**Binary Recursion Tree**

The recursive Shannon expansion corresponds to a binary recursion tree

\[
f = xf_1 + xf_2
\]

\[
= x(yf_{xy} + yf_{x\bar{y}}) + x(yf_{\bar{x}y} + yf_{\bar{x}\bar{y}})
\]

Example:

Path \( \pi(v) \) to node \( v \) corresponds to cube \( c^{\pi(v)} \)

Example: \( c^{\pi(v)} = x_1 \bar{x}_2 \bar{x}_3 \)

---

**Binary Recursion Tree**

- The root represents the original function \( f \).
- Each node \( v \) corresponds to \( f_{c^{\pi(v)}} \).
- If ever \( f_{c^{\pi(v)}} = 1 \) or \( 0 \) we can terminate the tree and replace \( v \) by 1 or 0. Such a node \( v \) is a leaf.
**Example**

\[ f = ab + ac \]

[Diagram of a tree with labeled nodes and arrows indicating variable splitting.]

**Implicit Enumeration - Branch and Bound**

Checking for tautology and many other theoretically intractable problems (co-NP complete) can be effectively solved using implicit enumeration:

- use recursive Shannon expansion to explore \( B^n \).
- In (hopefully) large subspaces of \( B^n \), prune the binary recursion tree by
  - exploiting properties of the node function \( f_c(x) \)
  - exploiting heuristic bounding techniques
- even though in the worst case the recursion tree may have \( 2^n \) nodes, in practice (in many cases), we typically encounter a linear number of nodes
**Implicit Enumeration – Branch and Bound**

- Thus we say that the $2^n$ min-terms of $f$ have been implicitly enumerated
- BDD’s (Binary Decision Diagrams) are alternate representations in which implicit enumeration is performed statically, and nodes with identical path cofactors are identified
  (very important -- will discuss later!)

**Example**

$f = ab + ac$

Not a tautology. In testing for tautology, we look for a cube subspace $c$ such that $f_c = 0$. If we can find it then $f$ is not the tautology.
• Can rule out complete cube subspace c' if $f_c = 1$

Means that $f$, in the subspace $c' = \overline{ac}$, is identically 1

• Tautology can be proved by finding $\{c_i\}$ such that $\Sigma c_i = 1$ and $f_{c_i} \equiv 1$ for all $c_i$. We don't need that $c_i, c_j = \emptyset$.

Definition 1 A function $f : B^n \rightarrow B$ is positive unate in variable $x_i$ iff

$$f_{\overline{x_i}} \subseteq f_{x_i}$$

This is equivalent to monotone increasing in $x_i$:

$$f(m^-) \leq f(m^+)$$

for all min-term pairs $(m^-, m^+)$ where

$$m^-_j = m^+_j, j \neq i$$

$$m^-_i = 0$$

$$m^+_i = 1$$

For example, $m^-_3 = 1001$, $m^+_3 = 1011$(where i=3)
Similarly for negative unate
monotone decreasing: \( f(m^-) \geq f(m^+) \)

A function is unate in \( x_i \) if it is either positive unate or negative unate in \( x_i \).

Definition 2 A function is unate if it is unate in each variable.

Definition 3 A cover \( f \) is positive unate in \( x_i \) iff \( \overline{x}_i \notin c_j \) for all cubes \( c_j \in \mathbb{F} \).

Example 1

\[ f = ab + \overline{bc} + ac \]

Positive unate in \( a, b \)
Negative unate in \( c \)

\( f(m^-) = 1 \geq f(m^+) = 0 \)
The Unate Recursive Paradigm

- In the EXPRESSO program, the key pruning technique is based on exploiting the properties of unate functions.
- In particular, the splitting variable is chosen so that the functions at lower nodes of the recursion tree become unate.

Unate covers F have many extraordinary properties:
  - If a cover F is minimal with respect to single-cube containment, all of its cubes are essential primes.
  - In this case F is the unique minimum cube representation of its logic function.
  - A unate cover represents the tautology iff it contains a cube with no literals, i.e. a single tautologous cube.

