De La Salle University

Posteriori Analysis of Algorithms Through Derivations of Growth Rate Based on Frequency Counts

User's Manual

A Thesis
presented to
the Faculty of the College of Computer Studies
De La Salle University

in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

by

Guzman, John Paul

Teresita Limoanco
Adviser

TABLE OF CONTENTS

1.0. INTRODUCTION

This program an implementation of an automated method of algorithm analysis developed in the thesis. It takes in an algorithm written in Python and generates approximations of its asymptotic behavior in Big Theta or in asymptotic equivalence.

1.1. SYSTEM REQUIRMENTS

- 1 gigahertz (GHz) or faster 32-bit (x86) or 64-bit (x64) processor
- 1 gigabyte (GB) RAM (32-bit) or 2 GB RAM (64-bit)
- 16 GB available hard disk space (32-bit) or 20 GB (64-bit)
- DirectX 9 graphics device with WDDM 1.0 or higher driver
- Operating System: Windows XP/7/8

1.2. INSTALLATION

1.2.1. Automated Installation

1. Navigate to the "Dependencies" folder.
2. Execute the "FULL_INSTALLATION.bat" batch file.
3. Follow the notes being displayed by the batch file.
4. If there are any problems with the automated installation, refer to the Manual Installation.

1.2. 2. Manual Installation

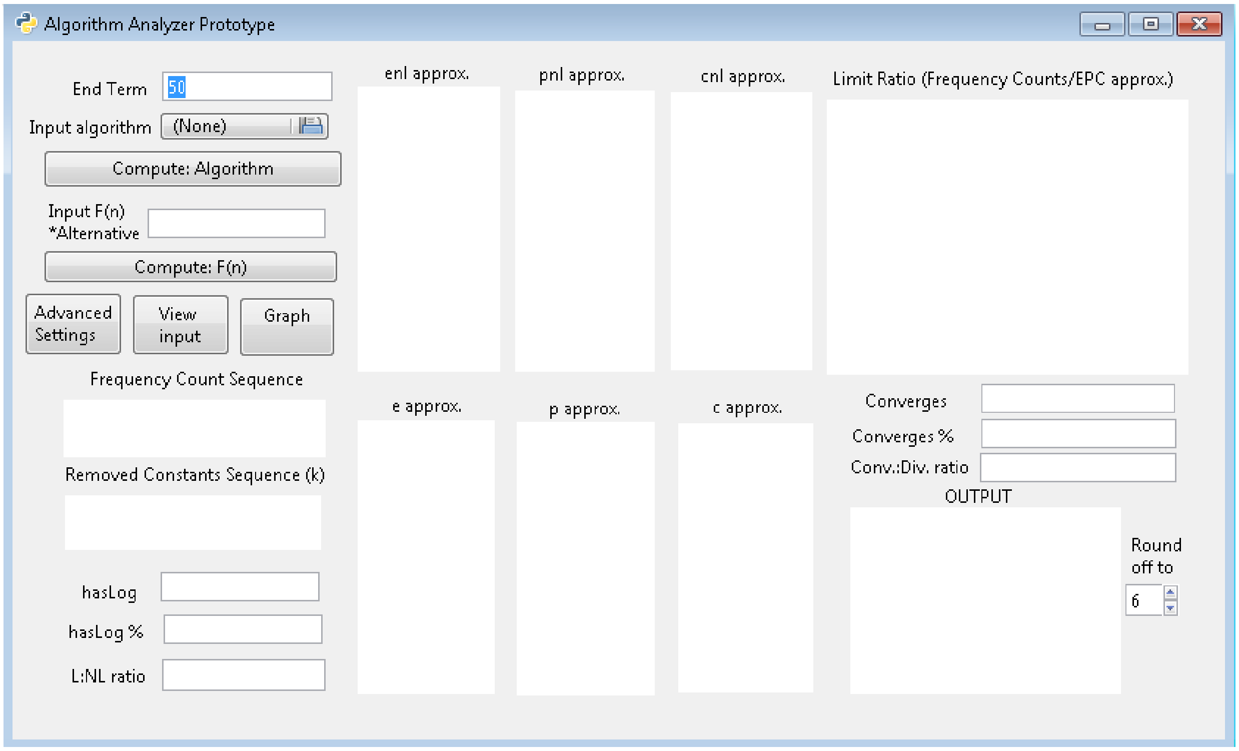
1. Navigate to the "Dependencies" folder.
2. Run and install "WinPython-32bit-2.7.10.3.exe".
3. Navigate to the installation folder of WinPython and open the "scripts" folder.
4. Execute the "register_python_for_all.bat" batch file. This will allow the PyGTK installer to detect Python.
5. Return to the "Dependencies" folder then, run and install "pygtk-all-in-one-2.24.2.win32-py2.7.msi".
6. Return to the WinPython's folder, open the "python-2.7.10" folder, locate the "python.exe" file and copy its full file path.
7. Open the command prompt, and change directory (cd) to the "Dependencies folder"
   Example: cd "C: \Dependencies"
8. Use the "editbin.exe" on "python.exe" and change its stack size usage to a desired value.
9. Execute the command: editbin.exe /stack:<stack size> "<full path of python.exe>"
   Example: editbin.exe /stack:50000000 "C:\Python 2.7.10\WinPython-32bit-2.7.10.3\python-2.7.10\python.exe"
10. Navigate to the "Executable" folder and open of the "path.txt" file. If it does not exist, create a new text file with the same name.
11. Edit "path.txt" to contain the path of the folder where "python.exe" is stored.
    Example: C:\Python 2.7.10\WinPython-32bit-2.7.10.3\python-2.7.10

