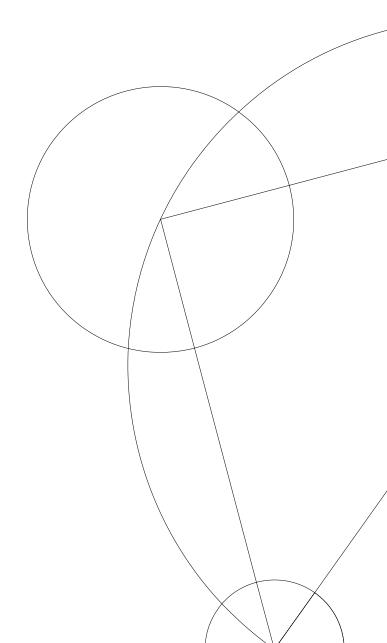# Project Outside the Course Scope

Closure Plots for Basic Arithmetics

Amr El Sayed
KuId - vwj159

August 17, 2016

Closure Plots for Basic Arithmetics

DIKU - Project Outside the Course Scope

Students Names	:	Amr El Sayed - VWJ159
University	:	University of Copenhagen
Institution	:	Department of Computer Science (DIKU)
General Supervisor	:	Michael Kirkedal Thmosen
Practical Supervisor	:	Oleksandr Shturmov
Period	:	Block 5
Year	:	2016
Pages	:	
Github	:	https://github.com/Amr116/ClosurePlots

Certificate

This is to certify that the work contained in the thesis entitled "Closure Plots for Basic Arithmetics" by Amr El Sayed has been carried out under our supervision and that this work has not been submitted elsewhere.

General Supervisor
Michael Kirkedal Thmosen
m.kirkedal@di.ku.dk
Department of Computer Science (DIKU)
University of Copenhagen

Practical Supervisor
Oleksandr Shturmov
oleks@di.ku.dk
Department of Computer Science (DIKU)
University of Copenhagen

1   Abstract

However, simulations with rounded arithmetic are simply guesses and guidance, not proof of anything.

The floating point has computation lacks mathematical rigor, and the simulations with rounded arithmetic are simply guesses and guidance, not proof of anything, and since a rounded number is by definition the substitution of an incorrect number for the correct one. Therefore The purpose of " Closure Plots for Basic Arithmetics" is to create visualization tool for illustrate basic arithmetics simulation to floating point during, and then highlight the bits lacks to those elementary operations.

Resume

Contents

2   Introduction

This paper covers the development process of Closure Plots for Basic Arithmetics on those aspects of floating-point that have a direct impact on designers of computer systems, and is intended for readers who have studied Computer Science or similar, or are simply interested in the field of IT. Readers are assumed to have basic programming knowledge and some familiarity with web development.

The purpose of this project is to create a visualization tool for illustrating the behaviour of basic arithmetic operations (e.g., Addition, Subtraction, Multiplication, Division) in floating-point arithmetic with different precision (e.g., Half, Single, Double, Quadruple).

Floating-point arithmetic is considered an esoteric subject by many people. This is rather surprising because floating-point is ubiquitous in computer systems. Almost every language has a floating-point datatype, computers from PCs to supercomputers have floating-point accelerators, most compilers will be called upon to compile floating-point algorithms from time to time, and virtually every operating system must respond to floating-point exceptions such as Overflow, Undeflow, Infinity, and Inexpressible.

In ny goal-oriented activity requiring, benefiting from, or creating a mathematical sequence of steps known as an algorithm(e.g. designing, developing and building hardware and software systems, processing, structuring, and managing various kinds of information, doing scientific research on and with computers, making computer systems behave intelligently, and creating and using communications and entertainment media), floating point is the formulaic representation that approximates a **real number** so as to support a trade-off between range and precision. A number is, in general, represented approximately to a fixed number of significant digits (the significand) and scaled using an exponent in some fixed base, the base for the scaling is normally two, ten, or sixteen. A number that can be represented exactly is of the following form:

3   Analysis & Design

3.1   Analysis

3.2   Design

4   Inspiration

5   Implementation

6   Future Development

7   Work progress

8   Conclusion

Footnotes