This type of implicit enumeration applies to many sub-problems (prime generation, reduction, complementation, etc.). Hence, we refer to it as the Unate Recursive Paradigm.
The Binate Select Heuristic

Tautology and other programs based on the unate recursive paradigm use a heuristic called BINATE_SELECT to choose the splitting variable in recursive Shannon expansion. The idea is for a given cover $F$, choose the variable which occurs, both positively and negatively, most often in the cubes of $F$.

The Binate Select Heuristic

Example 2 Unate and non-unate covers:

\[ G = ac + cd \]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

is unate

\[ F = ac + cd + bcd \]

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

is not unate

Choose $c$ for splitting.
The binate variables of a cover are those with both 1’s and 0’s in the corresponding column.

In the unate recursive paradigm, the BINATE_SELECT heuristic chooses a (most) binate variable for splitting, which is thus eliminated from the sub-covers.
**Examples**

\[ f = ab + cd + bcd \]

**Tautology**

\[
F = acd + bcd + ab\overline{d} + \overline{acd} + \overline{cd} + ac \\
+ \overline{ad} + \overline{bcd} + \overline{abd} + \overline{abc}
\]

*Is $F = 1$? NOT EASY!!*

\[
F = \\
1211 \\
2111 \\
0120 \\
0200
\]

\[
F = \\
2201 \\
1202 \\
1220 \\
2010 \\
0021 \\
0012
\]
Two Useful Theorems

Theorem 1 \[ F \equiv 1 \iff (F_{x_j} \equiv 1) \wedge (F_{x_j} \equiv 1) \]

Theorem 2 Let \( A \) be a unate cover matrix. Then \( A \equiv 1 \) if and only if \( A \) has a row of all 2's.

Proof:
If a row of all 2's is the tautology cube.
Only if. Assume no row of all 2's. Without loss of generality, suppose function is positive unate. Then each row has at least one "1" in it. Consider the point \((0,0,\ldots,0)\). This is not contained in any row of \( A \). Hence \( A \not\equiv 1 \).

Unate Reduction of Tautology Checking

Let \( F(x) \) be a cover. Let \((a,x')\) be a partition of the variables \( x \), and let
\[
F = \begin{bmatrix}
A & X \\
T & F'
\end{bmatrix}
\]

where
- the columns of \( A \) correspond to variables \( a \) of \( x \)
- \( T \) is a matrix of all 2's.

Theorem 3 Assume \( A \not\equiv 1 \). Then \( F \equiv 1 \iff F' \equiv 1 \)
**Example 3**

\[ F = \begin{bmatrix} A & X \\ T & F \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \\ 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 2 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 2 & 1 \end{bmatrix} \]

---

**Unate Reduction**

<table>
<thead>
<tr>
<th>( A_1 )</th>
<th>0 &amp; 1</th>
<th>( B_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 &amp; 2 &amp; 2 &amp; 2 &amp; 2 &amp; 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 &amp; 2 &amp; 2 &amp; 2 &amp; 2 &amp; 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 &amp; 2 &amp; 2 &amp; 2 &amp; 2 &amp; 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( A_2 )</th>
<th>0 &amp; 1</th>
<th>( B_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 &amp; 0 &amp; 1 &amp; 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Result: Only have to look at \( D_1 \) to test if this is a tautology.

Note: \( A_1, A_2 \) has no row of all 2's. Hence is a unate cover. Hence \((A_1, A_2) \neq 1\)
End of Lecture 2

\[ F = \begin{bmatrix} A & X \\ \hline T & F' \end{bmatrix} \quad \begin{array}{c} A \neq 1 \\ T = 2's \end{array} \]

**Theorem 1** Assume \( A \neq 1 \). Then \( F = 1 \iff F' = 1 \)

Proof: [if]: Assume \( F' = 1 \). Then we can replace \( F' \) by all 2's. Then last row of \( F \) becomes a row of all 2's, so tautology.
Proof (contd)

[Only if]: Assume $F' \neq 1$. Then there is a minterm $m_2$ such that $F'(m_2) = 0$, i.e. $m_2 \notin$ cube of $F'$. Similarly, $m_1$ exists where $A(m_1) = 0$, i.e. $m_1 \notin$ cube of $A$. Now the minterm $(m_1, m_2)$ in the full space satisfies $F(m_1, m_2) = 0$ since $m_1 m_2 \notin AX$ and $m_1 m_2 \notin TF'$.

