第三次Lab：

Dataflow Analysis

什么是def_use分析

- Def-Use 分析（Definition-Use Analysis）是程序分析中的一种重要数据流分析技术，用于确定程序中变量定义（def）和使用（use）之间的关系
  - 定义（Definition/Def）：对变量的赋值或写入操作
  - 使用（Use）：对变量值的读取操作
  - 示例:

```
1: x = 5      // Def of x
2: y = x + 3  // Use of x Def of y
3: x = 8      // Def of x
4: z = x * 2  // Use of x Def of z
```

Def-Use 分析会建立以下关系:

第1行的定义在第2行被使用

第3行的定义在第4行被使用

[1]

一、任务

- 本学期代码仓库:https://gitee.com/fdu-ssr/compiler2025spring
- GIR参考文档https://docs.qq.com/sheet/DTXBCSlZZS25mQnhQ?tab=urh0bh
- 本次实验需要补充compiler2025spring/lab3/code/src/lian/semantic/stmt_def_use_analysis.py
- 本次任务要求:
  - 编写指定GIR指令的def_use分析，标记指令中哪些符号被define了，那些符号被use了，包括call_expression if_stmt array_write array_read指令

[1]

二、代码运行方式

（1）运行scripts/lian.sh脚本

$./lian.sh <待分析代码文件路径> -l 语言名称

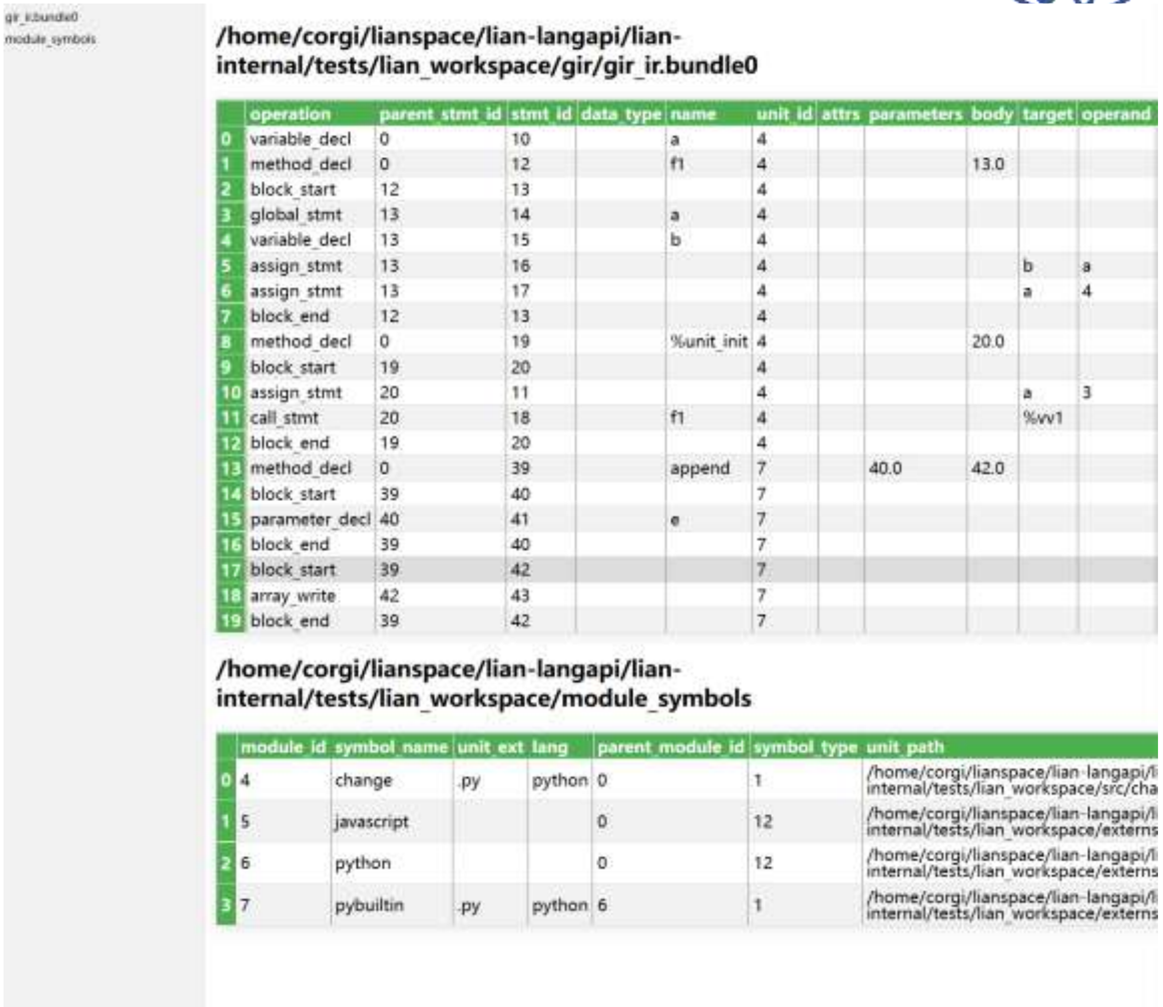
例如:

