Example- An engineering problem involving logical conditions with 4 inputs results in the Truth Table shown where "X" means we don't care about the output under these conditions (e.g., maybe this combination of inputs is physically impossible).

a	*b*	*c*	*d*	*F*
0	0	0	0	**0**
0	0	0	1	**0**
0	0	1	0	**0**
0	0	1	1	**1**
0	1	0	0	**1**
0	1	0	1	**X**
0	1	1	0	**X**
0	1	1	1	**1**
1	0	0	0	**0**
1	0	0	1	**0**
1	0	1	0	**X**
1	0	1	1	**1**
1	1	0	0	**0**
1	1	0	1	**1**
1	1	1	0	**X**
1	1	1	1	**X**

We will use a four variable K-Map to determine Boolean functions in the simplified sum-of-products and product-of-sums forms for this problem.

- First create K-Map directly from Truth Table.

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**X**	**1**	**X**
11	**0**	**1**	**X**	**X**
10	**0**	**0**	**1**	**X**

- Next, we will use this K-Map to directly express *F* in a simplified sum-of-products form. The following sequence allows the coverage of all squares/minterms in groups of four by taking advantage of the don't-care squares/minterms.

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**X**	**1**	**X**
11	**0**	**1**	**X**	**X**
10	**0**	**0**	**1**	**X**

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**X**	**1**	**X**
11	**0**	**1**	**X**	**X**
10	**0**	**0**	**1**	**X**

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**X**	**1**	**X**
11	**0**	**1**	**X**	**X**
10	**0**	**0**	**1**	**X**

The K-Map with the selected minterms filled-in.

ab\cd	**00**	**01**	**11**	**10**
00	~~m_0~~	~~m_1~~	$m_3 = a'b'c\,d$	~~m_2~~
01	$m_4 = a'b\,c'd'$	$m_5 = a'b\,c'd$	$m_7 = a'b\,c\,d$	$m_6 = a'b\,c\,d'$
11	~~m_{12}~~	$m_{13} = a\,b\,c'd$	$m_{15} = a\,b\,c\,d$	~~m_{14}~~
10	~~m_8~~	~~m_9~~	$m_{11} = a\,b'c\,d$	~~m_{10}~~

The simplified sum-of-products form is: $\underline{\boldsymbol{F_{sop} = a'\,b + b\,d + c\,d}}$;
a result with **three** terms and **six** literals.

- What if all the don't-care minterms/squares had been made "0"s?

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**0**	**1**	**0**
11	**0**	**1**	**0**	**0**
10	**0**	**0**	**1**	**0**

The following sequence allows the best coverage of all squares/minterms under this scenario.

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**0**	**1**	**0**
11	**0**	**1**	**0**	**0**
10	**0**	**0**	**1**	**0**

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**0**	**1**	**0**
11	**0**	**1**	**0**	**0**
10	**0**	**0**	**1**	**0**

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**0**	**1**	**0**
11	**0**	**1**	**0**	**0**
10	**0**	**0**	**1**	**0**

The simplified sum-of-products form is ***a′cd + b′cd + a′bc′d′ + abc′d*** ; a result with **four** terms and **14** literals. Not nearly as compact.

- Next, we will use this K-Map to express *F* in a simplified product-of-sums form(s). To get *F′* in simplified sum-of-products form(s), the following sequences allow the coverage of all the “0” squares/minterms in groups of two or four by taking advantage of the don’t-care squares/minterms.

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**X**	**1**	**X**
11	**0**	**1**	**X**	**X**
10	**0**	**0**	**1**	**X**

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**X**	**1**	**X**
11	**0**	**1**	**X**	**X**
10	**0**	**0**	**1**	**X**

and

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**X**	**1**	**X**
11	**0**	**1**	**X**	**X**
10	**0**	**0**	**1**	**X**

or

ab\cd	**00**	**01**	**11**	**10**
00	**0**	**0**	**1**	**0**
01	**1**	**X**	**1**	**X**
11	**0**	**1**	**X**	**X**
10	**0**	**0**	**1**	**X**

Here, $\underline{F' = b'c' + cd' + ac'd'}$ or $\underline{F' = b'c' + cd' + abd'}$.

