The variable A on the right-hand-side and the left-hand-side of the distributive law (7b) is then substituted by 0 and 1. When A = 0, both sides are equal to BC. When A = 1, both sides are equal to 1. Thus it can be concluded that the distributive law (7b) is valid.

Table 3.5 Proof of distributive law (7a) by perfect induction.

A B C	B + C	Left-hand-side of (7a) A (B + C)	A B	A C	Right-hand-side of (7a) AB + AC
0 0 0	0	0	0	0	0
0 0 1	1	0	0	0	0
0 1 0	1	0	0	0	0
0 1 1	1	0	0	0	0
1 0 0	0	0	0	0	0
1 0 1	1	1	0	1	1
1 1 0	1	1	1	0	1
1 1 1	1	1	1	1	1

Table 3.6 Proof of distributive law (7b) by the compact truth table method.

A	Left-hand-side of (7b) A + B C	Right-hand-side of (7b) (A + B)(A + C)
0	0 + B C = B C	(0 + B) (0 + C) = B C
1	1 + B C = 1	(1 + B) (1 + C) = 1•1 = 1

3.3 Sum-of-Products and Product-of-Sums Expressions

When a variable appears unprimed or primed in a switching expression, it is called a literal. An unprimed or a primed variable is also said to be, respectively, in true form or complemented form. If several literals are ANDed together, the result is called a product. Similarly, the ORing of several literals produces a sum. When several products are ORed together, the expression is called a sum-of-products (SOP) expression. In a sum-of-products expression, a product can have only one literal. Two examples for sum-of-products expression are given below. The first expression is a sum of three products. The second expression has four products, one of which is a single literal.

AB’ + BC + A’BD’

B’ + CD + A’C’D’ + AE’

A product-of-sums (POS) expression is the AND of several sums. In a product-of-sums expression, a sum may have just one literal. Two examples for product-of-sums expression are shown below. The first product-of-sums expression has three sums. The second expression also has three sums. But one of them is a single literal.

$$(A' + C')\,(A + C + D')\,(B + D')$$

$$C'\,(B' + D')\,\,(A + B + D)$$

Simplest (Minimal) Sum-of-Products and Product-of-Sums Expressions

Sum-of-products and product-of-sums are two fundamental expressions for Boolean functions. When a literal or a product is deleted from a sum-of-products expression for a switching function, the expression with deleted literal/product is no longer correct for the function. Then the sum-of-products expression is said to be simplest or minimal. In other words, a sum-of-products expression is simplest if and only if no literal or product can be deleted from the expression. Thus a simplest sum-of-products expression for a function consists of a minimum number of product terms and the total number of literals from all the product terms is also minimum. Two examples to illustrate what is a simplest sum-of-products expression are given in Section A.1 of Appendix.

Sum-of-products and product-of-sums expressions can be implemented as standard 2-level AND-OR and 2-level OR-AND circuits respectively as shown in Figure 6.1.

3.4 Theorems

The theorems given in this section are useful in simplifying switching expressions or in changing an expression to different forms. In proving a theorem by algebraic approach, only the basic laws and the theorems that have been proved will be used. When a basic law (L) or a theorem (T) is used, the basic law or the theorem number will be quoted inside a pair of square brackets at the end of a line.

(1) Combination theorem

(a) $A\,B + A\,B' = A$

(b) $(A + B)\,(A + B') = A$

Proof: (a)

$$\begin{aligned} \text{LHS} &= A\,B + A\,B' && \\ &= A\,(B + B') && [L7a] \\ &= A \bullet 1 && [L4b] \\ &= A = \text{RHS} && [L3a] \end{aligned}$$

(b)

$$\begin{aligned} \text{LHS} &= (A + B)\,(A + B') && \\ &= A + B\,B' && [L7b] \end{aligned}$$

$$\begin{aligned} &= A + 0 && [L2a'] \\ &= A = RHS && [L3b] \end{aligned}$$

The combination theorem allows two product (sum) terms to be combined to one product (sum) term. The number of literals in the resulting product (sum) is also reduced by one.

(2) Absorption theorem

(a) $A + A\,B = A$

(b) $A\,(A + B) = A$

Proof: (a)

$$\begin{aligned} LHS &= A + A\,B && \\ &= A \bullet 1 + A\,B && [L3a] \\ &= A\,(1 + B) && [L7a] \\ &= A \bullet 1 && [L3b'] \\ &= A = RHS && [L3a] \end{aligned}$$

(b)

$$\begin{aligned} LHS &= A\,(A + B) && \\ &= A\,A + A\,B && [L7a] \\ &= A + A\,B && [L2a] \\ &= A = RHS && [T2a] \end{aligned}$$

The following is an alternate proof of (b) using only basic laws.

