

accent

A compiler compiler for the entire class of context-free languages

`accent.compilertools.net`

`accent` ist ein C Parser-Generator, welcher an der Fraunhofer FIRST entwickelt und unter den Bedingungen der GPL2 zur Verfügung gestellt wird.

Im Gegensatz zu üblichen LL/LR-Parser-Generatoren wird bei `accent` die die Technik des *exhaustive parsing* umgesetzt. Somit können Parser für *beliebige* kontextfreie Grammatiken erzeugt werden.

Parser generieren und integrieren

Zur Erzeugung des Parsers wird vom Nutzer eine Spezifikationsdatei geschrieben, welche die Grammatik zusammen mit semantischen Aktionen enthält. Anhand dieser Spezifikation generiert `accent` eine C-Datei welche den Code des Parsers beinhaltet.

Die Integration des generierten Parsers in ein Programm erfolgt auf gleiche Weise wie bei `yacc`-generierten Parsern. Entsprechend sind von `accent` generierte Parser darauf ausgelegt, Lexer zu benutzen, welche mit `(f)lex` erzeugt wurden.

Grammatik-Spezifikation

Die Regeln der Grammatik werden in einer Syntax spezifiziert, welche an EBNF angelehnt ist. Dabei werden alle EBNF-Konstrukte (Alternative, Option, Wiederholung, Gruppierung) angeboten.

`accent` stellt zwei Konzepte zur Verfügung, um Mehrdeutigkeiten zu vermeiden: Erstens ist es möglich, *Prioritäten* auf Regel-Alternativen zu vergeben. Zweitens können *kürzest- bzw. längstmögliche Ableitungen* (bzgl. Anzahl aufgebrauchter Token) von Nichtterminalen erzwungen werden.

Semantische Aktionen

In eine Regel können an beliebiger Stelle Blöcke mit C-Code eingefügt werden. Beim Verarbeiten einer Eingabe werden diese Blöcke von einem *abstract syntax tree-walker* entsprechend der Reihenfolge ihres Auftretens in der Regel aufgerufen.

Nichtterminalen können beliebige Rückgabewerte und Eingabe-Paramameter zugewiesen werden, um Datenfluss zwischen den Semantik-Blöcken unterschiedlicher Regeln zu realisieren.

Bewertung

`accent` ist nicht nur bzgl. beherrschter Grammatiken mächtiger als sein Pedant `yacc`, sondern aufgrund von EBNF und erweiterten Möglichkeiten Semantik hinzuzufügen auch deutlich nutzerfreundlicher.

Dennoch ist `accent` nicht für alle Einsatzbereiche gut geeignet. So fehlen sinnvolle Konzepte zur Fehlerbehandlung. Das parsen kann nach Auftreten eines Syntaxfehlers oder einer Mehrdeutigkeit nicht fortgesetzt werden.

Auch der Bedarf an Speicherplatz und die benötigte Laufzeit des *exhaustive Parsing* schliessen den Einsatz aus, falls umfangreiche Eingaben effizient verarbeitet werden müssen.