Second Tison Method

Suppose you have a list of prime implicants and you want to enumerate all of the irredundant sums of $\mathcal{F}$?

Example:
Suppose we solve this as before with Petrick’s method.

<table>
<thead>
<tr>
<th>PI</th>
<th>cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = w'y$</td>
<td>2,3,6,7</td>
</tr>
<tr>
<td>$b = xy$</td>
<td>6,7,14,15</td>
</tr>
<tr>
<td>$c = yz$</td>
<td>3,7,11,15</td>
</tr>
<tr>
<td>$d = wxz'$</td>
<td>12,14</td>
</tr>
<tr>
<td>$e = wy'z'$</td>
<td>8,12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>minterm</th>
<th>Petrick</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$A$</td>
</tr>
<tr>
<td>3</td>
<td>$A + C$</td>
</tr>
<tr>
<td>6</td>
<td>$A + B$</td>
</tr>
<tr>
<td>7</td>
<td>$A + B + C$</td>
</tr>
<tr>
<td>8</td>
<td>$E$</td>
</tr>
<tr>
<td>11</td>
<td>$C$</td>
</tr>
<tr>
<td>12</td>
<td>$D + E$</td>
</tr>
<tr>
<td>14</td>
<td>$B + D$</td>
</tr>
<tr>
<td>15</td>
<td>$B + C$</td>
</tr>
</tbody>
</table>

After reduction we have

\[
p = ACE(B + D)
= ABCE + ACDE
\]
**Step 1:** Index each PI with a unique letter to form the initial list $L$. (We will use capital letters since we eventually get a Petrick expression.) For our example:

<table>
<thead>
<tr>
<th>PI</th>
<th>Element of $L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a = w'y$</td>
<td>$w'yA$</td>
</tr>
<tr>
<td>$b = xy$</td>
<td>$xyB$</td>
</tr>
<tr>
<td>$c = yz$</td>
<td>$yzC$</td>
</tr>
<tr>
<td>$d = wxz'$</td>
<td>$wxz'D$</td>
</tr>
<tr>
<td>$e = wy'z'$</td>
<td>$wy'z'E$</td>
</tr>
</tbody>
</table>
Step 2: For each biform variable, $v$, in the original PI list perform the following two operations:

Step 2.1: For every pair $S$ and $T$ in $L$ add to $L_{cons}(S, T, v)$. When computing the consensus treat the indices as if they were switching variables.

Step 2.2: Delete from $L$ all terms $S$ such that $S$ implies another term, $T$, in $L$, i.e., $T$ covers $S$.

Note for the example $w$, $y$, and $z$ are biform. $x$ is monoform.
**Process \( w \):**

The set \( L \) is:

\[
\begin{align*}
wxz'D & \quad w'yA & \quad xyB \\
wy'z'E & & yzC
\end{align*}
\]

\[
\begin{align*}
\text{cons}(wxz'D, w'yA) &= xyz'AD \\
\text{cons}(wy'z'E, w'yA) &= \emptyset
\end{align*}
\]

No covers so the new \( L \) is:

\[
\begin{align*}
wxz'D & \quad w'yA & \quad xyB \\
wz'y'z'E & \quad xyz'AD & \quad yzC
\end{align*}
\]
**Process $y$:**

The set $L$ is:

- $w'yA$
- $wy'z'E$
- $wxz'D$
- $xyz'AD$
- $yzC$
- $xyB$

$$
\text{cons}(w'yA, wy'z'E) = \emptyset \\
\text{cons}(xyz'AD, wy'z'E) = wxz'ADE \\
\text{cons}(yzC, wy'z'E) = \emptyset \\
\text{cons}(xyB, wy'z'E) = wxz'BE}
$$

$wxz'D$ covers $wxz'ADE$ so $L$ is now:

- $w'yA$
- $wy'z'E$
- $wxz'D$
- $xyz'AD$
- $wxz'BE$
- $xyB$
- $yzC$
Process $z$: The set $L$ is:

$$
\begin{array}{c}
yzC \quad wy'z'E \quad w'yA \\
wxz'D \quad xyB \\
xyz'AD \\
wxz'BE
\end{array}
$$

$$
\begin{align*}
\text{cons}(yzC, wy'z'E) &= \emptyset \\
\text{cons}(yzC, wxz'D) &= wxyCD \\
\text{cons}(yzC, xyz'AD) &= xyACD \\
\text{cons}(yzC, wxz'BE) &= wxyBCE
\end{align*}
$$

$xyB$ covers $wxyBCE$.