./lian.sh /python/change.py -l python

（2）运行结果为:

tests/lian_workspace/dataframe.html

将这个文件在网页中打开



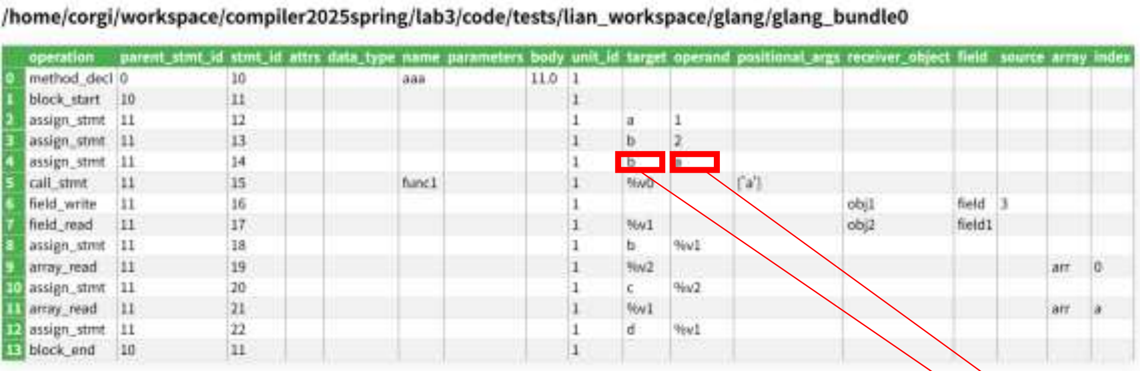
/home/corgi/lianspace/lian-langapi/lian-internal/tests/lian_workspace/gir/gir_ir.bundle0

	operation	parent_stmt_id	stmt_id	data_type	name	unit_id	attrs	parameters	body	target	operand
0	variable_decl	0	10		a	4					
1	method_decl	0	12		f1	4			13.0		
2	block_start	12	13			4					
3	global_stmt	13	14		a	4					
4	variable_decl	13	15		b	4					
5	assign_stmt	13	16			4				b	a
6	assign_stmt	13	17			4				a	4
7	block_end	12	13			4					
8	method_decl	0	19		%unit_init	4			20.0		
9	block_start	19	20			4					
10	assign_stmt	20	11			4				a	3
11	call_stmt	20	18		f1	4				%vv1	
12	block_end	19	20			4					
13	method_decl	0	39		append	7		40.0	42.0		
14	block_start	39	40			7					
15	parameter_decl	40	41		e	7					
16	block_end	39	40			7					
17	block_start	39	42			7					
18	array_write	42	43			7					
19	block_end	39	42			7					

/home/corgi/lianspace/lian-langapi/lian-internal/tests/lian_workspace/module_symbols

