

Sop_pos_zad_dom

May 12, 2024

```
[2]: import itertools
```

```
[3]: def generate_truth_table(inputs, outputs):  
    '''Generate a truth table from input combinations and corresponding outputs.'''  
    ↪'''  
    truth_table = []  
    for input_combination, output_bit in zip(inputs, outputs):  
        truth_table.append(input_combination + (output_bit,))  
    return truth_table
```

```
[1]: def generate_sop_from_truth_table(truth_table):  
    '''Generate a Sum of Products (SoP) expression from a truth table.'''  
    sop_terms = []  
    num_vars = len(truth_table[0]) - 1  
    for row in truth_table:  
        if row[-1] == 1:  
            term = []  
            for i, bit in enumerate(row[:-1]):  
                if bit == 1:  
                    term.append('x' + str(i+1))  
                elif bit == 0:  
                    term.append("¬x" + str(i+1))  
            sop_terms.append('(' + ' * '.join(term) + ')')  
    unique_terms = set(sop_terms)  
    return ' + '.join(sop_terms) if unique_terms else 'There are no conditions_'  
    ↪matching the SoP expression'
```

```
[4]: def generate_pos_from_truth_table(truth_table):  
    '''Generate a Product of Sums (PoS) expression from a truth table.'''  
    pos_terms = []  
    num_vars = len(truth_table[0]) - 1  
    for row in truth_table:  
        if row[-1] == 0:  
            term = []  
            for i, bit in enumerate(row[:-1]):  
                if bit == 0:  
                    term.append('x' + str(i+1))  
                elif bit == 1:
```

```

        term.append('¬x' + str(i+1))
        pos_terms.append('(' + ' + '.join(term) + ')')
    unique_terms = set(pos_terms)
    return ' * '.join(pos_terms) if unique_terms else 'There are no conditions_
    ↳matching the PoS expression'

```

```

[5]: outputs = [
    (1, 1, 0, 1),      # task1
    (0, 1, 1, 1),      # task2
    (0, 1, 0, 0),      # task3
    (0, 0, 0, 1),      # task4
    (0, 1, 1, 0),      # task5
    (1, 0, 0, 0),      # task6
    (1, 0, 1, 0),      # task7
    (0, 1, 0, 1),      # task8
    (1, 1, 1, 0),      # task9
    (1, 0, 1, 1),      # task10
    (0, 0, 0, 1, 0, 1, 1, 0),  # task11
    (1, 0, 1, 1, 0, 1, 0, 1),  # task12
    (1, 0, 1, 1, 1, 1, 1, 1),  # task13
    (0, 1, 1, 0, 1, 0, 0, 1),  # task14
    (0, 0, 1, 1, 1, 0, 0, 0),  # task15
    (1, 1, 1, 0, 1, 1, 0, 0),  # task16
    (0, 0, 1, 1, 0, 1, 0, 1),  # task17
    (1, 0, 0, 1, 0, 0, 0, 1),  # task18
    (0, 1, 1, 0, 0, 0, 0, 0),  # task19
    (1, 1, 0, 0, 0, 1, 1, 1),  # task20
    (0, 1, 0, 1, 0, 0, 0, 1),  # task21
    (1, 0, 1, 1, 1, 1, 1, 0),  # task22
    (1, 0, 1, 1, 1, 0, 1, 0),  # task23
    (0, 0, 0, 0, 0, 1, 0, 1),  # task24
    (0, 0, 1, 0, 1, 1, 1, 1),  # task25
    (0, 0, 1, 0, 0, 0, 1, 0),  # task26
    (0, 0, 0, 0, 1, 1, 1, 1),  # task27
    (1, 1, 0, 1, 1, 1, 0, 1),  # task28
    (0, 0, 1, 0, 1, 0, 1, 1),  # task29
    (1, 0, 1, 1, 1, 0, 1, 1),  # task30
    (0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1),  # task31
    (0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1),  # task32
    (0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0),  # task33
    (1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1),  # task34
    (1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0),  # task35
    (1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1),  # task36
    (1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1),  # task37
    (1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1),  # task38
    (1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0),  # task39
    (1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1),  # task40

```

```

(0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0), # task41
(1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0), # task42
(0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0), # task43
(1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1), # task44
(0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1), # task45
(1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0), # task46
(0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1), # task47
(0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0), # task48
(1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0), # task49
(1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0), # task50
]

