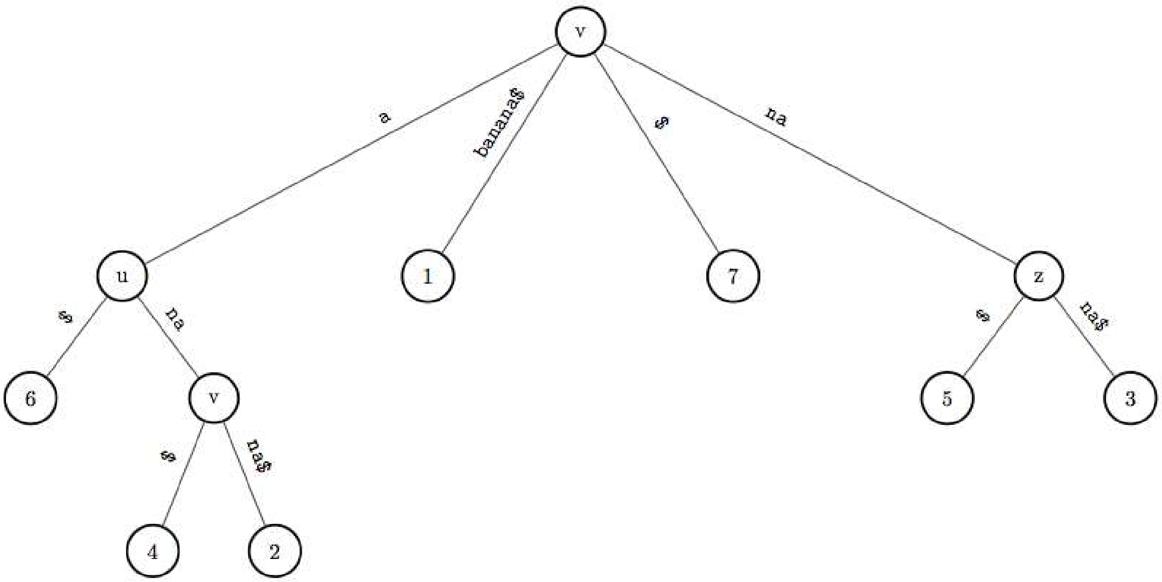
TranslateToLatexTree

This is a python script that compiles a simple text representation of a tree in a pretty latex figure.

Basic usage

Let's say you want to draw this tree:



Then you should create a txt file where each line describes a level of the tree, starting from the root:

```
line 1: v
```

In the second line you want to describe 4 nodes. Each node description is separated by the others with the symbol '|' and contains the label of the incoming edge (if any) and the label of the node, encoded with this syntax:

```
[label of the incoming edge] ; [label of the node]
```

Then the second will be:

```
line 2: a;u| banana$;1 | $;7 | na;z
```

In the third line, you have nodes with different parents: the description you will put at the beginning of the line will refer to the leftmost node at the upper level (in this case `u`). You can change the parent node using the symbol '@': the parent will become the next parent node on the right (in this case `1`). If the next parent node does not have child nodes, leave the description empty.

Third line will be:

```
line 3: $;6 |na;v @  @  @ $;5 | na$;3
```

Same for the fourth level/line, you have to specify the parent for each description, considering the node that you
created in the previous level: the previous level contains four nodes and only the second has child, then:

```
line 4: @ $;4| na$;2 @ @
```