1.3. CONVENTION

Variable names are contained within two double quotes to avoid confusion. The angled brackets along with their contents are to be replaced by the user.

2.0. GETTING STARTED

The prototype is run by executing "RUN_PROTOTYPE.bat" in the "Executables" folder.
It will open a console and display the GUI.



End Term 50

Input algorithm (None)

Round off to 6

2.1. BASIC FUNCTIONALITIES

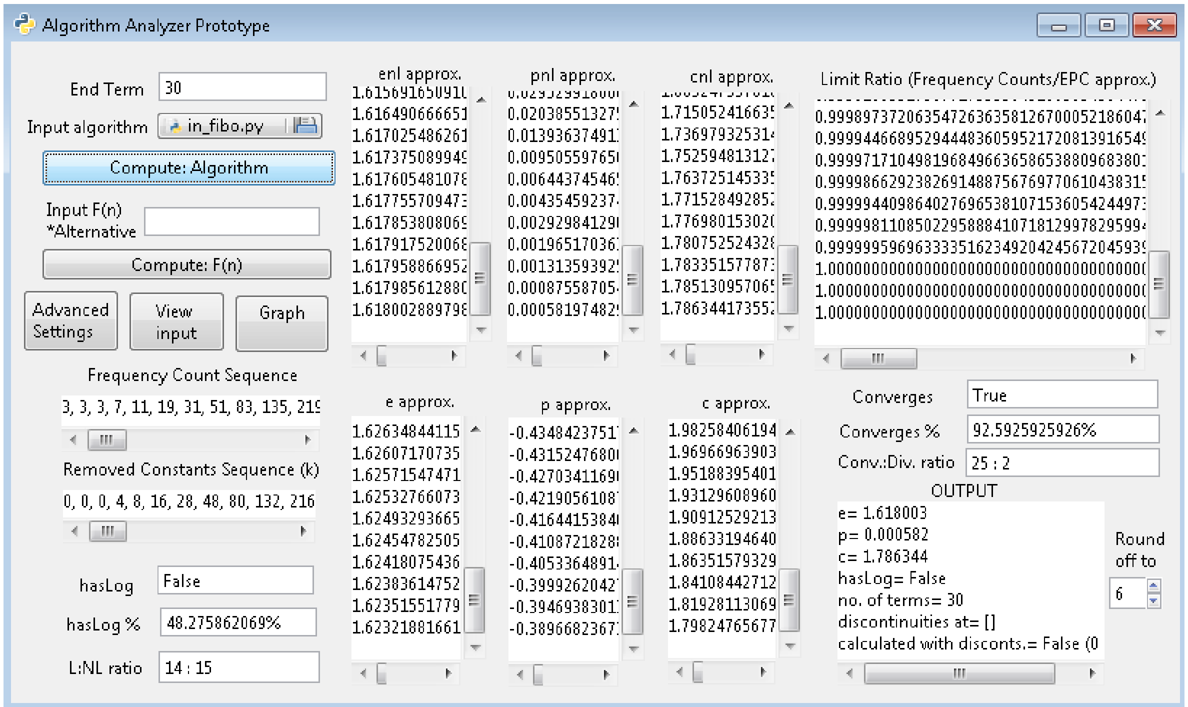
All but one of the interactive GUI are located at the upper left part of the window. There is one GUI element at the lower right part which rounds off the values in the "OUTPUT" TextView. The rest of the GUI elements used to display the results.

