

# 数据流和切片测试 工具的设计与实现

中期检查报告

阮中秋 19241030

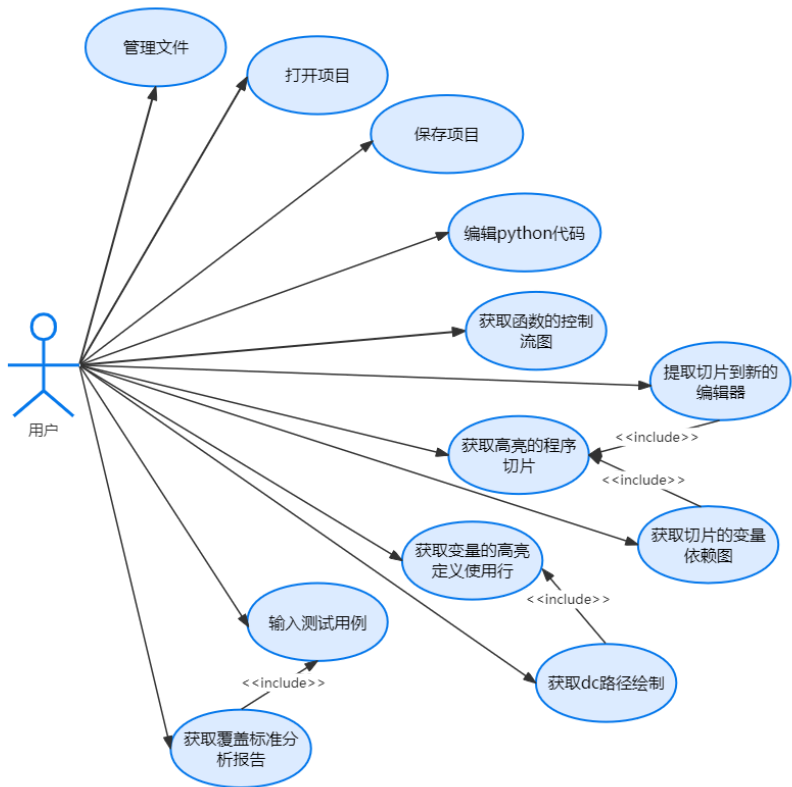
# 研究目标

基于数据流和切片测试思想，设计开发一个工具，能实现python代码高亮、可视化以及覆盖标准分析等功能，帮助学生理解数据流和切片测试的知识

- 任给一段代码，选中一个变量，可以按照切片测试思想，给出对应**代码片段高亮**
- 对如上高亮显示代码用图的形式绘制出方法和变量的包含关系，变量和变量的支持关系，等其他**可视化关系**
- 对数据量测试，给定一个变量，分别显示方法中变量的**定义和使用代码行**，并给出dc**路径的绘制**
- 任给一个测试用例，判断是否满足数据流测试的几种**覆盖标准**

