

# Hinteransichten eines Syntaxanalysewerkzeugs

Baltasar Trancón y Widemann  
TU Ilmenau,  
semantics GmbH Berlin

Markus Lepper  
semantics GmbH Berlin

Das weitgehend automatisierte Erzeugen von konkreter Software (Parser) aus abstrakten Spezifikationen (Grammatiken) hat im Bereich der Programmierwerkzeuge eine lange und fruchtbare Tradition. Dabei stehen nach wie vor mehrere grundverschiedene Paradigmen im Widerstreit:

- LR-Parsergeneratoren (Yacc etc.) und die von ihnen erzeugten monolithischen bottom-up Parser (Automaten) punkten mit Ausdrucksstärke im deterministischen Regime, Maschinennähe und guter Integration in imperative Sprachen.
- LL-Parsergeneratoren (ANTLR etc.) und die von ihnen erzeugten monolithischen top-down Parser (rekursiver Abstieg) punkten mit transparentem Code und guter Integration in objektorientierte Sprachen.
- Parserkombinatoren (Parsec etc.) punkten mit Kompositionalität, Inkrementalität und guter Integration in dynamische und funktionale Sprachen.

Wir berichten hier über erste Experimente mit einem kombinatorischen Ansatz, der das beste verschiedener Paradigmen zu vereinen versucht...

- zweistufige eingebettete domänenspezifische Sprache in Java:
  - syntaktische Kombinatoren
  - semantische Aktionen;
- geeignet zur direkten Benutzung als Kombinator-Bibliothek, oder als Zielsystem für einen LL-artigen Parsergenerator;

... sowie neue Lösungen einiger notorisch offener Probleme vorschlägt:

- volle Unterstützung von Nichtdeterminismus durch Backtracking, inklusive kompensierbarer semantischer Aktionen, im Stil monadischer funktionaler Parser;
- saubere Integration von Diagnosemeldungen und Nichtdeterminismus;
- kombinatorisches Beschneiden von Suchbäumen mit Heuristiken (LL-Direktoren etc.);
- robust gegenüber Grammatik-Transformationen.