



Logic Coverage

Lecture Objectives:

- 1) Define the terms predicate and clause.
- 2) Explain the concept of short circuit evaluation
- 3) Define the concept of predicate coverage and clause coverage.
- 4) Explain the concept of combinatorial coverage.
- 5) Define active clause coverage.
- 6) Apply active clause coverage to create test cases for an appropriate clause.
- 7) Explain inactive clause coverage.
- 8) Define the concept of infeasibility.

How was your
weekends?

or "Break" 休息
"Too long" 17777
interesting!

Predicates

- An expression which evaluates to a boolean value \rightarrow true/false

- Boolean variables

- \neg The negation operator

- \wedge The And operator

- \vee The or operator

- \rightarrow The implication operator

- \oplus The exclusive or operator

- \leftrightarrow The equivalence operator

~~$a = b$~~ $a = b$

! true

$a \wedge b$

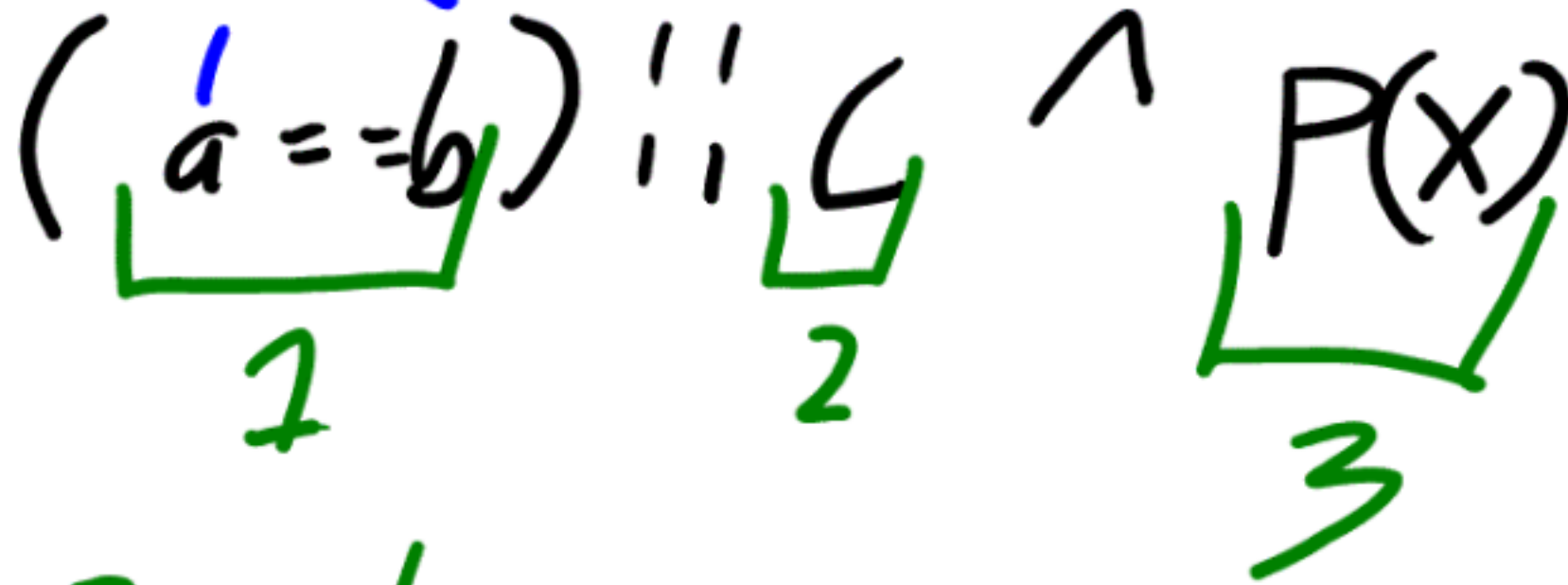
$a \vee b$

XOR

$a \wedge b$

- A clause is a predicate that does not contain any ^{boolean} logical operators

ints



check this

Clause

3 clauses