# 功能需求分析

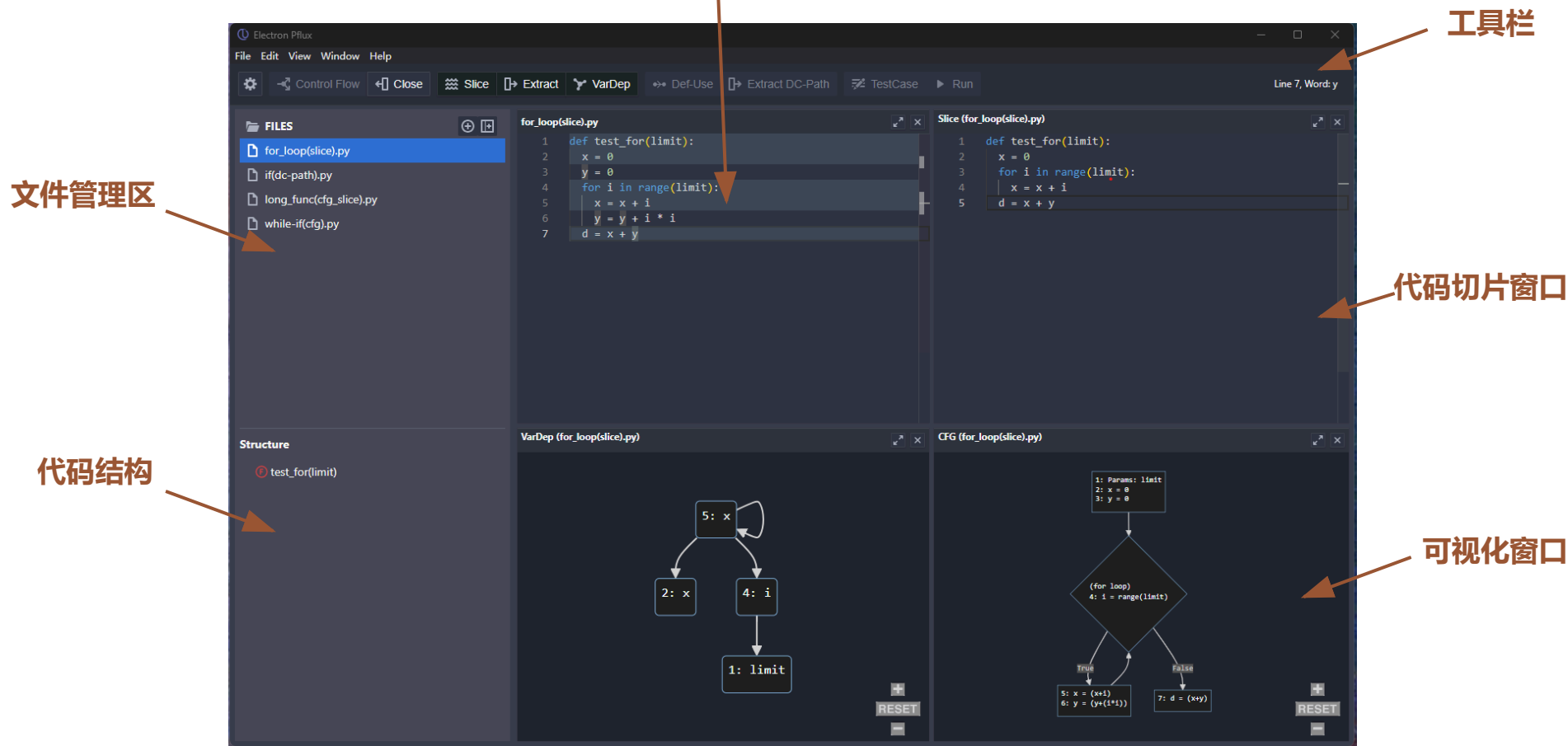
用例图



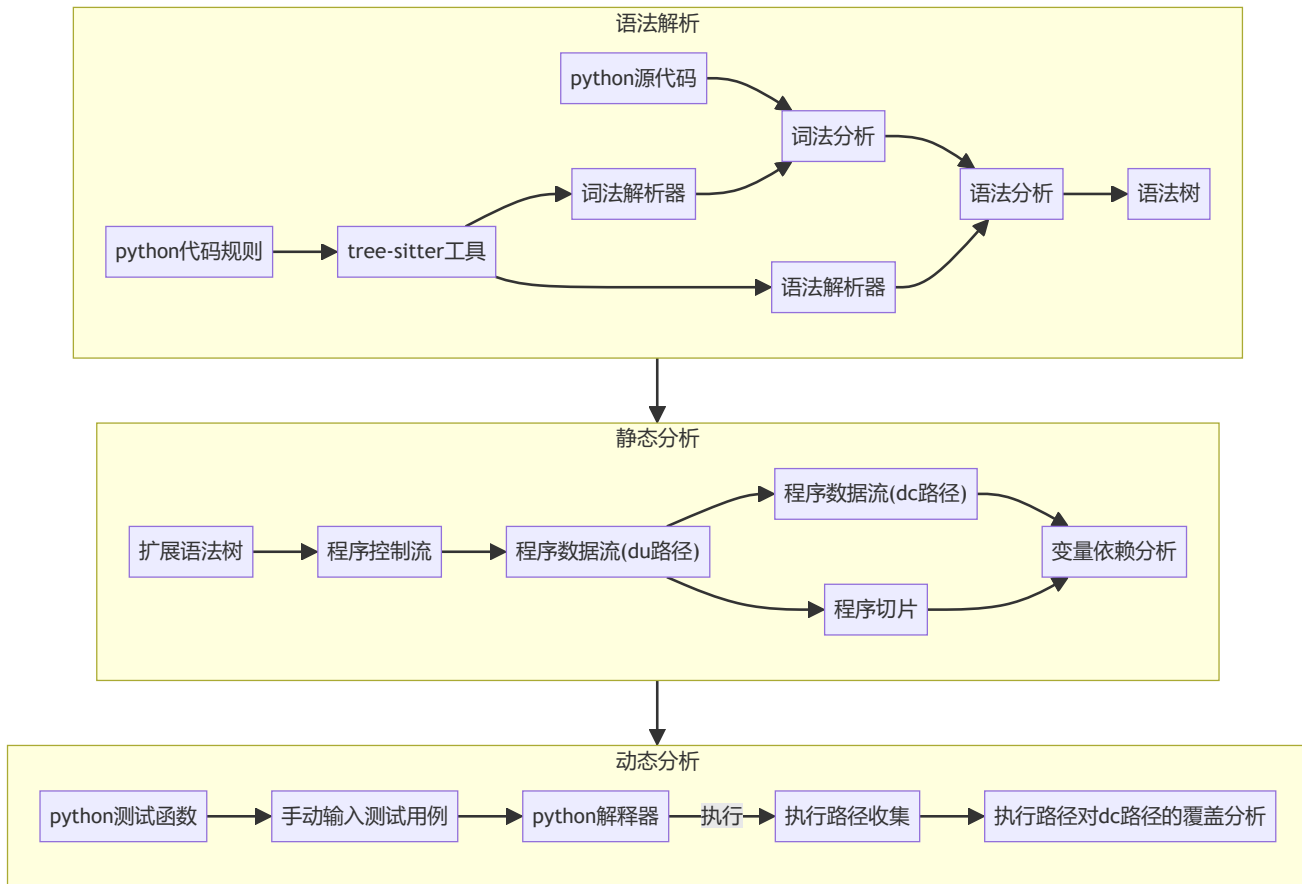
功能模块划分



# UI设计



# 功能实现逻辑



# 扩展语法树

## 赋值语句

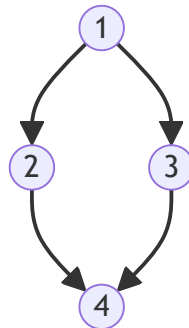
```
{  
  type: 'assign',  
  op: '+',  
  sources: [],  
  targets: [],  
  location: {  
    start_line: 1,  
    end_line: 1,  
    start_column: 1,  
    end_column: 10,  
  },  
  parent: {},  
  next_sibling: {},  
  last_sibling: {},  
}
```

# 程序控制流

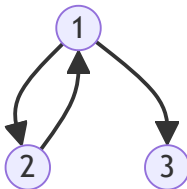
no control



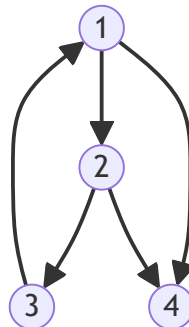
if-else



while(for)

