# FINDING FORMULA

Kira Clements, University of Bristol

# DISJUNCTIVE NORMAL FORM

| A | B | C | ? |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

We can look at each row that gives a 1 output, ∧ the input variables (¬ the input variable when it has a 0 value), and then ∨ each of the rows…

$? \equiv (\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge \neg B \wedge C)$

This gives us the formula in **disjunctive normal form (DNF)** i.e. the disjunction (∨) of one or more conjunctions (∧) of one or more variables.

¬ should only precede a variable

*But doesn't this look really complicated?*

# CONJUNCTIVE NORMAL FORM

Alternatively, we can find a formula in **conjunctive normal form (CNF)** i.e. the conjunction (∧) of one or more disjunctions (∨) of one or more variables.

| A | B | C | ? |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

We look at each row that gives a 0 output, ∧ the input variables (¬ the input variable when it has a 0 value), and then ¬ ∧ each of the rows…

$$? \equiv \neg(\neg A \wedge B \wedge C) \wedge \neg(A \wedge B \wedge \neg C) \wedge \neg(A \wedge B \wedge C)$$

In this case, we get a smaller formula (as there are less 0's in the output) but it is not in CNF yet.

# DE MORGAN’S LAWS

To achieve CNF, we need to only have disjunctions ($\lor$) within each bracketed expression and only have negations ($\neg$) preceding individual variables, rather than a whole expression. This can be achieved using De Morgan’s laws!

$\neg(\neg A \land B \land C)$

$\neg(A \land B \land \neg C)$

$\neg(A \land B \land C)$

**DE MORGAN**

$$\neg(A \land B) \equiv \neg A \lor \neg B$$
$$\neg(A \lor B) \equiv \neg A \land \neg B$$

Swap $\land \backslash \lor$ and $\neg$ variables

$(A \lor \neg B \lor \neg C)$

$(\neg A \lor \neg B \lor C)$

$(\neg A \lor \neg B \lor \neg C)$

Using De Morgan’s on each of our three terms from the previous slide, we get the following logical expression:

$$(A \lor \neg B \lor \neg C) \land (\neg A \lor \neg B \lor C) \land (\neg A \lor \neg B \lor \neg C)$$

This is now an expression in CNF that represents the previously given truth table.

# DISTRIBUTIVITY

Now we have 2 methods of finding formula from a given truth table, our next task is to look at how we can simplify these - the **distributivity** rule is ideal for this!

| DISTRIBUTIVITY | Analogous to factorising in mathematics, by taking a common variable out of each term |
|---|---|
| $(A \land B) \lor (A \land C) \equiv A \land (B \lor C)$<br>$(A \lor B) \land (A \lor C) \equiv A \lor (B \land C)$ | |

For example, taking our CNF expression from the previous slide:

$$(A \lor \neg B \lor \neg C) \land (\neg A \lor \neg B \lor C) \land (\neg A \lor \neg B \lor \neg C)$$

$$\neg B \lor ((A \lor \neg C) \land (\neg A \lor C) \land (\neg A \lor \neg C))$$

$$\neg B \lor ((A \lor \neg C) \land (\neg A \lor (C \land \neg C))$$

$$\neg B \lor ((A \lor \neg C) \land \neg A)$$

$$\neg B \lor ((A \land \neg A) \lor (\neg C \land \neg A))$$

$$\neg B \lor (\neg C \land \neg A)$$

C ∧ ¬C would always be false and ¬A ∨ 0 has the same truth value as ¬A

Distributivity is sometimes needed to expand so variables can be cancelled

# TRUTH TABLES

We can see how the boolean expression we generated links with the truth tables.

| A | B | C | ¬B ∨ (¬A ∧ ¬C) |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