Special Syntax for Discrete Mathematical Formula

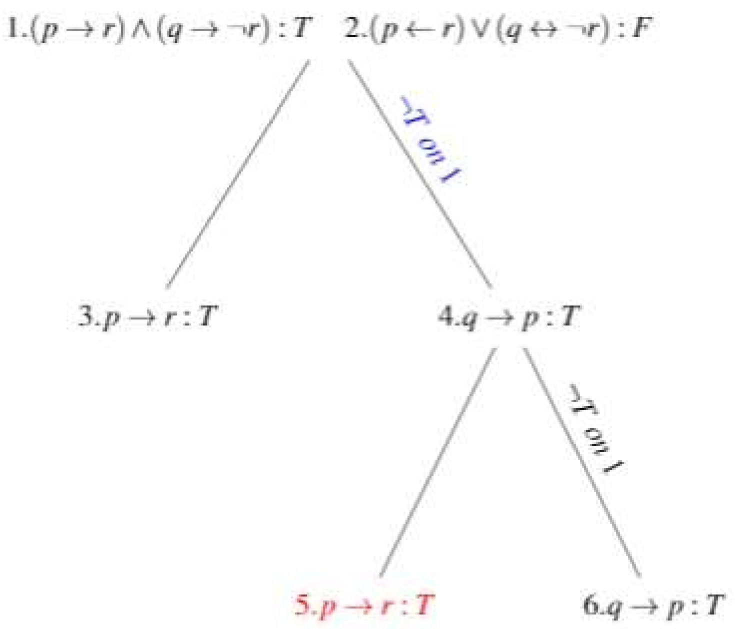
I want to generate some Formula of Discrete Mathematic. There are a lot of special sign here. Hence I defined some Syntax especially:

```
        dic = {"toRight": "\\rightarrow",
               "toLeft": "\leftarrow",
               "double": "\leftrightarrow",
               "or": "\vee ",
               "and": "\wedge ",
               "not": "\\neg",
               "sblue": "\\textcolor{blue}{", "eblue": "}",
               "sred": "\\textcolor{red}{", "ered": "}", "space": "\ ",
               "largeSpace": "\quad", "$$": " "}
```

An example:

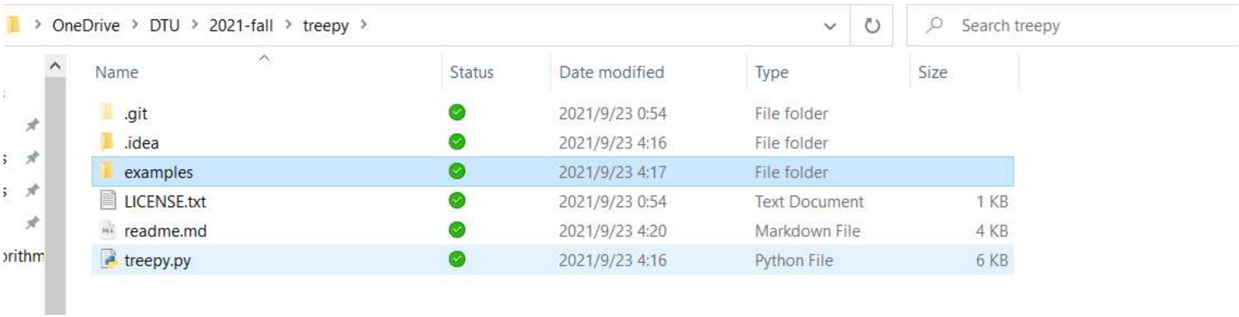
```
1.(p toRight r) and (q toRight not r):T largeSpace 2.(p toLeft r) or (q double not r):F
;3.p toRight r:T | sblue not  T space on space 1 eblue;4.q toRight p:T
@ ;sred 5. p toRight r :T ered| not  T space on space 1;6.q toRight p:T

```

The generated graph:


1.$(p \to r) \land (q \to \neg r) : T$
2.$(p \leftarrow r) \lor (q \leftrightarrow \neg r) : F$
$\neg T$ on 1
3.$p \to r : T$
4.$q \to p : T$
$\neg T$ on 1
5.$p \to r : T$
6.$q \to p : T$

How to run this script

My folder looks like this:

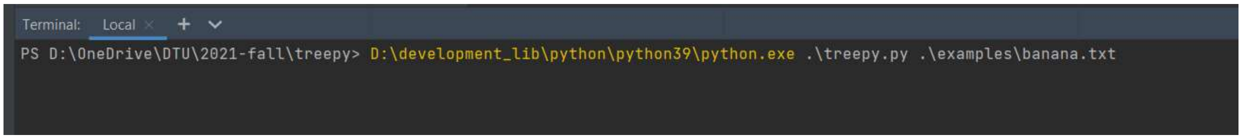


Name	Status	Date modified	Type	Size
.git	✓	2021/9/23 0:54	File folder	
.idea	✓	2021/9/23 4:16	File folder	
examples	✓	2021/9/23 4:17	File folder	
LICENSE.txt	✓	2021/9/23 0:54	Text Document	1 KB
readme.md	✓	2021/9/23 4:20	Markdown File	4 KB
treepy.py	✓	2021/9/23 4:16	Python File	6 KB

In folder `example` , there is a `txt` file which is used to describe a tree.

