Attribute grammar

From HaskellWiki

Contents

Introduction

Especially for functional programmers (with Haskell examples) -- see Wouter Swierstra: Why Attribute Grammars Matter (http://www.haskell.org/haskellwiki/The_Monad.Reader/Issue4/Why_Attribute_Grammars_Matter) (published in The Monad.Reader, Issue Four (http://www.haskell.org/haskellwiki/The_Monad.Reader/Issue4)). And also very deep connections with

- catamorphisms, category theory
- attribute grammars

are described there. And there are links from that article to other materials providing deep insights in these powerful theories: attribute grammar tools and articles of the Essential Haskell Compiler project.

It is no wonder that it is exactly a compiler project homepage that provides good circular programming and attribute grammar tutorials

- both on its project page (http://www.cs.uu.nl/wiki/Ehc/WebHome)
- and on a separate attribute grammar (http://www.cs.uu.nl/wiki/HUT/AttributeGrammarSystem) page:

the reason may be that a compiler project is complex enough to require good separation of concerns, modularity, reuse (goals of Aspect Oriented Programming, too), and attribute grammars are powerful tools to achieve these goals.

- Wikipedia article (http://en.wikipedia.org/wiki/Attribute_grammar)
- An Attribute Grammar tutorial by Swiestra Attribute grammars – A short tutorial (http://foswiki.cs.uu.nl/foswiki/pub/Swierstra/ResearchTalks/AgTutorial-org.pdf)
- Also Happy (http://www.haskell.org/happy/) contains AG-related capabilities, see Chapter 4. Attribute Grammars (http://www.haskell.org/happy/doc/html/sec-AttributeGrammar.html)
- Attribute Grammars were originally proposed be Donald E. Knuth in Semantics of Context-Free Languages (http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.455.1434&rep=rep1&type=pdf).
- Knuth describes the origin of the ideas behind Attribute Grammars in The Genesis of Attribute Grammars (http://www.dcs.warwick.ac.uk/~sk/cs325/gag.pdf). Furthermore he gives a rich bibliography and some error corrections for some of his papers there.

Separation of concerns

Both Swiestra's article (mentioned above) and the Essential Haskell Compiler Project tool (mentioned below) remarks the connection of attribute grammars to the topic of separation of concerns, a goal of aspect orinted programming. This goal can be achieved by multiple ways in functional programming, see the concepts monad and arrow). Swiestra'a article mentions analogies between

- attribute grammars,
- (Reader, Writer, State) monads,
- (fan-out operation of) arrows.

Portals or other rich resources

- Attribute Grammar page of Utrecht Univesity (http://foswiki.cs.uu.nl/foswiki/HUT/AttributeGrammarSystem)
- Description of Attibute Grammar aspects in the Happy parser generator system for Haskell (https://www.haskell.org/happy/doc/html/sec-AtrributeGrammarsInHappy.html)
- The FNC-2 Attribute Grammar System (http://www-sop.inria.fr/members/Didier.Parigot/www/fnc2/index.html)
- Resources for Attribute Grammar research by Didier Parigot (INRIA) (http://www-sop.inria.fr/members/Didier.Parigot/www/fnc2/AGtexte.html)
- Team OSCAR at INRIA is stopped since 1999 but the following page contains some older publications: Former Team OSCAR Home Page (https://www.inria.fr/en/teams/oscar)

Tools

Utrecht University's Attribute Grammar System (http://www.cs.uu.nl/wiki/HUT/AttributeGrammarSystem) tools include also an attribute grammar compiler, UUAGC. The concept of attribute grammar was used in their Essential Haskell Compiler (http://www.cs.uu.nl/wiki/Ehc/WebHome) project, which gives us not only a working programming language, but also a good didactical material about using attribute grammars, e.g. in writing compilers.

Albeits these materials are self-contained, they reveal that the theory of attribute grammars is related to other concepts (circular programming, catamorphism).

Robert Dockins has reported Attribute Grammar Support for Happy (http://haskell.org/communities/06-2006/html/report.html#happy-ag) in the Haskell Communities and Activities Report (10th edititon, June 2006)

Related concepts

- Circular programming
- Catamorphism, see Category theory

Retrieved from "https://wiki.haskell.org/index.php?title=Attribute_grammar&oldid=62300"

Categories: Theoretical foundations | Compiler tools

- This page was last modified on 26 January 2018, at 07:06.

- Recent content is available under simple permissive license.