Introductory question

- What is a decision table?
  - A simple non-technical answer?
  - A detailed technical answer?
A precise yet compact way to model complicated logic

Associate conditions with actions to perform

Can associate many independent conditions with several actions in an elegant way
## Decision Table Terminology

### Condition Stubs
- **c1**
  - Rule 1: True
  - Rule 2: True
  - Rules 3,4: True
  - Rule 5: False
  - Rule 6: False
  - Rules 7,8: False
- **c2**
  - Rule 1: True
  - Rule 2: True
  - Rules 3,4: False
  - Rule 5: True
  - Rule 6: True
  - Rules 7,8: False
- **c3**
  - Rule 1: True
  - Rule 2: False
  - Rules 3,4: False
  - Rule 5: True
  - Rule 6: True
  - Rules 7,8: False

### Action Stubs
- **a1**
  - Rule 1: False
  - Rule 2: False
  - Rules 3,4: False
  - Rule 5: True
  - Rule 6: True
  - Rules 7,8: False
- **a2**
  - Rule 1: False
  - Rule 2: False
  - Rules 3,4: False
  - Rule 5: True
  - Rule 6: True
  - Rules 7,8: False
- **a3**
  - Rule 1: False
  - Rule 2: False
  - Rules 3,4: False
  - Rule 5: True
  - Rule 6: True
  - Rules 7,8: False
- **a4**
  - Rule 1: False
  - Rule 2: False
  - Rules 3,4: False
  - Rule 5: True
  - Rule 6: True
  - Rules 7,8: False

### Condition Entries
- **c1**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **c2**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **c3**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **a1**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **a2**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **a3**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **a4**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X

### Action Entries
- **c1**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **c2**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **c3**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **a1**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **a2**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **a3**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X
- **a4**
  - Rule 1: X
  - Rule 2: X
  - Rules 3,4: X

**DTT–4**
Decision Table Terminology – 2

- Condition entries restricted to binary values
  - We have a **limited entry table**

- Condition entries have more than two values
  - We have an **extended entry table**

<table>
<thead>
<tr>
<th>condition stubs</th>
<th>condition entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>action stubs</td>
<td>action entries</td>
</tr>
</tbody>
</table>
# Printer Troubleshooting DT

## Conditions

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Y</th>
<th>Y</th>
<th>Y</th>
<th>N</th>
<th>N</th>
<th>N</th>
<th>N</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Printer does not print</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A red light is flashing</td>
<td></td>
<td></td>
<td></td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Printer is unrecognized</td>
<td></td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

## Actions

<table>
<thead>
<tr>
<th>Actions</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check the power cable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check the printer-computer cable</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensure printer software is installed</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check/replace ink</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Check for paper jam</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*A complete limited entry table*
How are condition entries in a decision table interpreted with respect to a program?
Conditions are interpreted as

- Input
- Equivalence classes of inputs
How are action entries in a decision table interpreted with respect to a program?
Actions are interpreted as
- Output
- Major functional processing portions
What is the relationship between decision tables and test cases?
Decision Table Interpretation – 5

- With a complete decision table
  - Have a complete set of test cases
# Triangle Decision Table

<table>
<thead>
<tr>
<th>C1:  (&lt;a, b, c &gt;) forms a triangle?</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>C3:  (a = b?)</td>
<td>–</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>C4:  (a = c?)</td>
<td>–</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>C5:  (b = c?)</td>
<td>–</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>A1: Not a Triangle</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2: Scalene</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>A3: Isosceles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A4: Equilateral</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td><strong>A5: Impossible</strong></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Action added by a tester showing impossible rules
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1-1: a &lt; b+c?</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>C1-2: b &lt; a+c?</td>
<td>-</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>C1-3: c &lt; a+b?</td>
<td>-</td>
<td>-</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>C2: a = b?</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>C3: a = c?</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>C4: b = c?</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

| A1: Not a Triangle | X | X | X | | | | | | | | |
| A2: Scalene | | | | X |
| A3: Isosceles | | | X | X | X |
| A4: Equilateral | | X |
| A5: Impossible | X | X | X | |

