MONASH University

FIT2107-Software Quality & Testing

Lecture 6 – Whitebox Testing II

8th September 2020

Dr Najam Nazar

Outline

- Some concepts -> Truth table
- Condition + Branch Coverage (C/DC)
- Path Coverage
- Modified Coverage/ Decision Coverage (MC/DC)

Announcement

- Assignment 2 is released
  - Whitebox testing
  - Unit Testing
  - CI/CD
  - Justification using the blackbox testing.

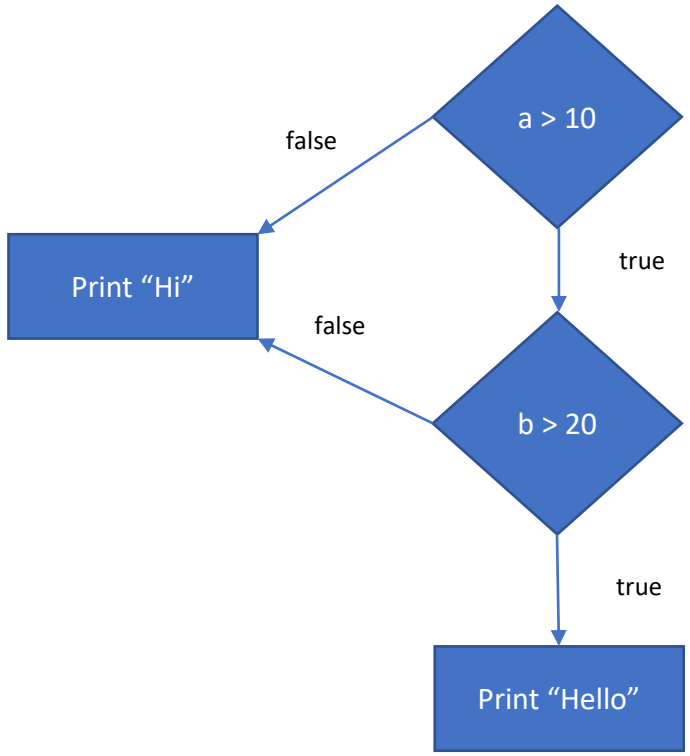
Truth Table

- A mathematical table.
- Show whether a propositional expression is true for all legitimate input.
- Used in:
  - Algebra
  - Computer science
  - Philosophy
  - …

P	Q	P ^ Q
T	T	T
T	F	F
T	F	F
F	F	F

Condition + Branch Coverage (C/DC)

```
def hello (a,b):
    if (a>10 and b>20):
        print("Hello")
    else:
        print("Hi")
```



- T1 = (20,10) a >10 is true and b >20 is false
- T2 = (5, 30) a > 10 is false and b > 20 is true
- Both T1 & T2 together achieve 100% coverage.
  - the final outcome of whole condition is false.
- 100% condition coverage but 50 % branch condition.
- In practice, when we use condition coverage, we actually do branch + condition coverage.

Path Coverage

- With condition we looked at each criterion (conditions and branch) individually.
- This gives us even more tests to generate (as seen in line coverage).
- Path coverage does not consider the conditions individually.
  - Rather it considers the (full) combination of the conditions in a decision
  - linearly independent paths in the program are executed at least once (all paths through CFG).

$$pathcoverage = \frac{paths\ covered}{Total\ paths} * 100$$

Example – Path Coverage

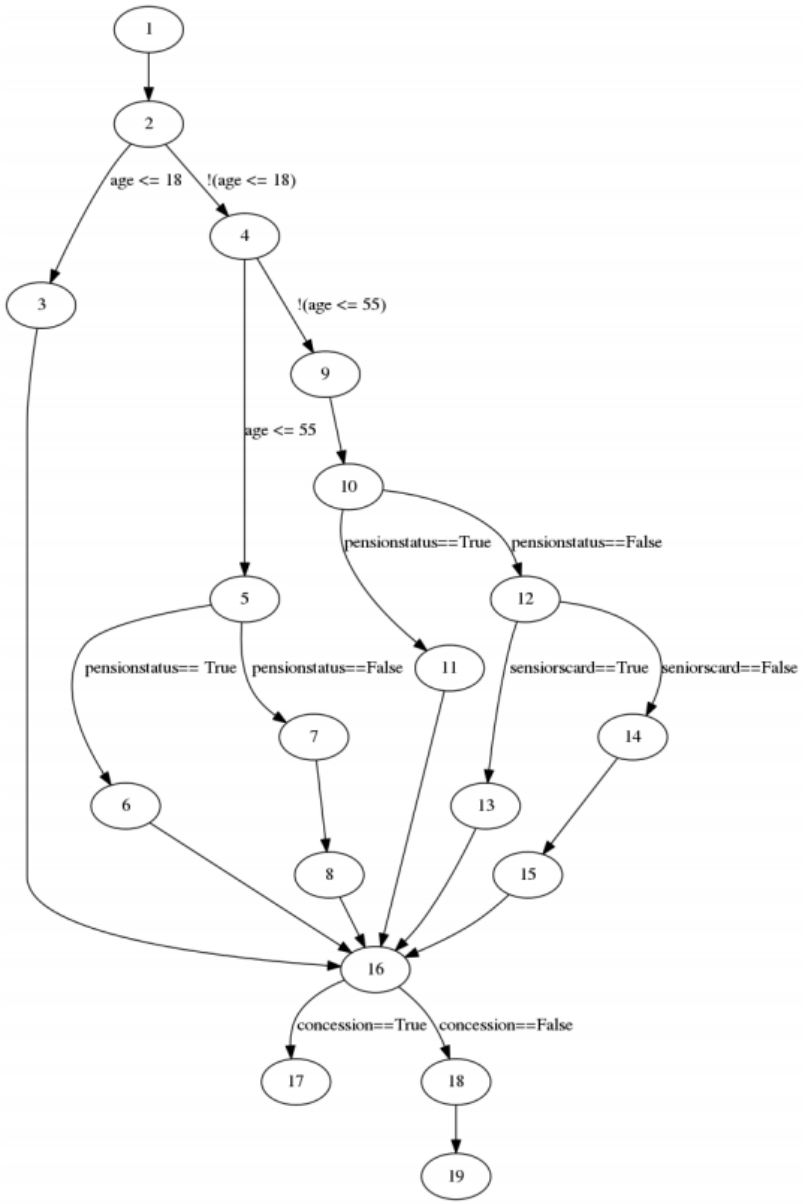
```
if (!Character.isLetter(str.charAt(i))
        & (last == 's' | last == 'r')) {
    words++;
}
```

- A = !Character.isLetter(str.charAt(i))
- B = last == ‘s’
- C = last == ‘r’
- Outcome: (A && (B || C))
- 8 Tests to cover the full path for if condition only