Testing

stop @ 1

8 clause _{about} \Rightarrow 2³ = 8

Order of operations

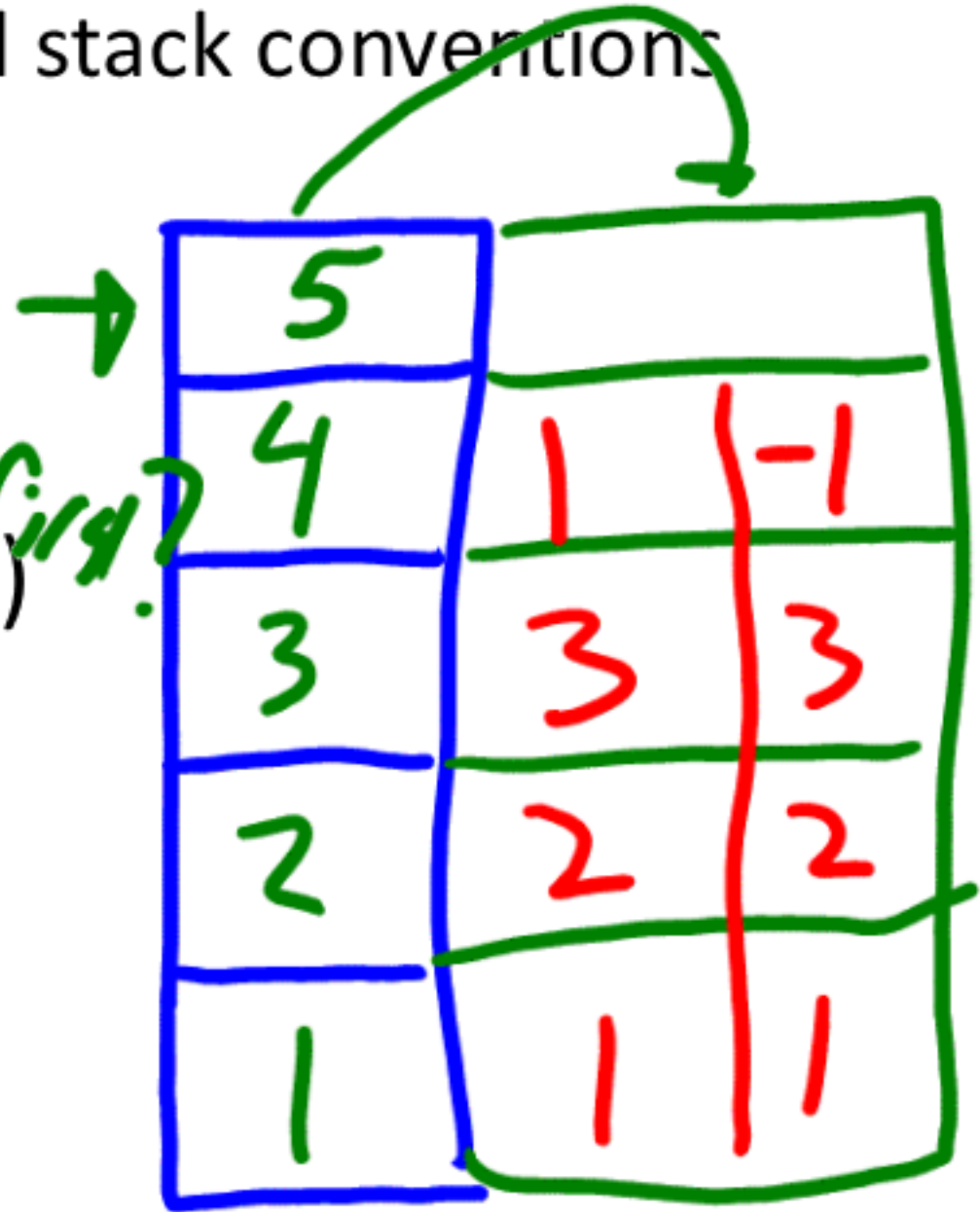
Challenges

- What is wrong with the following code?
 - Assuming normal stack conventions

- void push(int x);
- int pop();

- Push(pop()-pop())

*Which
expression?*



Short Circuit Evaluation (McCartyhy evaluation)

```
int denom = 0; Java  $\Rightarrow$  Compile  
if (denom && num/denom) { ... // error  
ensures that calculating num/denom  
never results in divide-by-zero error  
}
```

Short Circuit Evaluation (McCarthy evaluation)

```
int denom = 0; C
if (denom && num/denom) { ... //
ensures that calculating num/denom
never results in divide-by-zero error
}
```

↘ O skip the if
(Maybe)

```
if (denom)
{
  if (num / denom)
  {
  }
}
```



Short circuit evaluation

```
if (ishigh && (x==f(x)))
```

```
{  
.  
.  
.  
}
```

is high == 0

int f(int x)
{
 global x = x++;
 return x++;
}

Predicate Coverage

- Predicate Coverage:
 - two requirements:
 - p evaluates to true
 - p evaluates to false

Not very strong at detecting logical flaws.

