Evaluation Item	Unacceptable (0)	Acceptable (1)		Comments
Verifiability Note: If this item is "unacceptable", the rest of the assignment will not be marked.	The submission lacks a script for validating the parser, or the script provided could not be used to verify the parser.	The submission includes (a) a set of tests for verifying the parser and (b) a script to build your parser and to parse those tests. That scripts can be used on a lab machine to verify the parser outputs.		testconcat.py – hardcoded to a directory we don't have.
Evaluation Item	**Unacceptable (0)**	**Marginal (2)**	**Proficient (3)**	**Comments**
Correctness	The extension or optimization has broken the compiler's ability to generate code in many cases.	The extension or optimization has caused the compiler to fail in edge cases.	The compiler continues to generate legal code with the extension or optimization enabled.	
Robustness (new feature)	The extension or optimization only works on a very narrow set of cases. Alternatively, no extension or optimization is present.	The extension or optimization fails to account for some edge cases or rules out some common cases.	The extension or optimization is capable of handling unexpected or incorrect input gracefully.	

Evaluation Item	Unacceptable (0)	Marginal (2)	Proficient (3)	Comments
Documentation	The extension is not clearly identified or the document does not explain its purpose. Alternatively, there is no documentation to support the submission.	The extension is identified clearly, however, the documentation fails to distinctly illustrate/explain how it is implemented or why it is interesting.	The extension or optimization is identified clearly. The document explains what it does, how it is implemented, and why it is interesting.	You're doing dead code removal, in addition to just analysis. (2)

Evaluation Item	Unacceptable (0)	Marginal (3)	Proficient (5)	Comments
Overall Design	The compiler is implemented as a monolithic structure, with no clear delineation between the phases.	The overall design of the compiler could be improved by removing cross-coupling or factoring the optimization(s) or extension(s) into separate stages.	The overall design of the compiler is modular and clean. Cross-coupling of components has been minimized.	
Robustness (overall compiler)	The compiler only works in a narrow set of cases.	The compiler occasionally fails due to unexpected input.	Overall, the compiler is capable of handling unexpected or incorrect input gracefully.	

Additional Notes:

This is a cool idea that solves a real problem. We are impressed by how naturally the script reads. You did a good job with the language design step. We think there might need to be a bit more thought put into implied errors – errors that occur if an action cannot complete (because an item (like a button) isn't there, for example).

Group Members: **Vasin and Zhang** **Score:** **19** **/20**