Tests	A	B	C	Outcome
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

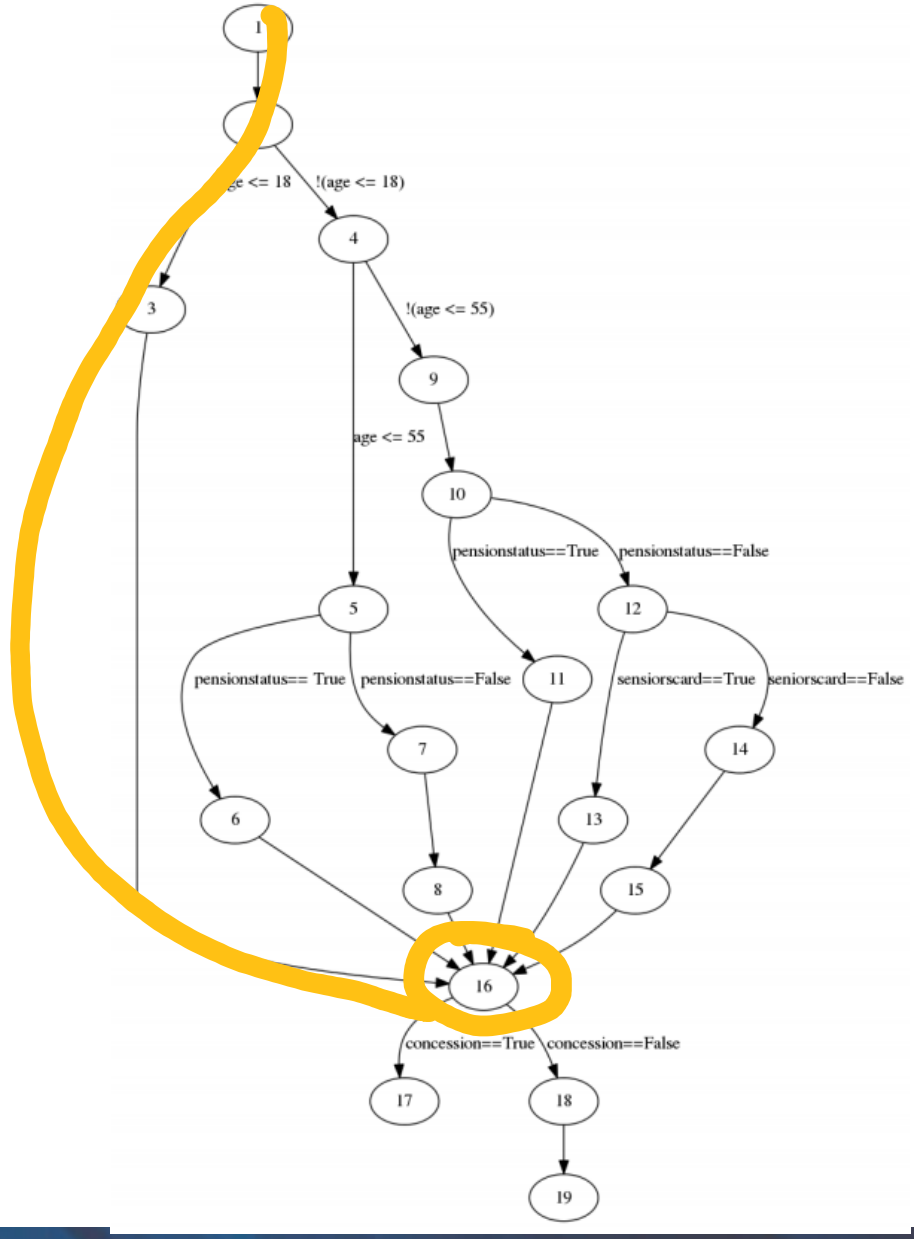
Path coverage using CFG

```python
def ticketprice(age, pensionstatus, seniorscard):
    if age <= 18:
        concession = True
    elif age <= 55:
        if pensionstatus:
            concession = True
        else:
            concession = False
    else:
        if pensionstatus:
            concession=True
        elif seniorscard:
            concession=True
        else:
            concession=False
    if concession:
        return 5.00
    else:
        return 10.00
```



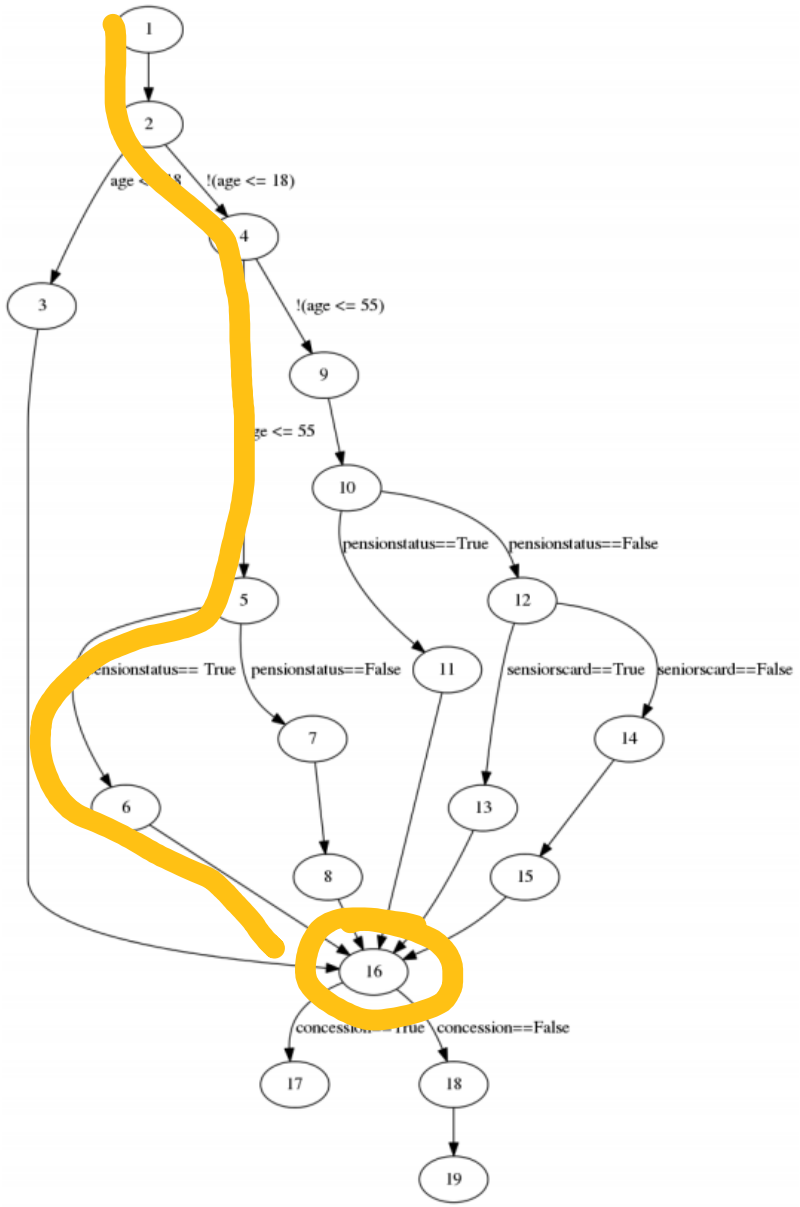
What Paths we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1,2,3,16