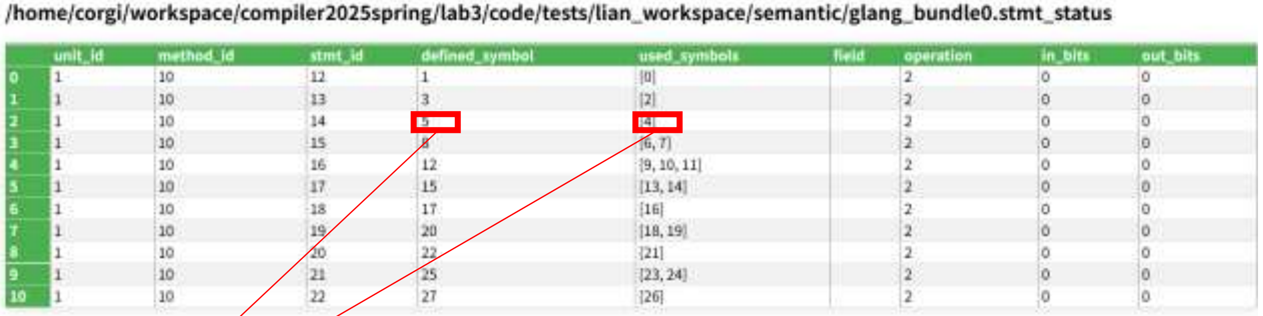
	module_id	symbol_name	unit_ext	lang	parent_module_id	symbol_type	unit_path
0	4	change	.py	python	0	1	/home/corgi/lianspace/lian-langapi/li internal/tests/lian_workspace/src/cha
1	5	javascript			0	12	/home/corgi/lianspace/lian-langapi/li internal/tests/lian_workspace/externs
2	6	python			0	12	/home/corgi/lianspace/lian-langapi/li internal/tests/lian_workspace/externs
3	7	pybuiltin	.py	python	6	1	/home/corgi/lianspace/lian-langapi/li internal/tests/lian_workspace/externs

三、结果的查看方式



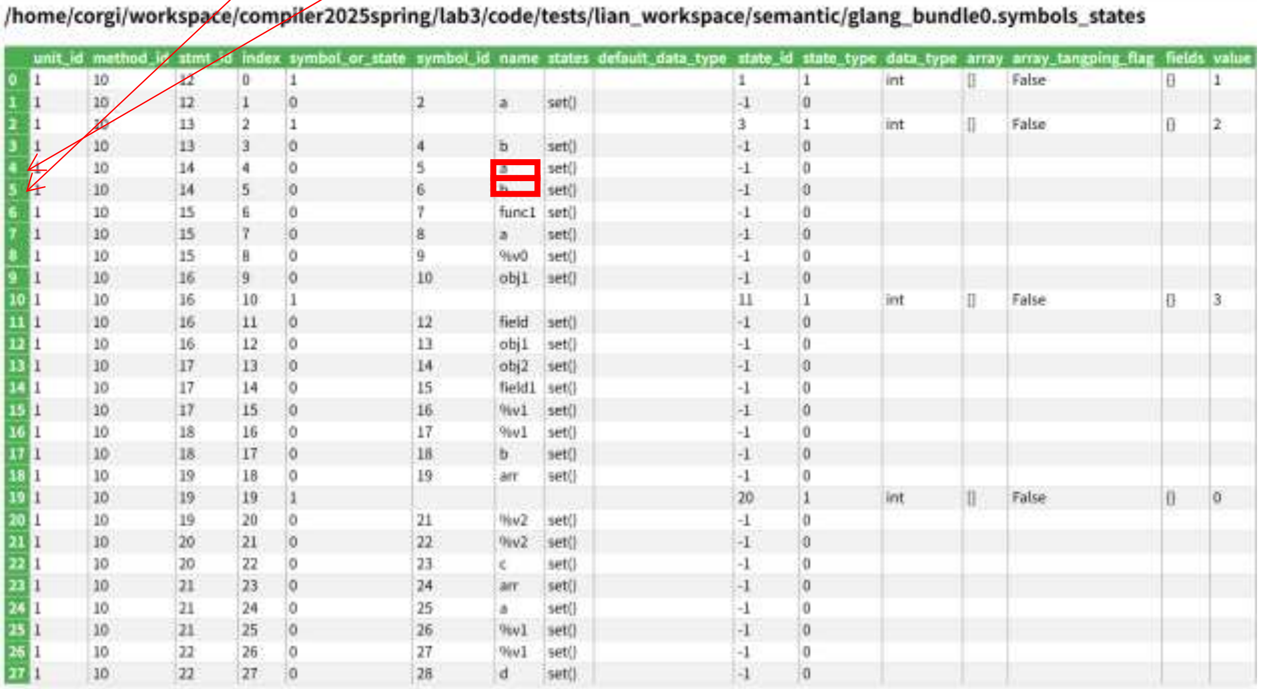
/home/corgi/workspace/compiler2025spring/lab3/code/tests/lian_workspace/glang/glang_bundle0

