University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 120: Introduction to Computing

Boolean Expression Terminology

Let's Review and Define Some New Terms

literal a variable or its complement

examples: **A**, **A'**, **B**, **B'**, **C**, **C'**

sum several terms ORed together

examples: **A + B**, **AB + B(C + D) + A'C**, **A'B' + D(A ⊕ B)(C + A')**

product several terms ANDed together

examples: **AB**, **(A + B)(B + CD)(A' + C)**, **(A' + B')(D + (A ⊕ B) + CA')**

Minterms Were Useful for Proving Logical Completeness

minterm on N inputs

a product in which each variable or its complement appears exactly once (no other factors)

examples: **AB'**, **A'B**, **AB** (on inputs **A**, **B**)

AB'C, **AB'C'**, **A'BC'**

(on inputs **A**, **B**, **C**)

A Maxterm Produces a Function with One Zero Row

maxterm on N inputs

a sum in which each variable or its complement appears exactly once (no other terms)

examples: **(A + B')**, **(A' + B)**, **(A + B)**

(on inputs **A**, **B**)

(A + B' + C), **(A + B' + C')**,

(A + 'B + C')

(on inputs **A**, **B**, **C**)

Sum-of-Products (SOP) Form is Quite Common

sum-of-products (SOP)
a sum (OR)
of products (AND)
of literals

examples: **AB + BC**,

AB' + C + A'C'D',

but NOT **A(B + C) + D**

Product-of-Sums (POS) Form is Also Common

product-of-sums (POS)
a product (AND)
of sums (OR)
of literals

examples: **(A + B)(B + C)**,

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

but NOT **(A + BC)D**

Canonical Forms Allow Easy Comparison, But Are Too Big

canonical SOP
a sum of minterms; the expression produced by the logical completeness construction

canonical POS
a sum of maxterms

What does canonical mean?

Unique (if we assume an ordering on variables).

Too many terms to be of practical value.

Do You Know Mathematical Implication?

What does A→B mean?

A implies B.

In other words: **if A is true, B is also true.**

What if **A is false?**

In that case, **is A→B true or false?**

If A is false, A→B is true.

So the Following Odd Statements are True

All **purple elephants** can fly.

(X is a **purple elephant** → X can fly.)

Students who score **above 125%** in ECE120 fail the class.

(X scored **above 125%** → X fails.)

In both, **the premise is false for any X**, so the **implications are true**.

One Function Can Imply Another

A function **G is an implicant of** a second function **F iff G** operates on the same variables as **F** and **G→F**.

In other words, every row
- with an output of 1 in **G**'s truth table
- also has an output of 1 in **F**'s truth table.

0 rows in **G**'s truth table do not matter.

For Our Purposes, Implicants are Products of Literals

In digital design, we only refer to products of literals as implicants.

So we will **assume that an implicant can be written as a product of literals**.

We Can Use Implicants to Simplify Functions

As a first step towards simplifying a function **F**, we can ask:

Given an implicant G of F, can we remove any of its literals and obtain another implicant of F?

For example, take **F = AB'C + ABC' + ABC**.

The first term (**AB'C**) is an implicant.

Can we remove any literals?

Try to Remove Each Literal to Find Only AC Implies F

Start from **AB'C** and try to remove each literal.

B'C is not an implicant.

AC is an implicant.

AB' is not an implicant.

A	B	C	F	B'C	AC	AB'
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	1
1	0	1	1	1	1	1
1	1	0	1	0	0	0
1	1	1	1	0	1	0

We Remove as Many Literals as We Can

So we can simplify **F** by replacing **AB'C** with **AC**:

F = AC + ABC' + ABC

Checking the second term (**ABC'**), we find that we can eliminate **C'** to obtain:

F = AC + AB + ABC

In the third term (**ABC**), we can eliminate **B** or **C**, but not both. Let's pick **B**.

F = AC + AB + AC

Prime Implicants Have a Minimal Number of Literals

F = AC + AB + AC

But now we have a duplicate term, which we can eliminate to arrive at a simple form for **F**:

F = AC + AB

We can remove no more literals.

One more definition: An implicant **G** of **F** is a **prime implicant of F** iff **none of the literals in G can be removed** to produce other implicants of **F**.

AB and AC are prime implicants of F.

To Simplify, Write Function as a Sum of Prime Implicants

The conclusion is obvious:

To simplify a function F,
write it as a sum of prime implicants.

Enjoy the algebra.

Good luck!

(Next time, we'll develop a graphical tool that lets us skip the algebra.)