$(a, x')$ is any row of first part
$$a(m_1) \land x(m_2) = 0 \land x'(m_2) = 0$$

$(t, f')$ is any row of the last part
$$t(m_1) \land f'(m_2) = t(m_1) \land 0 = 0$$

So $m_1 m_2$ is not in any cube of $F$.

Unate Reduction for Tautology

Procedure TAUTOLOGY($F, C$)
// $C$ is a cube returned if $F \neq 1$. Then $C$
// contains a minterm $m$ where $F(m) = 0$

\[ T \gets \text{SPECIAL\_CASES}(F) \]
if ($T \neq 1$) return $T$

\[ F \gets \text{UNATE\_REDUCTION}(F) \]
if ($F = \emptyset$) print $C$; return 0;

\[ j \gets \text{UNATE\_SELECT}(F) \]
\[ T \gets \text{TAUTOLOGY}(F_{x_j} \cup \{x_j\}) \]
if ($T = 0$) print $(C \cup \{x_j\})$, return 0

\[ T \gets \text{TAUTOLOGY}(F_{\bar{x}_j} \cup \{x_j\}) \]
if ($T = 0$) print $(C \cup \{x_j\})$, return 0
return 1

end
Unate Reduction for Tautology

Notes.
T=1(0) if F is a tautology (is empty), else T=-1

SPECIAL_CASES: (T=-1 unless)
- T=1: F contains a cube with no literals (all 2's)
- T=0: F contains same literal in every cube
- T=0 if number of minterms in onset is < 2^n
**Unate Recursive Paradigm**

- Create cofactoring tree stopping at unate covers
  - choose, at each node, the "most binate" variable for splitting
  - recurse till no binate variable left (unate leaf)
- "Operate" on the unate cover at each leaf to obtain the result for that leaf. Return the result
- At each non-leaf node, merge (appropriately) the results of the two children.

![Diagram of unate recursive paradigm]

- Main Idea "Operation" on unate leaf is easy
- Operations: complement, simplify, tautology, generate-primes...

---

**Operations on a Unate Cover: Complement**

- Map cube matrix $M$ into Boolean matrix $B$

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Thus non-2 $\rightarrow 1$

2 $\rightarrow 0$

![Matrix diagram]

$B = \begin{bmatrix}
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1
\end{bmatrix}$
Complement of a Unate Cover

Find all minimal column covers of B. (A column cover is a set of columns J such that for each row i, \( \exists j \in J \) such that \( B_{ij} = 1 \))

Example 4 \( (1,4) \) is a minimal column cover for

\[
\begin{array}{c|c|c|c|c}
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
\end{array}
\rightarrow
\begin{array}{c}
1 \\
1 \\
1 \\
1 \\
\end{array}
\]

All rows "covered" by at least one 1.

Complement of a Unate Cover

- For each minimal column cover create a cube with opposite column literal from \( M \).

Example 5 \( (1,4) \) \( \xrightarrow{\text{ad}} \) \( \tilde{c} \) is a cube of \( \tilde{f} \)

\[
\begin{array}{c|c|c|c|c}
2 & 1 & 2 & 0 & 2 \\
2 & 2 & 0 & 0 & 1 \\
1 & 1 & 2 & 2 & 1 \\
1 & 2 & 0 & 2 & 1 \\
\end{array}
\rightarrow
\begin{array}{c|c|c|c|c}
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
\end{array}
\]
**Complement of a Unate Cover**

The set of all minimal column covers = cover of $\overline{f}$.

**Example 6**

$$
\begin{array}{cccc}
2 & 1 & 2 & 0 & 2 \\
2 & 2 & 0 & 0 & 1 \\
1 & 1 & 2 & 2 & 1 \\
1 & 2 & 0 & 2 & 1
\end{array} \rightarrow
\begin{array}{cccc}
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1
\end{array}
$$

\{(1,4), (2,3), (2,5), (4,5)\} is the set of all minimal covers. This translates into:

$$
\overline{f} = \overline{a}d + \overline{b}c + \overline{b}e + d\overline{e}
$$

---

**Unate Complement Theorem**

Theorem 4 Let $M$ be a unate cover of $f$. The set of cubes associated with the minimal column covers of $B_M$ is a cube cover of $\overline{f}$.