	operation	parent_stmt_id	stmt_id	attrs	data_type	name	parameters	body	unit_id	target	operand	positional_args	receiver_object	field	source	array	index
0	method_decl	0	10			aaa		11.0	1								
1	block_start	10	11						1								
2	assign_stmt	11	12						1	a	1						
3	assign_stmt	11	13						1	b	2						
4	assign_stmt	11	14						1	b	a						
5	call_stmt	11	15			func1			1	%v0		['a']					
6	field_write	11	16						1				obj1	field	3		
7	field_read	11	17						1	%v1			obj2	field1			
8	assign_stmt	11	18						1	b	%v1						
9	array_read	11	19						1	%v2						arr	0
10	assign_stmt	11	20						1	c	%v2						
11	array_read	11	21						1	%v1						arr	a
12	assign_stmt	11	22						1	d	%v1						
13	block_end	10	11						1								



/home/corgi/workspace/compiler2025spring/lab3/code/tests/lian_workspace/semantic/glang_bundle0.stmt_status

	unit_id	method_id	stmt_id	defined_symbol	used_symbols	field	operation	in_bits	out_bits
0	1	10	12	1	[0]		2	0	0
1	1	10	13	3	[2]		2	0	0
2	1	10	14	5	[4]		2	0	0
3	1	10	15	8	[6, 7]		2	0	0
4	1	10	16	12	[9, 10, 11]		2	0	0
5	1	10	17	15	[13, 14]		2	0	0
6	1	10	18	17	[16]		2	0	0
7	1	10	19	20	[18, 19]		2	0	0
8	1	10	20	22	[21]		2	0	0
9	1	10	21	25	[23, 24]		2	0	0
10	1	10	22	27	[26]		2	0	0



/home/corgi/workspace/compiler2025spring/lab3/code/tests/lian_workspace/semantic/glang_bundle0.symbols_states

	unit_id	method_id	stmt_id	index	symbol_or_state	symbol_id	name	states	default_data_type	state_id	state_type	data_type	array	array_tangping_flag	fields	value
0	1	10	12	0	1					1	1	int	[]	False	{}	1
1	1	10	12	1	0	2	a	set()		-1	0					
2	1	10	13	2	1					3	1	int	[]	False	{}	2
3	1	10	13	3	0	4	b	set()		-1	0					
4	1	10	14	4	0	5	a	set()		-1	0					
5	1	10	14	5	0	6	b	set()		-1	0					
6	1	10	15	6	0	7	func1	set()		-1	0					
7	1	10	15	7	0	8	a	set()		-1	0					
8	1	10	15	8	0	9	%v0	set()		-1	0					
9	1	10	16	9	0	10	obj1	set()		-1	0					
10	1	10	16	10	1					11	1	int	[]	False	{}	3
11	1	10	16	11	0	12	field	set()		-1	0					
12	1	10	16	12	0	13	obj1	set()		-1	0					
13	1	10	17	13	0	14	obj2	set()		-1	0					
14	1	10	17	14	0	15	field1	set()		-1	0					
15	1	10	17	15	0	16	%v1	set()		-1	0					
16	1	10	18	16	0	17	%v1	set()		-1	0					
17	1	10	18	17	0	18	b	set()		-1	0					
18	1	10	19	18	0	19	arr	set()		-1	0					
19	1	10	19	19	1					20	1	int	[]	False	{}	0
20	1	10	19	20	0	21	%v2	set()		-1	0					
21	1	10	20	21	0	22	%v2	set()		-1	0					
22	1	10	20	22	0	23	c	set()		-1	0					
23	1	10	21	23	0	24	arr	set()		-1	0					
24	1	10	21	24	0	25	a	set()		-1	0					
25	1	10	21	25	0	26	%v1	set()		-1	0					
26	1	10	22	26	0	27	%v1	set()		-1	0					
27	1	10	22	27	0	28	d	set()		-1	0					

