TUTORIAL 6 · [COMPUTATION AND LOGIC]

OBJECTIVES

In this tutorial, you will:

- learn more about Karnaugh maps;
- convert logical expressions to CNF using Karnaugh maps;
- learn about minimal CNFs.

TASKS

Exercises 1–4 are mandatory. Exercises 3 and 4 have some optional tasks. Exercise 5 is optional.

SUBMIT a file `cl-tutorial-6` with your answers (image or pdf).

DEADLINE Saturday, 31st of October, 4 PM UK time

Good Scholarly Practice

Please remember the good scholarly practice requirements of the University regarding work for credit.

You can find guidance at the School page

http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct.

This also has links to the relevant University pages. Please do not publish solutions to these exercises on the internet or elsewhere, to avoid others copying your solutions.

EXERCISE 1

MANDATORY | BEFORE TUTORIAL SESSION

Watch the Week 6 videos and read Chapter 12 (*Karnaugh Maps*) of the textbook.

For each of the following expressions, say how many clauses there are in the expression, fill in a Karnaugh map like the one below and indicate the blocks of zeros corresponding to each clause:

1. $B \vee D$

2. $(\neg A \vee C) \wedge (A \vee \neg D)$

AB \ CD	00	01	11	10
00				
01				
11				
10				

SOLUTION TO EXERCISE 1

1. $B \vee D$

   1 clause

CD

AB	00	01	11	10
00	0	1	1	0
01	1	1	1	1
11	1	1	1	1
10	0	1	1	0

SOLUTION TO EXERCISE 1 (CONT.)

2. $(\neg A \vee C) \wedge (A \vee \neg D)$

   2 clauses: $\neg A \vee C$ and $A \vee \neg D$

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

For each of the following Karnaugh maps, give an expression that is true for exactly the states marked with x, and another expression that is true for exactly the states marked with y.

AB \ CD	00	01	11	10
00	x	x	x	x
01	x	x	x	x
11	x	y	y	x
10	x	y	y	x

AB \ CD	00	01	11	10
00	y	y	x	x
01	x	x	x	x
11	x	x	x	x
10	x	x	x	x

How would your answers change if the word “exactly” were omitted from the question?

SOLUTION TO EXERCISE 2

AB \ CD	00	01	11	10
00	x	x	x	x
01	x	x	x	x
11	x	y	y	x
10	x	y	y	x

$\neg A \vee \neg D$ is true for exactly the states marked with x

$A \wedge D$ is true for exactly the states marked with y

SOLUTION TO EXERCISE 2

AB \ CD	00	01	11	10
00	y	y	x	x
01	x	x	x	x
11	x	x	x	x
10	x	x	x	x

$A \vee B \vee C$ is true for exactly the states marked with x

$\neg A \wedge \neg B \wedge \neg C$ is true for exactly the states marked with y

If the word "exactly" were omitted, we could trivially answer the question by saying that $\top$ is true for all states (including those marked with x or y).

Give a Karnaugh map for each of the following expressions and use it to derive an equivalent CNF:

3. $r \leftrightarrow (a \wedge b)$
4. $r \leftrightarrow (a \rightarrow b)$

These are two of the expressions for which you derived CNFs in Exercise 5 from Tutorial 5.

How do the CNFs derived by the two methods compare?

Can you use your KM to find a minimal CNF for each expression? Where, by minimal we mean a CNF with as few occurrences of literals as possible – pay attention to the distinction between the number of literals in an expression and the number of occurrences of literals, as the same literal may occur several times.

SOLUTION TO EXERCISE 3 (CONT.)

3. $r \leftrightarrow (a \wedge b)$

An equivalent (minimal) CNF is:

$(r \vee \neg a \vee \neg b) \wedge (\neg r \vee a) \wedge (\neg r \vee b)$

r \ ab	00	01	11	10
0	1	1	0	1
1	0	0	1	0

This happens to be the same CNF that we obtained in Tutorial 5. However, the reduction method used in Tutorial 5 does not guarantee minimality.

Note that, in order to obtain a minimal CNF, you should consider maximal blocks. This is why we chose to overlap the blue and yellow blocks.

SOLUTION TO EXERCISE 3 (CONT.)

4. $r \leftrightarrow (a \rightarrow b)$

An equivalent (minimal) CNF is:

$(r \vee a) \wedge (r \vee \neg b) \wedge (\neg r \vee \neg a \vee b)$

r \ ab	00	01	11	10
0	0	0	0	1
1	1	1	1	0

Once again, this is the same CNF that we obtained in Tutorial 5.

