Advanced Information Data Analysis

lecture 5

Hiroyoshi Morita

May 13, 2019

5 Programs: Python and Jupyter

5.1 Installation of Programs:

Graphviz is a software library working as a kind of interface between user-programers and DOT language that is aimed to draw directed/undirected graphs. Refer to
http://graphviz.org/gallery/

5.1.1 Installations

- anaconda
- jupyter
- graphviz

anaconda   To install graphviz with Python, it is recommendable to install `anaconda` and use `conda` to install it as explained.
https://www.anaconda.com/distribution/
And the following Quita article would be helpful for anaconda installation.
https://qiita.com/t2y/items/2a3eb58103e85d8064b6 (in Japanese)
When you finish the install, please type the following commands on a terminal.

```
conda config --get channels
```

If you got 'conda-forge' in the output, then OK. Otherwise, you further type the followings.

```
conda config --append channels conda-forge
```

Then, you are ready to move on further.

jupyter   install jupyter lab with anaconda
https://jupyter.readthedocs.io/en/latest/install.html (in English)
https://datachemeng.com/anaconda_jupyternotebook_install/ (in Japanse)

graphviz-python   https://anaconda.org/conda-forge/python-graphviz

References To understand the details of graphviz, it would be helpful to have a look at commands of `graphviz`.

  http://www.graphviz.org/doc/info/attrs.html
  And the `dot` User's Manual is referred to
  https://www.graphviz.org/pdf/dotguide.pdf
  (already downloaded at ownCloud)

5.2 Demonstration of MFW.py

```
[1]: import MFW as m
```

```
[15]: T = "0101001101100$"
      st = m.SuffixTrie(T)
      st.view()
```

```
[ ]: adF = m.buildADT_FH(st)
     adF.view()
```

```
[ ]: adS = m.buildADT_SH(st, adF)
     adS.view()
```

Report 03 due: 13:00 on Monday, May 27, 2019

1. Install python-graphviz into your machine.
2. run the following program. And send three pictures obtained as report.

  Note Before running python, you must download MFW.py from ownCloud to the same directory as you run python.

```
[ ]: T = "010110"
     st = m.SuffixTrie(T)
     st.view()
     # 2.1 Save the picture named 'st-YourName'

     adF = m.buildADT_FH(st)
     adF.view()
     # 2.2 Save the picture named 'adF-YourName'

     adS = m.buildADT_SH(st, adF)
     adS.view()
     # 2.2 Save the picutre name 'adS-YourName'
```

3. By getting lots of pictures for various binary strings, list up your findings about MF-links and MFWs from them.