

ANTLR Tips & Tricks  
by Monty Zukowski  
Details available at <http://www.codetransform.com/tips.html>

General Principles

- look at generated code
- micro revisions, good log notes
- myriad mini tests
- goal is to be able to detect problems quickly and be able to revert to a known good build.

How to grow a grammar

- start by stealing lexer rules
- grab a body of code to test against if available
- transliterate BNF into ANTLR (rules on web for doing that)
- take it one test at a time
- continually regression test
- I like to get parser stable before building trees
- building trees properly may require factoring subrules into rules

Disambiguation strategy

- adding a new rule can cause ambiguities with other rules due to FOLLOW
- give example from GCC
- look at lookahead sets for a clue
- use syntactic predicates
- factor rules
- recursion v. closure

Tracing and debugging

- use -trace\* options
- condense output using smarter traceIn/traceOut methods
- you can use a debugger to inspect state too

Lexer issues

- prefix factoring
- token types out of sync
- parser filtering

Parser issues

- proper use of predicates
  - validating v. disambiguating
  - minimal syntactic predicates
- building trees
  - infinite loops
  - tree form
  - do in parallel with tree grammar
- don't be too strict, it may be easier to validate in a separate pass

Tree grammar issues

- tree design based on the application
- structuring for multiple passes
- how to represent two languages in one tree