Chapter 8
Lecture 16
Chris J. Myers

Synthesis of Finite State Machines

Simplified FSM Design Flow
An Incompletely Specified Moore Machine

Another Incompletely Specified Moore Machine
Machine Obtained by State Splitting

Reduced Machine for Previous Example
7

- Solve the binary covering problem.
- Find the prime compatible.
- Find the maximal compatible.
- Find all pairs of compatible states.

Minimization of Incomplete Specialized Machines

8

A Flow Table and its Compatibility Table
Constructing a Compatibility Table

\[ x^3 x + x^1 x + x^1 x \]

Convert POS formula to SOP form (i.e., complete sum).

\[ (x^3 + x^1)(x^3 + x^1)(x^3 + x^1) \]

Write a POS formula to express compatibility conditions.

If \( s_i \) and \( s_j \) are incompatible, then no maximal compatible.
exists a \( C \) and \( I \) such that \( C \subseteq C_1 \) and \( I \subseteq I_1 \).

A compatible \( C_1 \) with a class set \( I \) is prime iff there does not

\[ \{\{1, 1\}\} \]

Example class set of \( \{3, 4\} \) is prime.

Selection of implied compatible guarantees closure.

The set of implied compatible by a compatible is the class set.

Some stable pairs compatible only if other pairs are merged.

**Finding the Prime Compatibles**

<table>
<thead>
<tr>
<th>( a )</th>
<th>( b )</th>
<th>( c )</th>
<th>( d )</th>
<th>( e )</th>
<th>( f )</th>
<th>( g )</th>
<th>( h )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a' )</td>
<td>( b' )</td>
<td>( c' )</td>
<td>( d' )</td>
<td>( e' )</td>
<td>( f' )</td>
<td>( g' )</td>
<td>( h' )</td>
</tr>
<tr>
<td>( 0' )</td>
<td>( 0' )</td>
<td>( 0' )</td>
<td>( 0' )</td>
<td>( 0' )</td>
<td>( 0' )</td>
<td>( 0' )</td>
<td>( 0' )</td>
</tr>
<tr>
<td>( -)</td>
<td>( -)</td>
<td>( -)</td>
<td>( -)</td>
<td>( -)</td>
<td>( -)</td>
<td>( -)</td>
<td>( -)</td>
</tr>
<tr>
<td>( 0' )</td>
<td>( a' )</td>
<td>( b' )</td>
<td>( c' )</td>
<td>( d' )</td>
<td>( e' )</td>
<td>( f' )</td>
<td>( g' )</td>
</tr>
</tbody>
</table>

**Computation of Prime Classes**
Compatability Table for the Previous Flow Table
### Prime Compatibles for the Previous Flow Table

<table>
<thead>
<tr>
<th>Prime compatible classes</th>
<th>12</th>
<th>10</th>
<th>6</th>
<th>8</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>{r}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{q,r</td>
<td>q,r}</td>
<td></td>
<td></td>
<td>r</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{{r</td>
<td>q}</td>
<td>q,r}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{{q</td>
<td>p</td>
<td>c}</td>
<td>q,c}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>{{p</td>
<td>c}</td>
<td>c}</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>{p</td>
<td>c}</td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>{{q</td>
<td>p</td>
<td>c}</td>
<td>q</td>
<td>c}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>{q,r}</td>
<td>11</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>{{q</td>
<td>p</td>
<td>c}</td>
<td>q</td>
<td>c}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>{{q</td>
<td>p</td>
<td>c}</td>
<td>q,c}</td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>{{p</td>
<td>c}</td>
<td>c}</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>{q,r}</td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Setting up the Covering Problem

\[(a + b + c + d + f)(a + b + c + d + f)(a + b + c + d + f) -\]

\[
(a + b + c + d + f) \iff c_2 \\
(a + b + c + d + f) \iff c_2 \\
(a + b + c + d + f) \iff c_2 \\
(a + b + c + d + f) \iff c_2 \\
\]

- Prime 2 if \((p,q)\) requires \(\{(p,q)\}\) (requires any compatible in the closure of all compatible implied by any compatible in the closure - all states must be contained in a closed cover) -

- For state \(a: (c_1 + c_{11})\)

- Covering all states must be contained in a closed cover.

- Prime is part of the solution.

