Undoing Software Engineering: Demodularization of a SGLR Parser for Performance Gains

Background:

JSGLR2 is a Scannerless Generalized LR-parser implemented in Java. It is very modular, each piece of the algorithm can be swapped out with different implementations.
However support for modularity inflated the codebase significantly.

Questions:

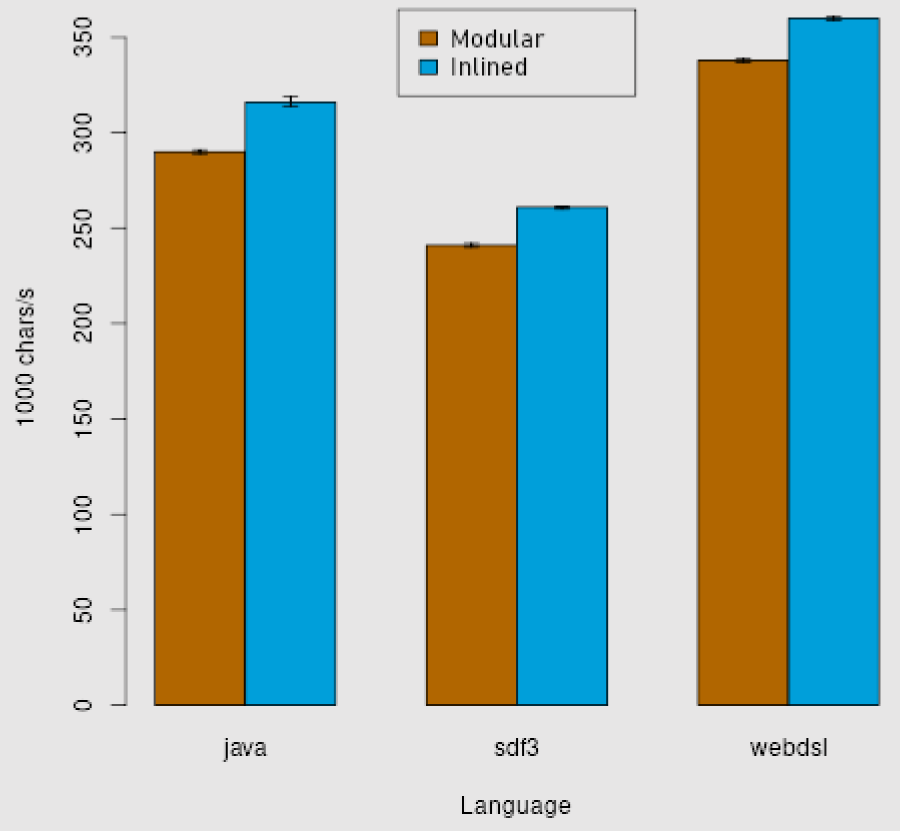
- What is the best way to strip out modularity from the parser?
- Does this "inlining" improve performance, and if so, how much?
- What components are the largest bottlenecks?

Methodology:

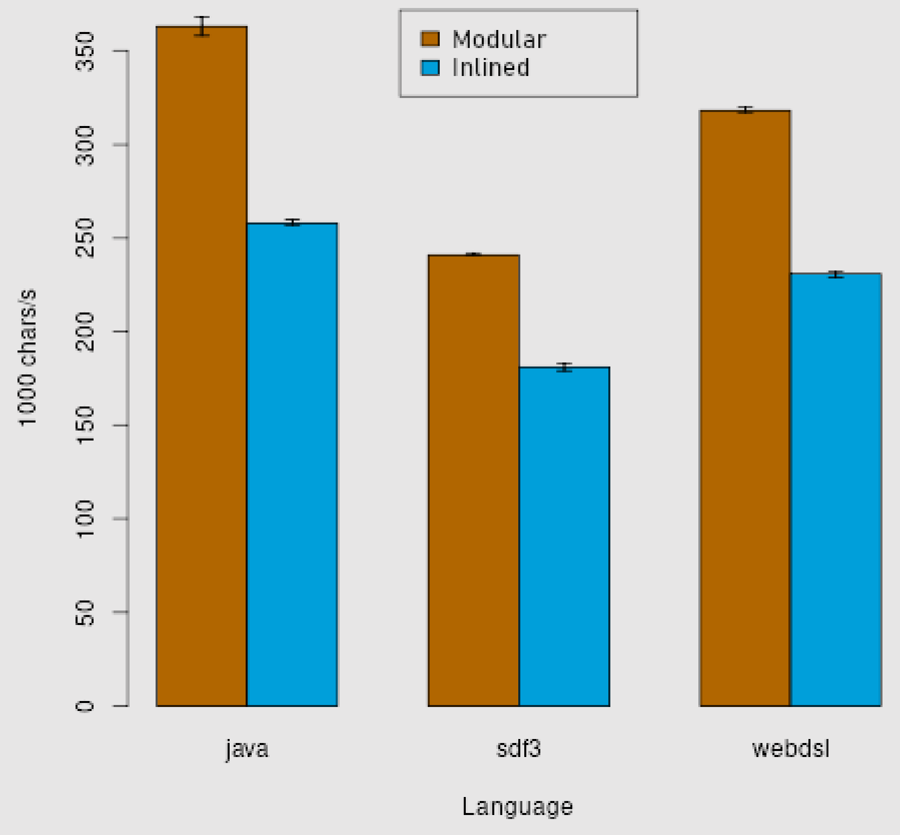
- Replace dependency-injected components with hardcoded instances.
- Remove inheritance
- Change observer mechanism

