12.5 DeMorgan's Law

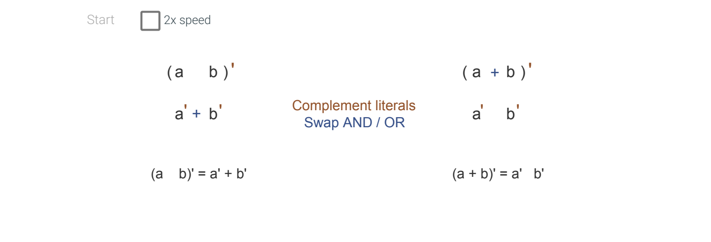
Basics

DeMorgan's Law is a Boolean algebra property for complementing an expression, coming in two forms: (a + b)' = a'b', and (ab)' = a' + b'. Each literal is complemented, and ANDs / ORs swap.

Table 12.5.1: DeMorgan's Law.

Property	Name	Description
(a + b)' = a'b'	DeMorgan's Law (for OR)	Each literal complemented, ORs become ANDs.
(ab)' = a' + b'	DeMorgan's Law (for AND)	Each literal complemented, ANDs become ORs.

PARTICIPATION ACTIVITY 12.5.1: DeMorgan's Law.



Start □ 2x speed

PARTICIPATION ACTIVITY 12.5.2: DeMorgan's Law.

1) (de)' = ?
   - ○ d'e'
   - ○ d' + e'
   - ○ d + e

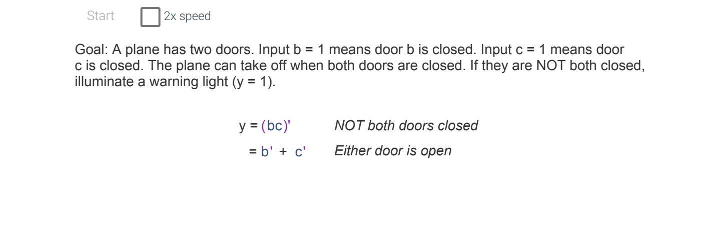
2) (f + g)' = ?
   - ○ f'g'
   - ○ fg
   - ○ f' + g'

3) (abc)' = ?
   - ○ a'b'c'
   - ○ a' + b' + c'
   - ○ DeMorgan's Law does not apply.

4) A play's cast has an understudy (a substitute). Which is equivalent to saying: If not all cast members show up, the understudy participates.
   - ○ If any cast member does not show up, the understudy participates.
   - ○ If all cast members show up, the understudy participates.

Examples

PARTICIPATION ACTIVITY 12.5.3: DeMorgan's Law for (ab)': Plane doors.



Start □ 2x speed

Goal: A plane has two doors. Input b = 1 means door b is closed. Input c = 1 means door c is closed. The plane can take off when both doors are closed. If they are NOT both closed, illuminate a warning light (y = 1).

PARTICIPATION ACTIVITY 12.5.4: DeMorgan's Law for (a + b)': Guards.

Start 2x speed

Goal: A building is protected by two guards. Input d = 1 means guard d is present. Input e = 1 means guard e is present. As long as at least one guard is present, all is good. If that is NOT the case, notify the manager (y = 1).

$y = (d + e)'$ *NOT either guard present*

$= d'e'$ *Both guards absent*

PARTICIPATION ACTIVITY 12.5.5: DeMorgan's Law: Basic examples.

Consider the examples above.

1) The plane can take off if both doors are _____.

○ closed

○ not closed

2) The plane's warning light illuminates if it ____ the case that both doors are closed.

○ is

○ is NOT

3) $y = (ab)'$ _____ in sum-of-products form.

○ is

○ is not

4) $y = a' + b'$ _____ in sum-of-products form.

○ is

○ is not

5) For the guards examples, all is good if any guard is present. The expression for that situation is _____.

○ $d + e$

○ $(d + e)'$

6) If NOT the case that any guard is present, the manager should be notified. The expression for notifying the manager is _____.

○ $d + e$

○ $(d + e)'$

7) Which is in sum-of-products form?

○ $(d + e)'$

○ $d'e'$

More complex cases

The laws apply to expressions beyond just literals. Examples:

- $(a'b)' = a'' + b'$, so $a + b$.
- $(ab + c)' = (ab)'c'$. The law can then be applied again: $(a' + b')c'$. Multiplying out yields $a'c' + b'c'$.

PARTICIPATION ACTIVITY 12.5.6: DeMorgan's Law for more complex examples.

1) $(d'e)' = ?$
Simplify answer.

Check Show answer

2) $(f' + g')' = ?$

Check Show answer

3) $((d + e)f)' = ?$
Apply DeMorgan's Law twice.

Check Show answer

4) $(ab + c)' = (?)c'$
Apply DeMorgan's Law twice. (Only type the ? part).

( ) c'

Check Show answer

DeMorgan's Law in programming

DeMorgan's Law is not just used in digital design, but also by computer programmers. Computer programs use expressions to control decisions. Ex: A program may proceed as long as the input is not 3 or 5. The programmer may write: If NOT(x==3 OR x==5) then proceed.

To simplify the expression, the programmer may apply DeMorgan's Law: If (x != 3 AND x != 5) then proceed. (!= means not equal). The NOT was applied to the two items (so == became != in two places), and the OR was changed to AND. The resulting expression may be simpler to read.

Provide feedback on this section