Proof. We first show that any such cube $c$ generated is in the offset of $f$, by showing that the cube $c$ is orthogonal (has empty intersection) with any cube of $M$. Note, the literals of $c$ are the complemented literals of $M$. (Since $M$ is a unate cover, the literals of $M$ are just the union of the literals of each cube of $M$). For each cube $m_i \in M, \exists j \in J$ such that $B_{ij} = 1$. ($J$ is the column cover associated with $c$). Thus, $(m_i)_j = x_j \implies c_j = x_j$ and $(m_i)_j = \overline{x}_j \implies c_j = \overline{x}_j$. Thus $m_i \cap c = \emptyset$. Thus $c \subseteq \overline{f}$. 

---

31

32
We now show that any minterm \( \mu \in \overline{f} \) is contained in some cube \( c \) generated. First \( \mu \) must be orthogonal to each cube of \( M \). So for each row of \( M \), there is at least one literal of \( \mu \) that conflicts with that row. The union of all columns (literals) where this happens is a column cover of \( B_M \); hence this union contains at least one minimal cover and the associated cube contains \( \mu \).

---

**Complement of a Unate Cover**

The set of all minimal column covers = cover of \( \overline{f} \).

Example 6

\[
\begin{array}{cccc}
2 & 1 & 2 & 0 \\
2 & 0 & 0 & 1 \\
1 & 2 & 2 & 1 \\
1 & 2 & 0 & 2
\end{array}
\]

\[
\begin{array}{cccc}
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 \\
1 & 1 & 0 & 0 \\
1 & 0 & 1 & 0
\end{array}
\]

\(\{(1,4), (2,3), (2,5), (4,5)\}\) is the set of all minimal covers.

This translates into:

\[
\overline{f} = \overline{ad} + \overline{bc} + \overline{d} \overline{e} + \overline{d} \overline{e}
\]

A minimal column cover is \((1,4) \rightarrow \overline{ad}

\[
\overline{ad} \quad \overline{f} = b \overline{d} + \overline{c} \overline{d} + abe + a \overline{c} \overline{e}
\]
Consider min-term \( \overline{abcd} \overline{e} \in \overline{f} \). It conflicts in literals \( \overline{a}, c, d, \overline{e} \). Thus \( \{1, 3, 4, 5\} \) is a column cover. It contains \( \{1, 4\} \) and \( \{4, 5\} \). Thus

\[
\overline{abcd} \overline{e} \in \overline{ad} \in d \overline{e}
\]

**Unate Covering**

**Definition 4** The problem, given a Boolean matrix \( B \), find a minimum column cover, is called a unate covering problem.

The problem of unate complementation was our first example of the unate covering problem and we will see it often in this course.

**Unate Covering Problem:**

Given \( B \), \( B_{ij} \in \{0, 1\} \) find \( x \), \( x_j \in \{0, 1\} \) such that

\[
Bx \geq 1
\]

and \( \sum_j x_j \) is minimum.

Sometimes we want to minimize

\[
\sum_j c_j x_j
\]

where \( c_j \) is a cost associated with column \( j \).
**Quine-McCluskey Procedure**

**(Exact)**

Given $G'$ and $D$ (covers for $F = (f,d,r)$ and $d$), find a minimum cover $G$ of primes where:

$$f \subseteq G \subseteq f + d$$

($G$ is a prime cover of $F$)

**Q-M Procedure**

1. Generate all the primes of $F$, $\{P_j\}$ (i.e. primes of $(f+d)=G+D$)
2. Generate all the minterms of $f=G \bigcap D$, $\{m_i\}$
3. Build Boolean matrix $B$ where
   
   $$B_{ij} = 1 \text{ if } m_i \in P_j$$
   $$= 0 \text{ otherwise}$$
4. Solve the minimum column covering problem for $B$ (unate covering problem)