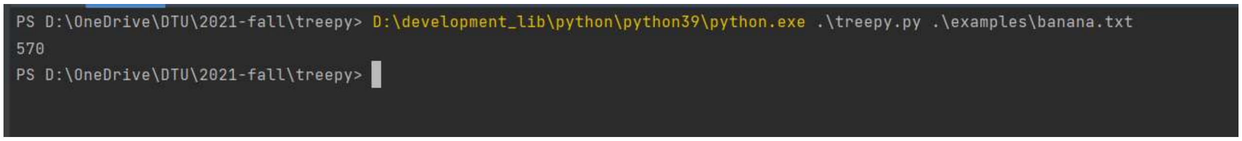
So I would try the command following:

```
{Python_path} .\treepy.py {tree_description_file}
```



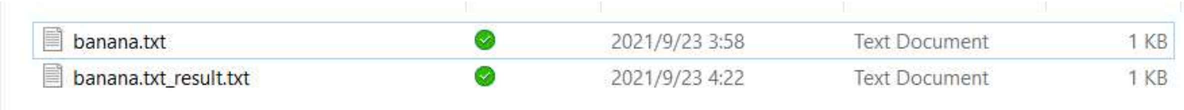
```
PS D:\OneDrive\DTU\2021-fall\treepy> D:\development_lib\python\python39\python.exe .\treepy.py .\examples\banana.txt
```

Run it, and you see:



```
PS D:\OneDrive\DTU\2021-fall\treepy> D:\development_lib\python\python39\python.exe .\treepy.py .\examples\banana.txt
570
PS D:\OneDrive\DTU\2021-fall\treepy> 
```

A `xxx_result.txt` file has been generated:



banana.txt	✓	2021/9/23 3:58	Text Document	1 KB
banana.txt_result.txt	✓	2021/9/23 4:22	Text Document	1 KB

The code in `xxx_result.txt` looks like:

```
\node[punkt] {$1.(p \rightarrow r)  \wedge  (q \rightarrow \neg r):T \quad 2.(p \leftarrow r)  \vee (q \leftrightarrow \neg r):F$}
    child {node[punkt] {$3.p \rightarrow r:T$}
edge from parent
 node[kant,above,pos=.4]{ }}
    child {node[punkt] {$4.q \rightarrow p:T$}
    child {node[punkt] {$\textcolor{red}{ 5. p \rightarrow r :T }$}
edge from parent
 node[kant,above,pos=.4]{ }}
    child {node[punkt] {$6.q \rightarrow p:T$}
edge from parent
 node[kant,above,pos=.4]{$\neg  T \  on \  1$}}
edge from parent
 node[kant,above,pos=.4]{$\textcolor{blue}{ \neg  T \  on \  1 }$}};
```

Copy it and paste it to your overleaf in the following way:

```
%start from here
\begin{tikzpicture}[
grow=down,
level 1/.style={sibling distance=5cm,level distance=4cm},
level 2/.style={sibling distance=4cm, level distance=4cm},
level 3/.style={sibling distance=4cm, level distance=4cm},
kant/.style={ text centered, sloped},
every node/.style={text ragged, inner sep=2mm},
punkt/.style={ shade, top color=white,
bottom color=white, draw=white, very thick }
]

%...insert your result here...%

\end{tikzpicture}

%end here
```

The tree is drawn using the tikz package. You will have to import the package *tikz* . As you can see the preamble of tikzpicture allows you to personalize your tree .

> @ Author : WenjieDTU
>
> Ref: https://github.com/diegoceccarelli/treepy