(b)

$$\begin{aligned} LHS &= A\,(A + B) && \\ &= (A + 0)\,(A + B) && [L3b] \\ &= A + 0 \bullet B && [L7b] \\ &= A + 0 && [L3a'] \\ &= A = RHS && [L3b] \end{aligned}$$

When applying the absorption theorem, a product (sum) term will be absorbed by another term and disappears completely. The absorbed term is in fact part of the absorbing term. This is illustrated by Venn diagrams in Section A.2 of Appendix.

❖ Example 3.1

The absorption theorem is used to simplify two expressions in this example.

(a) $AC + AB'CDE = (AC) + (AC)\,(B'DE) = AC$

(b) $B'\,(A + B')\,(B' + CD') = B'\,(B' + CD') = B'$

Example 3.1 (b) can also be simplified using basic laws.

$B'\,(A + B')\,(B' + CD') = (B' + 0)\,(A + B')\,(B' + CD') = B' + (0)(A)(CD') = B'$

(3) Elimination theorem

(a) $A + A'B = A + B$

(b) $A(A' + B) = AB$

Proof: (a)

LHS	$= A + A'B$	
	$= (A + A')(A + B)$	[L7b]
	$= 1 \bullet (A + B)$	[L4b]
	$= A + B = \text{RHS}$	[L3a]

(b)

LHS	$= A(A' + B)$	
	$= AA' + AB$	[L7a]
	$= 0 + AB$	[L4a]
	$= AB = \text{RHS}$	[L3b]

In the elimination theorem, a literal is eliminated from a product (sum) term that has more literals.

❖ Example 3.2

This example illustrates the simplification of Boolean expression using the elimination theorem.

(a) $AC' + AB'CDE' = A(C' + B'CDE') = A[C' + C(B'DE')]$
$= A(C' + B'DE') = AC' + AB'DE'$

(b) $(B + C')(A + B + C' + D + E)$
$= (B + C')[(B + C') + (A + D + E)]$
$= B + C'$

❖ Example 3.3

Simplify the sum-of-products expression $(AB' + BCD + A'B'D')$.

By applying the elimination theorem

$AB' + BCD + A'B'D'$
$= BCD + B'(A + A'D')$
$= BCD + B'(A + D')$
$= AB' + BCD + B'D'$

Since $(AB' + BCD + A'B'D')$ can be simplified by removing a literal, it is not a simplest sum-of-products expression.

Some of the basic laws, the combination theorem, absorption theorem, elimination theorem and DeMorgan’s theorem are illustrated by Venn diagrams of Set Theory in Section A.2 of Appendix.

(4) Consensus theorem

(a) $A\,B + A'\,C + B\,C = A\,B + A'\,C$

(b) $(A + B)\,(A' + C)\,(B + C) = (A + B)\,(A' + C)$

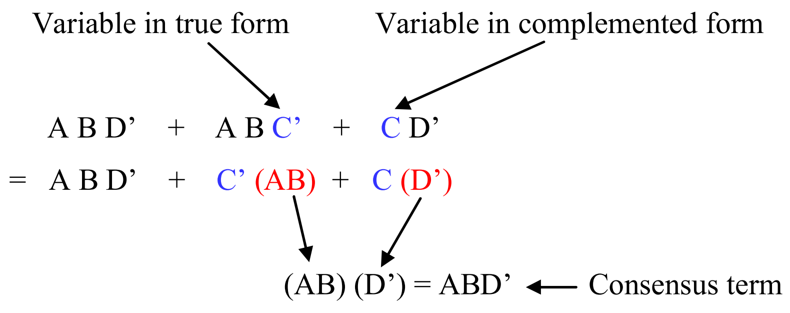
Proof: (a)

$$\begin{aligned}
\text{LHS} &= A\,B + A'\,C + B\,C \\
&= A\,B + A'\,C + 1\bullet B\bullet C && \text{[L3a]} \\
&= A\,B + A'\,C + (A + A')\,B\,C && \text{[L4b]} \\
&= A\,B + A'\,C + A\,B\,C + A'\,B\,C && \text{[L7a]} \\
&= (A\,B) + (A\,B)\,C + (A'\,C) + (A'\,C)\,B && \text{[L6a]} \\
&= A\,B + A'\,C = \text{RHS} && \text{[T2a]}
\end{aligned}$$

The proof of the second form of the consensus theorem is left as an exercise. The consensus theorem will remove a redundant term known as the “consensus term”. To find the consensus term, first look for a variable that appears in true form in one product and in complemented form in another product. This variable is A in the first form of the consensus theorem. A and A’ are ANDed with B and C respectively. The consensus term is BC. From the proof, it is seen that the consensus term BC is divided into two terms, one “absorbed” by AB and the other by A’C.

