Devin Yeung

✉ devin.cg.yeung.dev [at] outlook.com

 github.com/Devin-Yeung   blog

Education

BUN - HKBU United International College *2021.09 - 2025.06 (expected)*

Bachelor's Degree in Computer Science and Technology

- cGPA **3.93** / 4.00. Major cGPA **3.98** / 4.00. Rank (**1** / 147).
- **Dean's Award** in 2023-2024 (**Top 0.375%** schoolwide)   **Dean's Award** in 2021-2022 (**Top 0.375%** schoolwide)
- **National Scholarhsip** in 2023-2024 (**0.04%** nationwide)   **First Class Scholarship** in 2022-2023

Open Source Contribution

OXC Javascript Compiler *Byte Dance Web-Dev Team*

- Contributed more than 3k lines of code to the linter component of the `OXC` named `oxlint`, involving:
  - **Implemented ESLint Rules**: e.g. `no-thenable`, `no-unnecessary-await`, `ban-ts-comment`
  - **Improved the Continuous Integration**: include bug fixes and new features to enhance the development experience.

Bnfgen   Slides *Undergraduate Graduation Project*

- A general purposed BNF grammar generator, which can generate random strings from a given BNF grammar.
- Enhanced the standard BNF grammar with **custom extensions**, featuring:
  - Weighted productions for **probabilistic** generation   Seamless integration of **regular language**
  - Progressive **type system** for production rules   **Variable tracking** with declaration and reference
- Rich **semantic analysis**: include **duplicated** rules, **unreachable** rules and **dead loop** rules detection.

Loxide: A Lox Interpreter in Rust *Personal Project*

- Developed a Lox interpreter from scratch, including a **lexer**, recursive descent **parser**, **evaluator**, and **type checker**.
- Adopted Rust's exemplary **error reporting** techniques to generate clear, human-readable error messages.

Context-free Tools *Personal Project*

- Developed a comprehensive suite of tools for **analyzing context-free grammars**, including common algorithms like `First`, `Follow`, `LR(0)` set construction and `SLR` parsing table construction.
- Focused on **educational purposes**, the toolkit includes detailed visualizations and step-by-step explanations.

Competition

Wordle killer: The counterattack of information theory *2023.02*

- Analyzed the daily results of the Wordle puzzle from the New York Times (thesis).
  - Used **FB-Prophet** model and **Monte Carlo Simulation** to predict the future distributions of Wordle result.
  - Developed an algorithm based on **information entropy** to measure the difficulty level of a Wordle puzzle.
- Awarded the **Finalist** (**Top 1.8%** worldwide) in 2023 **Mathematical Contest in Modeling** (aka MCM / ICM).

Internship

**Teaching Assistant of Compiler Construction Course (2024 Fall)** *2024.09 - Now*

- Designed the **BNF grammar** and implemented the **autograder** (automatic grading system) for the course project.
- Developed tools to generate parser test cases and a dedicated **playground** for students to better understand the project.
- Maintained a high-performance **code similarity checker** to catch students' plagiarism behavior. (open source soon)

Project

Modeling and Control of 2-DOF Robot Arm *Course Project*

- Investigated control aspects of a two-degree-of-freedom robotic arm, emphasizing PID control, **kinematics**, **dynamics**, and **controller design**. Developed a `MATLAB` simulation to verify the correctness of algorithms. (thesis)

Skills

- **Technical Skills**:
  - **Programming Skills**: Not limited to specific language, **proficient in Rust**, familiar with Python, OCaml and C.
  - **Tools**: Comfortable with linux environment, familiar with git, skilled in leveraging AI.
- **Domain Specific Skills**
  - **Parser**: Comfortable with parser combinators and parser generators, familiar with creating hand-written parsers.

- **Formal Languages**: Proficient in regular expressions, context-free grammars, and automata theory.