0.1 UUAG

Report by:	Arie Middelkoop
Participants:	ST Group of Utrecht University
Status:	stable, maintained

UUAG is the *Utrecht University Attribute Grammar* system. It is a preprocessor for Haskell which makes it easy to write *catamorphisms* (that is, functions that do to any datatype what *foldr* does to lists). You can define tree walks using the intuitive concepts of *inherited* and *synthesized attributes*, while keeping the full expressive power of Haskell. The generated tree walks are *efficient* in both space and time.

New features are support for polymorphic abstract syntax and higher-order attributes. With polymorphic abstract syntax, the type of certain terminals can be parameterized. Higher-order attributes are useful to incorporate computed values as subtrees in the AST.

The system is in use by a variety of large and small projects, such as the Haskell compiler EHC, the editor Proxima for structured documents, the Helium compiler, the Generic Haskell compiler, and UUAG itself. The current version is 0.9.6 (April 2008), is extensively tested, and is available on Hackage.

We are currently improving the documentation, and plan to introduce an alternative syntax that is closer to the Haskell syntax.

Further reading

http://www.cs.uu.nl/wiki/bin/view/HUT/AttributeGrammarSystem
http://hackage.haskell.org/cgi-bin/hackage-scripts/package/uuagc-0.9.6