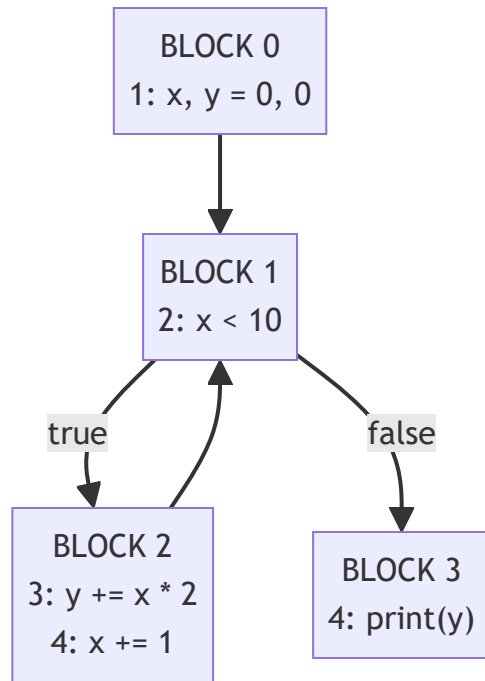


while with break

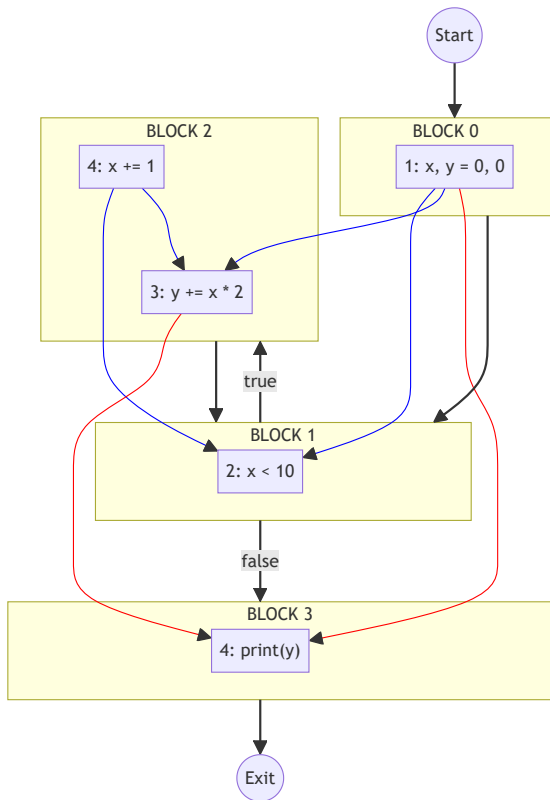


# 例子

```
1: x, y = 0, 0
2: while x < 10:
3:     y += x * 2
4:     x += 1
5: print(y)
```



# 程序数据流



# 程序切片

```
1: sum = 0
2: diff_sum = 0
3: for i in range(min(len(A), len(B))):
4:     sum += A[i] + B[i]
5:     diff_sum += A[i] - B[i]
6: print(sum, diff_sum)
```

## SLICE OF SUM

```
1: sum = 0
3: for i in range(min(len(A), len(B))):
4:     sum += A[i] + B[i]
6: print(sum, diff_sum)
```

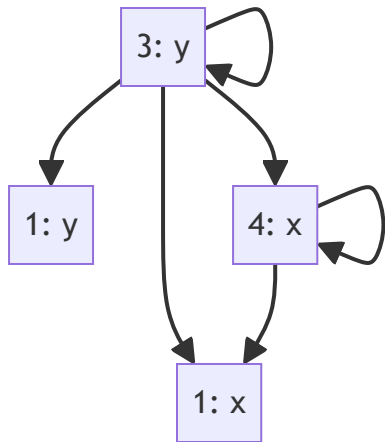
## SLICE OF DIFF\_SUM

```
2: diff_sum = 0
3: for i in range(min(len(A), len(B))):
5:     diff_sum += A[i] - B[i]
6: print(sum, diff_sum)
```



# 变量依赖

```
1: x, y = 0, 0
2: while x < 10:
3:     y += x * 2
4:     x += 1
5: print(y)
```



# 覆盖分析

## 测试函数

```
def func(limit):
    x, y = 0, 0
    while x < limit:
        y += x * 2
        x += 1
    print(y)
```

## 测试用例

limit

0

10

## 执行代码

```
# main.py
def func(limit):
    x, y = 0, 0
    while x < limit:
        y += x * 2
        x += 1
    print(y)
```

```
func(0)
func(10)
```