---

**Difficulty**

Note: Can be

- $\sim 2^n$ minterms
- $\sim 3^n/n$ primes

Thus $O(2^n)$ rows and $O(3^n/n)$ columns AND minimum covering problem is NP-complete. Hence can probably be double exponential in size of input, i.e., difficulty is $O(2^{3^n})$
**Example 8**

\[ F = \overline{xy}zw + \overline{xy}zw + xyzw + \overline{xyzw} \]

\[ D = yz + xyw + \overline{xyzw} + xzw + \overline{xyzw} \]

Primes: \( \overline{y} + w + \overline{x} \overline{z} \)

Covering Table

Solution: \( \{1,2\} \Rightarrow \overline{y} + w \) is minimum prime cover. (also \( w + \overline{x} \overline{z} \))

**Covering Table**

<table>
<thead>
<tr>
<th>( \overline{y} )</th>
<th>( w )</th>
<th>( \overline{x} \overline{z} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Minterms of \( f \)

<table>
<thead>
<tr>
<th>( \overline{xy}zw )</th>
<th>( \overline{xyzw} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Row singleton (essential minterm)

Essential prime

**Definition 5** An essential prime is any prime that uniquely covers a minterm of \( f \).
Covering Table
Row Equality: In practice, many rows are identical. That is there exist minterms that are contained in the same set of primes.

\[ m_1 \quad 0101101 \]
\[ m_2 \quad 0101101 \]

Any row can be associated with a cube -- called the signature cube.

\[ \text{signature \_ cube}(m_i) = \prod_{m_i \in P_j} P_j \]

e.g. \( m_1 \cup m_2 \subseteq P_2 P_4 P_5 P_7 \)

Row and Column Dominance

Definition 6 A row \( i_1 \) whose set of primes is contained in the set of primes of row \( i_2 \) is said to dominate \( i_2 \).

Example 9

\[ i_1 \quad 011010 \]
\[ i_2 \quad 011110 \]

\( i_1 \) dominates \( i_2 \)

We can remove row \( i_2 \), because we have to choose a prime to cover \( i_1 \), and any such prime also covers \( i_2 \). So \( i_2 \) is automatically covered.
**Row and Column Dominance**

Definition 7 A column $j_1$ whose rows are a superset of another column $j_2$ is said to dominate $j_2$.

Example 10

<table>
<thead>
<tr>
<th></th>
<th>$j_1$</th>
<th>$j_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$j_1$ dominates $j_2$

We can remove column $j_2$ since $j_1$ covers all those rows and more. We would never choose $j_2$ in a minimum cover since it can always be replaced by $j_1$.

---

**Pruning the Covering Table**

1. Remove all rows covered by essential primes (columns in row singletons). Put these primes in the cover $G$.
2. Group identical rows together and remove dominated rows.
3. Remove dominated columns. For equal columns, keep one prime to represent them.
4. Newly formed row singletons define $n$-ary essential primes.
5. Go to 1 if covering table decreased.

The resulting reduced covering table is called the cyclic core. This has to be solved (unate covering problem). A minimum solution is added to $G$ - the set of $n$-ary essential primes. The resulting $G$ is a minimum cover.
Example

<table>
<thead>
<tr>
<th>Essential Prime and Column Dominance G=P1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000000</td>
</tr>
<tr>
<td>1100001</td>
</tr>
<tr>
<td>0110000</td>
</tr>
<tr>
<td>0011100</td>
</tr>
<tr>
<td>0001011</td>
</tr>
<tr>
<td>0000110</td>
</tr>
<tr>
<td>0001101</td>
</tr>
<tr>
<td>0011110</td>
</tr>
</tbody>
</table>

| 1000000                                   |
| 1100001                                   |
| 0110000                                   |
| 0011100                                   |
| 0001011                                   |
| 0000110                                   |
| 0001101                                   |
| 0011110                                   |