```

```

[6]: # Generate inputs for 2 bits
inputs = list(itertools.product([0, 1], repeat=2))

for i, output in enumerate(outputs, start=1):
    truth_table = generate_truth_table(inputs, output)

    if i <= 5:
        expression = generate_sop_from_truth_table(truth_table)
        print(f'Task {i}. SoP: {expression}')
        print('=====')
    elif 6 <= i <= 10:
        expression = generate_pos_from_truth_table(truth_table)
        print(f'Task {i}. PoS: {expression}')
        print('=====')

```

```

Task 1. SoP: ( $\neg x_1 * \neg x_2$ ) + ( $\neg x_1 * x_2$ ) + ( $x_1 * x_2$ )
=====
Task 2. SoP: ( $\neg x_1 * x_2$ ) + ( $x_1 * \neg x_2$ ) + ( $x_1 * x_2$ )
=====
Task 3. SoP: ( $\neg x_1 * x_2$ )
=====
Task 4. SoP: ( $x_1 * x_2$ )
=====
Task 5. SoP: ( $\neg x_1 * x_2$ ) + ( $x_1 * \neg x_2$ )
=====
Task 6. PoS: ( $x_1 + \neg x_2$ ) * ( $\neg x_1 + x_2$ ) * ( $\neg x_1 + \neg x_2$ )
=====
Task 7. PoS: ( $x_1 + \neg x_2$ ) * ( $\neg x_1 + \neg x_2$ )
=====
Task 8. PoS: ( $x_1 + x_2$ ) * ( $\neg x_1 + x_2$ )
=====
Task 9. PoS: ( $\neg x_1 + \neg x_2$ )
=====
Task 10. PoS: ( $x_1 + \neg x_2$ )
=====

```

```
[7]: # Generate inputs for 3 bits
inputs = list(itertools.product([0, 1], repeat=3))

for i, output in enumerate(outputs, start=1):
    truth_table = generate_truth_table(inputs, output)

    if 11 <= i <= 20:
        expression = generate_sop_from_truth_table(truth_table)
        print(f'Task {i}. SoP: {expression}')
        print('=====')
    elif 21 <= i <= 30:
        expression = generate_pos_from_truth_table(truth_table)
        print(f'Task {i}. PoS: {expression}')
        print('=====')
```

Task 11. SoP: $(\neg x_1 * x_2 * x_3) + (x_1 * \neg x_2 * x_3) + (x_1 * x_2 * \neg x_3)$

=====

Task 12. SoP: $(\neg x_1 * \neg x_2 * \neg x_3) + (\neg x_1 * x_2 * \neg x_3) + (\neg x_1 * x_2 * x_3) + (x_1 * \neg x_2 * x_3) + (x_1 * x_2 * x_3)$

=====

Task 13. SoP: $(\neg x_1 * \neg x_2 * \neg x_3) + (\neg x_1 * x_2 * \neg x_3) + (\neg x_1 * x_2 * x_3) + (x_1 * \neg x_2 * \neg x_3) + (x_1 * \neg x_2 * x_3) + (x_1 * x_2 * \neg x_3) + (x_1 * x_2 * x_3)$

=====

Task 14. SoP: $(\neg x_1 * \neg x_2 * x_3) + (\neg x_1 * x_2 * \neg x_3) + (x_1 * \neg x_2 * \neg x_3) + (x_1 * x_2 * x_3)$

=====

Task 15. SoP: $(\neg x_1 * x_2 * \neg x_3) + (\neg x_1 * x_2 * x_3) + (x_1 * \neg x_2 * \neg x_3)$

=====

Task 16. SoP: $(\neg x_1 * \neg x_2 * \neg x_3) + (\neg x_1 * \neg x_2 * x_3) + (\neg x_1 * x_2 * \neg x_3) + (x_1 * \neg x_2 * \neg x_3) + (x_1 * \neg x_2 * x_3)$

=====

Task 17. SoP: $(\neg x_1 * x_2 * \neg x_3) + (\neg x_1 * x_2 * x_3) + (x_1 * \neg x_2 * x_3) + (x_1 * x_2 * x_3)$

=====

Task 18. SoP: $(\neg x_1 * \neg x_2 * \neg x_3) + (\neg x_1 * x_2 * x_3) + (x_1 * x_2 * x_3)$