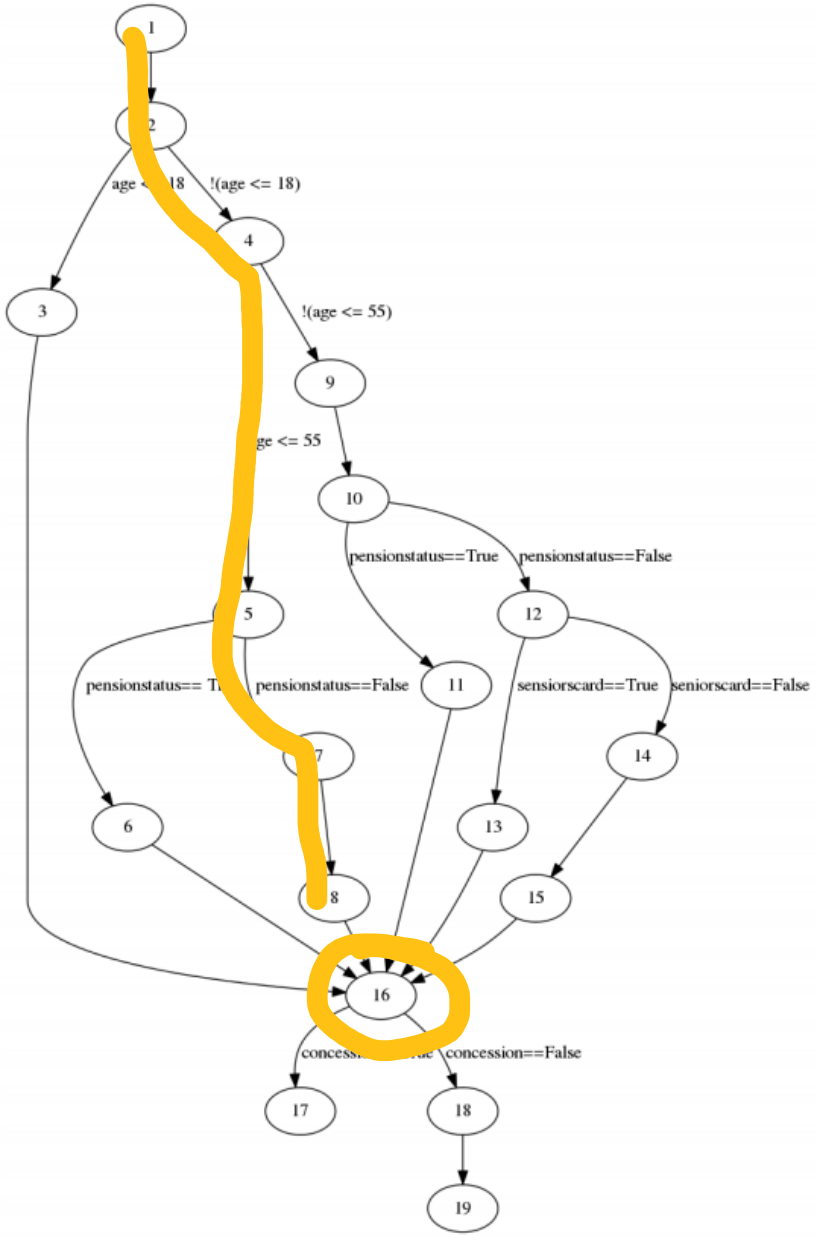
What Paths we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
  - 1, 2, 4, 5, 6, 16



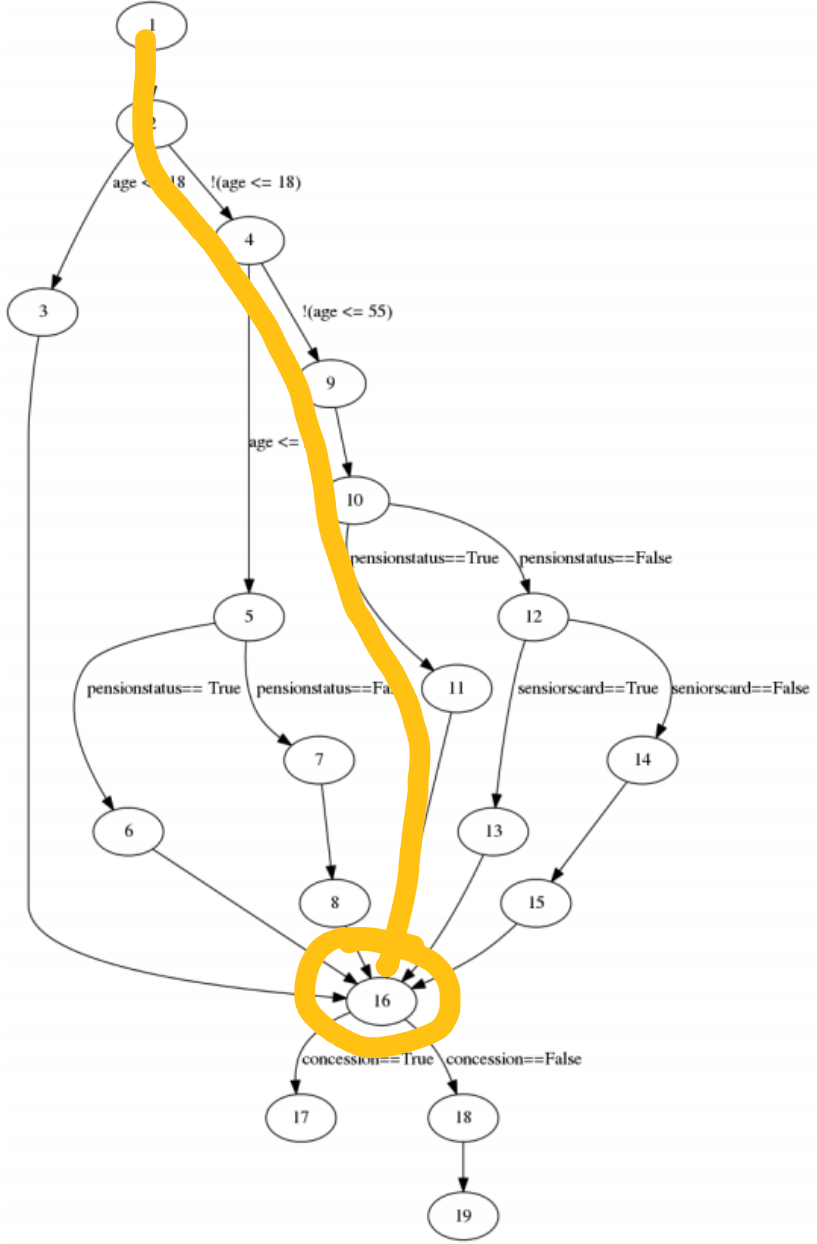
What Paths we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
  - 1, 2, 4, 5, 6, 16
  - 1, 2, 4, 5, 7, 8, 16



