Ethan Childs

Dallas, Texas | 817-675-9073 | EthanChildsbc@gmail.com | linkedin.com/in/ethanchilds | github.com/Ethan-Childs

Education

Texas Tech University Lubbock, Texas
Bachelor of Science Expected Graduation: May 2026
Major: Computer Science
Minor: Mathematics

- GPA: 3.62
- President's List: Spring 2023, Fall 2024
- Dean's List: Fall 2021, Spring 2022

Work Experience

Data Analyst Lubbock, Texas
NemaLife inc. January 2025 - Current

- Trained the company AI to enhance the accuracy of pharmaceutical test data on C. *elegans*.
- Performed video annotations to ensure accurate data was presented to clientele.
- Cataloged and managed over 1000 annotation and AI statistics in Excel Spreadsheets.
- Coordinated with 10 other Data Analysts to ensure project deadlines were met on time.

Sports Statistician Lubbock, Texas
Genius Sports (Seasonal) October 2024 - December 2024

- Updated live game statistics for Texas Tech football games on company software to determine betting odds.
- Compiled reports at the end of the game including average stats and future projections for key players.
- Worked diligently to have live game statistics updated before the start of the next sequence.

Projects

NASA Collaboration: Space Biology Research
NemaLife inc.

- Provided video annotations to track the health of the C. *elegans* and how they responded on different microchips.
- Coordinated with 10 other data analysts to ensure all data was correctly organized into Excel Spreadsheets.

Cellular Automaton Simulator
Multiprocessing

- Developed in Python this simulator performs iterations on files containing matrices full of cells represented by certain characters and returns the final matrix to the terminal and a new output file.
- Integrates Python's multiprocessing library for parallelizing row-wise matrix generations after out-of-bounds index wrapping with Modulo Arithmetic.
- Utilizes mathematical conditions and algorithms for determining the next state of the cell for the next iteration.

Front-end Compiler
Lexical Analyzer

- Developed in C this Lexical Analyzer preforms tokenization on 26 lexemes using deterministic finite automaton logic.
- Implements whitespace and newline handling to track the line of the file to ensure no matter what is on the file the data will be accurately analyzed.
- Flexible enough to run on its own without the Syntax Parser in an IDE or with a Makefile to read the given file and return it as tokens defined in the token.h file.

Syntax Parser

- Developed in C this Syntax Parser implements the Lexical Analyzer to add the function of letting the user know if there is a syntax error and what line the error is occurring on.
- This Syntax Parser accounts for the left-wise recursion problem in the BNF rules of Cooke's Programming Language.
- Integrated a custom built Makefile to compile faster in a Linux server environment when dealing with very large files.

Technical Skills

Programming Languages

- Python, C, Java, Bash, LaTex

Software Tools

- Excel Spreadsheets, Microsoft Word, Microsoft PowerPoint