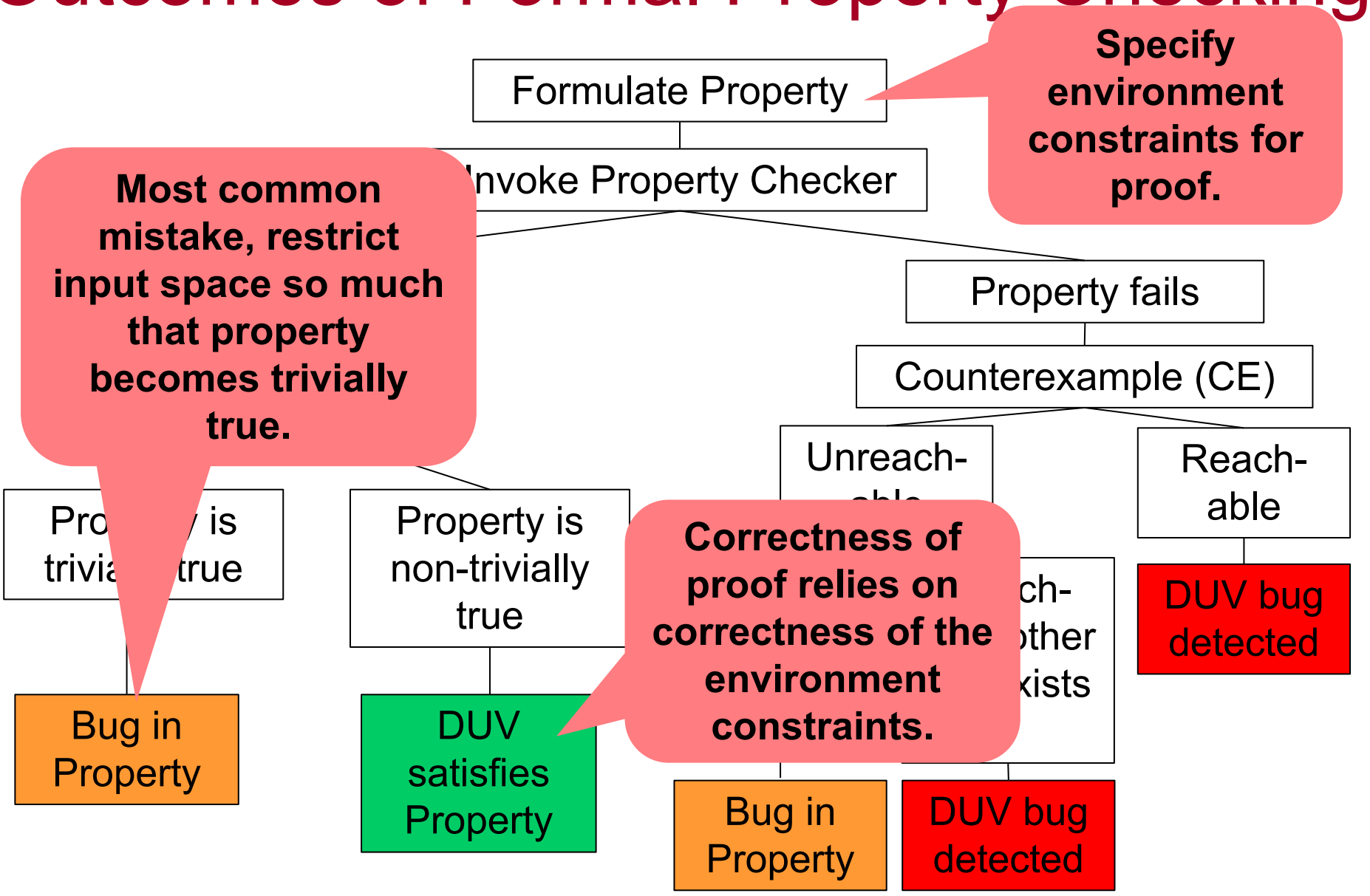
Outcomes of Formal Property Checking



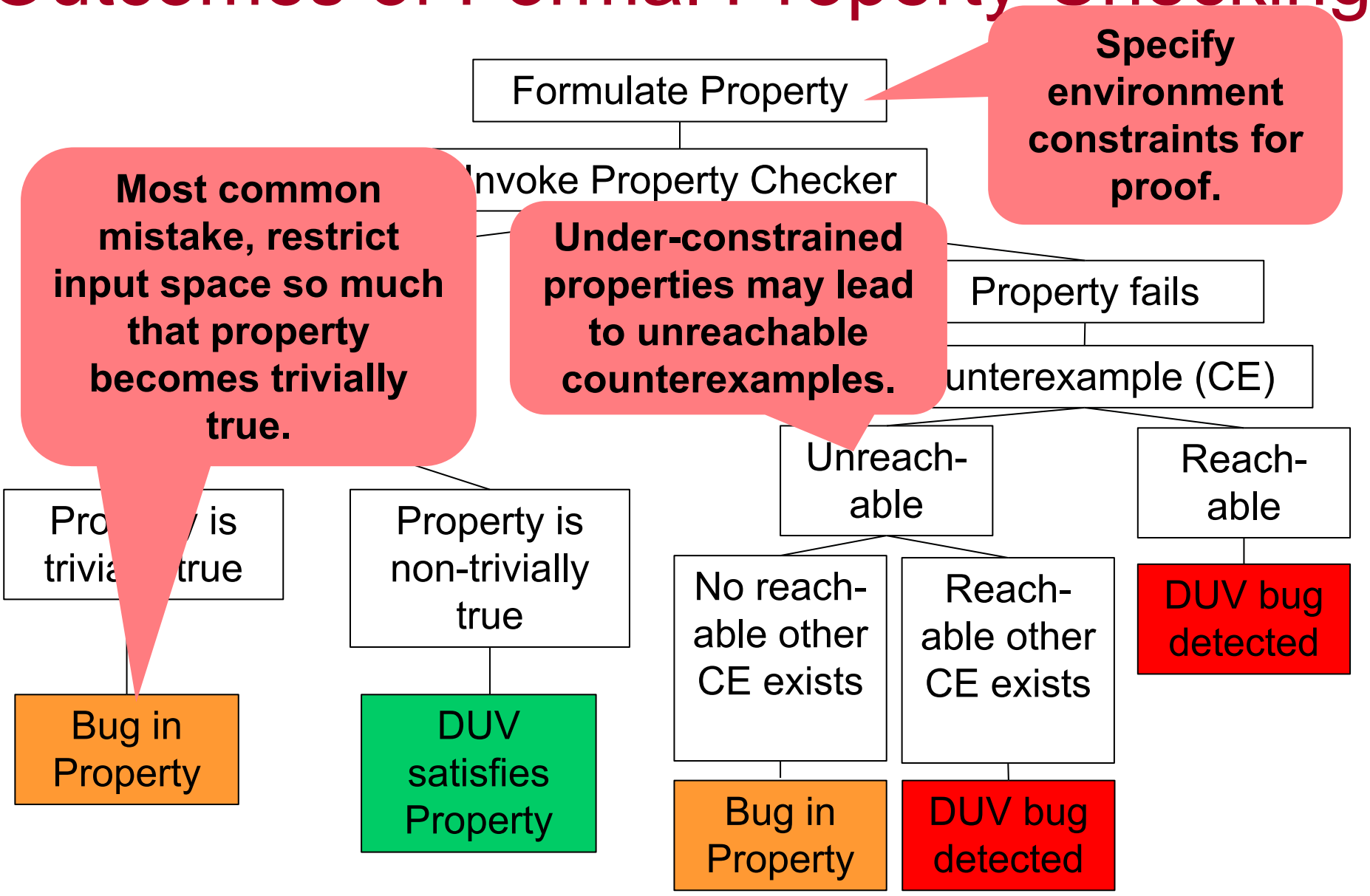
Outcomes of Formal Property Checking



Outcomes of Formal Property Checking



Outcomes of Formal Property Checking



How do you know you've encoded the property right?

