

3.2 Boolean algebra and Karnaugh maps

Name: _____ Class: _____ Date: _____ **Total: 23 marks**

KEY WORDS Boolean algebra • De Morgan’s theorem
• Karnaugh map • don’t-care

Objective

Simplify Boolean expressions with the standard laws, minimise functions of 3–4 variables on Karnaugh maps (using don’t-cares), and verify every answer by exhaustive evaluation over all input rows.

1 Key knowledge and worked examples

■ The laws that do the work

AND/OR/NOT with the laws: commutative, associative, distributive – plus the two that actually shorten expressions: **absorption** $A + A \cdot B = A$, and **De Morgan’s theorem** $(A \cdot B)' = A' + B'$, $(A + B)' = A' \cdot B'$ – break the bar, change the sign. XOR $A \oplus B$ is “exactly one”: $A'B + AB'$.

Worked example. Simplify $AB + AB'$: factor $A(B + B') = A \cdot 1 = A$. The check: both expressions evaluate identically over all four rows of (A, B) ✓.

A \ BC	00	01	11	10
0	0 m0	1 m1	1 m3	0 m2
1	0 m4	1 m5	1 m7	0 m6

group of four: only C stays 1, so $F = C$

■ Karnaugh maps: adjacency is simplification

The **Karnaugh map** lays out minterms so adjacent cells differ in one variable (the row/column order is Gray code: 00, 01, 11, 10). Rectangular groups of 2^k cells read as one product term with k variables eliminated. Take the **LARGEST** groups; every 1 must be covered.

Worked example. $F = \sum m(1, 3, 5, 7)$ of variables A, B, C: these are exactly the odd-index rows, i.e. every row where $C = 1$. The map shows a clean two-column group $\rightarrow F = C$. Exhaustive check: C matches all 8 rows ✓.

■ Don’t-cares enlarge groups

A **don’t-care** (input combination that cannot occur, or whose output does not matter) may be included in a group or not – whichever makes the group bigger. It never **STARTS** a group alone.

Worked example. $F(A, B, C) = \sum m(2, 3, 6)$ with don’t-cares $d(7)$: the 1s sit in cells 2,3,6; including 7 in the group $\{2,3,6,7\}$ reads B – two literals saved. Check: B matches rows 2, 3, 6 ✓ (row 7 is a don’t-care, so its value is unconstrained).

2 Practice

2.1 Simplify each: $A + A' \cdot B$, $A(A + B)$, $AB + A'B' + AB'$ – verify one by exhaustion. [3]

2.2 Apply De Morgan to $(A + B')'$ and to $(A' \cdot B)'$, and verify both by evaluating all four rows. [4]

2.3 Minimise $F(A, B, C) = \sum m(0, 2, 4, 6)$ (the even rows) on a Karnaugh map; state the one-literal answer and check by exhaustion. [3]

3 Exam-style questions

3.1 $F(A, B, C, D) = \sum m(0, 1, 2, 4, 5, 8, 9, 10)$ with don't-cares $d(3, 7, 11)$.

Draw (describe) the Karnaugh map grouping, use the don't-cares to minimise, and give the minimal SOP. Verify by evaluating your expression and the minterm list on all 16 rows (including the don't-care rows' freedom). [7]

3.2 A door-alarm sounds ($F = 1$) when the system is armed (A) and either a sensor trips (S) OR the door is ajar (D) while disarmed-on-purpose is false ($P = 0$); plus it always sounds when the test input (T) is held.

Write F in SOP from the description, simplify using the laws (name each law used), and verify the simplification exhaustively over the 32 rows. [6]

Solutions

2.1

- $A + A'B = (A + A')(A + B) = A + B$.
- $A(A + B) = A$ (absorption).
- $AB + A'B' + AB' = A(B + B') + A'B' = A + A'B' = A + B'$ (distributed, then the 2.1 identity).

2.2

- $(A + B')' = A' \cdot B$ (De Morgan, double-negation on B).
- $(A' \cdot B)' = A + B'$.
- Exhaustive check table shows both pairs identical over 00, 01, 10, 11 ✓.

2.3

- The even rows share $C = 0$... wait: rows 0,2,4,6 all have $C = 0$? Row 0 = 000, 2 = 010, 4 = 100, 6 = 110 – yes, all have $C = 0$.
- The group covers all cells with $C = 0$: $F = C'$.
- Exhaustive check over 8 rows ✓.

3.1

- Map: 1s at 0,1,2,4,5,8,9,10; don't-cares 3,7,11.
- Group 1: cells 0,1,8,9 (column pair, all have $B = 0$, D varying... reading: A varies, $B = 0$, $C = 0 \rightarrow$ term $B'C'$).
- Group 2: cells 0,2,8,10 $\rightarrow B'D'$.
- Group 3: cells 4,5 (+ don't-cares) $\rightarrow AB'$... reading cells 4 (0100), 5 (0101): $A = 0$, $B = 1$, $C = 0 \rightarrow$ term $A'BC'$.
- Minimal SOP: $F = B'C' + B'D' + A'BC'$ (7 of the 8 minterms directly; cell 0 covered twice, harmless).
- Verification: evaluating $B'C' + B'D' + A'BC'$ over 16 rows returns 1 exactly on $\{0,1,2,4,5,8,9,10\}$ and any value on $\{3,7,11\}$ ✓.

3.2

- SOP from description: $F = A \cdot (S + D \cdot P') + T$.
- Distribute: $AS + ADP' + T$ – three product terms; no further reduction applies (the terms share no single variable).
- Laws used: distributive (expanding), identity $A \cdot 1$; the T term is already a single literal.
- Exhaustive check over A, S, D, P, T (32 rows): original $A(S + DP') + T$ equals $AS + ADP' + T$ on every row ✓.