Assignment 1

Computer Organization

Deadline: 11:55pm, Monday, Oct 08, 2024

Student ID:\_\_\_\_\_2330016056\_\_\_\_\_\_\_\_\_\_\_\_ Name:\_\_\_\_\_\_\_Bohan YANG\_\_\_\_\_\_\_\_\_\_\_\_\_

1. 1) Define what a Turing machine is. 2) What does UTM stand for? Explain what it is. (10 points)
   1. A Turing machine is a mathematical model of computation describing an abstract machine that manipulates symbols on a strip of tape according to a table of rules.
   2. UTM is stand for Universal Turing Machine.

Turing described a Turing machine that could simulate all other Turing machines.

1. Describe the seven levels of transformations of a computer system. (10 points)

Level 6. User (Executable Programs)

Level 5. High-Level Language (C++, Java, etc.)

Level 4. Assembly Language (Assembly Code)

Level 3. System Software (Operating System, Library Code)

Level 2. Machine (Instruction Set Architecture)

Level 1. Control (Microcode or Hardwired)

Level 0. Digital Logic (Circuits, Gates, etc.)

The seven levels of transformations in a computer system show how we go from recognizing a problem to creating a physical device that solves it. It all begins with understanding the issue and crafting an algorithm. Then, we turn that algorithm into a program, which is linked to an instruction set architecture. This architecture is broken down into a microarchitecture, which further translates into circuits. Finally, those circuits come together to form the actual physical devices we use.

1. Explain the fetch-decode-execute cycle of the von Neumann Architecture. (10 points)
   1. the control unit fetch the next instruction from the memory
   2. the instruction is decoded into a language that the ALU understands
   3. data operands are fetched from the memory into the registers inside CPU
   4. the ALU executes the instruction and places the result into the registers or memory
2. Given 8 bits, represent the numbers +53 and -109 into binary using the following approach: 1) Signed-magnitude; 2) One’s complement; 3) Two’s complement. Show your steps. (12 points)
   1. Signed-magnitude
      1. +53 = (0110101)\_2, so +53’ s signed-magnitude is 0 0110101
      2. 109 = (1101101)\_2, so -109’s signed-magnitude is 1 1101101
   2. One’s complement
      1. +53 > 0, so one’s complement is 0 0110101
      2. -109 < 0, so one’s complement is 1 0010010
   3. Two’s complement
      1. +53 > 0, so one’s complement is 0 0110101
      2. -109 < 0, so one’s complement is 1 0010011
3. Convert -57.625 into binary using 32 bits floating number representation. Show your steps. (6 points)
   1. 57.625 = 111001.101 = 1.11001101 \* 2 ^ 5
   2. sign = 1
   3. exponent = 5 + 127 = 132 = 1000 0100
   4. mantissa = 1100 1101
   5. result = 1100 0010 0110 0110 1000 0000 0000 0000
4. The following Turing Machine is supposed to count in base 2.