=====

Task 19. SoP: $(\neg x_1 * \neg x_2 * x_3) + (\neg x_1 * x_2 * \neg x_3)$

=====

Task 20. SoP: $(\neg x_1 * \neg x_2 * \neg x_3) + (\neg x_1 * \neg x_2 * x_3) + (x_1 * \neg x_2 * x_3) + (x_1 * x_2 * \neg x_3) + (x_1 * x_2 * x_3)$

=====

Task 21. PoS: $(x_1 + x_2 + x_3) * (x_1 + \neg x_2 + x_3) * (\neg x_1 + x_2 + x_3) * (\neg x_1 + x_2 + \neg x_3) * (\neg x_1 + \neg x_2 + x_3)$

=====

Task 22. PoS: $(x_1 + x_2 + \neg x_3) * (\neg x_1 + \neg x_2 + \neg x_3)$

=====

Task 23. PoS: $(x_1 + x_2 + \neg x_3) * (\neg x_1 + x_2 + \neg x_3) * (\neg x_1 + \neg x_2 + \neg x_3)$

=====

Task 24. PoS: $(x_1 + x_2 + x_3) * (x_1 + x_2 + \neg x_3) * (x_1 + \neg x_2 + x_3) * (x_1 + \neg x_2 + \neg x_3) * (\neg x_1 + x_2 + x_3) * (\neg x_1 + \neg x_2 + x_3)$

Task 25. PoS: $(x_1 + x_2 + x_3) * (x_1 + x_2 + \neg x_3) * (x_1 + \neg x_2 + \neg x_3)$

Task 26. PoS: $(x_1 + x_2 + x_3) * (x_1 + x_2 + \neg x_3) * (x_1 + \neg x_2 + \neg x_3) * (\neg x_1 + x_2 + x_3) * (\neg x_1 + x_2 + \neg x_3) * (\neg x_1 + \neg x_2 + \neg x_3)$

Task 27. PoS: $(x_1 + x_2 + x_3) * (x_1 + x_2 + \neg x_3) * (x_1 + \neg x_2 + x_3) * (x_1 + \neg x_2 + \neg x_3)$

Task 28. PoS: $(x_1 + \neg x_2 + x_3) * (\neg x_1 + \neg x_2 + x_3)$

Task 29. PoS: $(x_1 + x_2 + x_3) * (x_1 + x_2 + \neg x_3) * (x_1 + \neg x_2 + \neg x_3) * (\neg x_1 + x_2 + \neg x_3)$

Task 30. PoS: $(x_1 + x_2 + \neg x_3) * (\neg x_1 + x_2 + \neg x_3)$

```
[8]: # Generate inputs for 4 bits
inputs = list(itertools.product([0, 1], repeat=4))

for i, output in enumerate(outputs, start=1):
    truth_table = generate_truth_table(inputs, output)

    if 31 <= i <= 40:
        expression = generate_sop_from_truth_table(truth_table)
        print(f'Task {i}. SoP: {expression}')
        print('=====')
    elif 41 <= i <= 50:
        expression = generate_pos_from_truth_table(truth_table)
        print(f'Task {i}. PoS: {expression}')
        print('=====')
```

Task 31. SoP: $(\neg x_1 * \neg x_2 * x_3 * \neg x_4) + (\neg x_1 * x_2 * \neg x_3 * \neg x_4) + (\neg x_1 * x_2 * \neg x_3 * x_4) + (\neg x_1 * x_2 * x_3 * \neg x_4) + (x_1 * \neg x_2 * \neg x_3 * \neg x_4) + (x_1 * \neg x_2 * \neg x_3 * x_4) + (x_1 * \neg x_2 * x_3 * \neg x_4) + (x_1 * \neg x_2 * x_3 * x_4) + (x_1 * x_2 * \neg x_3 * \neg x_4) + (x_1 * x_2 * \neg x_3 * x_4) + (x_1 * x_2 * x_3 * \neg x_4) + (x_1 * x_2 * x_3 * x_4)$