Now, we take the complement of F' to express F in a simplified product-of-sums form by using DeMorgan's Theorem (i.e., take dual and then complement each literal).

$$F_{\text{pos1}} = \left(F'\right)' = \left(b'c' + c\,d' + a\,c'd'\right)'$$
$$= (b+c)(c'+d)\left(a'+c+d\right)$$

a result with **three** terms and **seven** literals (two complemented). Or,

$$F_{\text{pos2}} = \left(F'\right)' = \left(b'c' + c\,d' + a\,b\,d'\right)'$$
$$= (b+c)(c'+d)\left(a'+b'+d\right)$$

a result with **three** terms and **seven** literals (three complemented). By one complement/NOT operation, the first option is slightly simpler.

- Finally, check these results using Truth Tables to show they agree for the "0"s and "1"s. We don't-care about the "X"s!

$F_{sop} = a'b + b\,d + c\,d$

a	*b*	*c*	*d*	*a'*	*a'b*	*b d*	*c d*	F_{sop}	***F***
0	0	0	0	1	0	0	0	**0**	**0**
0	0	0	1	1	0	0	0	**0**	**0**
0	0	1	0	1	0	0	0	**0**	**0**
0	0	1	1	1	0	0	1	**1**	**1**
0	1	0	0	1	1	0	0	**1**	**1**
0	1	0	1	1	1	1	0	**1**	**X**
0	1	1	0	1	1	0	0	**1**	**X**
0	1	1	1	1	1	1	1	**1**	**1**
1	0	0	0	0	0	0	0	**0**	**0**
1	0	0	1	0	0	0	0	**0**	**0**
1	0	1	0	0	0	0	0	**0**	**X**
1	0	1	1	0	0	0	1	**1**	**1**
1	1	0	0	0	0	0	0	**0**	**0**
1	1	0	1	0	0	1	0	**1**	**1**
1	1	1	0	0	0	0	0	**0**	**X**
1	1	1	1	0	0	1	1	**1**	**X**

$$F_{\text{pos1}} = (b+c)(c'+d)(a'+c+d)$$

a	b	c	d	a'	c'	$b+c$	$c'+d$	$a'+c+d$	F_{pos1}	***F***
0	0	0	0	1	1	0	1	1	**0**	**0**
0	0	0	1	1	1	0	1	1	**0**	**0**
0	0	1	0	1	0	1	0	1	**0**	**0**
0	0	1	1	1	0	1	1	1	**1**	**1**
0	1	0	0	1	1	1	1	1	**1**	**1**
0	1	0	1	1	1	1	1	1	**1**	**X**
0	1	1	0	1	0	1	0	1	**0**	**X**
0	1	1	1	1	0	1	1	1	**1**	**1**
1	0	0	0	0	1	0	1	0	**0**	**0**
1	0	0	1	0	1	0	1	1	**0**	**0**
1	0	1	0	0	0	1	0	1	**0**	**X**
1	0	1	1	0	0	1	1	1	**1**	**1**
1	1	0	0	0	1	1	1	0	**0**	**0**
1	1	0	1	0	1	1	1	1	**1**	**1**
1	1	1	0	0	0	1	0	1	**0**	**X**
1	1	1	1	0	0	1	1	1	**1**	**X**

$$F_{\text{pos2}} = (b+c)(c'+d)(a'+b'+d)$$

a	b	c	d	a'	b'	c'	$b+c$	$c'+d$	$a'+b'+d$	F_{pos2}	***F***
0	0	0	0	1	1	1	0	1	1	**0**	**0**
0	0	0	1	1	1	1	0	1	1	**0**	**0**
0	0	1	0	1	1	0	1	0	1	**0**	**0**
0	0	1	1	1	1	0	1	1	1	**1**	**1**
0	1	0	0	1	0	1	1	1	1	**1**	**1**
0	1	0	1	1	0	1	1	1	1	**1**	**X**
0	1	1	0	1	0	0	1	0	1	**0**	**X**
0	1	1	1	1	0	0	1	1	1	**1**	**1**
1	0	0	0	0	1	1	0	1	1	**0**	**0**
1	0	0	1	0	1	1	0	1	1	**0**	**0**
1	0	1	0	0	1	0	1	0	1	**0**	**X**
1	0	1	1	0	1	0	1	1	1	**1**	**1**
1	1	0	0	0	0	1	1	1	0	**0**	**0**
1	1	0	1	0	0	1	1	1	1	**1**	**1**
1	1	1	0	0	0	0	1	0	0	**0**	**X**
1	1	1	1	0	0	0	1	1	1	**1**	**X**