- Associate a with the ith prime such that \(c_i = 1\) implies the ith
\[ I = 6c = 9c = 1c = 10c \]

\[ I = (\overline{4c} + 6c)(6c + 8c + 8\rho)(9c + 2c + 8\rho)(8c + 2c + 8\rho) \]

\[ (\overline{4c} + 1c + 9c)(11c + 9c)(1c + 9c)(4c + 8\rho) \]

\[ (6c + 2c + 8\rho)(4c + 1c + 8\rho)(11c + 2c)(1c + 5\rho) \]

\[ (10c + 4c)(11c + 6c + 8c + 8\rho)(13c + 6c + 8c + 8\rho)(1c + 1c) \]

\[ (6c + 4c + 6c + 8c + 8\rho)(9c + 2c + 8\rho)(6c + 2c + 8\rho)(1c + 1c) \]

**Covering Problem for the Example**

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( x_5 )</th>
<th>( x_6 )</th>
<th>( x_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.6</td>
<td>0.6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
<td>0.2</td>
</tr>
<tr>
<td>5</td>
<td>0.6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td>1.6</td>
<td>0.6</td>
<td>0.6</td>
<td>1.6</td>
<td>0.2</td>
</tr>
<tr>
<td>1</td>
<td>0.6</td>
<td>1.6</td>
<td>0.6</td>
<td>0.6</td>
<td>1.6</td>
<td>0.2</td>
</tr>
</tbody>
</table>

**Reduced Flow Table**
Heuristic Choices for Next State Entries

<table>
<thead>
<tr>
<th>0.6</th>
<th>0.6</th>
<th>1.6</th>
<th>1.4</th>
<th>1.7</th>
<th>1.3</th>
<th>0.1</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>1.1</td>
<td>-1</td>
<td>-</td>
<td>1.1</td>
<td>1.1</td>
<td>0.1</td>
<td>2</td>
</tr>
<tr>
<td>1.1</td>
<td>-1</td>
<td>0.1</td>
<td>0.1</td>
<td>1.1</td>
<td>1.1</td>
<td>0.1</td>
<td>4</td>
</tr>
<tr>
<td>1.1</td>
<td>1.1</td>
<td>0.1</td>
<td>1.1</td>
<td>0.1</td>
<td>1.1</td>
<td>0.1</td>
<td>1</td>
</tr>
</tbody>
</table>

Branch and Bound Algorithm for Binate Covers
Given a Boolean formula in matrix form and a subset $S$ of columns of minimum cost such that for every row $i$, either $S$ has row $i$ or $\ell_i S \neq \ell_i D$.

**Formulation of BCP**

\[ f_{x, m} \overset{\text{min}}{=} \sum_{i} \ell_{i} S \]

where the objective function is:

\[ (S \ni \ell_{i} D) \lor (0 = \ell_{i} f) : \ell_{i} \in \mathbb{Z} \]

\[ (S \ni \ell_{i} D) \lor (1 = \ell_{i} f) : \ell_{i} \in \mathbb{Z} \]

**Example**

\[
\begin{bmatrix}
6 & - & 1 & 0 & 1 & - & - \\
5 & 0 & - & - & - & - & 0 \\
4 & 0 & - & - & - & - & - \\
3 & - & 1 & 1 & 0 & - & - \\
2 & 1 & - & - & 1 & - & - \\
1 & - & - & 1 & - & 1 & - \\
\end{bmatrix}
= \mathcal{A}
\]

\[
(9x + 4x + \varepsilon x)
(9x + \varepsilon x)(9x + 4x + \varepsilon x)(9x + \varepsilon x + \varepsilon x)(\varepsilon x + 1x)
= \mathcal{A}
\]
\[ \begin{bmatrix}
6 & 1 & 0 & 1 & - & - \\
3 & 1 & 1 & 0 & - & - \\
2 & - & 1 & - & 1 & - \\
1 & - & - & 1 & - & 1
\end{bmatrix} = \mathcal{P} \]

Definition 8.2.1: An essential row of \( \mathcal{P} \) is a row \( \mathcal{P} \) where only one coefficient is different from \( -1 \).

\[ \begin{bmatrix}
4 & 0 & 1 & 0 & 1 & 1 \\
3 & -1 & -1 & 1 & 1 \\
2 & -1 & 0 & 1 & 0
\end{bmatrix} \]

\[ \begin{bmatrix}
4 & 0 & 1 & 0 & 1 & 1 \\
3 & 1 & 1 & -1 & 0 & 1 \\
2 & -1 & 0 & 1 & 0 & 1
\end{bmatrix} \]

Definition 8.2.2: A row \( \mathcal{P} \) dominates another row \( \mathcal{P} \) if \( \mathcal{P} \) is satisfied whenever \( \mathcal{P} \) is satisfied, i.e., \( \mathcal{P} \geq \mathcal{P} \).
Column Dominance

Definition 8.2.3 Let $F'$ be two columns of the matrix $F$. $F'$ dominates $F$ if, for each clause $f'$ of $F$, one of the following occurs:

- $0 = \forall f$ and $0 = \forall f' - 1$
- $i \neq \forall f$ and $- = \forall f' - 1$
- $1 = \forall f'$

In binary case, only consider clauses w/o complemented literals.
Consider only rows with \( f_0 = 1 \).

