ESPRESSO Algorithm:
The “EXPAND” Step, continued
**“EXPAND” Step: Expansion Direction (REVIEW)**

**Basic idea:**
- Once cube \( c \) in cover \( F \) is selected for expansion, expand it in “good directions”

**3 Heuristic Expansion Steps (in sequence): to expand a cube \( c \)**

1. Expand \( c \) to completely contain (i.e. “swallow up”) as many other cubes of cover \( F \) as possible
   - delete these completely-contained cubes immediately!
2. Once #1 done -- continue to expand \( c \) to partially overlap as many other cubes of cover \( F \) as possible
   - the goal is to “induce redundancies”, which then get deleted in the IRRED step
3. Once #2 done -- continue to expand \( c \) into a “maximal size” prime implicant

➡️ New Focus: Step #2 and #3

---

**“EXPAND” Step: Expansion Direction**

**Example #1: illustrates Step #1 + Step #2**

*Suppose cube \( A \) has been picked for expansion*

![Diagram](image)
Example #1 (cont.): illustrates Step #1 + Step #2

STEP #1: expand cube A to fully contain as many other cubes as possible

- expand cube A to fully contain cube B
Example #1 (cont.): illustrates Step #1 + Step #2

**STEP #1:** expand cube A to fully contain as many other cubes as possible

- expand cube A to fully contain cube B
- delete cube B

**STEP #2:** expand cube A’, by 1 variable at a time, to overlap as many other cubes as possible
Example #1 (cont.): illustrates Step #1 + Step #2

**STEP #2:** expand cube A', by 1 variable at a time, to overlap as many other cubes as possible

- Option 1: expand cube A' in “z’ dimension”: overlaps 1 additional cube (C)
- Option 2: expand cube A’ in “y dimension”: overlaps 2 additional cubes (D, E)
“EXPAND” Step: Expansion Direction

Example #1 (cont.): illustrates Step #1 + Step #2

**STEP #2:** expand cube A’, by 1 variable at a time, to overlap as many other cubes as possible

Final choice = Option 2: expand cube A’ in ”y dimension”

```
00  01  11  10
--  --  --  --
```

Example #2: illustrates Step #2 + Step #3

Suppose cube A has been picked for expansion

```
00  01  11  10
--  --  --  --
```
Example #2 (cont.): illustrates Step #2 + Step #3

**STEP #2**: expand cube A, by 1 variable at a time, to overlap as many other cubes as possible

<table>
<thead>
<tr>
<th>wx</th>
<th>yz 00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Final choice = expand cube A in “y dimension” overlaps cubes B & C

**STEP #2**: expand cube A, by 1 variable at a time, to overlap as many other cubes as possible

<table>
<thead>
<tr>
<th>wx</th>
<th>yz 00</th>
<th>01</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Final choice = expand cube A in “y dimension”
“EXPAND” Step: Expansion Direction

Example #2 (cont.): illustrates Step #2 + Step #3

**STEP #3:** expand cube A’ into a maximal size prime implicant

<table>
<thead>
<tr>
<th></th>
<th>wx</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>yz</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

STEP #3: expand cube A’ into a maximal size prime implicant

**Option 1:** expand cube A’ in “z’ dimension”
- A’ becomes wx
Example #2 (cont.) illustrates Step #2 + Step #3

**STEP #3:** expand cube $A'$ into a maximal size prime implicant

**Option 2:** expand cube $A'$ in $w'/x'$ dimensions - $A'$ becomes $z$

Final choice = Option 2
- 'larger' prime ($A'' = z$) has fewer literals
ESPRESSO Algorithm:  
The “IRREDUNDANT” Step

Example Cover:  after “expand” = prime cover

Given: cover F (after EXPAND)

Goal: make F irredundant = delete max # of implicants while still maintaining a valid cover
The “IRREDUNDANT” Step

Approach #1 (SUBOPTIMAL): greedily remove one redundant cube at a time

Before:

After deleting P3: irredundant suboptimal cover

Approach #2 (OPTIMAL): form & solve an exact optimization problem

This is approach used in ESPRESSO...:

Goal: set up a simplified PI table, and solve it exactly, to determine min # of cubes to keep (delete all others)
## The "IRREDUNDANT" Step

### Simplified PI Table:

<table>
<thead>
<tr>
<th></th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>1011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>1110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

#### Approach:
- solve PI table exactly

#### Optimal Solution:
- select \{P1, P2, P5, P6\}
- discard rest...
The “IRREDUNDANT” Step

An Issue:
- How can afford to compute an exact solution to prime implicant table (like the expensive Quine-McCluskey method!) in the inner loop of a fast heuristic algorithm (espresso)?

Justification:
- This is not the same as the QM method!

Key Observation: columns in this table are only the primes in current cover, -- not all the primes of the function!

Hence, there is no prime generation step, and often a (much) smaller table!

Approach #2 (OPTIMAL): final solution

before:

after: irredundant optimal cover

#25

#26
ESPRESSO Algorithm: The “REDUCE” Step

The “REDUCE” Step

Example Cover: after “expand” and “irredundant” = prime irredundant cover
The “REDUCE” Step

**Example Cover:** after “expand” and “irredundant” = prime irredundant cover

**Given:** cover F (after EXPAND/IRRED)

**Goal:** maximally reduce all cubes of cover F, in some order, while still maintaining a valid cover

---

The “REDUCE” Step

**Key Observation:** reduce operation is order-dependent

**Cube Reduction Order #1:** B, A, C
- only B can be reduced
The “REDUCE” Step

Key Observation: reduce operation is *order-dependent*

Cube Reduction Order #1: B, A, C
- cubes C and B can be reduced

Cube Reduction Order #2: C, B, A
- cubes C and B can be reduced