"End Term" text field is where the user specifies the number of terms to be generated by the experiments. The "Input algorithm" button opens a window which allows the user to navigate through the machine's directories and choose a file. These first two elements give sufficient control for general purpose algorithm analysis. See sample run below.

The input algorithm file must follow these constraints:

- It must compile with Python 2.7.
- The main function to be called in the algorithm should be defined as f.
- The algorithm must only have 1 variable as parameter.

- The algorithm may call other algorithms provided that they are defined within the same file.
- The "elif" keyword and multiline commenting should not be used due to a difficulty in parsing. Instead, use a nested "if" inside an "else" and use " # " for commenting.



End Term: 30

Input algorithm: in_fibo.py

Input F(n) *Alternative:

Frequency Count Sequence: 3, 3, 3, 7, 11, 19, 31, 51, 83, 135, 21

Removed Constants Sequence (k): 0, 0, 0, 4, 8, 16, 28, 48, 80, 132, 216

hasLog: False

hasLog %: 48.275862069%

L:NL ratio: 14 : 15

```
enl approx.        pnl approx.        cnl approx.
1.616490666651     0.02038551327      1.71505241663
1.617025486261     0.01393637491      1.73697932531
1.61737508994      0.00950559765      1.75259481312
1.617605481078     0.00644374546      1.76372514533
1.61775570947      0.00435459237      1.77152849285
1.61785380806      0.00292984129      1.77698015302
1.61791752006      0.00196517036      1.78075252432
1.61795886695      0.00131359392      1.78335157787
1.61798561288      0.00087558705      1.78513095706
1.61800288979      0.00058197482      1.78634417355
```

```
Limit Ratio (Frequency Counts/EPC approx.)
0.999897372063547263635812670005218604
0.999944668952944483605952172081391654
0.999971710498196849663658653880968380
0.999986629238269148875676977061043831
0.999994409864027696538107153605424497
0.999998110850229588841071812997829599
0.999999596963333516234920424567204593
1.000000000000000000000000000000000000
1.000000000000000000000000000000000000
1.000000000000000000000000000000000000
```

```
e approx.          p approx.          c approx.
1.62634844115      -0.4348423751      1.98258406194
1.62607170735      -0.4315247680      1.96966963903
1.62571547471      -0.4270341169      1.95188395401
1.62532766073      -0.4219056108      1.93129608960
1.62493293665      -0.4164415384      1.90912529213
1.62454782505      -0.4108721828      1.88633194640
1.62418075436      -0.4053364891      1.86351579329
1.62383614752      -0.3999262042      1.84108442712
1.62351551779      -0.3946938301      1.81928113069
1.62321881661      -0.3896682367      1.79824765677
```

