



A8.5 Simplifying expressions

Boolean algebra and logic circuits · A level · OCR H446 1.4.3, AQA 7517

4.6.5.1, Eduqas A500QS 1.2 · about 25 min

Name _____

Class _____

Date _____

What this lesson is about

A method for simplifying by algebra, worked exam-style examples, and checking the result.

- What "simpler" means
- A method
- Worked examples
- Checking each step by machine
- Simplifying a robot's condition

Questions

5 marks in all

1. Simplify $(A \vee B) \wedge (A \vee \neg B)$. [1 mark]

- ☐ A A
- ☐ B B
- ☐ C $A \vee B$
- ☐ D 1

2. Simplify $\neg(A \vee B) \vee (\neg A \wedge B)$. [1 mark]

- ☐ A $\neg A$
- ☐ B $\neg B$
- ☐ C $\neg A \wedge \neg B$
- ☐ D $A \vee B$

3. Simplify $(A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge \neg B)$. [1 mark]

- ☐ A A
- ☐ B $A \wedge B$
- ☐ C $A \wedge C$
- ☐ D $B \vee C$

4. Put the steps simplifying $A \cdot B + A \cdot \bar{B} + \bar{A} \cdot B$ in order. [1 mark]

Number the lines 1 to 4 to put them in the right order.

- ____ $A + \bar{A} \cdot B$
- ____ $A \cdot 1 + \bar{A} \cdot B$
- ____ $A + B$
- ____ $A \cdot (B + \bar{B}) + \bar{A} \cdot B$

5. A student writes $\neg A \vee \neg B = \neg A \wedge \neg B$ as a step. What is wrong?

[1 mark]

- ☐ A The operator was changed without the bar over the whole that De Morgan needs
- ☐ B Nothing, it is De Morgan's law
- ☐ C Commutation was used instead of association
- ☐ D The NOTs should have cancelled

The task: simplify, then prove it

The function `original(a, b, c)` computes $Q = \neg(A \vee \neg B) \vee (A \wedge B) \vee (A \wedge \neg B \wedge C)$. Its inputs are bits (0 or 1) and it returns 0 or 1. Simplify Q on paper, then write `simplified(a, b, c)`, taking and returning the same, as a single `return` line that uses at most two of `and`, `or` and `not` in total. It may not call `original` or use the bitwise operators `&`, `|`, `^` or `~`. Print eight lines, one per row in binary order (A outermost, then B, then C), in exactly this form, with every value 0 or 1: `A=0 B=1 C=0 original=1 simplified=1`

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def original(a, b, c):
    return int((not (a or not b)) or (a and b) or (a and not b and c))

def simplified(a, b, c):
    return 0
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/a8-5-simplifying-expressions/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Simplify $(A \wedge B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (A \wedge \neg B)$ by algebra. Check it with `same`, extended to three inputs.
2. How many two-input gates does Q need as written, and how many after simplifying? Count a NOT as a gate.
3. Write the working for the task as a `steps` list and check every step by machine.