首先查看GIR，例如stmt_id=14的指令，在symbol_states表中，绿色列对应着当前行的index，index=4和index=5的行是stmt_id=14的GIR对应的symbol

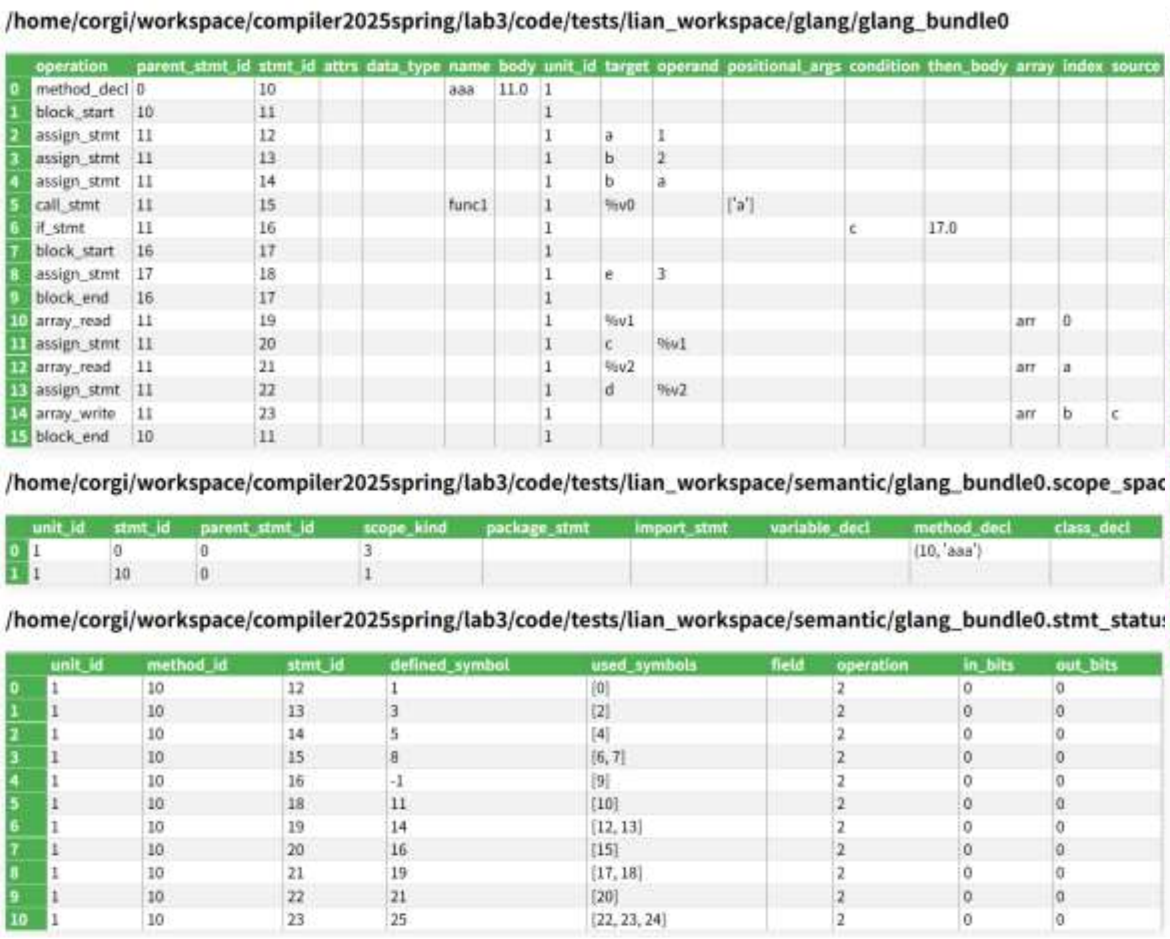
stmt_status记录了每条指令def_use的关系，define_symbol列
中记录了这条指令define了在symbol_states表中index=5的symbol，也就是“b”