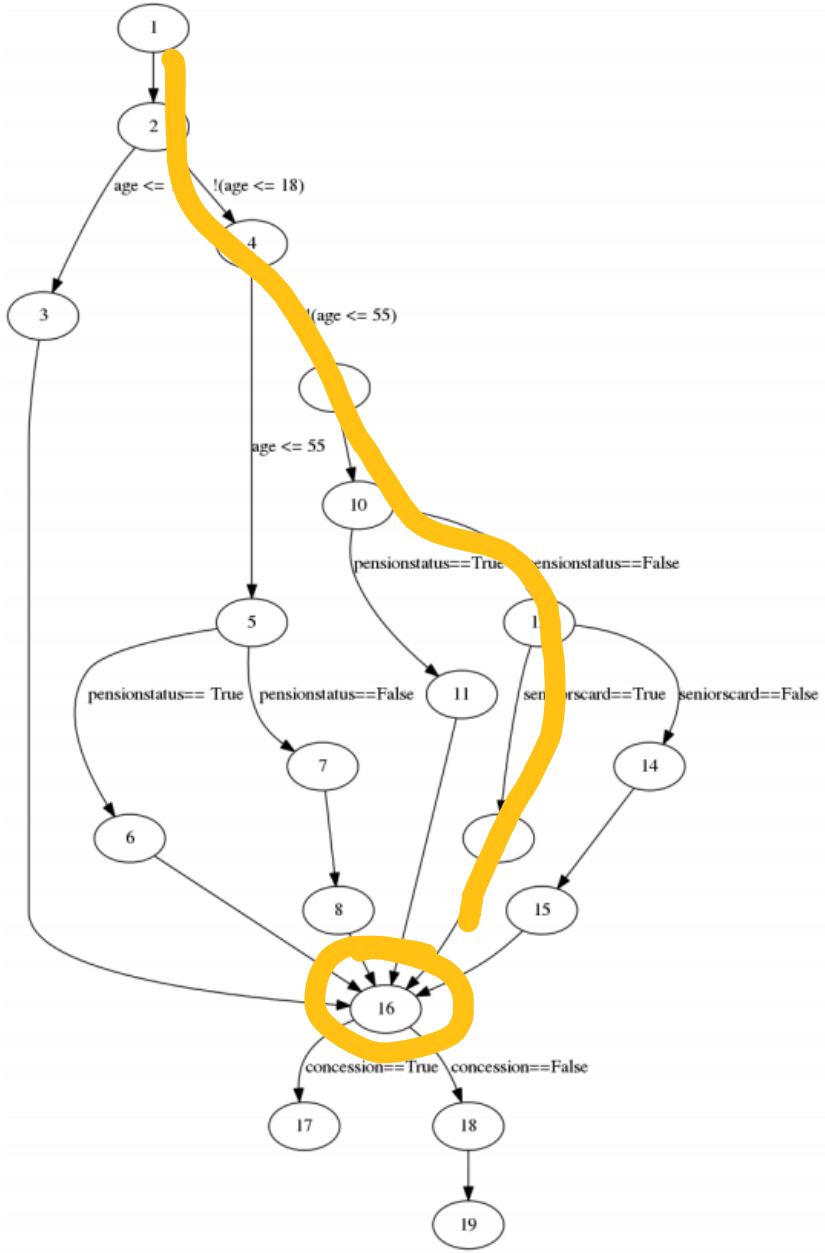
What Paths we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
  - 1, 2, 4, 5, 6, 16
  - 1, 2, 4, 5, 7, 8, 16
  - 1, 2, 4, 9, 10, 11, 16



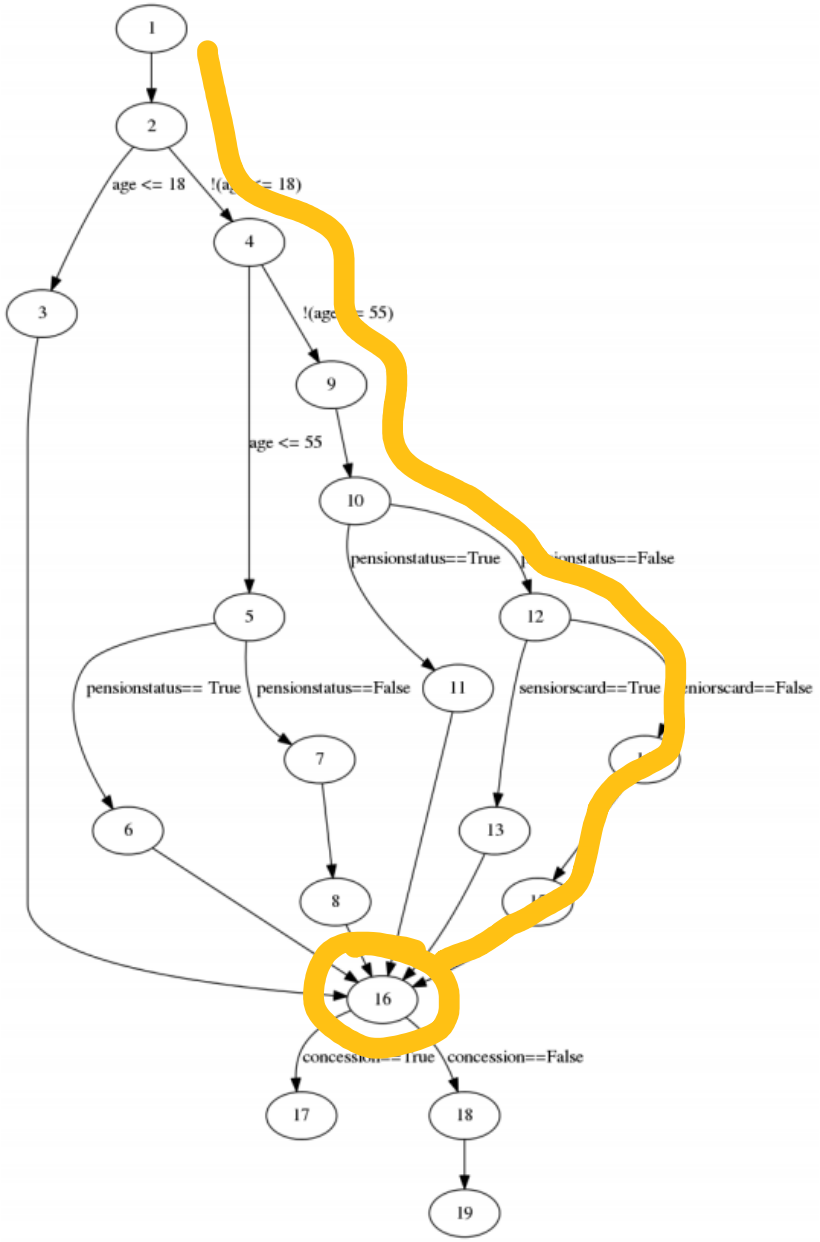
What Paths we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
  - 1, 2, 4, 5, 6, 16
  - 1, 2, 4, 5, 7, 8, 16
  - 1, 2, 4, 9, 10, 11, 16
  - 1, 2, 4, 9, 10, 12, 13, 16



