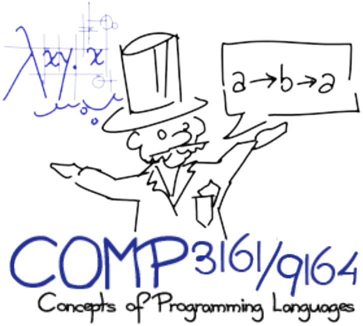

Revision Lecture

Rob Sison and Thomas Sewell
UNSW
Term 3 2024

That's it

As of Tuesday, we have now covered all the content in COMP3161/COMP9164. Thanks for sticking with the course.

- **Syntax Foundations**
  Concrete/Abstract Syntax, Ambiguity, HOAS, Binding, Variables, Substitution, λ-calculus
- **Semantics Foundations**
  Static Semantics, Dynamic Semantics (Small-Step/Big-Step), Abstract Machines, Environments, Stacks, Safety, Liveness, Type Safety (Progress and Preservation)
- **Features**
  - Algebraic (Sum/Product) Data Types, Recursive Types
  - Errors, Exceptions
  - Polymorphism (Universal Types), Type Inference, Unification
  - Abstract (and Existential) Data Types, Overloading, Subtyping
  - Concurrency, Session Types

MyExperience

Please fill out the survey. It helps tremendously.

`https://myexperience.unsw.edu.au`

Further Learning

- UNSW courses:
  - `COMP3131` — Programming Languages and Compilers
  - `COMP3153` — Algorithmic Verification
  - `COMP4141` — Theory of Computation
  - `COMP4161` — Advanced Topics in Software Verification
- Online Learning
  - Oregon Programming Languages Summer School Lectures (`https://www.cs.uoregon.edu/research/summerschool/archives.html`) Videos are available from here! Also some on YouTube.

What's next?

The exam is on **Friday, 29th of November 2024**, in the morning session (between 9:30am-1:30pm).

- You'll have 2 hours and 10 minutes within the 4-hour window.
- Online with a login – we'll provide you info on credentials.
- We have posted some sample exams with revision questions.
  (Note: Ignore sample questions on "most general unifiers" and STM.)
- The final exam will run similar to the sample exams.
- Open book: can use any passive resource (books, slides, google, etc), but:
  - **Not** allowed to ask for help from anyone.
  - **Not** allowed AI assistance for technical support (e.g. ChatGPT).
- If there are clarification questions, make **private** threads on Ed or **email us** at: `mailto:cs3161@cse.unsw.edu.au`.

Provisional: TS Hiring

Thomas & Rob work for/with CSE's Trustworthy Systems group.

For more info on our group: `https://trustworthy.systems`

Provisional: TS Hiring

We're looking to hire research assistants or supervise thesis students for software verification projects related to seL4.

- **Pancake language**: implement and verify compiler improvements
- **Microkit**: library code verification using deductive verification tools
- **component subpolicies**: verify userland code satisfies local security policies using deductive verification tools
- **inter-component protocols**: develop tool to validate correspondence between protocol model and C implementation
- **worst-case execution time**: develop formal reasoning framework for timeliness of seL4 userland application code
- **specification gap**: verify seL4's system calls behave the way the manual says they do using interactive theorem proving
- **time-protection extensions**: verify seL4 prevents data leakage between users via timing channels using interactive theorem proving

A background in OS or other theory/verification courses also helps.

If any of this sounds interesting to you, get in touch!

Your Requests and Sample Exams

We can now go through any questions you have, including from the sample exams.