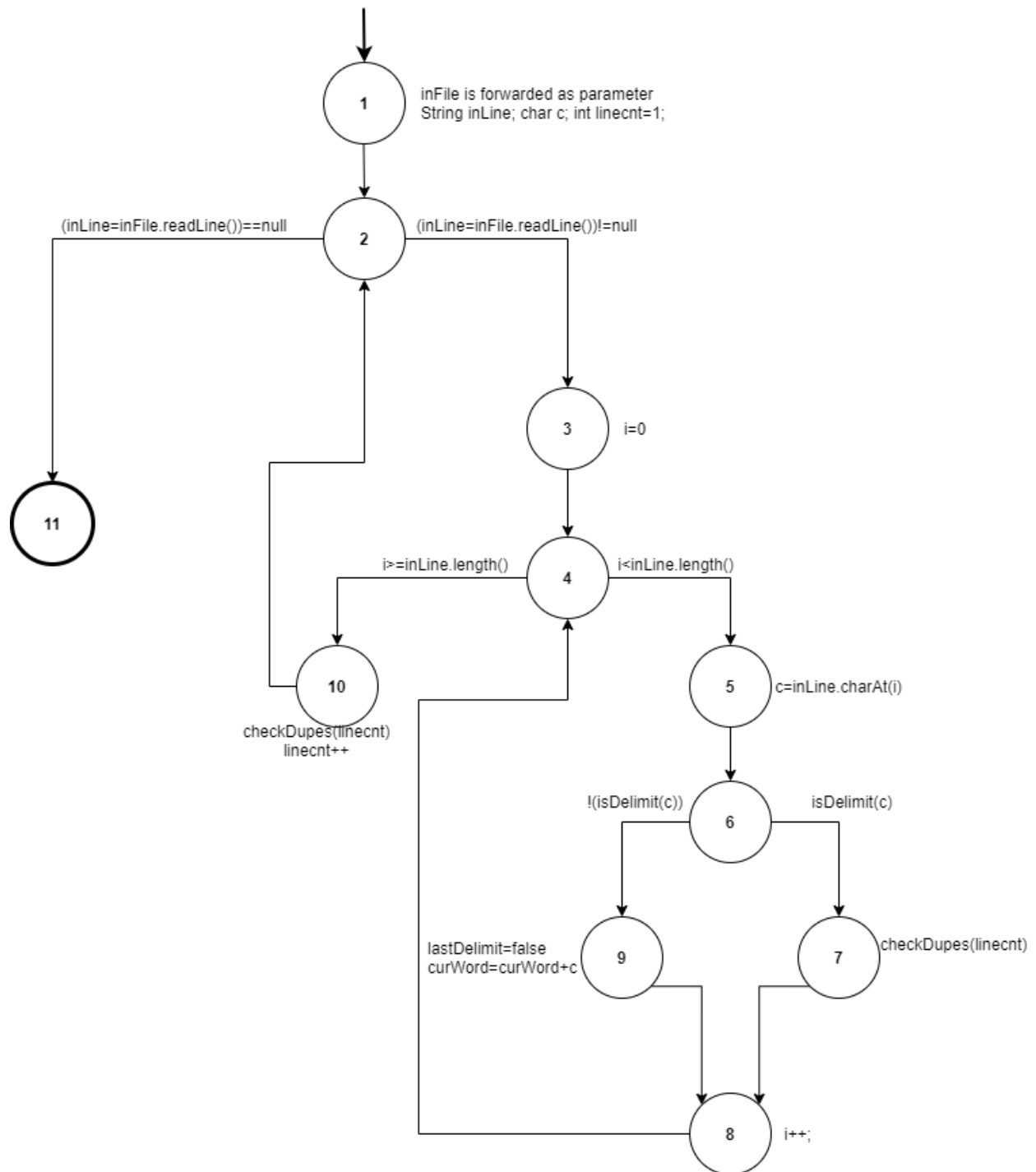


Софтверски квалитет и тестирање

Домашна задача 2

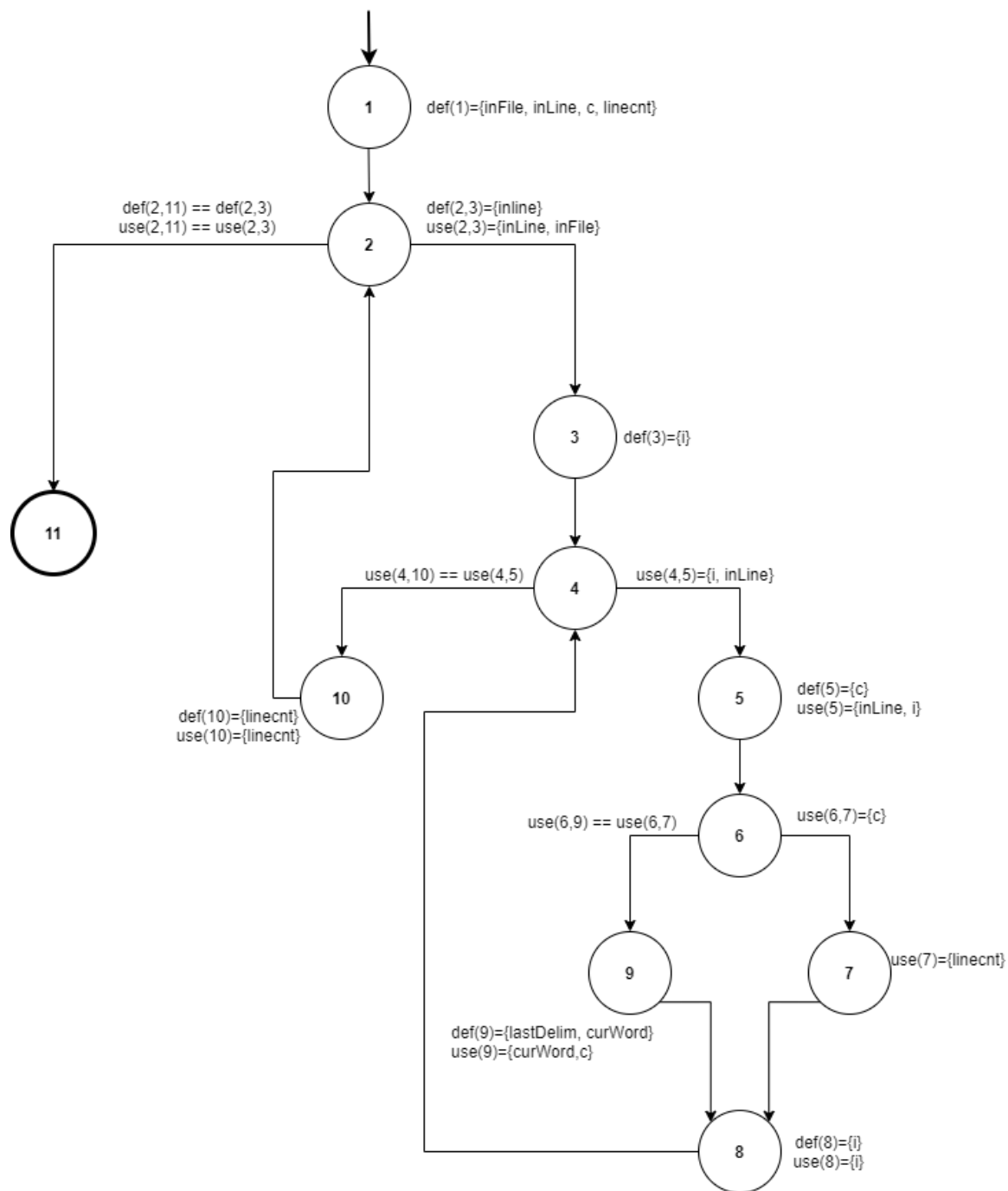
183016

Соодветниот граф на контрола на тек, за методата `stut` од класата `Stutter.java` е даден на слика 1:



слика 1 CFG за `stut` методата

На следната слика е прикажан истиот граф, чиито ребра и јазли се обележани со соодветните множества на дефиниции и употреби од променливите кои се дефинирани/употребени на конкретниот јазол/ребро.



слика 2 *du-pairs* за секој јазол/ребро

За подобра прегледност, во следните две табели се дадени сите *def* и *use* множества за секој јазол од графот (во првата табела) и сите *def* и *use* множества за секое ребро од графот (во втората табела).

Табела 1 Defs и Uses во секој јазол од CFG за stut методата

Јазол	def	use
1	{inFile, inLine, c, linecnt}	/
2	/	
3	{i}	/
4	/	/
5	{c}	{inLine, i}
6	/	/
7	/	{linecnt}
8	{i}	{i}
9	{lastDelim, curWord}	{curWord, c}
10	{linecnt}	{linecnt}
11	/	/

Табела 2 Defs и Uses на секое ребро од CFG за stut методата

Ребро	def	use
(1,2)	/	/
(2,3)	{inLine}	{inLine, inFile}
(2,11)	def(2,3)	use(2,3)
(3,4)	/	/
(4,5)	/	{i, inLine}
(4,10)	/	use(4,5)
(10,2)	/	/
(5,6)	/	/
(6,7)	/	{c}
(6,9)	/	use(6,7)
(7,8)	/	/
(9,8)	/	/
(8,4)	/	/

Во следната табела, за секоја од променливите е наведено множеството du-pairs т.е сите можни комбинации на почетна локација каде променливата се дефинира (def) и со крајна локација каде променливата ќе биде искористена (use).