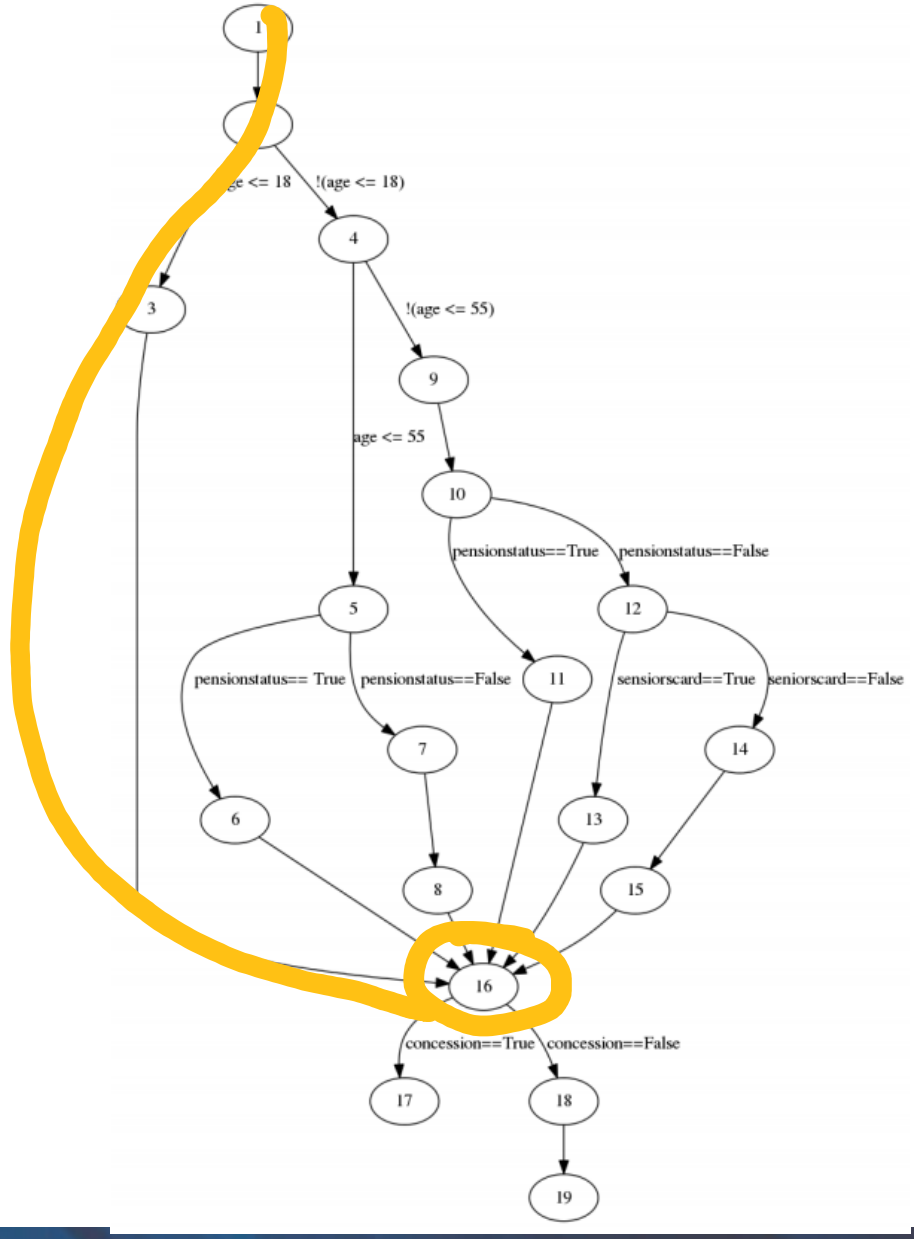
What Paths we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
  - 1, 2, 4, 5, 6, 16
  - 1, 2, 4, 5, 7, 8, 16
  - 1, 2, 4, 9, 10, 11, 16
  - 1, 2, 4, 9, 10, 12, 13, 16
  - 1, 2, 4, 9, 10, 12, 14, 15, 16



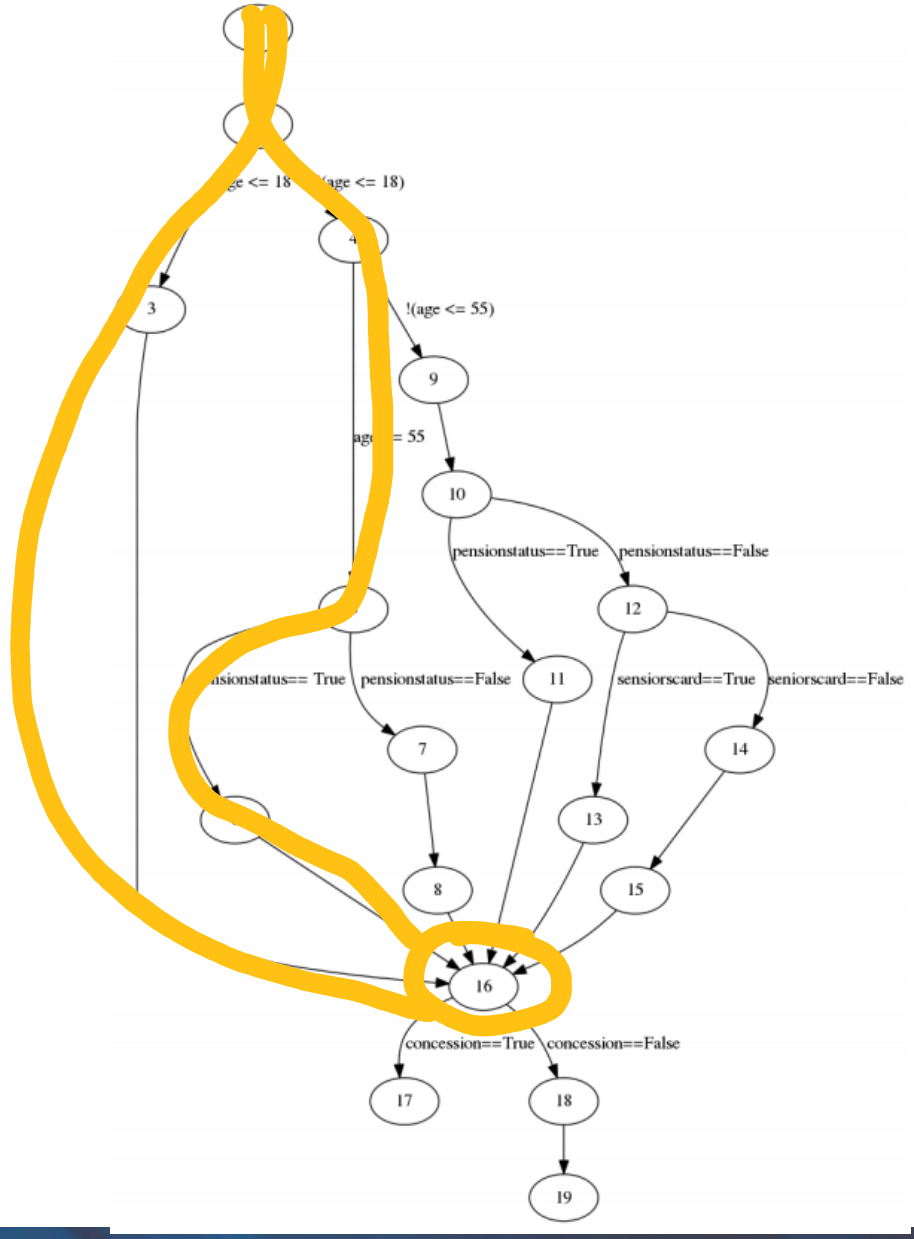
So What constraints we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
    - (age ≤ 18) ∧ (pension= T)