{

"name": "Count in Base 2",

"max\_state": 2,

"symbols": "0,1",

"tape": "0",

"position": 0,

"rules": [

[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_],

[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_],

[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_],

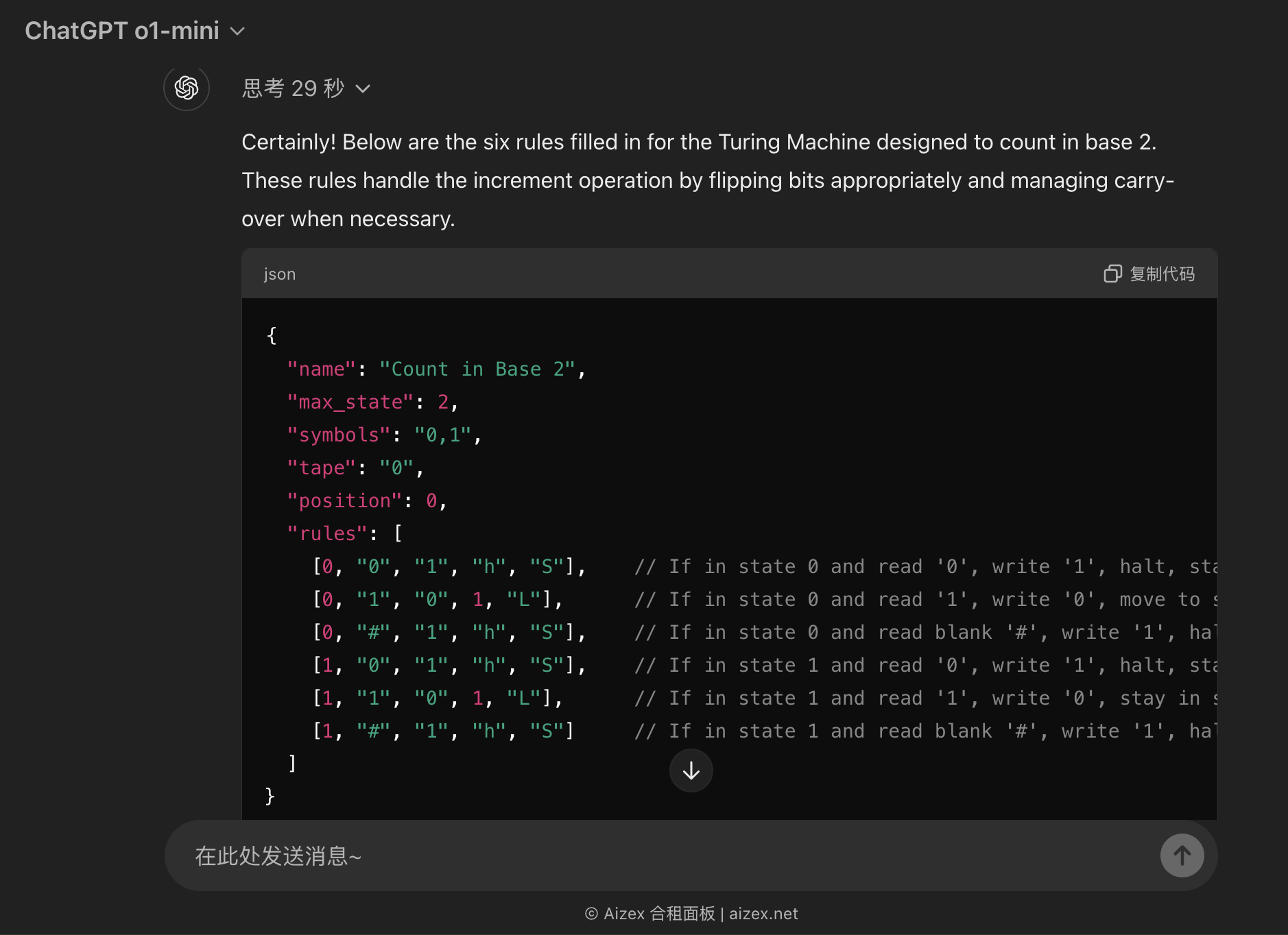
[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_],

[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_],

[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_]

]

}

1. Please use ChatGPT or other generative AI tools to fill in the missing rules and attach a screenshot of your attempt. (5 points)
   1. 
2. Do you think the answer provided is correct? If yes, test it on the website we used in the lab. If not, modify it accordingly. After making your adjustments, paste the runnable JSON content below. (6 points)

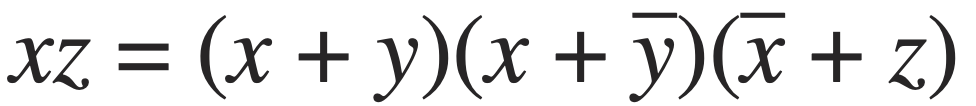
|  |
| --- |
| {  "name": "Count in Base 2",  "max\_state": 2,  "symbols": "0, 1",  "tape": "0",  "position": 0,  "rules": [  [ 0, "#", "1", 1, "R" ],  [ 0, "0", "1", 1, "R" ],  [ 0, "1", "0", 0, "L" ],  [ 1, "#", "#", 0, "L" ],  [ 1, "\*", "\*", 1, "R" ]  ]  } |

1. Compare this Turing Machine program with Lab0 Exercise 2. Are there any differences? (4 points)
   1. Yes, there is a difference in the initial tape contents.

In Lab0 Exercise 2, the Turing machine program starts with an empty tape.

In this Turing machine program, it starts with a tape containing “0”.

Because the max\_state is only 2. If max\_state is 3, this Turing machine program can start with an empty tape and count “0”.