Табела 3 DU парови за секоја променлива од CFG графот за stut методата

Променлива	DU парови
inFile	[1, (2, 3)] , [1, (2, 11)]
inLine	[1,5] , [1, (2, 3)] , [1, (2, 11)] , [1, (4, 5)] , [1, (4, 10)] , [(2, 3), 5] , [(2, 11), 5] , [(2, 3), (2, 3)] , [(2, 3), (2, 11)] , [(2, 3), (4, 5)] [(2, 3), (4, 10)] , [(2, 11), (2, 3)] [(2, 11), (2, 11)] , [(2, 11), (4, 5)] [(2, 11), (4, 10)]
c	[1,9] , [5,9] , [1, (6, 7)] , [1, (6, 9)] , [5, (6, 7)] [5, (6, 9)]
linecnt	[1,7] , [1,10] , [10,7] , [10,10]
i	[3,5] , [3,8] , [8,5] , [8,8] , [3, (4, 5)] , [3, (4, 10)] , [8, (4, 5)] , [8, (4, 10)]
lastDelim	/
curWord	[9,9]

Следно е потребно за секоја од променливите да се дефинираат du-paths коишто според дефиницијата се *едноставни патеки* (во кои нема внатрешни циклуси) и се *def-clear* (долж ваквите патеки вредноста на конкретната променлива нема да се промени – т.е. не постои def за променливата во рамките на таквата патека). Од овие патеки понатаму ќе се дефинираат конкретните **тест патеки** (преку комбинирање на du патеки), коишто пак ќе бидат одредени од избраниот критериум за покривање.

Табела 4 DU патеки за секоја од променливите од CFG

Променлива	DU патеки
inFile	[1,2,11] , [1,2,3]
inLine	[2,3,4,5] , [2,3,4,10]
c	[5,6,7] , [5,6,9]
linecnt	[1,2,3,4,10] , [1,2,3,4,5,6,7] [10,2,3,4,10] , [10,2,3,4,5,6,7]
i	[3,4,5] , [3,4,10] , [3,4,5,6,9,8] , [3,4,5,6,7,8] [8,4,5] , [8,4,10] , [8,4,5,6,9,8] , [8,4,5,6,7,8]
lastDelim	/
curWord	[9,8,4,5,6,9]

Одбраниот критериум за покривање е All-DU-paths којшто побарува да биде покриен секој можен пат помеѓу сите def и сите uses, за секоја променлива и тоа преку сите можни du патеки – поедноставно кажано, потребно е да биде покриена секоја du патека. Ова се прави преку комбинирање на веќе добиените du патеки во целосни тест патеки: секоја ваква патека мора да започне во иницијалниот јазол и да заврше во крајниот јазол. Ваквите тест патеки се наведени во *табела 5*.

Табела 5 All-DU Paths Coverage

Променлива	All-DU Paths Coverage
inFile	[1,2,11] , [1,2,3,4,10,2,11]
inLine	[1,2,3,4,5,6,7,8,4,10,2,11] [1,2,3,4,10,2,11]
c	[1,2,3,4,5,6,7,8,4,10,2,11] [1,2,3,4,5,6,9,8,4,10,2,11]
linecnt	[1,2,3,4,10,2,11] [1,2,3,4,5,6,7,8,4,10,2,11] [1,2,3,4,10,2,3,4,10,2,11] [1,2,3,4,10,2,3,4,5,6,7,8,4,10,2,11]
i	[1,2,3,4,5,6,9,8,4,10,2,11] [1,2,3,4,10,2,11] [1,2,3,4,5,6,7,8,4,10,2,11] [1,2,3,4,5,6,9,8,4,5,6,9,8,4,10,2,11] [1,2,3,4,5,6,9,8,4,10,2,11] [1,2,3,4,5,6,9,8,4,5,6,7,8,4,10,2,11]
lastDelim	/
curWord	[1,2,3,4,5,6,9,8,4,5,6,9,8,4,10,2,11]

Во следната табела се наведени сите тест патеки кои ќе го задоволат одбраниот критериум, заедно со конкретни тест случаи кои ќе резултираат со извршување на соодветната тест патека.

Табела 6 Конкретни тест случаи според All-DU paths критериумот

Тест случај (input, expected output)	Тест патека
(*празна тхт датотека*, "empty txt file")	[1,2,11]
(*тхт датотеката има 1 празна линија*, "enters while – inLine is not null")	[1,2,3,4,10,2,11]
(".", "enters while – inLine is not null; enters for")	[1,2,3,4,5,6,7,8,4,10,2,11]
("a", "enters while – inLine is not null; enters for")	[1,2,3,4,5,6,9,8,4,10,2,11]
(*тхт датотеката има 2 празни линии*, "enters while – inLine is not null; enters while – inLine is not null")	[1,2,3,4,10,2,3,4,10,2,11]
(*тхт датотека со првата линија празна, а во втората линија има .*, "enters while – inLine is not null; enters while – inLine is not null; enters for")	[1,2,3,4,10,2,3,4,5,6,7,8,4,10,2,11]
("aa", "enters while – inLine is not null; enters for; enters for")	[1,2,3,4,5,6,9,8,4,5,6,9,8,4,10,2,11]
("a.", "enters while – inLine is not null; enters for; enters for")	[1,2,3,4,5,6,9,8,4,5,6,7,8,4,10,2,11]