Columns that intersect many short rows favored.

Choice of the Branching Column

Infeasible Problems
An Example of Reductions

\[
\begin{bmatrix}
0 & 1 \\
1 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 1 \\
1 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 1 \\
1 & 1
\end{bmatrix}
\]

Therefore, all practical encoding methods are heuristic.

\[
i(\hat{y}d - \gamma\hat{z})/i(\hat{y} - \gamma\hat{z})
\]

If permutation or complementation is the same assignment:

\[
i(d - \gamma\hat{z})/i(\gamma\hat{z})
\]

If one uses as bits to encode d states, there are

State Encoding
\begin{equation}
\forall x_i x_j f_i = \forall x_i x_j f_i + \forall x_i x_j f_i
\end{equation}

\begin{align*}
\ldots + \forall x_i x_j f_i + \forall x_i x_j f_i + \ldots &= 1O \\
\ldots + \forall x_i x_j f_i + \forall x_i x_j f_i + \ldots &= \exists \Lambda \\
\ldots + \forall x_i x_j f_i + \forall x_i x_j f_i + \ldots &= \exists \Lambda
\end{align*}

\begin{center}
\begin{tabular}{cccc}
I & II & 00 & 10 \\
I & II & 10 & 10 \\
1O & $\exists \Lambda \lambda$ & $\forall x_i x_i$ & $\exists x_i x_i$
\end{tabular}
\end{center}

**Pan-out Oriented Algorithm**

---

\begin{equation}
\forall x_i x_j f_i + \ldots = \exists \Lambda
\end{equation}

\begin{align*}
\ldots + \forall x_i x_j f_i + \ldots &= \exists \Lambda \\
\ldots + \forall x_i x_j f_i + \forall x_i x_j f_i + \ldots &= \exists \Lambda
\end{align*}

\begin{center}
\begin{tabular}{cccc}
0 & II & 10 & II \\
I & 0I & 10 & 0I \\
1O & $\exists \Lambda \lambda$ & $\forall x_i x_i$ & $\exists x_i x_i$
\end{tabular}
\end{center}

**Pan-in Oriented Algorithm**


FSM and its Attraction Graph

\[
\varphi = [1] \cdot [0] + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \cdot [1, 0, 1] \cdot z = (3', 3) M
\]

\[
\frac{f Z}{f} = \frac{f S}{f} + S \cdot q_N
\]

\[
\begin{bmatrix}
1 & \varepsilon_d \\
0 & \varepsilon_d \\
1 & \varepsilon_d \\
z & \varepsilon_u
\end{bmatrix} = Z
\]

Panout-Oriented Algorithm
$x^1_i = z$

$x^2_i + x^3_i + x^4_i = z^1_i$

$x^2_i = 1_i$

An Assignment Derived by the Pantou Algorithm

$\frac{\ell X^1 X + \ell S \cdot \ell S \cdot \ell N}{\ell S \cdot \ell S \cdot \ell S \cdot \ell N}$

$X = \begin{bmatrix}
1 & 1 & z^u_s \\
0 & 1 & z^u_s \\
z & 1 & 1^u_s
\end{bmatrix}$

$x = \begin{bmatrix}
\varepsilon^d_s \\
\varepsilon^d_s \\
\varepsilon^d_s
\end{bmatrix}$

Pantou-Oriented Algorithm
\[ x_{\bar{f}} = z \]
\[ x_{\bar{f}} \bar{y} = \bar{z}_X \]
\[ x_{\bar{f}} \bar{y} + x_{\bar{f}} \bar{y} = 1_X \]

An Assignment Derived by the Parnin Algorithm

\[ (w x^i \cdots i x^i \bar{f}^i, \bar{f}^i) f = 1_X \]
\[ (w x^i \cdots i x^i \bar{f}^i, \bar{f}^i) \emptyset = 0_X \]
\[ (w x^i \cdots i x^i \bar{f}^i, \bar{f}^i) \bar{f} = 0_X \]
\[ (w x^i \cdots i x^i \bar{f}^i, \bar{f}^i) f = 1_X \]

Decomposition and Encoding
\begin{align*}
\{1', 2', 3, 6, 7' 8\} &= \{1', 2', 3, 6, 7' 8\} \\
\{1', 2', 3, 6, 7' 8\} &= S
\end{align*}