Results:

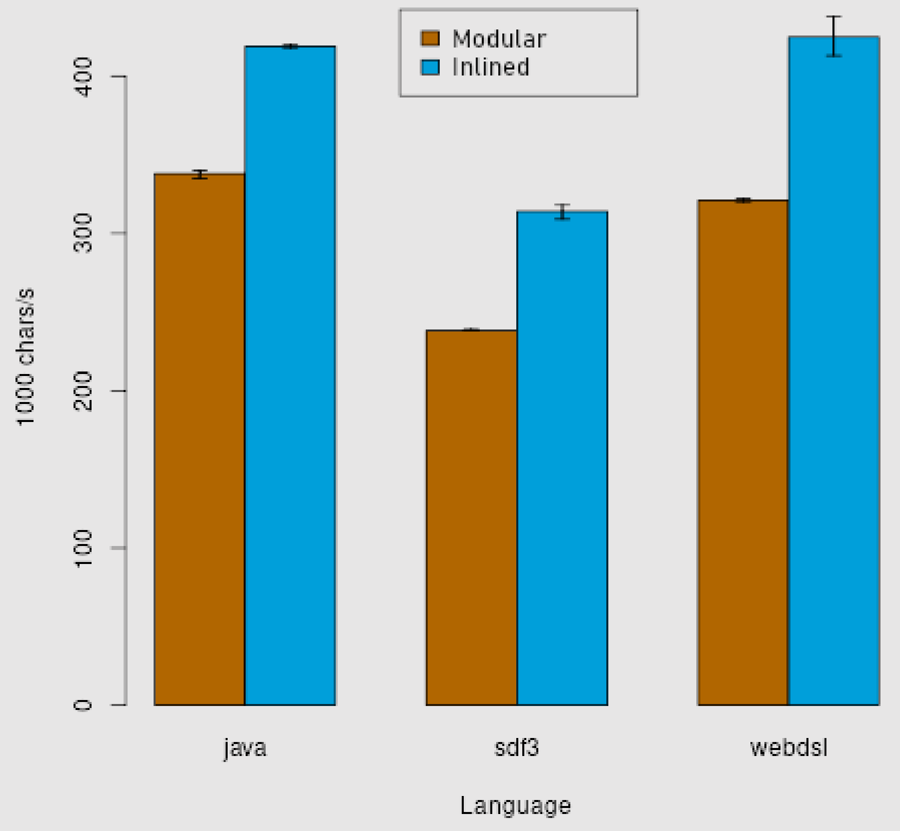
Parsing Througput: With inlined observers



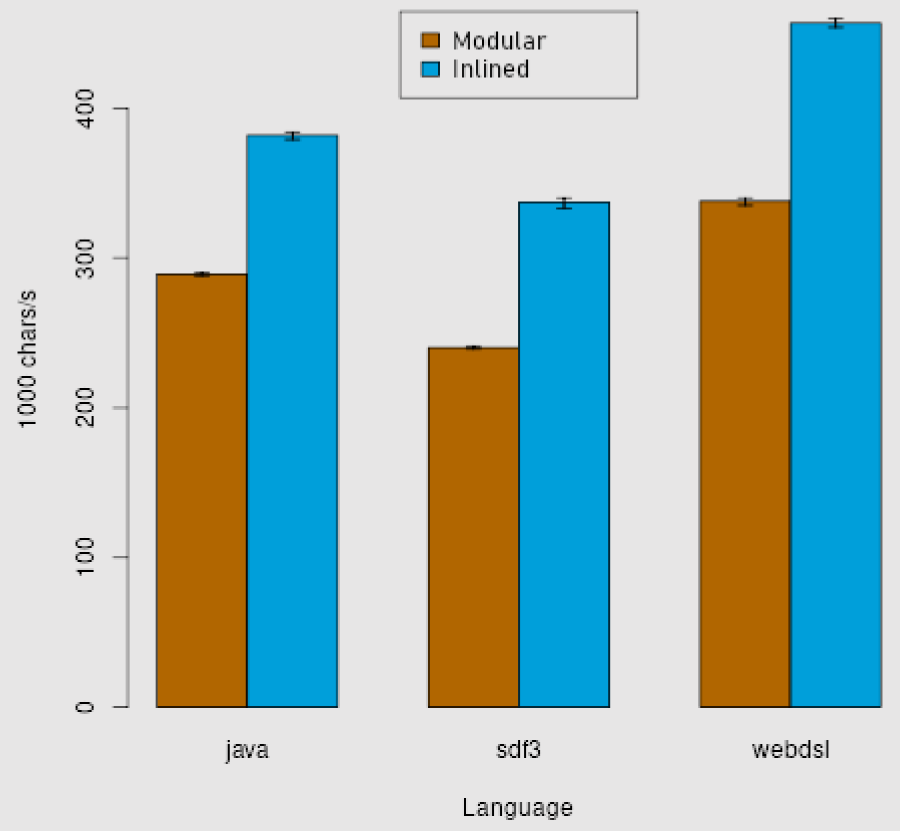
Parsing Throuhput: With integrated measurements



Parsing Throughput: No observers



Parse Throuput: No observers + classes marked final



Conclusion:

- Inheritance removal gives improved performance
- Removing observers also gives better speeds.

Nik Kapitonenko
M.Kapitonenko@tudelft.nl

Supervisors:
Jasper Denkers
Daniel Pelsmaeker