HaMLet

"To Be Or Not To Be Standard ML" ;-)

Version 2.0.0

What is it?

HaMLet is a faithful and complete implementation of the Standard ML programming language (SML'97). It aims to be

- an accurate reference implementation of the language specification,
- a platform for experimentation with the language semantics or extensions to it,
- a useful tool for educational purposes.

The implementation is intended to be as direct a translation of the language formalisation found in the Definition of Standard ML [1] as possible, modulo bug fixes. It tries hard to get all details of the Definition right. The HaMLet source code

- implements complete Standard ML,
- closely follows the structure of the Definition, with lots of cross references,
- conforms to the latest version of the Standard ML Basis Library [2],
- is written entirely in Standard ML, with the ability to bootstrap,
- may readily be compiled with SML/NJ, MLton, Moscow ML, Poly/ML, Alice ML, the ML Kit, or SML#.

HaMLet can perform different phases of execution — like parsing, elaboration (type checking), and evaluation — selectively. In particular, it is possible to execute programs in an untyped manner, thus exploring a universe where even ML programs "can go wrong".

It should be emphasized that HaMLet is by no means a development system, but has been solely written with the aforementioned goal of experimentation in mind. Interpretation is highly inefficient (since it is a direct implementation of the semantic rules) and error messages are rather basic. However, HaMLet is able to bootstrap itself.

As a byproduct, the HaMLet documentation contains a comprehensive list of all known bugs and `grey areas' in the current version of the SML language definition, which may be interesting on its own.

What's new?

Release 2.0 (2013/10/10) brings a major revamp of the internal AST representation. In particular, elaboration now stores its results in the AST, which should make HaMLet more useful as an experimental compiler front-end. As a proof of concept, the release also integrates a simple compiler to JavaScript.

The most significant changes are:

- Restructured AST to include annotations in the form of typed property list (breaks all code based on HaMLet 1, sorry `:(`).
- Elaboration stores result of each rule as annotation in AST.
- Other restructurings to better support compilers, e.g., separate Elab/EvalProgram modules, and split StaticLibrary and DynamicLibrary.
- Added simple JavaScript compiler and runtime as a proof of concept, accessible via the newly added `-j` mode.
- Various bug fixes, improvements, code clean-ups and beautification.

See the change log for more details.

Download

The HaMLet sources are available as a tarball, zipfile or Debian package:

- Sources (gzipped tar or zip) — includes PDF version of the documentation
- GitHub repository
- Standalone documentation (Postscript or hyperref'd PDF)
- A list of defects in the Definition — derived from Appendix A of the HaMLet documentation (Postscript or PDF)
- Old version 1.3.1 (gzipped tar or zip)

Contact

For questions, comments and bug reports please contact the author at

- rossberg@mpi-sws.org

Feedback is always welcome.

Successor ML

There also is a special "HaMLet S" that incorporates proposals for Successor ML (sML). It represents a testbed and sort of a personal vision of where sML might ~~go~~ have gone. Its most interesting features are:

- Extensible records.
- More expressive pattern matching.
- Views.
- Higher-order modules and nested signatures.
- Local and first-class modules.
- Miscellaneous fixes to known issues with SML.

Downloads:

- Overview of all extensions — for details see Appendix B of the documentation, and the Successor ML Wiki.
- A few program examples demonstrating some of the extensions.
- Sources (gzipped tar or zip)
- GitHub repository branch

See changes for a version history. Note that HaMLet S is still based on HaMLet 1.3.

Other Implementations of Standard ML

SML implementations more suitable as proper development systems are:

- Standard ML of New Jersey
- MLton
- Moscow ML
- Alice ML
- Poly/ML
- ML Kit
- SML.NET
- SML#

HaMLet evolved as a byproduct of the Alice project, and owes much of its existence to the first version of the ML Kit, which took a very similar approach.

References

1. Robin Milner, Mads Tofte, Robert Harper, David MacQueen.
   "The Definition of Standard ML" (Revised).
   The MIT Press, 1997

2. Emden Gansner, John Reppy.
   "The Standard ML Basis Library".
   Cambridge University Press, 2004
   http://www.standardml.org/Basis/
