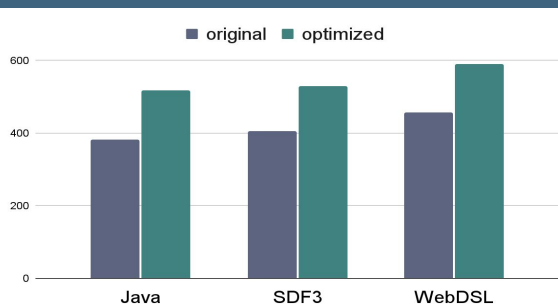# Optimizing SGLR Parser Performance

TUDelft

1. Background

- SGLR enables declarative and modular language definitions
- SGLR relatively slow:
  - Generalized parsing has a cubic runtime
  - Scannerless parsing requires more work
- JSGLR2 is a modular SGLR implementation introduced to simplify optimization

2. Method

A. Identify

- Identify bottlenecks using a profiler
- Identify usage patterns using measurements

B. Optimize

- Apply a known optimization
- Find a way to eliminate unnecessary work
- Optimizations implemented modularly

D. Evaluate

- Optimizations evaluated separately
- Performance compared to original implementation
- Evaluation using languages and sources from real-world projects

C. Verify

- Verify correctness of optimizations using integration tests

3. Results

- 29% to 35% faster non-deterministic parsing
- Reaching a practical throughput of 500k chars/s

original ■ optimized

Justin de Ruiter
j.deruiter-3@student.tudelft.nl

Supervisors   Jasper Denkers
Eelco Visser   Daniël Pelsmaeker