Task 32. SoP: $(\neg x_1 * \neg x_2 * x_3 * \neg x_4) + (\neg x_1 * x_2 * \neg x_3 * x_4) + (\neg x_1 * x_2 * x_3 * \neg x_4) + (x_1 * \neg x_2 * x_3 * \neg x_4) + (x_1 * x_2 * \neg x_3 * \neg x_4) + (x_1 * x_2 * \neg x_3 * x_4) + (x_1 * x_2 * x_3 * \neg x_4) + (x_1 * x_2 * x_3 * x_4)$

Task 33. SoP: $(\neg x_1 * x_2 * \neg x_3 * \neg x_4) + (x_1 * x_2 * \neg x_3 * x_4)$

Task 34. SoP: $(\neg x_1 * \neg x_2 * \neg x_3 * \neg x_4) + (\neg x_1 * \neg x_2 * x_3 * \neg x_4) + (\neg x_1 * \neg x_2 * x_3 * x_4) + (\neg x_1 * x_2 * \neg x_3 * \neg x_4) + (\neg x_1 * x_2 * x_3 * \neg x_4) + (\neg x_1 * x_2 * x_3 * x_4) + (x_1 * \neg x_2 * \neg x_3 * \neg x_4) + (x_1 * \neg x_2 * \neg x_3 * x_4) + (x_1 * \neg x_2 * x_3 * \neg x_4) + (x_1 * \neg x_2 * x_3 * x_4) + (x_1 * x_2 * \neg x_3 * \neg x_4) + (x_1 * x_2 * \neg x_3 * x_4) + (x_1 * x_2 * x_3 * \neg x_4) + (x_1 * x_2 * x_3 * x_4)$

$\neg x_4) * (x_1 + \neg x_2 + x_3 + x_4) * (x_1 + \neg x_2 + \neg x_3 + x_4) * (x_1 + \neg x_2 + \neg x_3 + \neg x_4) * \\ (\neg x_1 + x_2 + x_3 + x_4) * (\neg x_1 + x_2 + \neg x_3 + \neg x_4) * (\neg x_1 + \neg x_2 + x_3 + \neg x_4) * (\neg x_1 + \\ \neg x_2 + \neg x_3 + \neg x_4)$

=====

Task 47. PoS: $(x_1 + x_2 + x_3 + x_4) * (x_1 + \neg x_2 + x_3 + x_4) * (x_1 + \neg x_2 + x_3 + \neg x_4) \\ * (x_1 + \neg x_2 + \neg x_3 + \neg x_4) * (\neg x_1 + x_2 + x_3 + \neg x_4) * (\neg x_1 + x_2 + \neg x_3 + \neg x_4) * (\neg x_1 + \\ + \neg x_2 + x_3 + x_4) * (\neg x_1 + \neg x_2 + x_3 + \neg x_4) * (\neg x_1 + \neg x_2 + \neg x_3 + x_4)$

=====

Task 48. PoS: $(x_1 + x_2 + x_3 + x_4) * (x_1 + \neg x_2 + x_3 + \neg x_4) * (x_1 + \neg x_2 + \neg x_3 + \\ x_4) * (x_1 + \neg x_2 + \neg x_3 + \neg x_4) * (\neg x_1 + x_2 + \neg x_3 + x_4) * (\neg x_1 + x_2 + \neg x_3 + \neg x_4) * \\ (\neg x_1 + \neg x_2 + x_3 + x_4) * (\neg x_1 + \neg x_2 + x_3 + \neg x_4) * (\neg x_1 + \neg x_2 + \neg x_3 + x_4) * (\neg x_1 + \\ \neg x_2 + \neg x_3 + \neg x_4)$

=====

Task 49. PoS: $(x_1 + x_2 + x_3 + \neg x_4) * (x_1 + \neg x_2 + \neg x_3 + x_4) * (\neg x_1 + x_2 + \neg x_3 + \\ x_4) * (\neg x_1 + \neg x_2 + \neg x_3 + \neg x_4)$

=====

Task 50. PoS: $(x_1 + \neg x_2 + x_3 + x_4) * (x_1 + \neg x_2 + \neg x_3 + \neg x_4) * (\neg x_1 + x_2 + x_3 + \\ x_4) * (\neg x_1 + x_2 + \neg x_3 + \neg x_4) * (\neg x_1 + \neg x_2 + x_3 + \neg x_4) * (\neg x_1 + \neg x_2 + \neg x_3 + x_4) * \\ (\neg x_1 + \neg x_2 + \neg x_3 + \neg x_4)$

=====