A predicate

$$((a > b) \cup C) \cap p(x)$$

And

or

	a	b	c	p(x)
1	5	4	true	true
2	4	5	true	false
	4	6	false	false

Little mouse



Clause Coverage

- Clause Predicate Coverage: For each p in P , TR contains two requirements: p evaluates to true and p evaluates to false

Predicate: $((a > b) \cup C) \cap p(x)$

$A > b$ T F
 $C =$ T F
 $P(x)$ T F

	a	b	C	P(x)
1	5	4	t	t
2	5	6	f	f

Clause Coverage

- Predicate Coverage: For each p in P , TR contains two requirements: p evaluates to true and p evaluates to false

CL not having PC

Predicate: $((a > b) \cup C) \cap p(x)$

Clause Coverage: Yes

Predicate Coverage: No

	a	b	C	$p(x)$	
A	5	4	true	False	F
B	4	5	false	True	F

Problems with PC and CC

- PC does not fully exercise all the clauses, especially in the presence of short circuit evaluation
- CC does not always ensure PC
- That is, we can satisfy CC without causing the predicate to be both true and false
- This is definitely not what we want !

- Predicate: $(a \cup b) \cap c$ — Boolean

Combinatorial Coverage & Test Coverage

a	b	c	R
F	F	F	F
F	F	T	F
F	T	F	F
F	T	T	T
T	F	F	F
T	F	T	T
T	T	F	F
T	T	T	T

	a		b	
	TT	TF	FT	FT
T	T	T	F	T
F	F	F	F	F

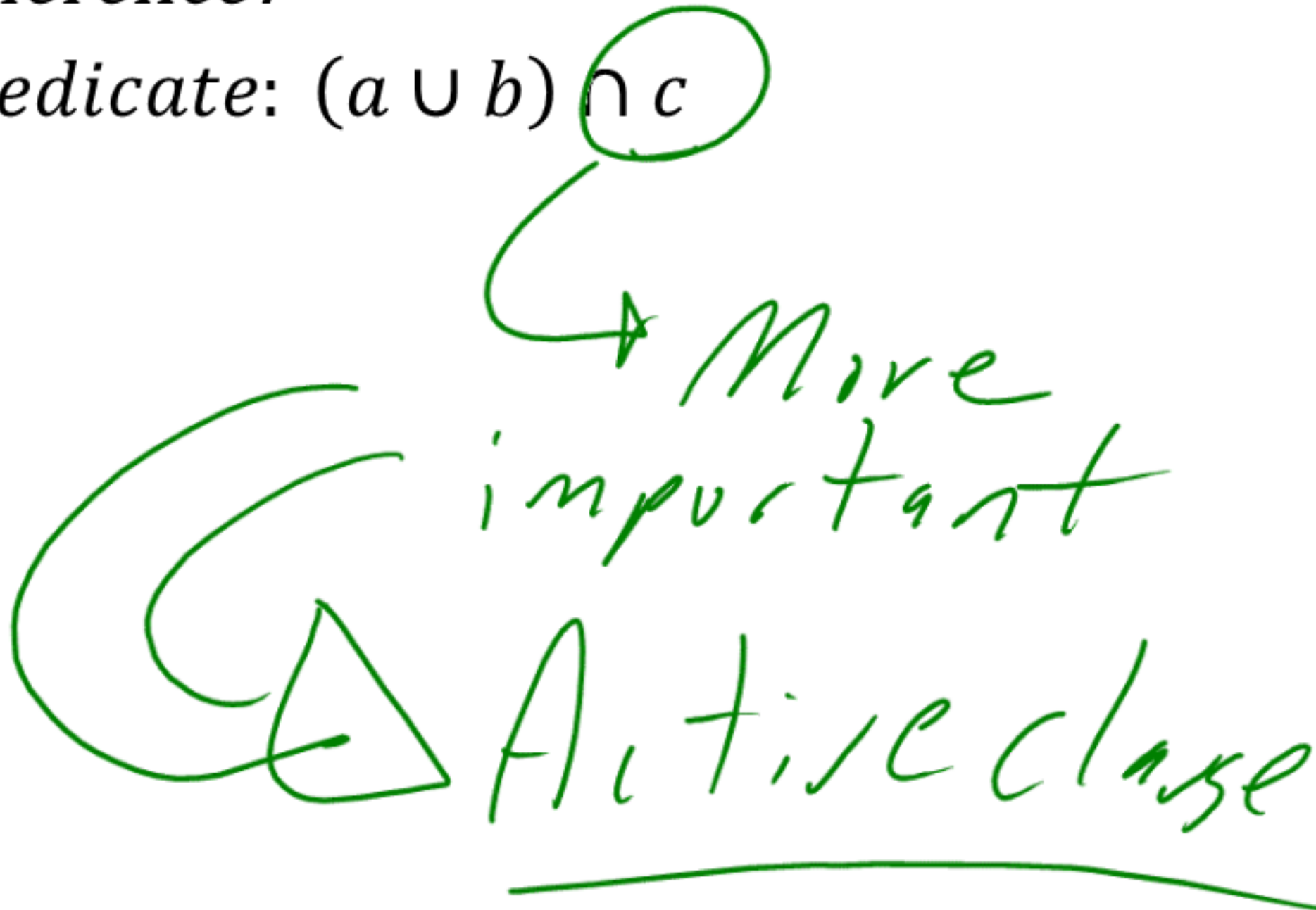
SE2832 Introduction to Software Verification