1. Show that
2. Using truth table; (5 points)
3. Using Boolean identities; (5 points)
4. Use ChatGPT or other generative AI tools to attempt the proof and attach a screenshot of your attempt. (5 points)

(1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | z | xz | (x+y)(x+!y)(!x+z) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

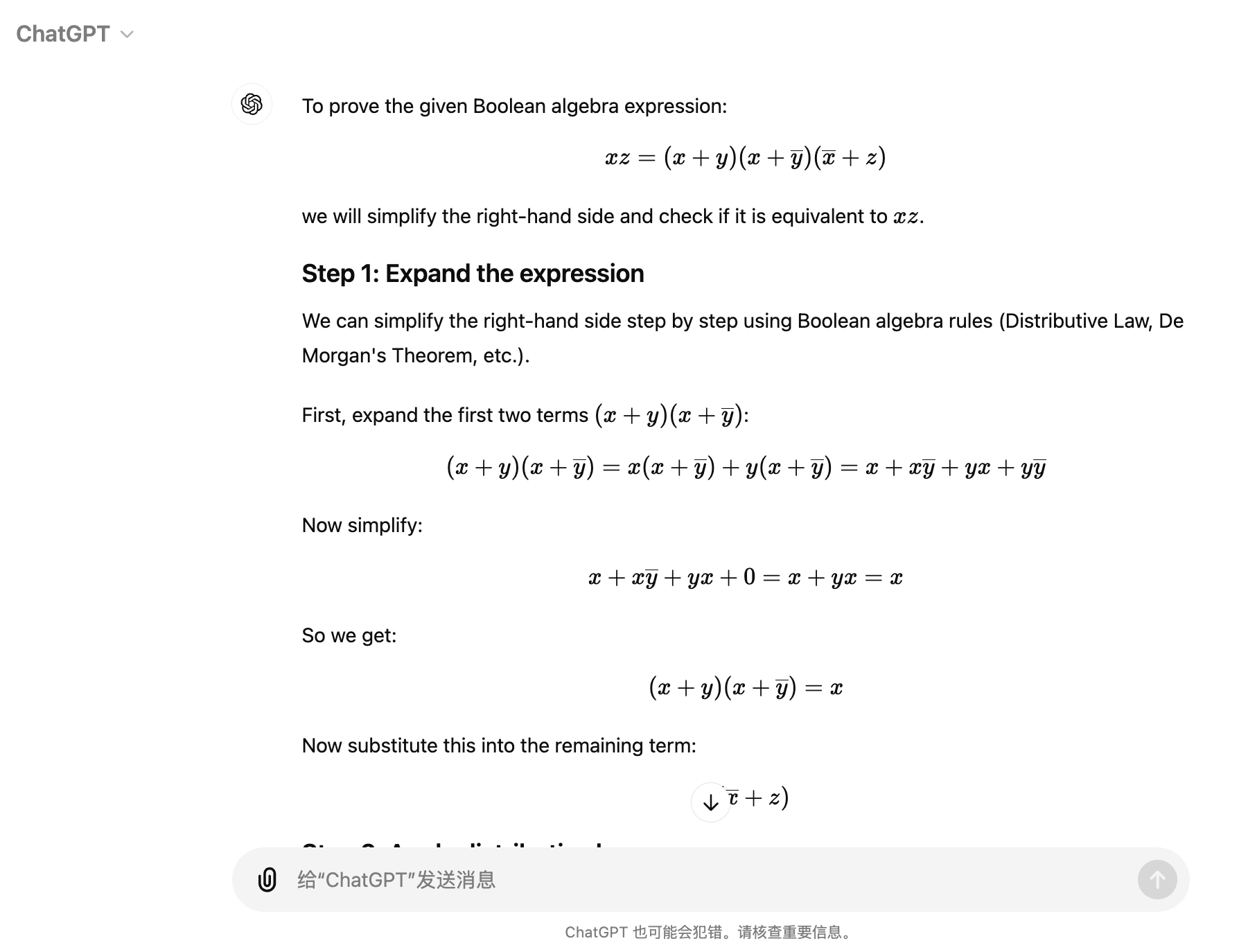
(2)