\text{Example:}
\begin{align*}
S &= \{\mathbb{a}B\} \cap \\
\mathbb{G} \neq \mathbb{a} &\text{ for } \emptyset = \mathbb{G} \cup \mathbb{a} \\
\{\mathbb{a}B\} &= \mathbb{v}
\end{align*}

\(S\) whose set union is \(S\)

\(\mathbb{G}\) partition on a set \(S\) is a collection of disjoint subsets of \(S\)

\text{Partitions}

\begin{align*}
\{1', 2', 3, \mathbb{G}' \} &\geq \{1', 2', 3, \mathbb{G}' \} \\
\mathbb{T} \mathbb{G} &\geq \mathbb{T} \\
\{1', 2', 3, \mathbb{G}' \} &= \mathbb{T} \\
\{1', 2', 3, \mathbb{G}' \} &\geq 0 \\
\{1', 2', 3, \mathbb{G}' \} &= S
\end{align*}

\text{Partitions as a Partial Order}
\[ \{6, 2, 3, 4, 5, 6, 7, 8, 9\} = \nu + 1 \]
\[ \{6, 2, 3, 4, 5, 6, 7, 8, 9\} = \nu \cdot 1 \]
\[ \{6, 2, 3, 4, 5, 6, 7, 8, 9\} = \nu \]
\[ \{6, 2, 3, 4, 5, 6, 7, 8, 9\} = 1 \nu \]
\[ \{6, 2, 3, 4, 5, 6, 7, 8, 9\} = S \]

**Example**

\[ \text{Theorem 8.4.1} \]
A FSM has a non-trivial parallel decomposition if and only if there exist two non-trivial S' partitions that imply that
\[ I \in \forall (\nu) (s' \in \forall) \equiv (s' \in S) \]
\[ \text{have the substitution property} \]
\[ \text{A partition} \notin \text{on the set of states} \notin S \text{of an FSM is said to} \]

**Partitions with Substitution Property**
Example FSM with Parallel Decomposition

\[ \begin{array}{cccc}
1 & 2 & 3 & 4 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 \\
2 & 2 & 2 & 2 \\
3 & 3 & 3 & 3 \\
4 & 4 & 4 & 4 \\
\end{array} \]

\[ \begin{array}{cccc}
1 & 2 & 3 & 4 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 \\
2 & 2 & 2 & 2 \\
3 & 3 & 3 & 3 \\
4 & 4 & 4 & 4 \\
\end{array} \]

Components of the FSM

\{ \{0,1\} = \Pi, \{1,4\} = \Pi, \{2,3\} = \Pi, \{3,4,5\} = \Pi \}
Structure of Parallel Decomposition

Structure of Serial Decomposition
Example of FSM with Serial Decomposition

Independent component for the FSM

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

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
### Construction of the Dependent Component

- **Table 1:**

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>0</th>
<th>II</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
<td>1</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>III</td>
<td>III</td>
</tr>
</tbody>
</table>

- **Table 2:**

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>0</th>
<th>0</th>
<th>I</th>
<th>II</th>
<th>II</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
<td>1</td>
<td>0</td>
<td>I</td>
<td>II</td>
<td>II</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>I</td>
<td>0</td>
<td>I</td>
<td>III</td>
<td>II</td>
</tr>
</tbody>
</table>

- **Table 3:**

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>0</th>
<th>0</th>
<th>I</th>
<th>I</th>
<th>I</th>
<th>III</th>
<th>III</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>A,0</td>
<td>B,0</td>
<td>B,1</td>
<td>A,1</td>
<td>A,1</td>
<td>B,0</td>
<td>I</td>
<td>I</td>
</tr>
</tbody>
</table>
### Computation of the S.P. Partitions

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\{1, 4; 2, 3\} = \nu
\]
\[
\{1, 4; 3, 4\} = \mu
\]
\[
\{1, 4; 2, 3\} = \xi
\]
\[
\{1, 4; 3, 5\} = \varepsilon
\]
\[
I = \lambda
\]
S.P. Partition Lattice

If \( (\mathcal{P}, \mathcal{Q}) \) is a partition pair, then \( \not\exists \) has the S.P.

\[
\forall I \exists (\mathcal{P}, \mathcal{Q}) \equiv (\mathcal{Q}, \mathcal{P}) \equiv I
\]

A partition pair \((\mathcal{P}, \mathcal{Q})\) is an ordered pair of partitions s.t.
Schematic for the Encoding of the Machine

Summary

- Described relationship between partitions and encoding.
- Introduced an approach to PNN encoding.
- Discussed method for minimizing incompletely specified