

Team 2
David Tan, Naser Sharifi
Comp 471 Programming Languages
Dr Konstantin Läufer
Project 2b, March 3 2014

A BRIEF SUMMARY

Part1

Upon completion of this project we have evaluated between Project 2a and Project 2b, the following quality attributes [2]:

Attributes	Project 2a	Project 2b
Modifiability is the ability of a system to be extended to accomplish additional functionality.	More stringent approach. Will need rewrite again.	The shapeF factory permits such feacture to build on top of the existing system
Integrability is the ability to easily integrate separate systems or components of a system.		The cata is an example that it can be easily integrated to the systems.
Portability is the ability of a system to execute on different hardware and software platforms		Once the shapeF factory is established user can re-import these functions and use them.
Performance is the measure of how well the computer system responds to its inputs. Common measures are response time, resource utilization, and throughput.		With respect to this, the codes have been shorten and certainly enhanced its performance.
Reliability is the ability of the system to sustain operations. A common measure is mean time between failures. Ease of creation is the difficulty of constructing the system. This is often measured in labor hours.		Indeed, it took lesser amount of line of codes to once the shapeF factory is established.
Reusability is the reuse of existing code in a current development. This is only one of a variety of types of reuse, but it is the one most connected with the architecture of the system under development.		Again codes can be re-used for the different roles.
Highlight attribute : Scalability is		

the ability of a system to support modifications that dramatically increase the size of the system		
--	--	--

Part2

Catamorphism (“cata”) provides a simple route to designing fold-like functions on other algebraic data structures, like various sorts of trees. It takes a simple algebraic function and creates a recursive evaluator for a nested data structure (the fix point of the functor in question). This is a generalization of list folding to arbitrary recursive data structures. As shown in Figure 1.1 [catamorphism](#) [1], cata is a generalized foldRight function.

With regards to our project, as shown in the test size and depth, cata can be easily implemented and provide a suitable test bed.

(TODO)Will it work for implementing boundingBox? have not tested.

Conclusion

As shown from the above table, for a simple project, the method used in Project 2a is ample in functionality. However, with regards to a extensive architected structure, Project 2b has the attributes to go on full scale. The example of cata fully demonstrates the robustness of Project 2b

References

- [1] Bartosz Milewski, “Understanding F-Algebras”
- [2] Rick Kazman, University Of Waterloo, Len Bass, Software Engineering Institute, “Toward Deriving Software Architectures From Quality Attributes”