Abstract

This paper discusses a Domain Specific Language (DSL) that has been developed to enable implementation of concepts of discrete mathematics. A library of data types and functions provides functionality which is frequently required by users. Covering the areas of Mathematical Logic, Set Theory, Functions, Graph Theory, Number Theory, Linear Algebra and Combinatorics, the language’s syntax is close to the actual notation used in the specific fields.

References
- Fowler, M. 2010, "Domain-Specific Languages", Addison-Wesley Professional.
- Taha, W. 2008, "Domain Specific Languages", IEEE International Conference on Computer Engineering and Systems (ICESS).
- Mernik, M., Heering, J., and Sloane, A. M. 2005, "When and How to Develop Domain-Specific Languages", ACM Computing Surveys (CSUR).
- Ghosh, D. 2011, "DSLs in Action", Manning Publications Co.
- Karsai, G., Krahn, H., Pinkernell, C., Rumpe, B., Schindler, M., and Volkel, S. 2009, "Design Guidelines for Domain Specific Languages", Proceedings of DSM 2009.
- Hughes, J., "Why Functional Programming Matters", The Computer Journal, 1989.
- Goldberg, B., "Functional Programming Languages", ACM Computing Surveys, Vol. 28, No. 1, March 1996.
- Hudak, P. 1996. "Building domain-specific embedded languages", ACM Computing Surveys, December 1996.
- http://www.haskell.org/haskellwiki/Introduction. Introduction to Haskell, retrieved on October 22, 2012.
- http://www.haskell.org/ghc/docs/7.0.4/html/users_guide/index.html. The Glorious Glasgow Haskell Compilation System User's Guide, Version 7.0.4

**Index Terms**

Computer Science  
Programming Languages

**Keywords**

Domain-Specific Language  
Glasgow Haskell Compiler  
Haskell  
Preprocessor