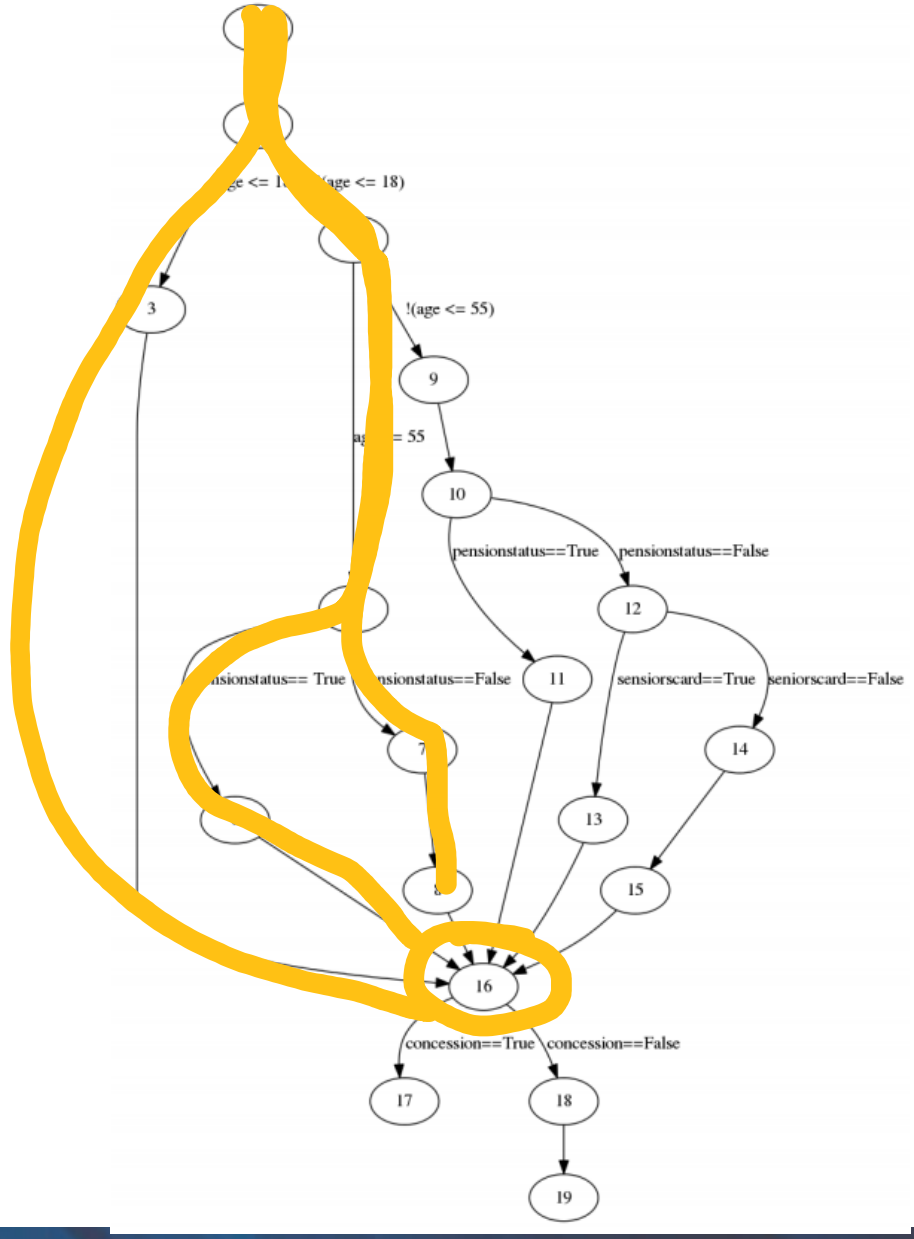
So What constraints we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
    - (age ≤ 18) ∧ (pension= T)
  - 1, 2, 4, 5, 6, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧(pension= T)