Similar to equivalence classes we can refine the conditions
## Triangle Test Cases

<table>
<thead>
<tr>
<th>Case ID</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>Expected Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>Not a Triangle</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>Not a Triangle</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>Not a Triangle</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>Equilateral</td>
</tr>
<tr>
<td>5</td>
<td>???</td>
<td>???</td>
<td>???</td>
<td>Impossible</td>
</tr>
<tr>
<td>6</td>
<td>???</td>
<td>???</td>
<td>???</td>
<td>Impossible</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>Isosceles</td>
</tr>
<tr>
<td>8</td>
<td>???</td>
<td>???</td>
<td>???</td>
<td>Impossible</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>Isosceles</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>Isosceles</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>Scalene</td>
</tr>
</tbody>
</table>
Admission to University

- Students in
  - The range [80%, 100%] gpa are admitted and receive a scholarship.
  - The range [70%, 80%) gpa are admitted but have no scholarship.
  - The range [60%, 70%) gpa are admitted if they have no failures.
  - Otherwise they are not admitted.
Don't Care Entries and Rule Counts

- Limited entry tables with N conditions have $2^N$ rules.

- Don't care entries reduce the number of explicit rules by implying the existence of non-explicitly stated rules.

  - **How many rules does a table contain including all the implied rules due to don't care entries?**
- Each don't care entry in a rule doubles the count for the rule
- For each rule determine the corresponding rule count
- Total the rule counts
### Don't Care Entries and Rule Counts – 3

<table>
<thead>
<tr>
<th>Rule</th>
<th>Expression</th>
<th>Rule Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a &lt; b + c$?</td>
<td>32</td>
</tr>
<tr>
<td>2</td>
<td>$b &lt; a + c$?</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>$c &lt; a + b$?</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>$a = b$?</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>$a = c$?</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>$b = c$?</td>
<td>1</td>
</tr>
</tbody>
</table>

**Rule count total:** 64

**Total rules:** $2^6$
Don't Care Entries and Rule Counts – 4

- How many rules do extended entry tables have?
- What is the rule count with don't care entries?
  - See DDT-27..28 (NextDate 2'nd try)
  - See DDT-30-31 (NextDate 3'rd try)
Don't Care Entries and Rule Counts – 5

- Is it useful to count the rules in a decision table?
- Why?
Don't Care Entries and Rule Counts – 6

- Less rules than combination rule count
  - Indicates missing rules

- More rules than combination rule count
  - Could indicate redundant rules
    - See Table 7.9, page 107
  - Could indicate inconsistent table
    - See Table 7.10, page 108
The NextDate problem illustrates the correspondence between equivalence classes and decision table structure.

The NextDate problem illustrates the problem of dependencies in the input domain.

- Decision tables can highlight such dependencies.
- Impossible dates can be clearly marked as a separate action.
NextDate Equivalence Classes – for 1st try

\[
\begin{align*}
M1 &= \{ \text{month} : 1 \ldots 12 \mid \text{days(month)} = 30 \} \\
M2 &= \{ \text{month} : 1 \ldots 12 \mid \text{days(month)} = 31 \} \\
M3 &= \{ \text{month} : \{2\} \} \\
D1 &= \{ \text{day} : 1 \ldots 28 \} \\
D2 &= \{ \text{day} : \{29\} \} \\
D3 &= \{ \text{day} : \{30\} \} \\
D4 &= \{ \text{day} : \{31\} \} \\
Y1 &= \{ \text{year} : 1812 \ldots 2012 \mid \text{leap\_year (year)} \} \\
Y2 &= \{ \text{year} : 1812 \ldots 2012 \mid \text{common\_year (year)} \} 
\end{align*}
\]