六、编写def_use

- 本次实验用到的api有：
    - add_def_use_symbols（stmt_id, def_symbol, used_symbols, op)
    - def_symbol是该条指令被定义的符号
    - used_symbols是该条指令被使用的符号列表
    - 在使用该api时只用补充正确的stmt_id, def_symbol, used_symbol即可
- 注意事项
    - call_stmt会use函数名与参数，参数只需考虑positional_args，positional_args是一个列表，记录所有实参名
    - array_write不仅会define array(a[b]的a部分), 也会use array

[1]

参考结果-def_use关系正确即可



/home/corgi/workspace/compiler2025spring/lab3/code/tests/lian_workspace/glang/glang_bundle0

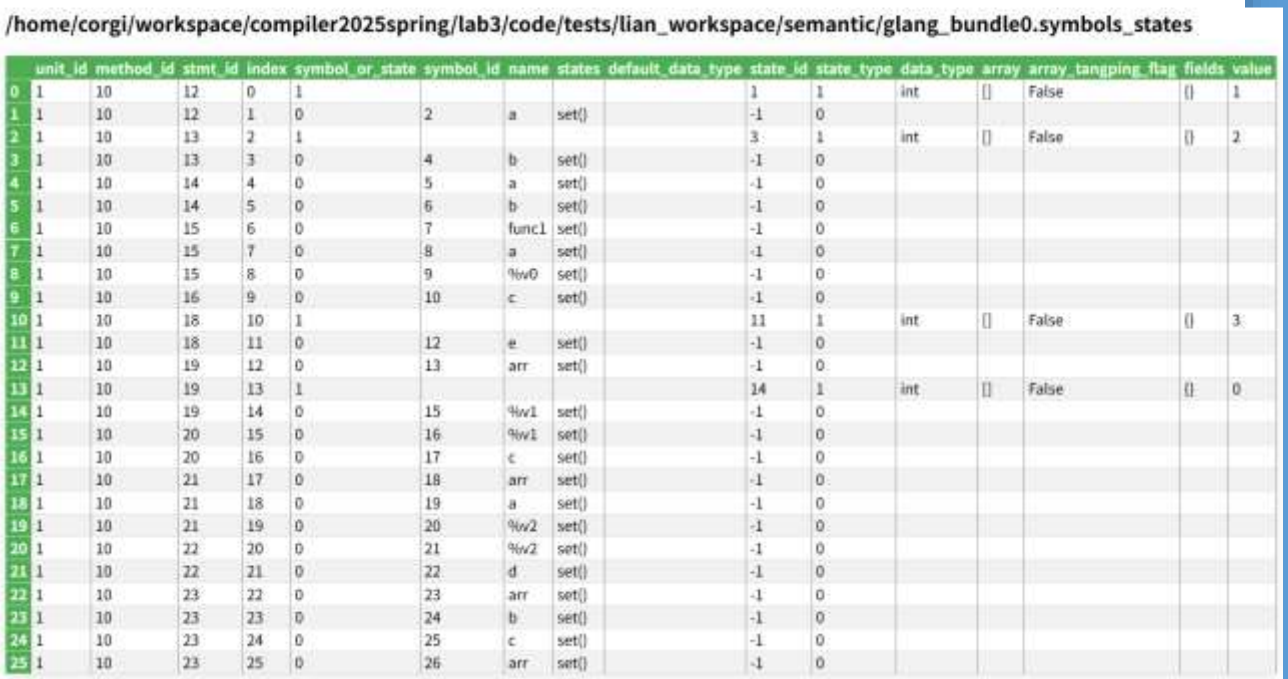
	operation	parent_stmt_id	stmt_id	attrs	data_type	name	body	unit_id	target	operand	positional_args	condition	then_body	array	index	source
0	method_decl	0	10			aaa	11.0	1								
1	block_start	10	11					1								
2	assign_stmt	11	12					1	a	1						
3	assign_stmt	11	13					1	b	2						
4	assign_stmt	11	14					1	b	a						
5	call_stmt	11	15			func1		1	%v0		['a']					
6	if_stmt	11	16					1				c	17.0			
7	block_start	16	17					1								
8	assign_stmt	17	18					1	e	3						
9	block_end	16	17					1								
10	array_read	11	19					1	%v1					arr	0	
11	assign_stmt	11	20					1	c	%v1						
12	array_read	11	21					1	%v2					arr	a	
13	assign_stmt	11	22					1	d	%v2						
14	array_write	11	23					1						arr	b	c
15	block_end	10	11					1								

