

Class 15

- Questions/comments
- Testing continued
- Assign (see Schedule for links)
 - Problem Set 6 discuss
 - Readings

1

Discussion

- Coincidental correctness
- Coverage criteria
 - Statement
 - Branch/decision
 - Multiple condition
 - MCDC

MC/DC: Example

(a && b && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

MC/DC: Example

(a && b && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

MC/DC: Example

(**a** && b && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

MC/DC: Example

(a && b && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

1	True	True	True	True
5	False	True	True	False

MC/DC: Example

(a && **b** && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

1	True	True	True	True
5	False	True	True	False

MC/DC: Example

(a && b && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

1	True	True	True	True
5	False	True	True	False
3	True	False	True	False

MC/DC: Example

(a && b && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

1	True	True	True	True
5	False	True	True	False
3	True	False	True	False

MC/DC: Example

(a && b && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

1	True	True	True	True
5	False	True	True	False
3	True	False	True	False

MC/DC: Example

(a && b && c)

Test case	a	b	c	outcome
1	True	True	True	True
2	True	True	False	False
3	True	False	True	False
4	True	False	False	False
5	False	True	True	False
6	False	True	False	False
7	False	False	True	False
8	False	False	False	False

1	True	True	True	True
5	False	True	True	False
3	True	False	True	False
2	True	True	False	False

Modified Condition/Decision Coverage (MC/DC)

- Tradeoff between number of required test cases and thoroughness of the test
- Required by both US and European quality standards in aviation, and used in other industries (e.g., automobile, submarine, medical)

Discussion

- Coincidental correctness
- Coverage criteria
 - Statement
 - Branch/decision
 - Multiple condition
 - MCDC
 - Basis path

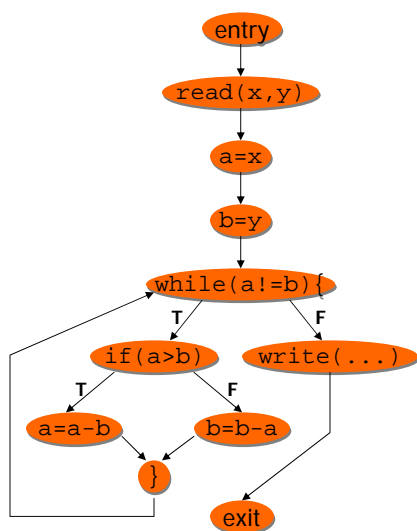
Discussion

- Coverage criteria (new today)
 - Loop
 - Data-flow
 - Why was it developed? Three main groups presented criteria
 - What are different data-flow criteria?
 - How can you compute test-requirements?
 - What is the subsumption hierarchy?
 - What are practical considerations for data-flow testing?
 - How would you instrument for data-flow coverage?

Loop Coverage

Test requirements (given n): all paths that contain at most n iterations of any loop in the program

Loop Coverage: Example



- $n=0$ body of while is never executed
- $n=1$ all statements are executed
- $n=2$ body of while is executed twice (intuitively, more thorough)
- $n=3$...
- n must be defined based on some rationale

Loop Coverage: Problems

```
...
int a[k], elem=x;
for(i=0;a[i]!=elem;i++){
    ...
}
...
```

For the example, $n \geq k$ may be a good choice, but

- n may be different for different loops in the program
- n may be very high

- No algorithm can identify an optimal value for n
- The cost of the criterion grows exponentially with n

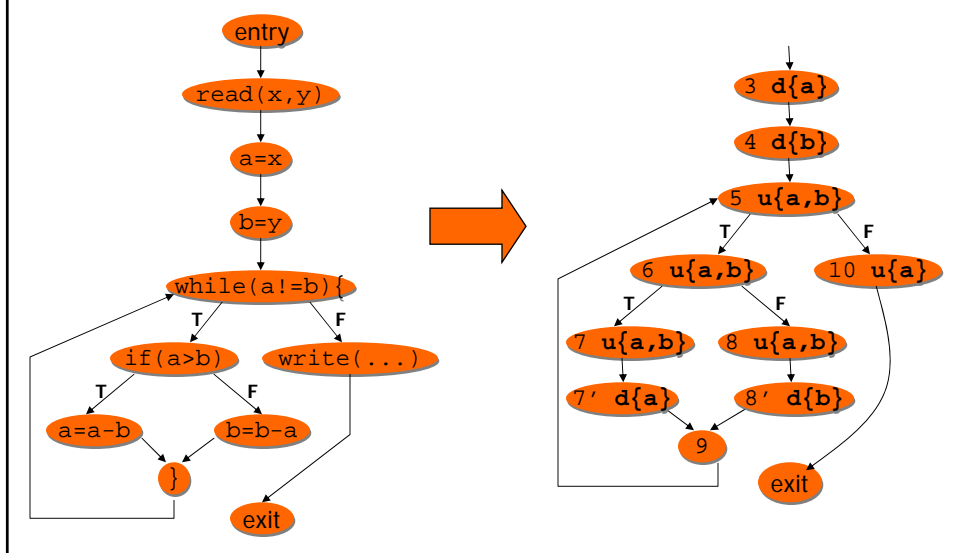
Data-Flow Coverage Criteria

Basic idea: Test case selection is based on coverage of definitions to uses (du-pairs)

Test requirements: sets of du-pairs with respect to variable v

- a **definition** of v is a reference to v where the value of v is changed (e.g., assignment to v , input a value of v)
- a **use** v is a reference to v where the value of v is fetched by not changed (e.g., v on right-hand side of assignment, v is output)
- a **def-clear subpath** for a definition d of v and a use u of v is a subpath in the CFG between d and u on which v is not redefined
- two kinds of uses: **computation** (c-use) and **predicate** (p-use)

Data-Flow Coverage Criteria: Example



Data-Flow Coverage Criteria: Example

