Parsing Tools Research Proposal: ArkTS Parser

Semyon Grigorev

Saint Petersburg State University

2024

Why not existing tools?

- ANTLR
  - ☹ Not incremental: it is unlikely possible to support accurate incremental parsing without totally rework
  - ☹ No advanced error recovery
- TreeSitter
  - ☹ Problems with error recovery
  - ☹ Returns new ATS on input update: all your caches became useless
- Babel + ANTLR = Current solution
  - ☹ Poor performance
  - ☹ Hard to extend grammar
  - ☹ Not incremental
  - ☹ No advanced error recovery

Prpoposed solution

ArkTS parser on our own parsing infrastructure

Challenges:

- Nontrivial grammar (ambiguities, etc.)
- Performance
- Error recovery
- Incremental parsing

Current State[1]

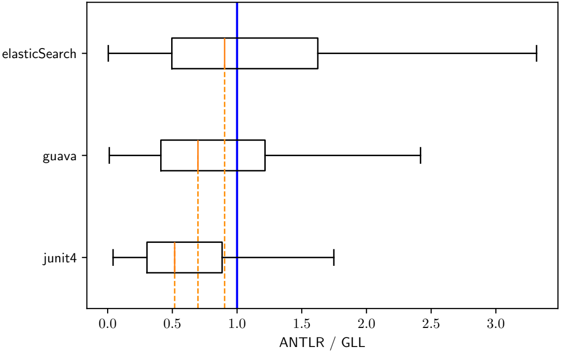
- ✓ Basic parser development tool
  - ✓ Grammar description DSL
  - ✓ Preliminary performance evaluation
- ✓ Error recovery mechanism
  - ⚙ Preliminary performance evaluation
- ⚙ User-friendly AST generation

- ⌛ ArkTS parser development
  - ▸ ANTLR grammar of TypeScript
- ⌛ Documentation
- ⌛ Performance tuning
- ⌛ Advanced incremental parsing

[1] https://github.com/FormalLanguageConstrainedPathQuerying/UCFS/tree/cf_solver

Preliminary Evaluation Result

- Java grammar
- 3 real-world projects
  - junit4: 425 files, avg. size 3KB (40KB max)
  - guava: 1 416 files, avg size 8KB (198KB max)
  - elasticSearch: 14 685 files, avg size 6KB (242KB max)



Research Tasks

- Implement JavaScript subset parser
  - Evaluate performance of implemented parser
- Implement ArkTS subset parser
  - Evaluate performance of implemented parser
- Performance tuning
- Grammars tuning
- Error recovery evaluation and tuning
- Incremental parsing implementation