$$(a \cap c) \cup (b \cap c)$$

Active Clause

- Which clause makes a bigger difference?
- Predicate: $(a \cup b) \cap c$



Active Clause

- *Which clause makes a bigger difference?*
- *Predicate: $(a \cup b) \cap c$*
- ***To really test the results of a clause, the clause should be the determining factor in the value of the predicate***

MSOE Deans List and Honor Roll

- Deans List
 - Must have at least 30 credits completed
 - Must have a cumulative GPA greater than 3.2
 - Must be a full time student
 - ✂ Must have a term GPA > 2.0
- High Honors Deans List
 - Must have at least 30 credits completed
 - Must have a cumulative GPA greater than 3.7
 - ✂ Must have a term GPA > 2.0
 - Must be a full time student
- Honors List
 - Must have a cumulative GPA > 2.0 but < 3.20
 - Must be a full time student
 - Must have a term GPA > 3.20 ✂

Code

```
public static void printHonorRollStatus(double cumulativeGPA,  
    double termGPA, int creditsCompleted, boolean fullTimeStatus) {  
    // Determine if the student is on the deans list.  
    if ((creditsCompleted > 30) && (cumulativeGPA > 3.20)  
        && (fullTimeStatus == true) && (termGPA > 2.0)) {  
        System.out.println("You are on the dean's list.");  
    } else if ((creditsCompleted > 30) && (cumulativeGPA > 3.70)  
        && (fullTimeStatus == true) && (termGPA > 2.0)) {  
        System.out.println("You are on the high honors dean's list.");  
    } else if ((creditsCompleted > 30) && (cumulativeGPA > 2.0)  
        && (fullTimeStatus == true) && (termGPA > 3.2)) {  
        System.out.println("You are on the honor list.");  
    } else {  
        System.out.println("You are on a hit list somewhere.");  
    }  
}
```

All Andel ...

259

Another Example

$2^4 \Rightarrow 16$

```

public double schoolClosingProbability(int temperature, int windspeed,
    int inchesOfSnowOnGround, boolean isAnotherLargeSchoolNearbyClosed) {
    if ((temperature < -20) && (windspeed > 20)
        && (inchesOfSnowOnGround > 12)) {
        return HIGH_PROBABILITY;
    }
    if (((inchesOfSnowOnGround > 12) && (windspeed > 10))
        || (isAnotherLargeSchoolNearbyClosed == true)) {
        return HIGH_PROBABILITY;
    } else {
        return LOW_PROBABILITY;
    }
}

```

Not here

temp wind inch of snow nearby