As in discussion for equivalence classes.
<table>
<thead>
<tr>
<th>Condition</th>
<th>A1: Impossible</th>
<th>A2: Next Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1: month in M1?</td>
<td>T</td>
<td>–</td>
</tr>
<tr>
<td>C2: month in M2?</td>
<td>–</td>
<td>T</td>
</tr>
<tr>
<td>C3: month in M3?</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Because a month is in an equivalence class we cannot have T for more than one entry. The do not care entries are really F.
### NextDate DT (1st try - partial)

#### How many rules
- for a complete table?
- with don't care entries?

| C1: month in M1? | T | T | T | T | T | T | T | T |
| C2: month in M2? | T | T | T | T | T | T | T | T |
| C3: month in M3? |
| C4: day in D1?  | T | T | T | T | T | T | T | T |
| C5: day in D2?  | T | T | T | T | T | T | T | T |
| C6: day in D3?  |
| C7: day in D4?  | T | T | T | T | T | T | T | T |
| C8: year in Y1? | T | T | T | T | T | T | T | T |
| C9: year in Y2? | T | T | T | T | T | T | T | T |
| A1: Impossible | X | X |
| A2: Next Date   | X | X | X | X | X | X | X | X | X |
NextDate Equivalence Classes – for 2nd try

\[ M1 = \{ \text{month} : 1 \ldots 12 \mid \text{days(month)} = 30 \} \]
\[ M2 = \{ \text{month} : 1 \ldots 12 \mid \text{days(month)} = 31 \} \]
\[ M3 = \{ \text{month} : \{2\} \} \]
\[ D1 = \{ \text{day} : 1 \ldots 28 \} \]
\[ D2 = \{ \text{day} : \{29\} \} \]
\[ D3 = \{ \text{day} : \{30\} \} \]
\[ D4 = \{ \text{day} : \{31\} \} \]
\[ Y1 = \{ \text{year} : \{2000\} \} \]
\[ Y2 = \{ \text{year} : 1812 \ldots 2012 \mid \text{leap_year(year)} \land \text{year} \neq 2000 \} \]
\[ Y3 = \{ \text{year} : 1812 \ldots 2012 \mid \text{common_year(year)} \} \]

Handle leap year better
NextDate DT (2nd try - part 1)

This table has 16 rules. How many rules for a complete table?

<table>
<thead>
<tr>
<th>C1: month in</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>C2: day in</td>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
</tr>
<tr>
<td>C3: year in</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>A1: Impossible</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>A2: Increment day</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3: Reset day</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4: Increment month</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5: reset month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6: Increment year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Extended entry table – more refined actions
### NextDate DT (2nd try - part 2)

<table>
<thead>
<tr>
<th>C1: month in</th>
<th>M3</th>
<th>M3</th>
<th>M3</th>
<th>M3</th>
<th>M3</th>
<th>M3</th>
<th>M3</th>
<th>M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2: day in</td>
<td>D1</td>
<td>D1</td>
<td>D1</td>
<td>D2</td>
<td>D2</td>
<td>D2</td>
<td>D3</td>
<td>D3</td>
</tr>
<tr>
<td>C3: year in</td>
<td>Y1</td>
<td>Y2</td>
<td>Y3</td>
<td>Y1</td>
<td>Y2</td>
<td>Y3</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>A1: Impossible</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>A2: Increment day</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3: Reset day</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4: Increment month</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5: reset month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6: Increment year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
New Equivalence Classes – for 3rd try

\[ M_1 = \{ \text{month} : 1 \ldots 12 \mid \text{days(month)} = 30 \} \]
\[ M_2 = \{ \text{month} : 1 \ldots 12 \mid \text{days(month)} = 31 \land \text{month} \neq 12 \} \]
\[ M_3 = \{ \text{month} : \{12\} \} \]
\[ M_4 = \{ \text{month} : \{2\} \} \]
\[ D_1 = \{ \text{day} : 1 \ldots 27 \} \]
\[ D_2 = \{ \text{day} : \{28\} \} \]
\[ D_3 = \{ \text{day} : \{29\} \} \]
\[ D_4 = \{ \text{day} : \{30\} \} \]
\[ D_5 = \{ \text{day} : \{31\} \} \]
\[ Y_1 = \{ \text{year} : 1812 \ldots 2012 \mid \text{leap\_year (year)} \} \]
\[ Y_2 = \{ \text{year} : 1812 \ldots 2012 \mid \text{common\_year (year)} \} \]