/home/corgi/workspace/compiler2025spring/lab3/code/tests/lian_workspace/semantic/glang_bundle0.scope_spac

	unit_id	stmt_id	parent_stmt_id	scope_kind	package_stmt	import_stmt	variable_decl	method_decl	class_decl
0	1	0	0	3				(10, 'aaa')	
1	1	10	0	1					

/home/corgi/workspace/compiler2025spring/lab3/code/tests/lian_workspace/semantic/glang_bundle0.stmt_statu:

	unit_id	method_id	stmt_id	defined_symbol	used_symbols	field	operation	in_bits	out_bits
0	1	10	12	1	[0]		2	0	0
1	1	10	13	3	[2]		2	0	0
2	1	10	14	5	[4]		2	0	0
3	1	10	15	8	[6, 7]		2	0	0
4	1	10	16	-1	[9]		2	0	0
5	1	10	18	11	[10]		2	0	0
6	1	10	19	14	[12, 13]		2	0	0
7	1	10	20	16	[15]		2	0	0
8	1	10	21	19	[17, 18]		2	0	0
9	1	10	22	21	[20]		2	0	0
10	1	10	23	25	[22, 23, 24]		2	0	0



/home/corgi/workspace/compiler2025spring/lab3/code/tests/lian_workspace/semantic/glang_bundle0.symbols_states

	unit_id	method_id	stmt_id	index	symbol_or_state	symbol_id	name	states	default_data_type	state_id	state_type	data_type	array	array_tangping_flag	fields	value
0	1	10	12	0	1					1	1	int	[]	False	{}	1
1	1	10	12	1	0	2	a	set()		-1	0					
2	1	10	13	2	1					3	1	int	[]	False	{}	2
3	1	10	13	3	0	4	b	set()		-1	0					
4	1	10	14	4	0	5	a	set()		-1	0					
5	1	10	14	5	0	6	b	set()		-1	0					
6	1	10	15	6	0	7	func1	set()		-1	0					
7	1	10	15	7	0	8	a	set()		-1	0					
8	1	10	15	8	0	9	%v0	set()		-1	0					
9	1	10	16	9	0	10	c	set()		-1	0					
10	1	10	18	10	1					11	1	int	[]	False	{}	3
11	1	10	18	11	0	12	e	set()		-1	0					
12	1	10	19	12	0	13	arr	set()		-1	0					
13	1	10	19	13	1					14	1	int	[]	False	{}	0
14	1	10	19	14	0	15	%v1	set()		-1	0					
15	1	10	20	15	0	16	%v1	set()		-1	0					
16	1	10	20	16	0	17	c	set()		-1	0					
17	1	10	21	17	0	18	arr	set()		-1	0					
18	1	10	21	18	0	19	a	set()		-1	0					
19	1	10	21	19	0	20	%v2	set()		-1	0					
20	1	10	22	20	0	21	%v2	set()		-1	0					
21	1	10	22	21	0	22	d	set()		-1	0					
22	1	10	23	22	0	23	arr	set()		-1	0					
23	1	10	23	23	0	24	b	set()		-1	0					
24	1	10	23	24	0	25	c	set()		-1	0					
25	1	10	23	25	0	26	arr	set()		-1	0					

[1]