

Lecture 29

Lab 8: Replace method

Assume every L-system Rule has a unique find string

Self. rules = [['F', 'FF'] , ['F', '+F']]

rule 1

find replace

NOT allowed!!

Example:

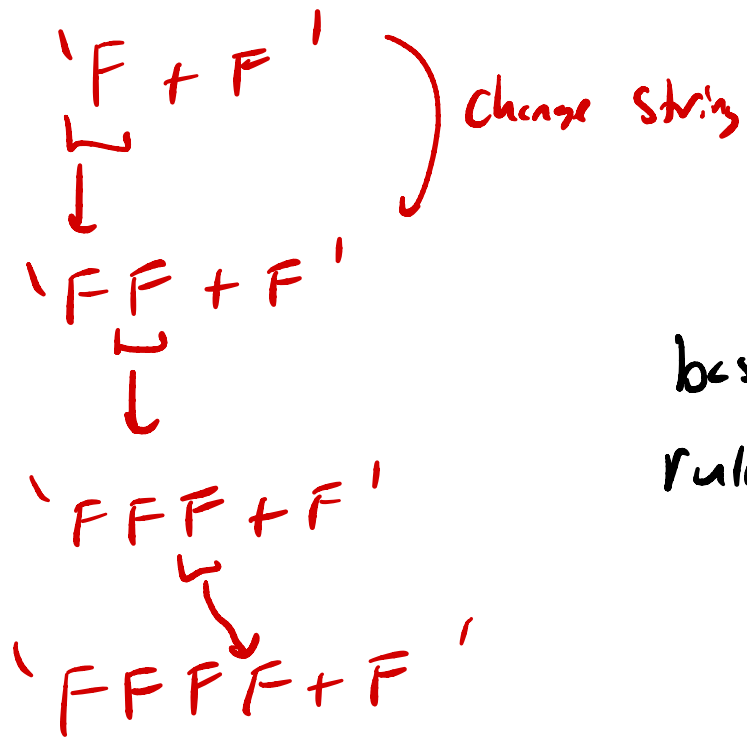
Base String: 'F + F'

Rules: [['F', 'FF']]

Approach: Walk char-by-char left $\rightarrow$ right,
Apply rules one by one.

'F + F'
└─┬─┘
└─┬─┘
'FF + FF'

We Don't Want:



base string: 'F++F++F'

rules: $\left[\left[_, 'F+F' \right], \left['F', 'FF' \right] \right]$

Rule 1

Rule 2

eg. 'F++F++F'

```
def replace(self, currString):  
    newString = ''
```

```
    for char in currString:  
        # looping thru currString char-by-char  
        found = False
```

```
        for rule in self.rules: # rule: sublist  
            # looping thru rules  
            # ['-', 'F+F']
```

```
            if char == rule[0]:
```

```
                # we have a match!
```

```
                newString += rule[1]
```

```
                found = True
```

```
                break
```

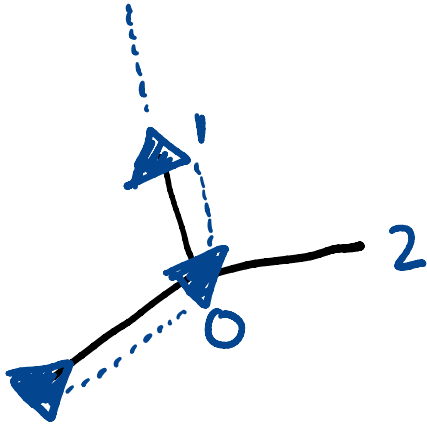
```
        # no rule matches
```

```
        if found is not True:
```

```
            newString += char
```

```
    return newString
```

Drawing trees with L-systems: branching



Why not use turtle goto method? $1 \rightarrow O$.

problem: Where to go — what is the x, y ?
— what heading do I use?

Solution: Why don't we save our state before embarking down one branch?