New $L$ is:

$$
\begin{array}{c}
yzC \quad wy'z'E \quad w'yA \\
wxyCD \quad wxz'D \quad xyB \\
xyACD \quad xyz'AD \quad wxz'BE
\end{array}
$$
Step 3: Remove from $L$ any term whose variables portion is not a PI, i.e., ignore the indices.

Final list from Step 2 is

$$L = (wzx'BE, wy'z'E, wxyCD, wxz'D, w'yA, xyB, xyACD, xyz'AD, yzC)$$

Only the following are PI's in the variables portion

$L = (wzx'BE, wy'z'E, wxz'D, w'yA, xyB, xyACD, yzC)$
Step 4: Form a Petrick expression for each PI that consists of the sum of the indices with which the PI appears in \( L \). Form the product of these sums into a single Petrick expression for \( \mathcal{F} \).

\[
\begin{align*}
w'y & : A \\
x'y & : B + ACD \\
yz & : C \\
w'xz & : D + BE \\
w'y'z' & : E
\end{align*}
\]

Form the product

\[
\mathcal{F} : ACE(B + ACD)(D + BE)
\]
Step 5: Obtain a sum of product form for the Petrick expression for $\mathcal{F}$ using distribution, idempotence, and absorption. Each term of the resulting simplified sum corresponds to an irredundant sum of $\mathcal{F}$ and all irredundant sums must have a corresponding term.

\[
ACE(B + ACD)(D + BE) = ACE(A + B)(B + C')
\]
\[
(B + D)(B + D)(D + E)
\]
\[
= ACE(B + D)
\]
\[
= ABCE + ACDE
\]

So we have two irredundant sums for $\mathcal{F}$

\[
\mathcal{F} = w'y + xy + yz + wy'z'
\]
\[
\mathcal{F} = w'y + wxz' + yz + wy'z'
\]

The first is minimal due to literal count.
Consensus and Cover tests can be implemented via the binary representation of the algebraic expression of the PI’s and subsequently created terms.

**Consensus:** Suppose we are working with 4 variables \( w, x, y, \) and \( z. \)

\[
\begin{align*}
    w'xyz' & \rightarrow 0110 \\
x'y'z' & \rightarrow x010
\end{align*}
\]

To take the consensus with respect to \( x \) we must have two patterns such that one has a 0 in bit 2 and the other a 1.

In order for to satisfy the consensus condition the rest of the bits must match when \( x \) is considered a don’t care condition. (If there was a position that had 1 and 0 in the two patterns it means that some variable other than \( x \) appeared complemented and uncomplemented in the two expressions.)

The bit pattern of the consensus is merely the common bit pattern where 0 or 1 are chosen over an \( x \) when an \( x \) is present. The bit position where the variable that generated the consensus is set to \( x \) as is any position which was \( x \) in both patterns.

The consensus with respect to \( x \) of the patterns above is \( w'y'z' \rightarrow 0x10. \)
Implication Detection

Consider the three expressions and their binary representations:

\[
\begin{align*}
w'xyz' & \rightarrow 0110 \\
x'yz' & \rightarrow x010 \\
yz' & \rightarrow xx10
\end{align*}
\]

If \( S \rightarrow T \) then \( S = TR \) for some product \( R \), i.e., if all literals in \( T \) appear in \( S \) then \( T \) covers \( S \).