<table>
<thead>
<tr>
<th>Essential Prime and Column Dominance G=P1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000000</td>
</tr>
<tr>
<td>1100001</td>
</tr>
<tr>
<td>0110000</td>
</tr>
<tr>
<td>0011100</td>
</tr>
<tr>
<td>0001011</td>
</tr>
<tr>
<td>0000110</td>
</tr>
<tr>
<td>0001101</td>
</tr>
<tr>
<td>0011110</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n-ary Essential Prime and Column Dominance G=P1 + P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>456</td>
</tr>
<tr>
<td>101</td>
</tr>
<tr>
<td>011</td>
</tr>
<tr>
<td>110</td>
</tr>
<tr>
<td>111</td>
</tr>
</tbody>
</table>

Cyclic Core

<table>
<thead>
<tr>
<th>456</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
</tr>
<tr>
<td>011</td>
</tr>
<tr>
<td>110</td>
</tr>
</tbody>
</table>

Row dominance

Solving the Cyclic Core

Best known method (for unate covering) is branch and bound with some clever bounding heuristics.

Independent Set Heuristic:
Find a maximum set of "independent" rows I. Two rows $B_{i1}, B_{i2}$ are independent if $\frac{i}{j}$ such that $B_{i1j} = B_{i2j} = 1$. (They have no column in common)

Example 11: Covering matrix B rearranged with independent sets first.
Lemma 1
\[ |\text{Solution of Covering}| \geq |I| \]

**Heuristic**

Let \( I = \{I_1, I_2, \ldots, I_k\} \) be the independent set of rows
- choose \( j \in I_i \) which covers the most rows of \( A \).
  - Put \( j \rightarrow J \)
- eliminate all rows covered by column \( j \)
- \( I \leftarrow I \backslash \{I_i\} \)
- go to 1 if \( |I| > 0 \)
- If \( B \) is empty, then done (in this case we have the guaranteed minimum solution - IMPORTANT)
- If \( B \) is not empty, choose an independent set of \( B \) and go to 1

Can you think of some improved heuristics?
Generating Primes

We use the unate recursive paradigm. The following is how the merge step is done. (Assumption: we have just generated all primes of $f_{x_i}$ and $f_{\bar{x}_i}$.)

Theorem 5 $p$ is a prime of $f$ iff $p$ is maximal among the set consisting of

- $P = x_iq$, $q$ is a prime of $f_{x_i}, q \subsetneq f_{\bar{x}_i}$
- $P = x_ir$, $r$ is a prime of $f_{\bar{x}_i}, r \subsetneq f_{x_i}$
- $P = qr$, $q$ is a prime of $f_{x_i}, r$ is a prime of $f_{\bar{x}_i}$

End of lecture 3
Generating Primes

Example 12 Assume \( q = abc \) is a prime of \( f_{x_i}^{x_i} \).
Form \( p = x_i abc \). Suppose \( r = ab \) is a prime of \( f_{x_i}^{x_i} \).
Then \( x_i a b \) is an implicant of \( f \)
\[
f = x_i abc + x_i ab + abc + L.
\]
Thus \( abc \) and \( x_i ab \) are implicants, so \( x_i abc \) is not prime.

Note: \( abc \) is prime because if not, \( ab \subseteq f_x \) (or \( ac \) or \( bc \)) contradicting \( abc \) prime of \( f_{x_i}^{x_i} \).

Note: \( x_i ab \) is prime, since if not then either \( ab \subseteq f \) or \( x_i a \subseteq f \). The first contradicts \( abc \) prime of \( f_{x_i}^{x_i} \) and the second contradicts \( ab \) prime of \( f_{x_j}^{x_j} \).
New Exact Methods

- Implicit Q-M based on BDD’s (Coudert and Madre, McGeer, Swamy, Brayton)
  - form characteristic BDD of all primes
  - form characteristic BDD of all minterms of $f (F=(f,d,r))$
  - formulate row dominance and column dominance elimination as BDD operations
  - iterate dominance and prime/minterm elimination until no further decrease
  - generate covering table (cyclic core at this point) and solve

New Exact Methods

GREAT RESULTS!
In both cases, superior to ESPRESSO-EXACT (both in speed and in number of problems solved)

ESPRESSO has a suite of about 130 examples, of which ESPRESSO-EXACT can solve about 110. The others solve almost all of the 130.