2.2.3: Manipulation of Algebraic Functions

An algebraic expression (or function) consists of
- binary variables *A*, *B*, *C*, *w*, *x*, *y*, *z*, etc
- constants **0** and **1**
- and logic operation symbols +, • and ’

Examples of algebraic functions:
$F = x\,y' + x'\,y + y\,z\,\mathbf{1}$ $G = A\,B\,C' + B'D + C\,D + \mathbf{0}$

Goal: given an algebraic function, use the properties of Switching Algebra to change it to an **equivalent** function. (Equivalent → same truth table)
This is done to either: (1) simplify, or (2) expand the original function.

First, some definitions:

- ***Literal***: a variable appearing in its true (***x***) or complement form (***x’***). **Each appearance** is counted.

  $F = a\,b' + b\,c'd + a'd + e'$ How many literals? _____

  High number of literals means complex algebraic expression.

- ***Product Term***: Single literals connected with **AND** operator.
  Examples: *a b’* *b c’d* *a’d* *w’x y z’* *e’*

  $F = w'x\,y\,z' + w\,x'y'z' + w\,x\,y + z'$ How many ***product*** terms? _____

  $F = x + w'y + w\,x\,y'z$ How many ***product*** terms? _____

  $F = a' + b + c$ How many ***product*** terms? _____

  $F = AB'$ How many ***product*** terms? _____

  $F = z$ How many ***product*** terms? _____

  w’ (*x* + *z’*) is not a product term. But *w’ x* + *w’z’* has of 2 product terms.

- ***Sum Term***: one or more <u>single</u> literals connected with **OR** operator.
  Examples: $a+b'$ $b+c+d'$ $x'+z$ z'

$F = (a+b') \bullet (b+c'+d) \bullet (a')$ How many ***sum*** terms? _____

$F = (w+x) \bullet (w+y)$ How many ***sum*** terms? _____

$F = w \bullet (x+y)$ How many ***sum*** terms? _____

$F = (w+x'+y'+z') \bullet (w'+x+y+z)$ How many ***sum*** terms? _____

In general, few (product or sum) terms and few literals → simpler function.

Two <u>standard</u> forms of algebraic functions:

- ***Sum of Products*** (***SOP***): one or more ***product*** terms connected with **OR** operators.
- ***Product of Sums*** (***POS***): one or more ***sum*** terms connected with **AND** operators.

NOTE: an expression could also appear in **neither** of these standard forms. In its given form:

$F = x'y + x\,y' + x\,y\,z$ SOP or POS? _____

$F = (x+y')\,(x'+y)\,(x'+z')$ SOP or POS? _____

$F = x' + y + z$ SOP or POS? _____

$F = x\,y\,z'$ SOP or POS? _____

$F = x\,(w' + yz)$ SOP or POS? _____

$F = z' + wx'y + v\,(x\,z' + w')$ SOP or POS? _____

Canonical Forms of Algebraic Expressions

An algebraic expression in which every input variable appears in each term.

Two canonical forms:

- ***Canonical Sum of Products* (*SOP*)**

$F(x,y,z) = x'y\,z + x\,y'z + x\,y\,z$	Canonical SOP form
$F(a,b) = a'b + ab'$	Canonical SOP form
$F(x,y,z) = x'y\,z + y'z$	Not Canonical form (missing x)

- ***Canonical Product of Sums* (*POS*)**

$F(x,y,z) = (x+y+z')(x'+y'+z')$	Canonical POS form
$F(a,b,c) = (a+b'+c)$	Canonical POS form
$F(x,y,z) = (x+y')\,(x'+y)\,(x'+z')$	Not Canonical form

Simplification of Algebraic Functions

Given an algebraic function, use the properties of Switching Algebra to simplify it to an **equivalent** expression, usually in *either* SOP or POS form.

Ex1. Simplify $F = a\,b + a\,b'$ into 1 term with 1 literal. [See P9a]

Ex2. Simplify $F = a\,(a' + b)$ into 1 term with 2 literals. [See P10b]

Ex3. Simplify $G = x'y + xy' + xyz$ into 3 terms with 6 literals.

Ex4. Simplify $H = a'b'c' + a' c' + a'b + a$.

Ex5. Simplify $F = (a'+b') (a'+b) (a'+c') (a)$.

Ex6. Simplify $H = a'b'c' + a'b\,c' + a'b\,c + a\,b'c'$

Why simplify algebraic functions? Because simpler functions result in simpler logic circuits!

2.3: Implementation of Algebraic Functions with Logic Gates

Once you have an algebraic function, you can design a logic circuit.

Ex1. Design a logic circuit for the function $F = x'y\,z + x\,y'\,z + x\,z'$

Ex2. Design a logic circuit for the function $H = (x + y)\,(x' + y' + z)$

NOTE: SOP or POS functions always lead to *two-level circuits*. (Even if the *NOT* gates are needed, these are still called "two-level circuits.")

Functions in neither SOP nor POS form result in a *multi-level circuit*.
Ex3. Design a logic circuit for the function $F = z' + wx'y + v(xz' + w')$

2.4: The Complement of a Function

The complement of a function F is another algebraic function denoted F', where all ones and zeros of the truth tables are inverted.

a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

a	b	c	G
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$F(a,b,c)$ and $G(a,b,c)$ are complements of each other.

That is, $G(a,b,c)$ is $F'(a,b,c)$

To find the complement of a function, use *DeMorgan's Theorem*.

P11a. $(a + b)' = a' \cdot b'$
P11aa. $(a + b + c \ldots)' = a' \cdot b' \cdot c' \ldots$

P11b. $(a \cdot b)' = a' + b'$
P11bb. $(a \cdot b \cdot c \ldots)' = a' + b' + c' \ldots$

See textbook pages 58-59 for proof. (Might show up as extra credit.)

Note that the prime is not distributive!!! $(a \cdot b)' \neq a' \cdot b'$ (Very typical error)
$(a + b)' \neq a' + b'$

Ex1. $F(w,x,y,z) = w\,x'y + x\,y' + w\,x\,z$. Find its complement function $F'(w,x,y,z)$.

Ex2. $G(a,b) = a\,b + a'b'$. Find its complement function $G'(a,b)$.

Ex3. $H(a,b,c) = (a+b+c')(a+b')$. Find its complement function $H'(a,b,c)$.

Ex4. Simplify the function $F(a,b,c) = (a\,c' + b\,c)'\,(a\,c + b')'$.

Shortcut for finding the complement of an algebraic function:
(1) Complement each variable (change *a* to *a'*, *a'* to *a*, *x* to *x'*, *z'* to *z*, etc)
(2) Interchange the elements 0 and 1 (change 0 to 1, and 1 to 0)
(3) Replace AND by OR, and OR by AND, being sure to preserve order of operations. Keep parenthesis intact. Sometimes, additional parentheses are required.

However, you must always remember how to use DeMorgan's theorem!

Ex1. $G(a,b) = a\,b + a'b'$. Find its complement function $G'(a,b)$.

Ex2. $F(x,y,z) = x + y'z'$. Find its complement function $F'(a,b,c)$.

Ex3. $F(a,b,c,d,e) = ab'(c+d'e) + a'bc'$. Find its complement function.