Input Domain Partitioning

Dr. Mark Sebern
SE2832-2014-09
What's a Domain?

- Are you the monarch of something?
- Does your software serve a certain audience?
- Remember anything from (discrete or other) math?
Inputs and Domains
Input Partitioning

- Builds on idea of equivalence classes
- Breaks up the overall input space
- Set of partitions
- But first, a review of descriptive terms for equivalence classes …
Weak EC

At least one test with each range of each variable ...

Good enough?

Equivalence class: set of values treated “the same”
Now, we cover every combination of valid input variable ranges.
Robust EC

Now, we cover every combination of valid input variable ranges, plus “outside” values

Best?
Robust BV

Not just every combination of input variable ranges

Add test inputs “near” boundaries?
Input Partitioning

<table>
<thead>
<tr>
<th>Domain</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>q</td>
</tr>
<tr>
<td>Blocks</td>
<td>b</td>
</tr>
</tbody>
</table>

- **Disjoint**
  \[ \forall i, j \mid i \neq j \cdot b_i \cap b_j = \emptyset \]
- **Complete**
  \[ \bigcup_{b \in B_q} b = D \]

Can you translate this into English?
Input Domain?

- All possible program inputs
- Input parameters
- Method arguments
- Input data files (including history = DB)
- User inputs
## IDM Example

### Input Domain Model(ing)

### Block $b_i$ like EC?

<table>
<thead>
<tr>
<th>File Ordering State</th>
<th>Sorted ascending</th>
<th>Sorted descending</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>Y</td>
<td>?</td>
</tr>
<tr>
<td>$b$</td>
<td>?</td>
<td>Y</td>
</tr>
<tr>
<td>$b$</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File Ordering State</th>
<th>Sorted ascending</th>
<th>Sorted descending</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b$</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>$b$</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>$b$</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>$b$</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Kinds of IDM

- Interface-based
  - Each input parameter in isolation
- Functionality-based
  - Considers all (or group) of input parameters together
- Each input parameter or group has its own partition ($q_k$)
Strategies

• All combinations coverage (ACoC)
• Each choice coverage (ECC)
• Pair-wise coverage (PWC)
• T-wise coverage (TWC)
• Base choice coverage (BCC)
• Multiple base choices (MBCC)
Your thoughts?