Converges: True

Converges %: 92.5925925926%

Conv.:Div. ratio: 25 : 2

OUTPUT

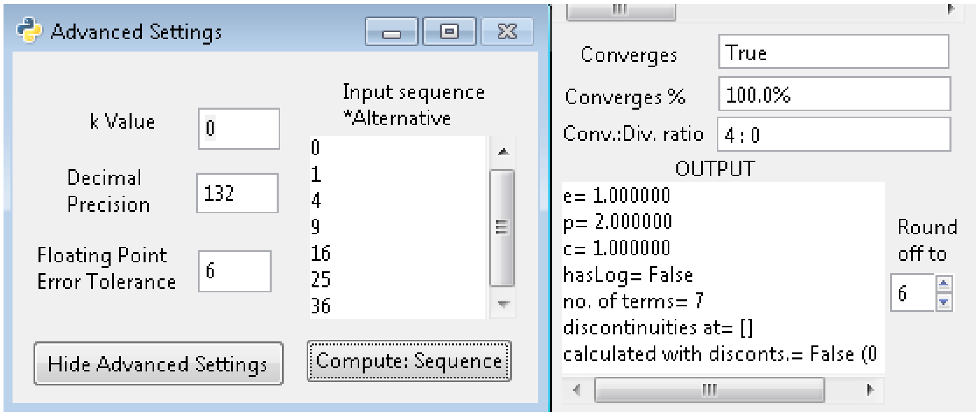
```
e= 1.618003
p= 0.000582
c= 1.786344
hasLog= False
no. of terms= 30
discontinuities at= []
calculated with disconts.= False (0
```

Round off to: 6

The other controls are used for more abstract purposes. The "Compute: F(n)" is an option that allows the user to input a function like "2*n*log(n)+20" to generate the data instead of using an algorithm. However, this feature has three constraints. First, "n" must be the only variable present in the string. Second, +,-,*,/ (arithmetic operators) are the only special characters allowed. Third, only specific functions are allowed and these are log(), exp(), and pow(). Other functions such as sqrt() is allowed, but it is not as precise as pow(n, 0.5).

2.2. ADVANCED FUNCTIONALITIES

The "Advanced Settings" button opens up a window containing more abstract options. The "Compute: Sequence" option allows the user to input a numeric sequence that replaces generated data. The number as line 1 is stored in the 0th index of the generated data array, line 2 is stored at the 1st index, line 3 is stored at the 2nd index, and so on.



k Value: 0

Decimal Precision: 132

Floating Point Error Tolerance: 6

Input sequence *Alternative:

```
0
1
4
9
16
25
36
```

Converges: True

Converges %: 100.0%

Conv.:Div. ratio: 4 : 0

```
e= 1.000000
p= 2.000000
c= 1.000000
hasLog= False
no. of terms= 7
discontinuities at= []
calculated with disconts.= False (0
```

Round off to: 6

The k Value is used in generating the "Removed Constants Sequence(k)". If a(x) is the "Frequency Count Sequence", then the removed constants sequence is equal to a(x)-a(0)+k. The reasoning behind this can be seen in Section 5.4 of the thesis document. Lastly, "Decimal Precision" is the number of significant digits to be calculated while the "Floating Point Error Tolerance" is the number of digits at the end of the number to be rounded off. This will resolve issues where the inaccuracy of the floating point representation would interfere with heuristics.

3.0. MESSAGES

- ERR_FILE = "Unable to open file error. No file has been selected. "
  - Displays when no file has been selected or the file has been moved when attempting to compute using an algorithm.
- ERR_COMPILE = "Unable to continue due to compilation error. "
  - Displays when there are errors in the input algorithms like syntax errors that prevent the input algorithm to be compiled.
- ERR_ALGORITHM = "Unable to run algorithm properly. "
  - Displays when runtime errors occur while executing the augmented algorithm.
- ERR_FUNCTION = "Unable to evaluate function value error. "
  - Displays when the input function string could not be evaluated.
- ERR_NUMBER = "Unable to convert sequence terms to decimal error. "
  - Displays when the input numeric sequence contains a string that is not a number.
- ERR_MATH_REPLACE = "Used a place holder value due to math operation error. "
  - Displays when math operator errors occur (e.g. when evaluating log(0)) and 0 is used place of the evaluated value.
- ERR_ROUND_OFF = "Prevented rounding off. "
  - Displays when attempting to round off a number where all the significant digits are not fractional.
- ERR_MATH_IGNORE = "Approximation ignored due to math operation error. "
  - Displays when math operator errors occur.
- ERR_GRAPH = "Unable to graph plot points error. No plot points have been calculated."
  - Displays when attempting to draw a graph prior to doing any computations.