PRESENTER
Thomas E. Hansen

Background

Correctness is hard! You can use types to help, but this relies on the programmers choosing the right types for the problem. For example, using Integers for package numbering in a protocol would allow for negative packages.

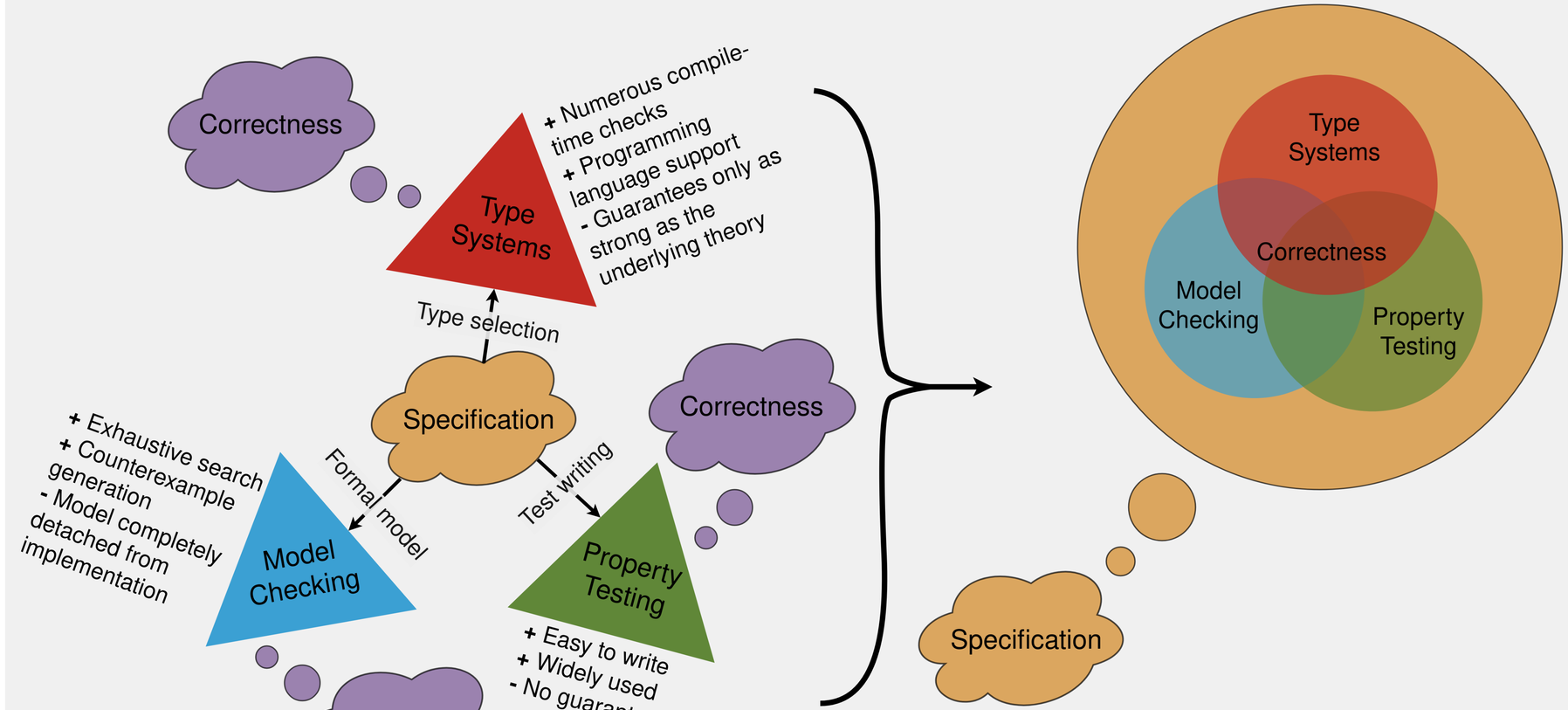
Existing Methods

- **Testing –** Very cheap
  There are no formal guarantees.
- **Type checking** – Relatively cheap
  Relies on correct modelling.
- **Model checking –** Very expensive
  Models often do not represent the full implementation.
- **Formal proofs** – Very expensive
  Proof systems tend to be separate from implementation.

The idea

Use termination- and type-checking to reduce the search-space for model checkers, and use model checkers to verify that our chosen types capture the desired behaviour. With tests being used to quickly iterate and gain confidence in the implementation.

Type systems can help with software correctness, but well-typed programs can go wrong! How do we know we've picked the right types?



University of St Andrews

THOMAS E. HANSEN, Edwin Brady