KEY

Notes for...

Evaluating More Complex Boolean Expressions. One tool that can be used to evaluate a Boolean is expression is a Truth Table.

Truth Table for the AND, OR, and NOT logical operators. P is expression 1 Q is expression 2

P	Q	P && Q	P || Q	!P	(P && Q) || !P
True	True	T	T	F	T || F T
True	False	F	T	F	F || F F
False	True	F	T	T	F || T
False	False	F	F	T	F || T T

Operator Precedence

! unary operator (highest)

&&

|| (lowest)

Evaluate the following.

! (!B) is equivalent to B

Look at last page of notes for Unit 3 precedence chart

① insert values

② !F then b == c then && then ||

① !F && F == T || F

②b T && F || F

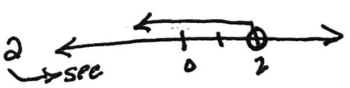
②c F || F

②a T && F == T || F

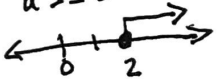
!a && b == c || d where c is TRUE and a, b, d are FALSE F

Some Useful Boolean Equivalences

example a < 2 → see



!(a<2) rest of line! a >= 2



Let a and b represent algebraic value

!(a < b) is equivalent to a >= b // not a < b

!(a == b) is equivalent to a != b // not a equals b

!(a >= b) is equivalent to a < b

c is true, a,b,d false answer the following carefully:

a||b!=c||d==b t

!(a==b) && b || c==d f

a || b || d && c f

!(a || b || d) && c t

!(a || b || d) && !c f

f || f != t || f == f ⇒ f || t || t ⇒ t

f && f || f

!t && f || f

f && t

!f && t → t && t

!f && !t

t && f

f || f && t

f → f || f

a || d && c f

!f && t

!(a || d) && c t

(a&&d) || true t

f || t

false || !d t

true && !d t

t && !f

t && t

Let p and q and r represent boolean values

!(p && q) is equivalent to !p || !q

!(p || q) is equivalent to !p && !q

DeMorgan's Laws

THEY DO:

Some sample problems.

1) Simplify !(3 <= x && x <= 5)

!(x >= 3 && x <= 5) → x < 3 || x > 5

2) For what values of p and q is this expression TRUE?

(p && q) || !(p || q)

p && q || (!p && !q)

both true or both false

T && T !F && !F See?

3) What value(s) will this print?

System.out.println(!(x > 10 || x < 20));

x <= 10 && x >= 20 no values ever, so false

4) Simplify !(x != 0 || y != 0)

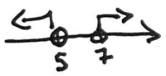
x == 0 && y == 0

5) Simplify !(x < 5) || !(x >= 9)

x >= 5 || x < 9

since or, this will always be true!

6) When is this expression TRUE?



x > 7 && x < 5 || x > 10

F || x > 10 so x > 10 only

Short-Circuit Evaluation

The JVM sometimes knows the value of a Boolean expression before it has evaluated all of its parts. For instance, in the expression

(p && q)

if p is False, then the whole expression must be False (and the JVM will not bother to evaluate q).

In the same way, in this expression:

(p || q)

if p is true, then the whole expression must be true (and the JVM will not bother to evaluate q).

1)	If x has a value of zero, what happens? runtime error 10/0 Arithmetic exception / by zero	int x;
		// x is assigned a value
		if (x > 4 || 10.0 / (x - 5) > 0)
		System.out.println("A");
		else if (10 / x < 0 || x > 10)
		System.out.println("B");
		else if (x > 5 && 1.0/(x - 3) > 0)
		System.out.println("C");
		else
		System.out.println("D");

2)	If x has a value of 5, what happens? 5 > 4 so true short circuit	
3)	If x has a value of 3, what happens? D	

10/0 runtime error

1.0 no runtime error bc double division

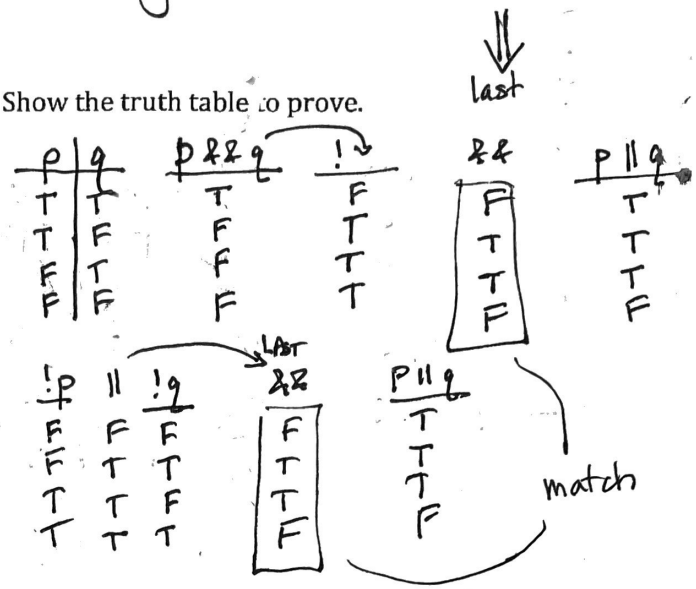
Higher precedence: && or || you must know this! && has higher precedence! and comes before or

if you struggle on this page do not worry, this is more than you will need to know ☺

Writing equivalent statements. Show the truth table to prove.