Handle end of month and year better
NextDate DT (3rd try - part 1)

### A 22 rule table

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C1:</strong></td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td><strong>C2:</strong></td>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
</tr>
<tr>
<td><strong>C3:</strong></td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td><strong>A1:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>A2:</strong></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>A3:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td><strong>A4:</strong></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td><strong>A5:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>A6:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A 22 rule table

**C1:** month in

**C2:** day in

**C3:** year in

**A1:** Impossible

**A2:** Increment day

**A3:** Reset day

**A4:** Increment month

**A5:** Reset month

**A6:** Increment year
NextDate DT (3rd try - part 2)

<table>
<thead>
<tr>
<th></th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1: month in</td>
<td>M3</td>
<td>M3</td>
<td>M3</td>
<td>M3</td>
<td>M3</td>
<td>M4</td>
<td>M4</td>
<td>M4</td>
<td>M4</td>
<td>M4</td>
<td>M4</td>
<td>M4</td>
</tr>
<tr>
<td>C2: day in</td>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
<td>D1</td>
<td>D2</td>
<td>D2</td>
<td>D3</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
</tr>
<tr>
<td>C3: year in</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>Y1</td>
<td>Y2</td>
<td>Y1</td>
<td>Y2</td>
<td>–</td>
</tr>
<tr>
<td>A1: Impossible</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>A2: Increment day</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3: Reset day</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4: Increment month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5: reset month</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A6: Increment year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DTT–32
- It has been shown that equivalence classes and decision tables can be closely related.

  - What benefit do we get from using equivalence classes in place of decision tables?

  - What benefit do we get from using decision tables in place of equivalence classes?
The specification is given or can be converted to a decision table.

The order in which the predicates are evaluated does not affect the interpretation of the rules or resulting action.

The order of rule evaluation has no effect on resulting action.

Once a rule is satisfied and the action selected, no other rule need be examined.
The order of executing actions in a satisfied rule is of no consequence.

The restrictions do not eliminate many applications.

- In most applications, the order in which the predicates are evaluated is immaterial.
- Some specific ordering may be more efficient than some other but in general the ordering is not inherent in the program's logic.
You have constructed a decision table

Before deriving test cases, what properties should the decision table have?
Before deriving test cases, ensure that

- The rules are complete
  - Every combination of predicate truth values is in the decision table

- The rules are consistent
  - Every combination of predicate truth values results in only one action or set of actions
Guidelines and Observations

- Decision Table testing is most appropriate for programs where one or more of the conditions hold.

- **What are those conditions?**
Decision Table testing is most appropriate for programs where

- There is a lot of decision making
- There are important logical relationships among input variables
- There are calculations involving subsets of input variables
- There is complex computation logic (high cyclomatic complexity)
- There are cause and effect relationships between input and output – ???
What are some problems with using decision tables?
Decision tables do not scale up very well
  - May need to
    - Use extended entry decision tables
    - Algebraically simplify tables

Decision tables need to be iteratively refined
  - The first attempt may be far from satisfactory
    - Similar to using equivalence classes
Guidelines and Observations – 5

- Look for redundant rules
  - More rules than combination count of conditions
  - Actions are the same
  - Too many test cases
  - See Table 7.9, page 107

- Look for inconsistent rules
  - More rules than combination count of conditions
  - Actions are different for the same conditions
  - See Table 7.10, page 108

- Look for Missing rules
  - Incomplete table