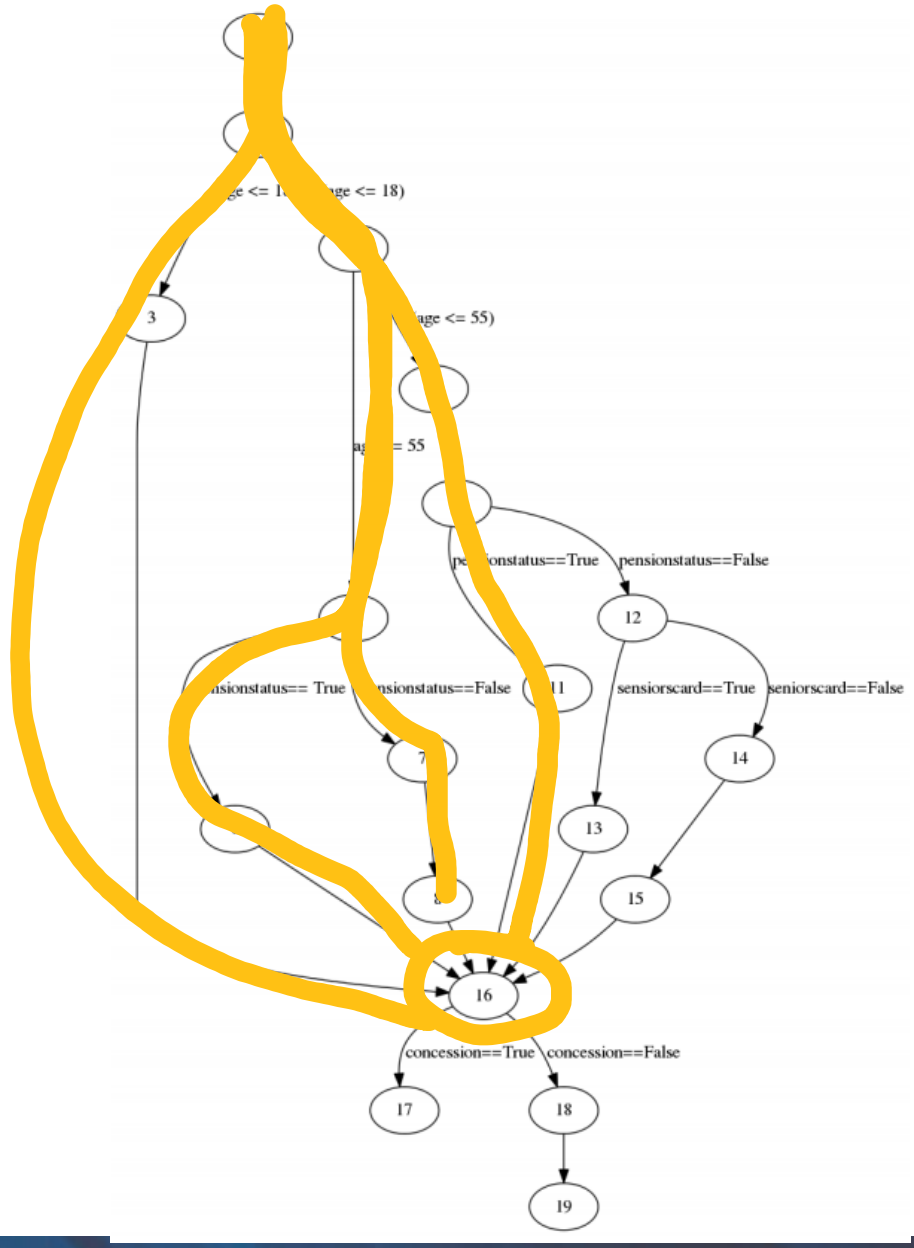
So What constraints we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
    - (age ≤ 18) ∧ (pension= T)
  - 1, 2, 4, 5, 6, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧(pension= T)
  - 1, 2, 4, 5, 7, 8, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧ (pension= F)



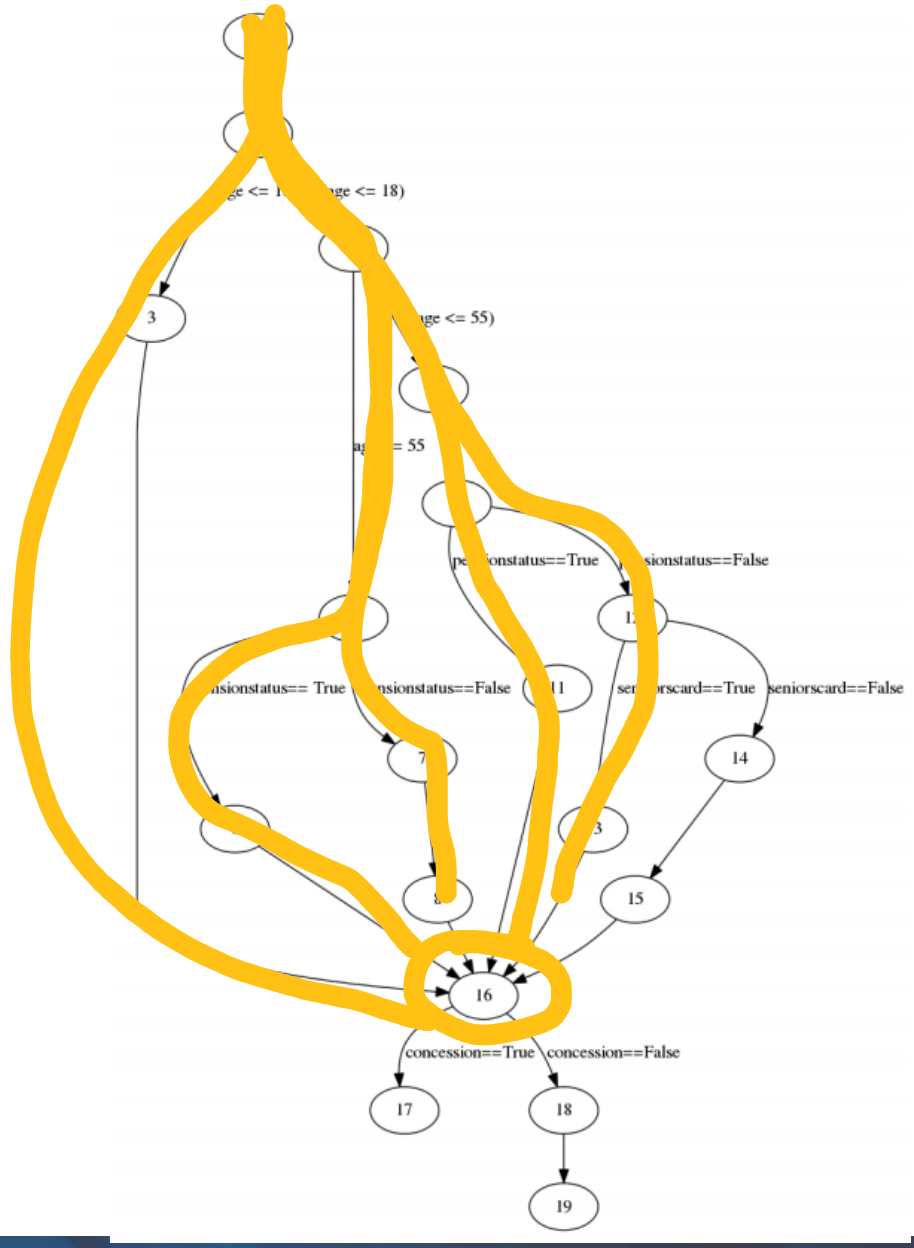
So What constraints we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
    - (age ≤ 18) ∧ (pension= T)
  - 1, 2, 4, 5, 6, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧(pension= T)
  - 1, 2, 4, 5, 7, 8, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧ (pension= F)
  - 1, 2, 4, 9, 10, 11, 16
    - ¬(age ≤ 18) ∧ ¬(age ≤ 55) ∧ (pension= T)