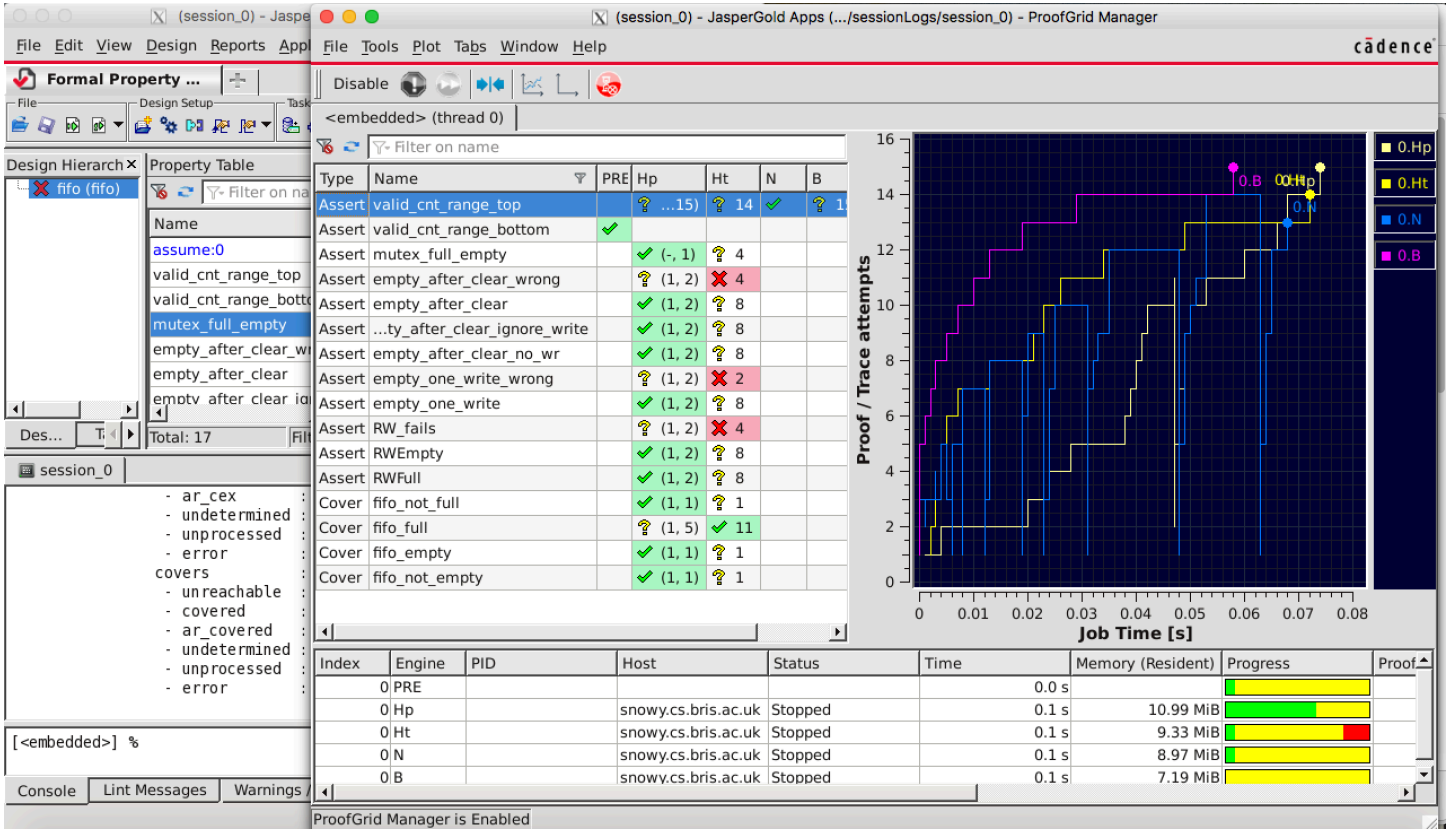
- Keep properties and sequences **simple**; build complex properties from simple, short properties.
- **Peer review** properties you write.
- **Know what to expect**, e.g. create failing conditions.
- If the property fails (when you expect it to succeed), then **investigate the counterexample:**
  - Is it reachable or not?

- **But if the property succeeds, how do you know whether you've encoded the property right?**

HANDS-ON FORMAL PROPERTY CHECKING DEMO

Formal Property Checking

- Jasper DEMO
  - **DUV:** FIFO design from ABV lecture
  - Verification of selected FIFO properties from ABV



The demo session includes

- **Automatic generation of basic properties using "Visualize":**
  - Basic functionality of the DUV
  - Range checks of signals
- **Verification of SVA properties:**
  - "Empty and full are never asserted together."
  - "After clear the FIFO is empty."
  - "On empty after one write the FIFO is no longer empty."
- **Inspect and understand counterexamples:**
  - Debug several failed properties

Note:
- Close link to coverage closure (by construction).
- Link from env_constraints to simulation assertions.

Next

- Recordings of lectures for **Week 8:**
  - ✓ Closing the Cycle
  - ✓ SoC Verification Guest Lecture (live on Wednesday)
  - – uobdv.github.io/Design-Verification/ shows a **weekly schedule of topics** to watch, ideally BEFORE the next online session
  - – Recordings are available from Blackboard unit page
- Tasks for you this week:
  - – Watch ”Closing the Cycle” and study associated notes on Analysis and Adaptation with a focus on Failure Analysis
  - – Get ready for **Coursework** part of the unit (if applicable)
    - to be released on Monday at 9am
    - EDA tool support
    - “feed forward” session in Week 10

Apple Guest Lecture 2021

Pete Vigil on *Design Verification for GPUs - Practice and Challenges*

Wednesday, 15 December 2021, 10am GMT

This talk covers the broad range of verification tools, techniques and processes, introduces aspects of GPU architecture, and explores the unique challenges of verification for GPU designs.

Speaker Bio: Pete Vigil is the site lead and design verification manager for the GPU group at Apple's St. Albans site. He has worked in verification for over 30 years, from start-ups to large companies, and on many types of designs including CPUs, GPUs, and other IP. Pete holds a BSCSE from Santa Clara University, has co-founded two verification start-ups. and is co-author on a number of design patents.

Questions

https://www.bristol.ac.uk/tsl