$\Rightarrow$ Save position (x, y) and heading, at O ,

$\Rightarrow$ Draw down branch 1,

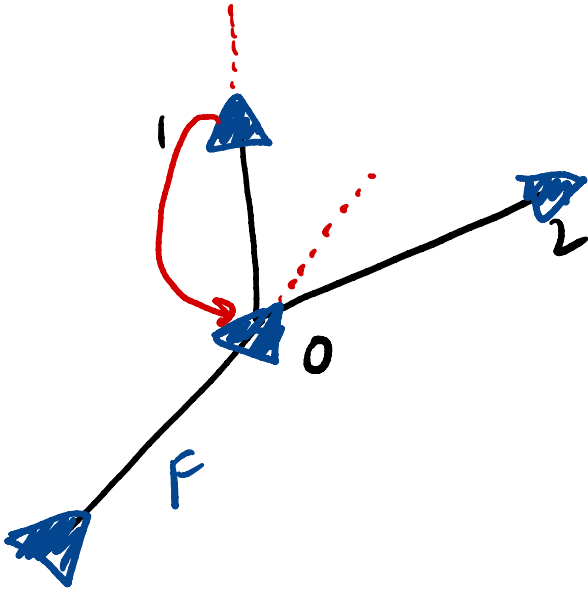
$\Rightarrow$ restore state before going down ①

[restore previous position when at O
and heading]

$\Rightarrow$ draw branch 2.

How to implement Save/restore?

- [Save current state (heading + position)
-] Restore previous saved state (heading + position)

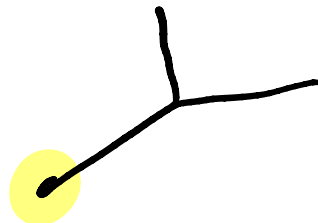


System string:

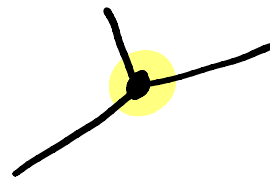
$F [+ F] - F$
get to 0 branch 1 branch 2

How to implement '[' ']' Symbols to Save State?

Stack = []



Stack.append(heading)
Stack.append(position)



Stack = [heading, position]

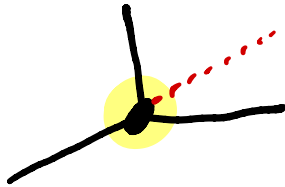
```
prevpos = Stack.pop()
```

Stack = [heading]

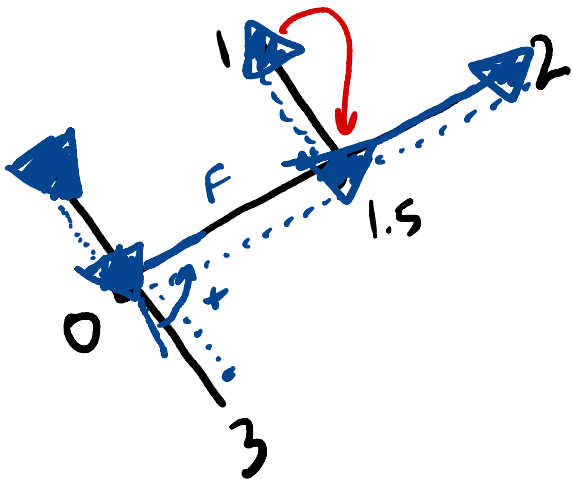
```
prevHeadings = Stack.pop()
```

Stack = []

$\Rightarrow$ go prev pos, set heading to prevHeading



Practice: What L-system string will draw this tree branch?



$$F[+F[+F]F]F$$

Annotations:

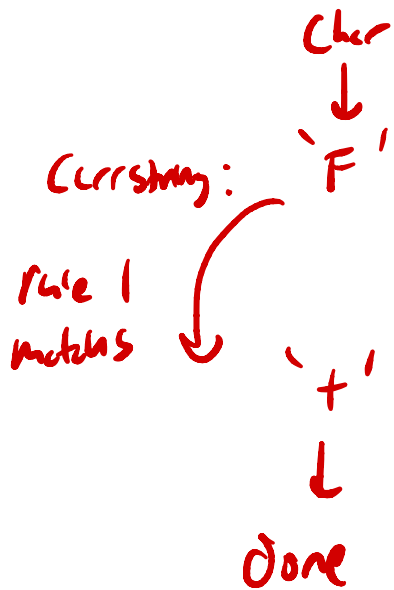
- go back to 1.5 (pointing to the first '[')
- draw 2 (pointing to the second '[')
- draw 1 (under the innermost '[')
- draw 3 (under the outermost '[')
- Save state @ 0 (under the first '[')

Why not String replace method?

base = 'F'

self.rules = [['F', '+'], ['+', '-']]

Loop char-by-char



String replace