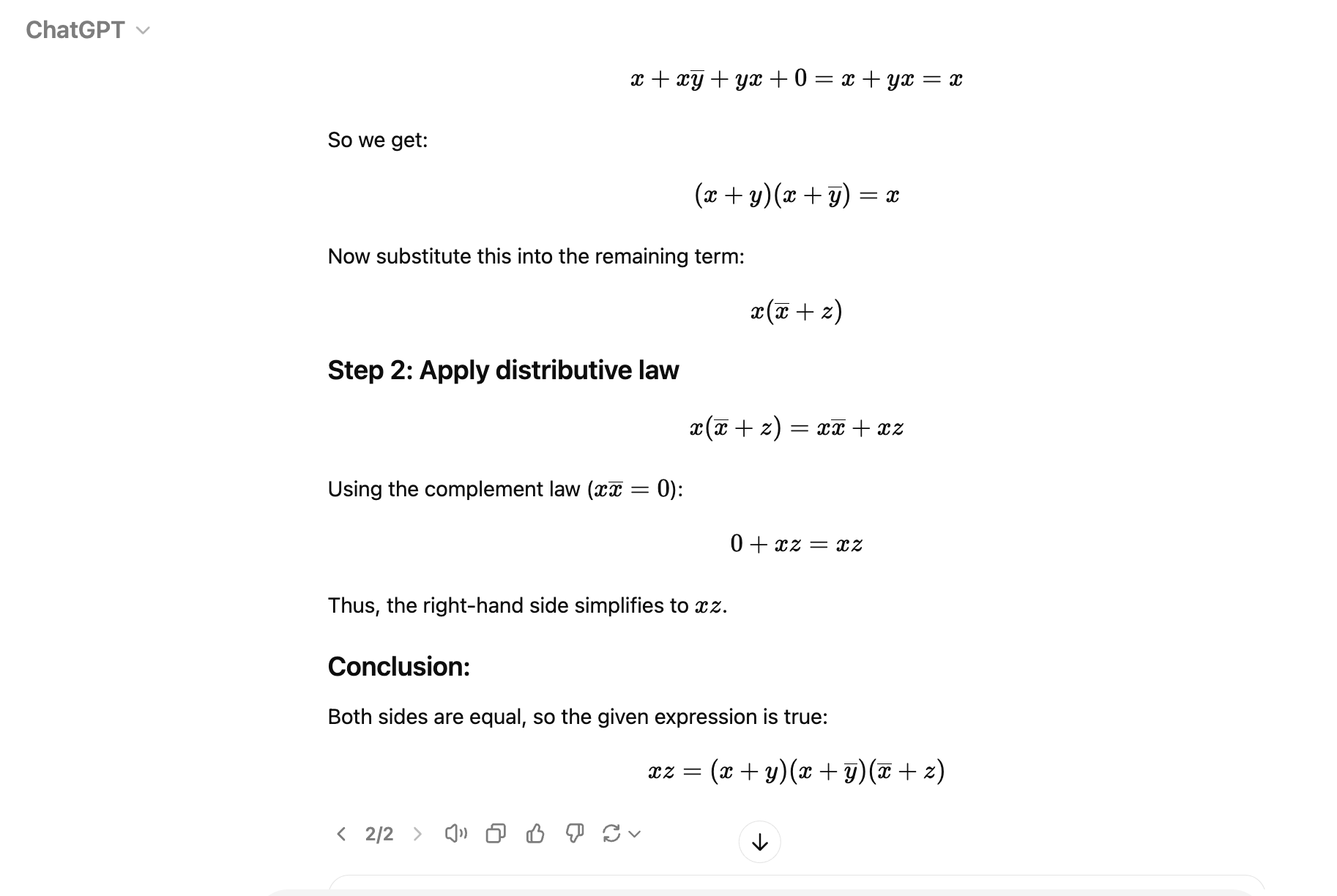
RHS = (x + y)(x + y’)(x’ + z) = x (x’ + z) = x’x + xz = xz

LHS = xz

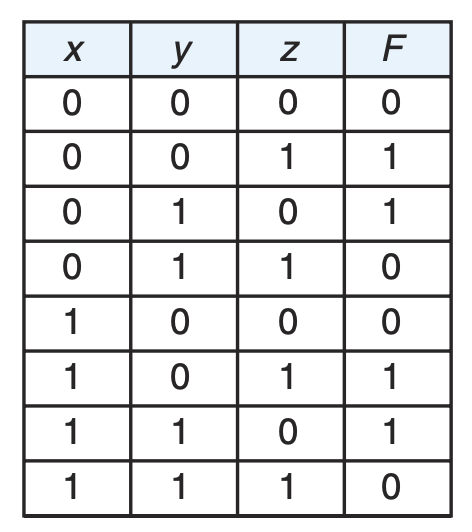
LHS = RHS

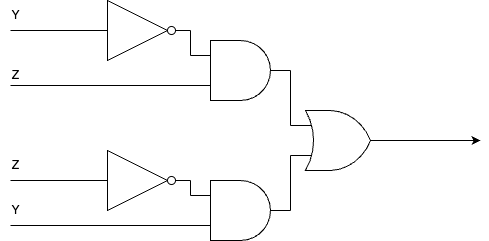
Q.E.D.

(3) 



1. The truth table for a Boolean expression is shown below. (16 points)
   1. Write the Boolean expression in sum-of-products form. (4 points)
   2. Write the Boolean expression in product-of-sums form. (4 points)
   3. Simplify the sum-of-products form using Boolean identities; (4 points)
   4. Draw the logical circuit diagram for the simplified Boolean expression;(4 points)



1. 
2. Simplify the above Boolean expression using K-MAP by hand. (6 points )