❖ Example 3.4

Simplify the sum-of-products expression (A B D’ + A B C’ + C D’) by eliminating a consensus term.



$= ABC' + CD'$

❖ Example 3.5

$B\,D' + A\,B'\,C' + A'\,C'\,D'$

The above expression is used to show how the consensus theorem can be used to simplify a Boolean expression in a manner different from that in Example 3.6. In

Example 3.6, a consensus term exists in a Boolean expression and is removed. In this example, the expression cannot be simplified before a consensus term is added to the expression.

$$\begin{tikzcd}
\text{Consensus term from A and A'} \arrow[r] & (B'C')\,(C'D') = B'C'D' \\
& B\,D' + A\,B'\,C' + A'\,C'\,D' \arrow[u] \arrow[u, shift left=3] \arrow[d] \arrow[d, shift right=3] \\
\text{Consensus term from B and B'} \arrow[r] & D'\,(A\,C') = AC'D'
\end{tikzcd}$$

Two consensus terms, B’C’D’ and AC’D’, can be generated from the variables A and B respectively. None of them exists in the expression. However, AC’D’ is added to the expression so that

$$BD' + AB'C' + A'C'D' = BD' + AB'C' + A'C'D' + AC'D'$$

By using the combination theorem,

$$A'C'D' + AC'D' = (A' + A)\,C'D' = C'D'$$

The expression is simplified to

$$BD' + AB'C' + C'D'$$

❖ Example 3.6

In this example, the application of consensus theorem is illustrated by adding a consensus term to a Boolean expression in order to eliminate two other terms in the original expression.

$$A'C + BCD + AC'D + AB'C'$$

In the above expression, a consensus term ABD can be generated from BCD and AC’D and are added to the expression.

$$A'C + BCD + AC'D + AB'C' = A'C + BCD + AC'D + AB'C' + ABD$$

BCD is a consensus term from A’C and ABD and can be eliminated from the expression.

$$A'C + BCD + AC'D + AB'C' + ABD = A'C + AC'D + AB'C' + ABD$$

AC’D is a consensus term derived from AB’C’ and ABD. Thus

$$A'C + AC'D + AB'C' + ABD = A'C + AB'C' + ABD$$

(5) Interchange Theorem

$$A\,B + A'\,C = (A + C)\,(A' + B)$$

Proof:

$$\begin{aligned}
\text{RHS} &= (A + C)\,(A' + B) \\
&= A\,A' + A\,B + A'\,C + B\,C && [L7a] \\
&= 0 + A\,B + A'\,C + B\,C && [L4a] \\
&= A\,B + A'\,C + B\,C && [L3b] \\
&= A\,B + A'\,C = \text{LHS} && [T4a]
\end{aligned}$$

The interchange theorem is not for simplification of Boolean expressions. It is used for the conversion of an expression from a sum-of-products to a product-of-sums, or vice versa. In applying this theorem from the conversion between a SOP expression and a POS expression, a variable should appear in true form in one product and in complemented form in the other product. Or the true form of a variable in one sum and its completed form in another sum. If such a variable does not exist, the theorem is not applicable. The conversion involves a process of interchanging literals. If the variable appears in true and complemented forms is A. The literals associated with A and A' in the sum-of-products (LHS of Theorem 5) are B and C respectively. To change the expression to a product-of-sums (RHS of Theorem 5), B will associate with A' and C becomes a partner of A. The process of interchanging associating literals is also applicable to the conversion from product-of-sums expression to sum-of-products expression.

Conversions between Sum-of-Products and Product-of-Sums Expressions

When a product-of-sums expression is converted to a sum-of-product expression, the process is called "multiplying-out". On the contrary, the change of a sum-of-products expression to a product-of-sums expression is called "factoring".

The distributive laws are always used in conversions between sum-of-products and product-of-sums. In fact, there are two opposite processes in the distributive laws. These two processes are shown below. Distribution is to distribute a literal to each and every literal in another term. Collection is to collect a common literal from each and every term. One process is the reverse of the other process.

Distributive law (7a)

$$A \bullet (B + C) \xrightarrow[\text{Distribution}]{} A \bullet B + A \bullet C$$

$$A \bullet (B + C) \xleftarrow[\text{Collection}]{} A \bullet B + A \bullet C$$