In binary form this means that if we view the \( x \)'s as don't cares if one pattern equals a second in all positions that are not \( x \) then the expression corresponding to the first pattern covers the expression corresponding to the second.

\( x'yz' \) does not cover \( w'xyz' \) because they differ in bit 2.

\( yz' \) covers both \( x'yz' \) and \( w'xyz' \) because its pattern is a subset of the other two, i.e., all three have 10 in bits 3 and 4 and \( yz' \) has \( x \)'s everywhere else.
Now consider what the First Tison method does to some simple K-map examples and compare it to what the QM tabulation procedure does.

\[ F(x,y,z) = \sum (4,5,6,7) \]

\[ = xy'z' + xy'z + xyz' + xyz \]

0–cubes for QM
(always start with minterms in QM)
2-cube 4,5,6,7. $x$ is found but it is produced repeatedly.

4,5 and 6,7: $xy' + xy = x$

4,6 and 5,7: $xz' + xz = x$

Clearly there is some unnecessary work in QM tabulation.
First Tison on Same Example

\[ F(x,y,z) = \sum (4,5,6,7) \]

\[ = xy'z' + xy'z + xyz' + xyz \]

Start with minterms and take consensi w.r.t. \( y \)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

\[ x \ y \ y' \ z \quad x \ y \ y' \ z' \quad \text{yields} \quad x \ z' \quad 4,6 \]

\[ x \ y \ z \quad x \ y' \ z \quad x \ z \quad 5,7 \]

\[ \text{take consensus w.r.t. } z \]

\[ \text{cons}(x \ z', x \ z, z) = x \]

\[ 4,5,6,7 \]
F(x,y,z) = \sum (4,5,6,7)
= xy'z' + xy'z + xyz' + xyz

Start with minterms and take consensi w.r.t. z

x y' z
x y z
x y' z'  yields  x y  6,7
x y z  x y' z'  x y'  4,5

Take consensus w.r.t. y
cons( x y , x y' ) = x
4,5,6,7
• Either choice of biform variable elimination yielded the same final PI.

• Not all of the 1-cubes were formed in First Tison only those needed to get $x$.

• In general, First Tison does not produce all implicants like QM.

• We could have started with a sum-of-products other than the canonical sum of minterms. Either of the following two starting points would have worked (as well as others):

\[
F(x, y, z) = xz' + xz \\
= xy + xy'
\]
Consider the following example via QM tabulation.

\[
\begin{array}{c|c|c|c|c}
XY & 00 & 01 & 11 & 10 \\
\hline
Z \ 0 & 0 & 1 & 2 & 3 \\
\hline
1 & 1 & 3 & 7 & 1 \\
\hline
\end{array}
\]

\[F(x, y, z) = \sum (2, 5, 6, 7)\]
\[= x'y'z + xy'z + x'yz + xyz\]

3 1-cubes that are also the PI’s are produced.

2,6 \quad yz' \\
5,7 \quad xz \\
6,7 \quad xy
F(x, y, z) = yz' + xz

2,6 and 5,7 cover F

Start at a different point that is not a sum of minterms.

Note we are missing a PI and the two we have do not line up nicely as before.

Take consensus w.r.t. z

cons(yz', xz) = xy which is 6,7

Also note that there is no deletion due to implication in step 2.1.
\[ F(x, y, z) = y z' + x y + x y' z \]

2, 6, 6, 7, 5

A minterm and 2 PI's. All implicants due to SOP.

\[ L = (y z', x y, x y' z) \]

\begin{align*}
\text{cons} \ (y z', x y' z, z) &= \emptyset \quad \text{no useful combination} \\
\text{cons} \ (y z', x y' z, y) &= \emptyset \\
\text{cons} \ (x y, x y' z, y) &= x z \quad \text{note K-map does not apply}
\end{align*}

\[ x z \text{ covers } x y' z \text{ so } L = (y z', x y, x z) \]

![K-map diagram](image-url)
Now consider the Second Tison method for enumerating irredundant sums applied to

\[ \mathcal{F}(w, x, y, z) = y'z + w'x'y' + w'xz + wxy' + wx'z \]

Which as can be seen on the K-map below is a redundant sum of PI's.