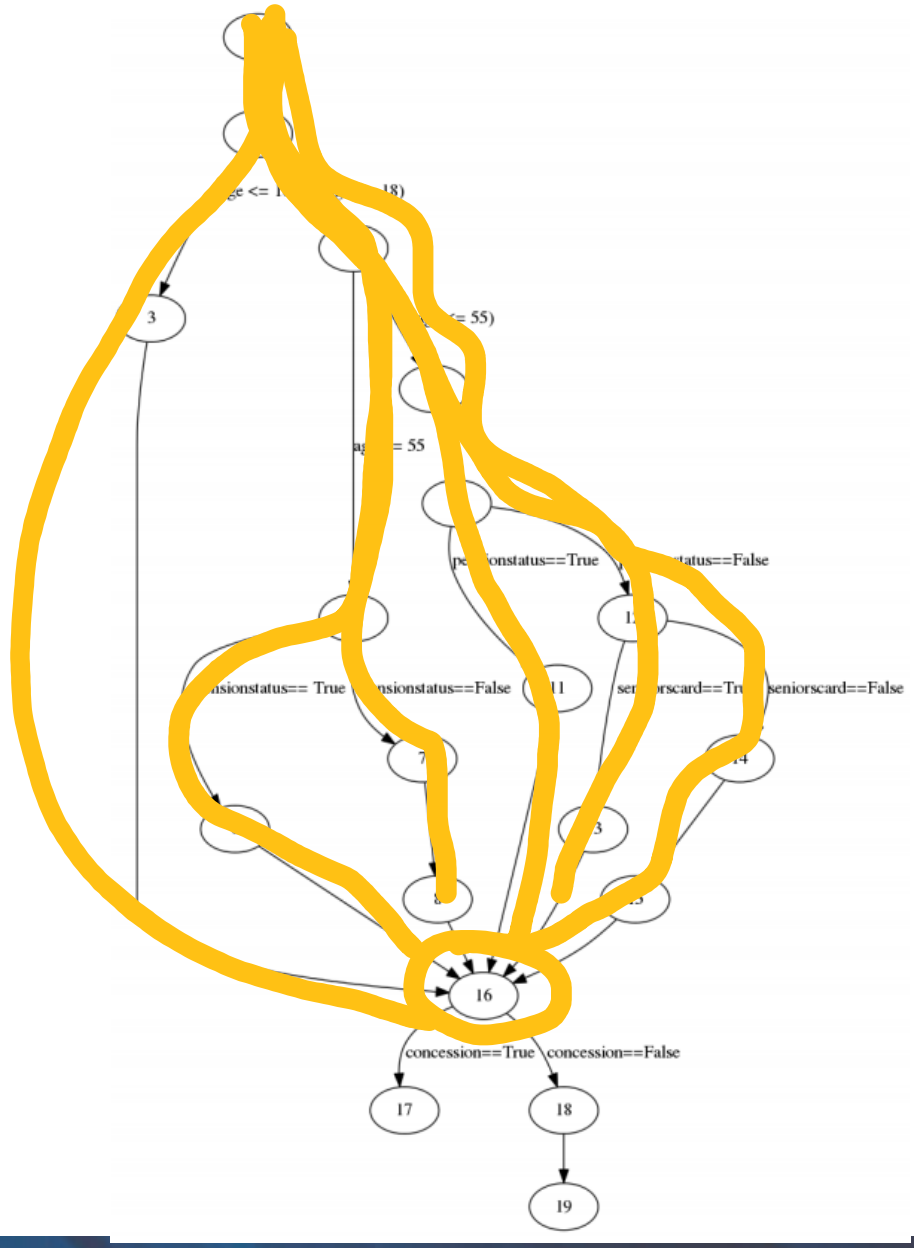
So What constraints we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
    - (age ≤ 18) ∧ (pension= T)
  - 1, 2, 4, 5, 6, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧(pension= T)
  - 1, 2, 4, 5, 7, 8, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧ (pension= F)
  - 1, 2, 4, 9, 10, 11, 16
    - ¬(age ≤ 18) ∧ ¬(age ≤ 55) ∧ (pension= T)
  - 1, 2, 4, 9, 10, 12, 13, 16
    - ¬(age ≤ 18) ∧ ¬(age ≤ 55) ∧ (pension= F) ∧ (seniors= T)



So What constraints we have?

- Let's consider the branch 16->18, the potential paths are:
  - 1, 2, 3, 16
    - (age ≤ 18) ∧ (pension= T)
  - 1, 2, 4, 5, 6, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧(pension= T)
  - 1, 2, 4, 5, 7, 8, 16
    - ¬(age ≤ 18) ∧ (age ≤ 55) ∧ (pension= F)
  - 1, 2, 4, 9, 10, 11, 16
    - ¬(age ≤ 18) ∧ ¬(age ≤ 55) ∧ (pension= T)
  - 1, 2, 4, 9, 10, 12, 13, 16
    - ¬(age ≤ 18) ∧ ¬(age ≤ 55) ∧ (pension= F) ∧ (seniors= T)
  - 1, 2, 4, 9, 10, 12, 14, 15, 16
    - ¬(age ≤ 18) ∧ ¬(age ≤ 55) ∧ (pension= F) ∧ (seniors= F)



MC/DC

- exercise each condition in a way that it can, independently of the other conditions, affect the outcome of the entire decision.
- every possible condition of each parameter must have influenced the outcome at least once.

```
def admission (degree, experience, character):
    if character and (degree or experience):
        print("Admitted")
    else:
        print("Rejected")
```

- Whether the applicant has a good character (true or false),
- Whether the applicant has a degree (true or false),
- Whether the applicant has experience in a field of work (true or false)

MC/DC - Example

- MC/DC will give?
  - Character = {1, 5}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Character = {1, 5} {2,6}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Character = {1, 5} {2,6} {3,7}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Character = {1, 5} {2,6} {3,7}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Degree = {}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Degree = {2,4}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Degree = {2,4}
  - ??

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Experience = {3,4}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Experience = {3,4}
  - ??

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Character = {1, 5} {2,6} {3,7}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Character = {1, 5} {2,6} {3,7}
  - Degree = {2,4}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- MC/DC will give?
  - Character = {1, 5} {2,6} {3,7}
  - Degree = {2,4}
  - Experience = {3,4}

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

MC/DC - Example

- So how many test?
  - Character = {1, 5} {2,6} {3,7}
  - Degree = {2,4}
  - Experience = {3,4}
- So T = {2,3,4,6} => 100% MC/DC
- N+1 tests = 4 Tests
- Which is better than $2^3 = 8$
- Created at Boeing and is required for aviation software.

Tests	Character	Degree	Experience	Decision?
1	T	T	T	T
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	F
6	F	T	F	F
7	F	F	T	F
8	F	F	F	F

Summary

- Condition coverage reports the true or false outcome of each condition and measure them independently of each other.
- Condition + Branch are done together.
- Every Path should be executed at least one.
- MC/DC
    - every condition shown to independently affect a decision outcome (by varying that condition only)
    - a condition independently affects a decision when, by flipping that condition and holding all the others fixed, the decision changes

QUESTIONS???