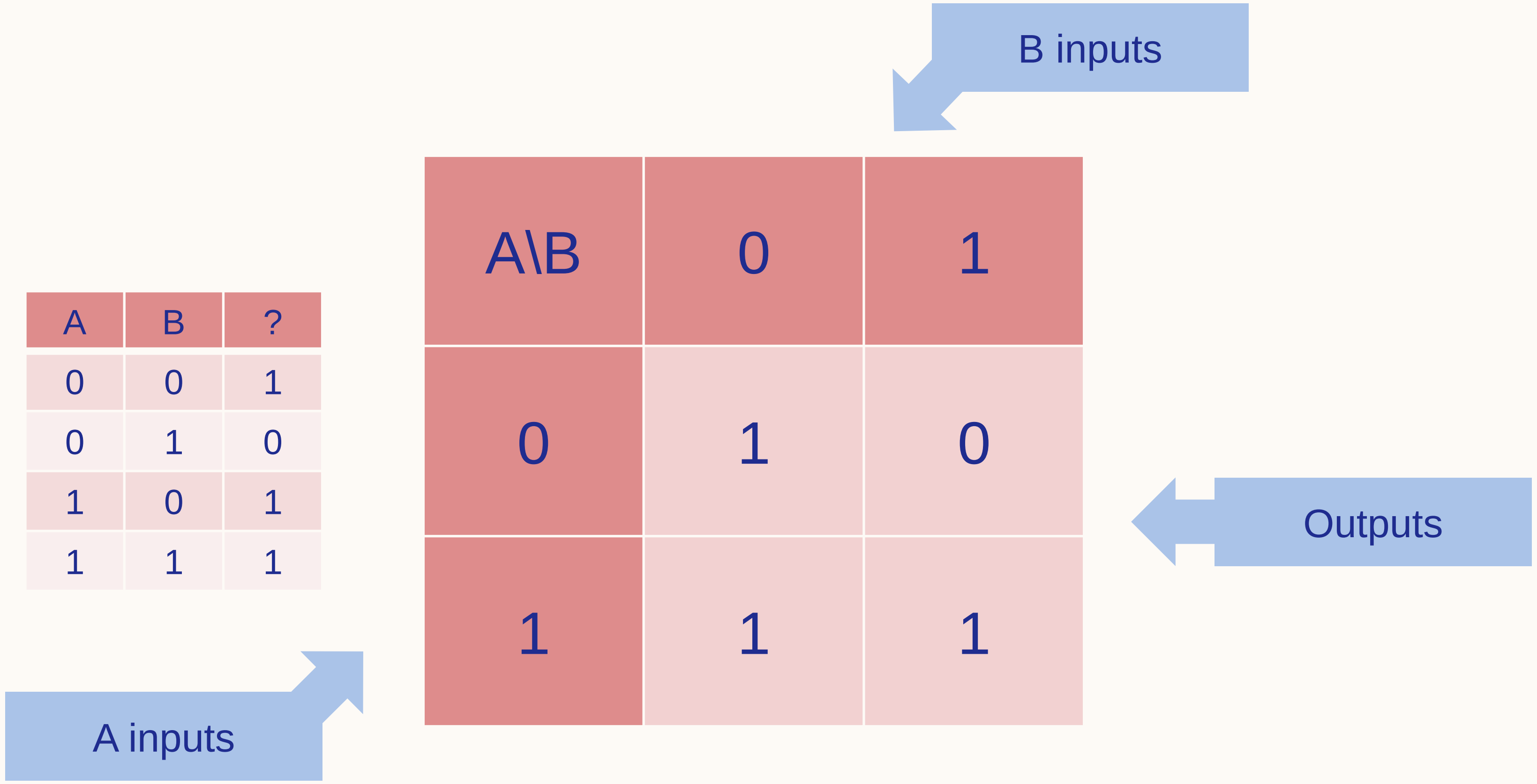
De Morgan’s laws and distributivity can also be useful for proving logical equivalence, as if you can manipulate one formula into another, then they are logically equivalent.

However, one flaw with these rules is that it can be hard to determine when the simplest form has been reached.

# KARNAUGH MAPS

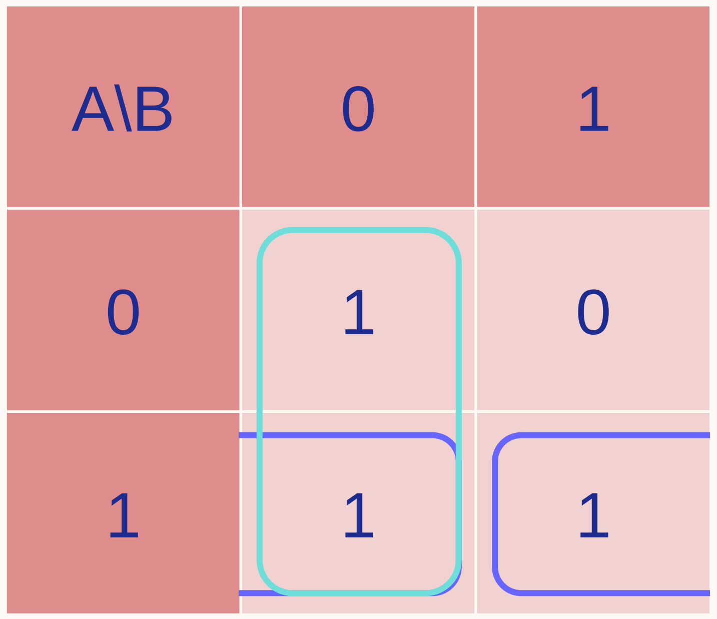
Karnaugh maps are a useful method of identifying a minimal form for a logical expression.

| A | B | ? |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



| A\B | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |

# KARNAUGH MAPS



| A\B | 0 | 1 |
|---|---|---|
| 0 | 1 | 0 |
| 1 | 1 | 1 |

To find a minimal form, simply follow these steps:

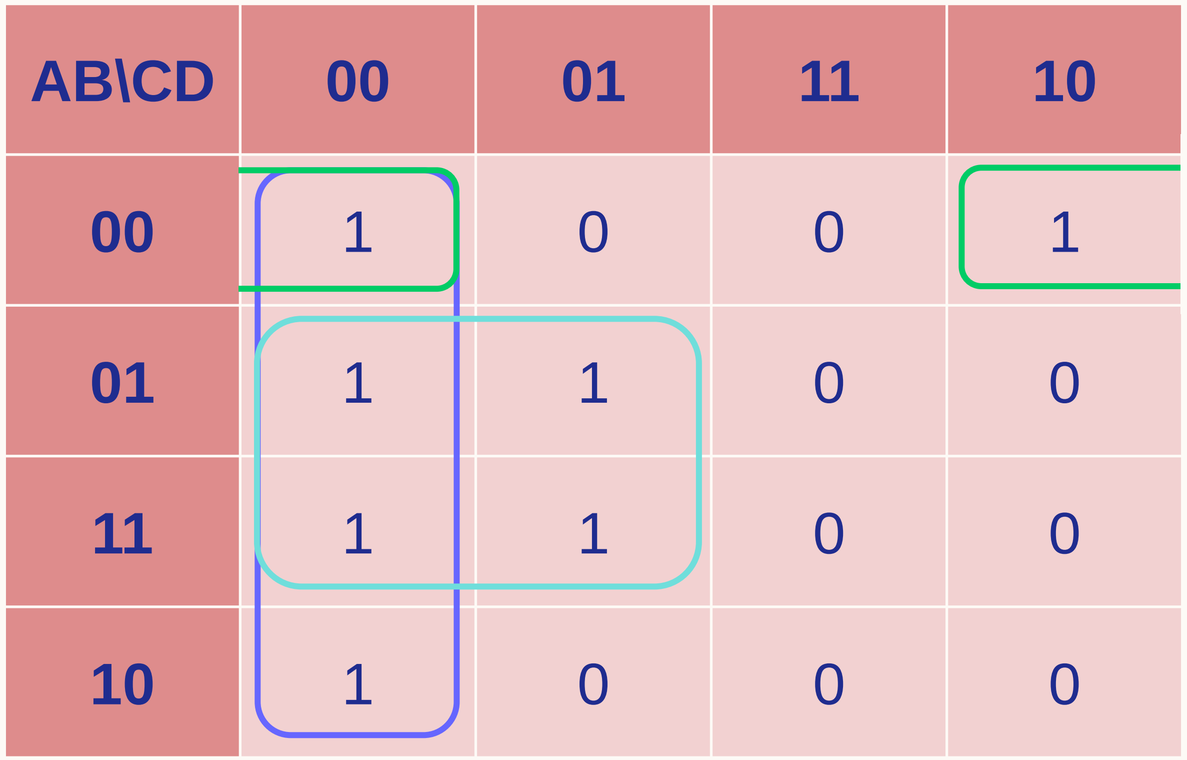
- Group 1s together in boxes with edge lengths $2^n$.
    - Boxes can wrap around.
    - Boxes can (and should) overlap, as *the bigger the box, the simpler the expression!*
- For each box, ∧ the variables that don't change (¬ them if 0).
- Finally, ∨ each of the boxes' expressions.

$A \vee \neg B$

# K-MAPS (EXAMPLE 1)

When there is more than 1 variables on a side, the input values are listed using **binary-reflected Gray code**, where adjacent values only differ by one bit (binary digit).

Start with all bits zero and successively flip the right-most bit that produces a new string



| AB\CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 1 | 1 | 0 | 0 |
| 10 | 1 | 0 | 0 | 0 |

$(\neg C \land \neg D) \lor (B \land \neg C) \lor (\neg A \land \neg B \land \neg D)$

# K-MAPS (EXAMPLE 2)

Looking at our original example, we can validate our answer and see that the same formula can be reached just using a k-map!

| AB\C | 0 | 1 |
|---|---|---|
| 00 | 1 | 1 |
| 01 | 1 | 0 |
| 11 | 0 | 0 |
| 10 | 1 | 1 |

| A | B | C | ¬B ∨ (¬A ∧ ¬C) |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

# MORE VARIABLES

For 5 variables, the most common method is to construct two 4-variable K-maps, representing the two values of the 5th variable. Boxes can then be matched or “overlayed” between the K-maps to simplify the logic.

**E = 0**

| AB\CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| **00** | 0 | 0 | 0 | 0 |
| **01** | 1 | 1 | 1 | 1 |
| **11** | 1 | 1 | 1 | 1 |
| **10** | 0 | 0 | 0 | 0 |

**E = 1**

| AB\CD | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| **00** | 0 | 0 | 0 | 0 |
| **01** | 1 | 1 | 0 | 0 |
| **11** | 1 | 1 | 0 | 0 |
| **10** | 0 | 0 | 0 | 0 |

$(B \land \neg E) \lor (B \land \neg C)$