## 运行

```
> python main.py
```

## 收集执行路径

```
1→2→3→4....
```



Control Flow



Slice



Extract



VarDep



Def-Use



Extract DC-Path



TestCase



Run

Line 1, Word: def

FLUX1



add.py

cfg-test.py

dc-path-test.py

main.py

test.py

Structure

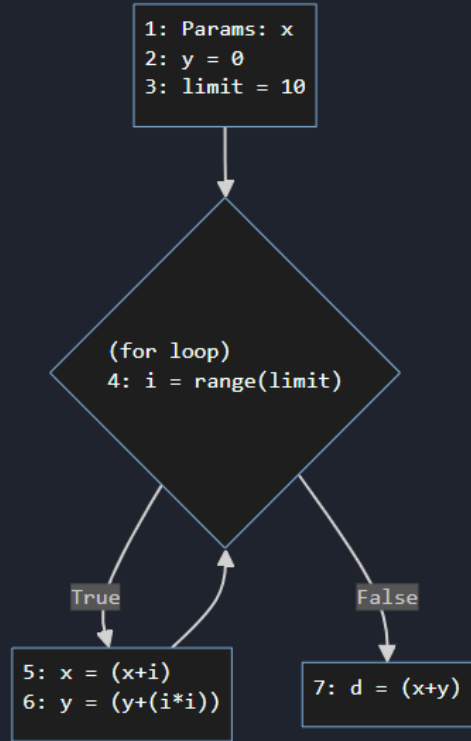
func1(x)

cfg-test.py



```
1 def func1(x):  
2     y = 0  
3     limit = 10  
4     for i in range(limit):  
5         x = x + i  
6         y = y + i * i  
7     d = x + y
```

CFG (cfg-test.py)





Control Flow

Slice

Extract

VarDep

Def-Use

Extract DC-Path

TestCase

Run

Line 7, Word: y

FLUX1



add.py

cfg-test.py

dc-path-test.py

main.py

test.py

cfg-test.py



```
1 def func1(x):  
2     y = 0  
3     limit = 10  
4     for i in range(limit):  
5         x = x + i  
6         y = y + i * i  
7     d = x + y
```

Slice (cfg-test.py)

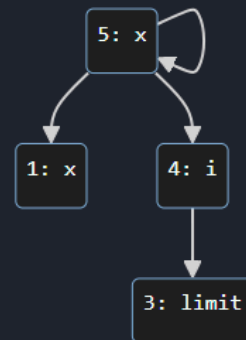


```
1 def func1(x):  
2     limit = 10  
3     for i in range(limit):  
4         x = x + i  
5         d = x + y
```

Structure

func1(x)

VarDep (cfg-test.py)





Control Flow

Slice

Extract

VarDep

Def-Use

Extract DC-Path

TestCase

Run

Line 7, Word: x

FLUX1



add.py

cfg-test.py

dc-path-test.py

main.py

test.py

cfg-test.py

Def

Use

```
1 def func1(x):
2     y = 0
3     limit = 10
4     for i in range(limit):
5         x = x + i
6         y = y + i * i
7     d = x + y
```

DCPath (cfg-test.py)

|   | StartLine | EndLine | UseType |
|---|-----------|---------|---------|
| 1 | 1         | 7       | C-use   |
| 2 | 1         | 5       | C-use   |
| 3 | 5         | 7       | C-use   |
| 4 | 5         | 5       | C-use   |

## Structure

func1(x)



Control Flow



Close



Slice



Extract



VarDep



Def-Use



Extract DC-Path



TestCase



Run

Line 1, Word: test\_coverage

## FILES

for\_loop(slice).py

if(dc-path).py

long\_func(cfg\_slice).py

while-if(cfg).py



if(dc-path).py



```
1 def test_coverage(a, b):
2     if b > 1:
3         a = a + 7
4     if a > 10:
5         b = a + b
6     print(a, b)
```

TestCase (if(dc-path).py)



## Function: test\_coverage(a, b)

|         | a  | b |   |
|---------|----|---|---|
| 1       | 4  | 2 | × |
| 2       | 11 | 1 | × |
| New Row |    |   |   |

Coverage Analysis (if(dc-path).py)



## Function: test\_coverage(a, b)

Coverage Standard ☒ Covered

All Uses

All C-Uses  
some P-UsesAll P-Uses  
some C-Uses

All C-Uses

All Defs

All P-Uses

## Execution Path

1 1-&gt;2-&gt;3-&gt;4-&gt;5-&gt;6

2 1-&gt;2-&gt;4-&gt;5-&gt;6

## Structure

test\_coverage(a, b)

# 未完成的工作及安排

- 将dc路径的绘制叠加到控制流图上，使其更加直观；
- 项目测试尚未完成；
- ui优化，添加更多自定义设置；

| 时间                 | 工作   |
|--------------------|--|
| 2024.4.3-2024.4.31 | 使用测试框架jest完成项目测试，其他在原有功能上修改和优化，然后撰写完整的毕设论文 |