PI 1,5,9,13  y'z is clearly redundant
0,1 5,7 12,13 9,11 are all essential
Petrick expression: \((A+BCDE)BCDE = BCDE\)

Multiple consensus operations used to build different forms of \(y'z\)

Labels track the leaves involved in producing any node in the consensus tree

The Petrick expression generated from only original PI's enumerates the different ways one PI can be covered by combinations of the other PI's

Algebraic reduction of the Petrick expression exposes and eliminates redundant representations of PI's and yields all irredundant sums
The tree also shows how to construct an algebraic derivation of the fact that $y'z$ is redundant given the other PI’s are present in a sum.

\[
\begin{align*}
a &+ (b + c) + (d + e) \\
a &+ (b + c + w'y'z) + (d + e) \\
a &+ (b + c + w'y'z) + (d + e + wy'z) \\
(a + wy'z + wy'z) &+ (b + c) + (d + e) \\
(w'y'z + wy'z) &+ (b + c) + (d + e) \\
(b + c + w'y'z) &+ (d + e + wy'z) \\
(b + c) &+ (d + e)
\end{align*}
\]
In the previous example, we saw the conclusion that PI’s b, c, d, and e covered PI a built up via several consensi.

Consider

\[(w'x'y', w'xz, wxy', wxz)\]

\(w\) and \(x\) are biform variables (these are the variables with respect to which we took consensi)

\(y\) and \(z\) are monoform variables

Note that each \(T_i\) is written \(T_i = B_iX_i\) where \(B_i\) is a product of biform variables and \(X_i\) is a set of monoform variables.

Combining all of the consensi into a single operation defines an object that is the fundamental link of the concept prime implicants to an arbitrary SOP for a switching function.
**Definition:** Let \((T_1, \ldots, T_p)\) be a set of product terms of the form \(T_i = B_i X_i\) where \(B_i\) is a product of biform variables and \(X_i\) is a product of monoform variables. Set \(X = X_1 X_2 \cdots X_p\) (omitting duplicate literals).

\(X\) is the **generalized consensus**, \(GC(T_1, \ldots, T_p)\), of \(T_1, \ldots, T_p\) if

\[ B_1 + B_2 + \cdots + B_p = 1 \]

irredundantly, i.e., there are no extra terms in the sum. (Note: redundant terms are determined without performing any additional Boolean transformations to the sum.)

**Example 1:**

\[ y'z = GC(w'x'y', w'xz, wxy', wx'z) \]

since \(w'x' + w'x + wx + wx' = 1\) irredundantly.

**Example 2:**

\((wx, w'y, w'z)\)

do not have a GC since since \(w + w' + w' = 1\) is redundant.
The example of the first portion of Tison’s Second method conforms to the definition of generalized consensus proposed earlier. It merely adds the indices in to Example 1 as monoform variables:

**Example 1 (modified):**

\[ y'z BCDE = GC((w'x')(y'B), (w'x)(zC), (wx)(y'D), (wx')(zE)) \]

Only the monoform variables have increased so the GC changes but the irredundant sum

\[ w'x' + w'x + wx + wx' = 1 \]

stays the same.
Some interesting facts become easy to prove given Tison’s methods.

**Fact 1:** Suppose \( f = T_1 + T_2 + \cdots + T_n \) is a sum of products form for the switching function \( f \). Assume that nothing more than the equality above is known about the sum, i.e., do not assume that it is minimal or irredundant or complete or anything else special. Suppose the literal \( x \) appears in the sum but \( x' \) does not. It follows that \( x' \) does not appear in any irredundant sum of \( f \).