Remember that, to obtain a minimal CNF, you should consider maximal blocks. This is why we chose to overlap the red and blue blocks.

EXERCISE 4

MANDATORY | BEFORE TUTORIAL SESSION

For each of the following pairs of clauses, draw a Karnaugh map and show the two blocks of zero states corresponding to the two clauses:

5. $A \vee \neg B,\ \neg A \vee \neg D$
6. $A \vee \neg B,\ A \vee B \vee C$

Use these Karnaugh maps to identify new clauses Δ_1 and Δ_2, different from both premises, such that the following sequents are valid:

7. $A \vee \neg B,\ \neg A \vee \neg D \models \Delta_1$
8. $A \vee \neg B,\ A \vee B \vee C \models \Delta_2$

How many different solutions can you find for clauses Δ_1 and Δ_2?

SOLUTION TO EXERCISE 4

5. $A \vee \neg B$, $\neg A \vee \neg D$

CD

AB	00	01	11	10
00	1	1	1	1
01	0	0	0	0
11	1	0	0	1
10	1	0	0	1

SOLUTION TO EXERCISE 4 (CONT.)

6. $A \vee \neg B$, $A \vee B \vee C$

CD

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

SOLUTION TO EXERCISE 4 (CONT.)

7. $A \vee \neg B, \neg A \vee \neg D \models \Delta_1$

For example, $\Delta_1 = \neg B \vee \neg D$

We can choose Δ_1 from:

- 8 1×1 blocks
- 6 1×2 blocks
- 4 2×1 blocks
- 1 2×2 block

Hence, in total, there are 19 possible solutions (clauses) for Δ_1.

		CD			
		00	01	11	10
AB	00	1	1	1	1
	01	0	0	0	0
	11	1	0	0	1
	10	1	0	0	1

SOLUTION TO EXERCISE 4 (CONT.)

8. $A \vee \neg B,\ A \vee B \vee C \models \Delta_2$

For example, $\Delta_2 = A \vee C$

We can choose Δ_2 from:

- 6 1×1 blocks
- 5 1×2 blocks
- 2 2×1 blocks
- 1 2×2 block

Hence, in total, there are 14 possible solutions (clauses) for Δ_2.

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

EXERCISE 4 (CONT.)

OPTIONAL | BEFORE TUTORIAL SESSION

Read about race conditions on the Wikipedia Karnaugh maps page and on how race conditions can be avoided by adding clauses.

For each of the Karnaugh Maps you have drawn (i.e. for 5 and 6), identify any race conditions and derive both a minimal CNF and a minimal race-free CNF.

SOLUTION TO EXERCISE 4 (CONT.)

5. $A \vee \neg B$, $\neg A \vee \neg D$

There is a race condition when moving from the blue block to the red block (which are adjacent but disjoint).

To eliminate the hazard, we add a new clause, $\neg B \vee \neg D$, corresponding to the yellow block.

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

A minimal CNF is $(A \vee \neg B) \wedge (\neg A \vee \neg D)$.

A minimal race-free CNF is $(A \vee \neg B) \wedge (\neg A \vee \neg D) \wedge (\neg B \vee \neg D)$.

SOLUTION TO EXERCISE 4 (CONT.)

6. $A \lor \neg B$, $A \lor B \lor C$

There is a race condition when moving from the blue block to the red block (which are adjacent but disjoint).

To eliminate the hazard, we add a new clause, $A \lor C$, corresponding to the yellow block.

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

A minimal CNF is $(A \lor \neg B) \land (A \lor C)$, which is also race free.

EXERCISE 5

Discuss with your table colleagues and solve together this exercise.

Using an example, explain why Karnaugh maps use the order 00, 01, 11, 10 for the values of A, B and C, D rather than truth table order.

Hint: Think of which groups of cells you can/can't identify using conjunctions of literals.

Use the following A-B-C criteria to fine-tune your answer:

Accessibility – your explanation should be accessible to someone who has not taken INF1;
Brevity – shorter explanations are better;
Clarity – the explanation should be clear and unambiguous.

At the end of the session you will name a representative to share your answer with the class.

EXERCISE 5 (CONT.)

Gray codes are useful in dealing with electromechanical switches, error correction in digital communications, and measuring angles, among many other things.

You can read more about their use in rotary encoders for angle-measuring devices here.

Discuss with your colleagues how to write in Haskell a function

```
gray :: Int -> [[Int]]
```

that returns, for any input n, an n-bit Gray code.