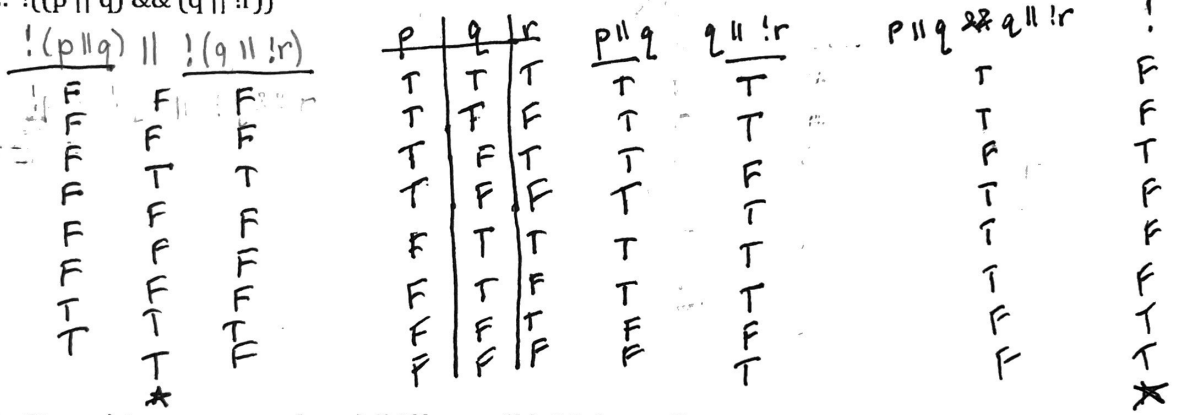
(!p || !q) && (p || q)

1. !(p && q) && (p || q)



parenthesis show that && should be last.

2. !((p || q) && (q || !r))



3. Given this statement: (a < c) || !((c == a*b) && (c < a))

what are possible values of a, b, c that will make this true? (ie.. a>c is false, etc)

will a truth table help? or assign values to parts of the statement?

if a < c is true, short circuits to true

or if c != a*b or c >= a, also works

4. Rewrite this statement more explicitly to show order:

!A && B || C

(!A && B) || C

6) Complete the Truth Table

P	Q	R	Q && R	P || (Q && R)
True	True	False	F	T
False	False	True	F	F

7) Complete the Truth Table

P	Q	R	P && Q	P && R	(P && Q) || (P && R)
True	False	True	F	T	T
False	False	True	F	F	F

8) List all the values of p, q, and r (where p, q, and r are boolean variables) that make this expression TRUE. Can use a truth table.

!p && (q || r)

if p is false and q=t r=t
q=t r=f
q=f r=t

9) List all the values of x that make this expression TRUE. Assume x is an int.

!(x > 5 && x <= 15)

x<=5 || x>15 $(-\infty, 5] \cup (15, \infty)$

10) Simplify (where p and q are boolean variables)

!(!p && q) p || !q

11) Simplify (where p and q are boolean variables)

!(p || !q) !p && q

12) If a boolean expression involved four different independent boolean variables/expressions (e.g. p, q, r, and s), how many different rows (i.e. combinations of values) would the Truth Table contain?

2 var $2^2 = 4$ rows 3 var $2^3 = 8$ rows $2^4 = 16$ rows

13) The boolean expression !A && B || C is equivalent to

a) !A && (B || C)
b) ((!A) && B) || C
c) (!A) && (B || C)
d) !(A && B) || C
e) !(A && B || C)

14) Assume that a and b are integers. The boolean expression

!(a <= b) && (a*b > 0) a > b && a*b > 0 → a & b both >0 or both <0

will always evaluate to true given that

a) a = b no bc what if a=0=b
b) a > b a>b -2 > -4 works 0 > -1 doesn't
c) a < b
d) a > b and b > 0
e) a > b and b < 0

15) Given that a, b, and c are integers, consider the boolean expression

(a < b) || !((c == a*b) && (c < a)) → (a<b) || ((c != a*b) || (c ≥ a))

Which of the following will guarantee that the expression is true? if a<b is true

a) c < a is false
b) c < a is true
c) a < b is false
d) c == a * b is true
e) c == a * b is true, and c < a is true

16) If *c* and *d* are boolean variables, which one of the answer choices is NOT equivalent to the following expression?

(c && d) != (c || d)
t when same
f when diff

a) (c && !d) || (!c && d)
b) (c || d) && (!c && !d)
c) (c || d) && (!c || !d)
d) (c || d) && !(c && d)
e) c != d

17) Select the TRUE statement.	int x = 0;
	boolean alive = true;
	if (alive || 50 / x != 0)
	  System.out.println("ok");

a) there will be a compiler error due to 50/x b) there will be a run-time error due to 50/x c) the code will run and "ok" will be displayed. short circuit d) the code will run and "ok" will NOT be displayed.	

18) m1 and m2 must each return a boolean.	True	False	if (m1() && m2())
			  // code A
			else
			  // code B

19) If m1 returns true, then method m2 will not be called.	True	False	
20) If m1 returns false, then method m2 will not be called.	True	False	

↳ short circuit