**Proof:** Run Tison’s first method on \( f = T_1 + T_2 + \cdots + T_n \) to get the list of PI’s. No new literals are created by consensi operations so if \( x' \) did not appear in the original sum it does not appear in an PI. Any irredundant sum is a sum of PI’s and therefore \( x' \) does not appear in any irredundant sum. \( \square \)
Fact 2: Suppose for a function $f(w, x, y, z)$ one irredundant sum contains the literal $x'$. It follows that $x'$ must appear in every irredundant sum of $f$.

Proof: Suppose $x'$ appears in a prime implicant $P$ in some irredundant sum. Suppose there exists another irredundant sum $f = P_1 + \cdots + P_k$ which does not contain $P$ or $x'$. Enumerate the prime implicants of $f$ via first Tison applied to $(P_1, \ldots, P_k)$. $P$ must be on the list and therefore the list must contain $x'$. But recall that first Tison only works with the literals in the terms on the list, i.e., $(P_1, \ldots, P_k)$. However, $x'$ was assumed to not be in that list so there is a contradiction and $x'$ must appear in any irredundant sum. □
Fact 3: Suppose \( f(x_1, \ldots, x_n) \) has PI's \( a_1, \ldots, a_k \) and a prime implicant function as described in the notes.

\[
p(A_1, \ldots, A_k) = I_1 + \cdots + I_{n_s}
\]

where each \( I_j \) is a product of some of the \( A \)'s and corresponds to the selection of a set of PI's that make an irredundant sum. It follows that the \( I_j \)'s are the PI's of the prime implicant function \( p(A_1, \ldots, A_k) \).

Proof: Recall that no complement, \( A'_j \), appears in the prime implicant function SOP. Furthermore, if \( p \) has been reduced via absorption and equality to the terms corresponding to irredundant sums then there is no \( I_j \) which is covered by another term \( I_k \). Therefore, by Tison's first method the \( I_j \)'s are the PI's since no consensi exist and no terms must be removed from the list due to coverage. \( \square \)
Definition: Suppose $f(x_1,\ldots,x_n)$ has a SOP representation whose literals have the following properties:

- If the literal $x_i$ appears in the SOP then the literal $x'_i$ does not.

- If the literal $x'_i$ appears in the SOP then the literal $x_i$ does not.

$f$ is called a unate function. (Note that you do not necessarily have to know the specific SOP—only that it exists.)
**Fact 4:** If $f$ is a unate function then its complete sum is also its unique irredundant (and therefore unique minimal) sum.

**Proof:** Since $f$ is unate it must have an SOP representation which satisfies the conditions above. This definition assures that all $x_i$ are monoform in the SOP. Therefore if we simplify by equality and absorption as far as possible we are left with an SOP

$$f = a_1 + \cdots + a_k$$

which Tison’s first method will not alter. (No coverage and no consensi.) It is therefore the complete sum.

Now consider Tison’s second method. Since the complete sum consists of PI’s with only monoform variables, no consensi exist and the resulting Petrick expression for the prime implicant function $p$ consists of a single term

$$p = A_1 A_2 \ldots A_k$$

which implies a single irredundant sum which is also the complete sum and the unique minimal sum. □.
Recognizing unate functions can save work. For example, consider the following two functions:
\[
\begin{align*}
\mathcal{F}(w, x, y, z) &= (w + z')(x + y') + wxy + x'y'z' \\
\mathcal{G}(w, x, y, z) &= w'(x' + y + z')
\end{align*}
\]
and determine the multiple output prime implicants.

The PI’s can be easily deduced by multiplying out the functions and simplifying by absorption. We get
\[
\begin{align*}
\mathcal{F} &= wx + wy' + xz' + y'z' \\
\mathcal{G} &= w'x' + w'y + w'z'
\end{align*}
\]
Since there are no biform variables in either the terms are the PI’s by Tison’s first method.

Multiplying the two complete sums yields
\[
\mathcal{F}\mathcal{G} = w'xz' + w'y'z'
\]
whose terms are also the PI’s due to Tison’s first method.

The algebraic properties of unate functions are used heavily in the package ESPRESSO II for synthesizing two-level network with near minimal cost.