Decision Table Syntax

- User Stories, Use Cases
- Collecting Semantics
- Completeness, Consistency, ...
- Kinds of Requirements
- Software Definition:
- Domain Modelling
- Software Engineer/Designer: Content
- Vocabulary
- Apropos (Non-)Determinism
- Conflict Axiom, ...
- Domain Model
- Decision Table
- Detour
Definition.

A decision table \( T \) is called complete if and only if the disjunction of all rules' premises is a tautology, i.e. if

\[
\bigvee_{r \in T} \text{pre}(r).
\]
Completeness: Example

T: decision table

<table>
<thead>
<tr>
<th>r_1</th>
<th>...</th>
<th>r_n</th>
<th>else</th>
</tr>
</thead>
</table>

c_1 \ldots \ v_{1,n} |
| ... |
| ... |
| c_m \ldots \ v_{m,n} |

a_1 \ldots \ w_{1,n} \ e |
| ... |
| ... |
| a_k \ldots \ w_{k,n} \ e |

F_{pre}(r_1) \lor F_{pre}(r_2) \lor F_{pre}(r_3) = (c_1 \land c_2 \land \neg c_3) \lor (c_1 \land \neg c_2 \land c_3) \lor (\neg c_1 \land \text{true} \land \text{true})

is not a tautology.

Syntax:

F or Convenience: The 'else' Rule

• Syntax:

T: decision table

r_1 \ldots r_n | else | c_1 \ldots v_{1,n} | \ldots | c_m \ldots v_{m,n} |

a_1 \ldots w_{1,n} \ e | \ldots | a_k \ldots w_{k,n} \ e |

• Semantics:

F(else) := \neg (\lor \ r \in T \setminus \{\text{else}\} F_{pre}(r)) \land F_{eff}(w_1,e,a_1) \land \ldots \land F_{eff}(w_k,e,a_k)

Proposition.

If decision table T has an 'else'-rule, then T is complete.

Definition. [Uselessness]

Let T be a decision table. A rule r \in T is called useless (or: redundant) if and only if there is another (different) rule r' \in T:

• whose premise is implied by the one of r and
• whose effect is the same as r's,

i.e. if \exists r' \neq r \in T: |

F_{pre}(r) \Rightarrow F_{pre}(r') \land F_{eff}(r) \iff F_{eff}(r')

r is called subsumed by r'.

• Again: uselessness is decidable; reduces to SAT.
Rule \( r_4 \) is subsumed by \( r_3 \).

Rule \( r_3 \) is not subsumed by \( r_4 \).

Useless rules "do not hurt" as such.

Yet useless rules should be removed to make the table more readable, yielding an easier usable specification.

\[ \text{Determinism Definition.} \]
\[ T \text{ is called deterministic if and only if the premises of all rules are pairwise disjoint, i.e. if } \forall r_1 \neq r_2 \in T \exists \neg (F_{\text{pre}}(r_1) \land F_{\text{pre}}(r_2)). \]

Otherwise, \( T \) is called non-deterministic.

And again: determinism is decidable; reduces to SAT.

\[ \text{Determinism: Example} \]
\[ T: \text{room ventilation} \]

\[ \begin{array}{c|c|c|c}
\hline
\text{button pressed?} & \text{ventilation off?} & \text{ventilation on?} & \text{start ventilation} \\
\hline
\times & \times & - & \times \\
\times & \times & - & \times \\
\hline
\end{array} \]

\[ \text{go start ventilation} \]

\[ \text{stop stop ventilation} \]

Is \( T \) deterministic? Yes.

Determinism: Another Example

\[ T_{\text{abstr}}: \text{room ventilation} \]

\[ \begin{array}{c|c|c|c}
\hline
\text{button pressed?} & \text{start ventilation} & \text{stop ventilation} & \text{go start ventilation} \\
\hline
\times & \times & - & \times \\
\times & \times & - & \times \\
\hline
\end{array} \]

Is \( T_{\text{abstr}} \) deterministic? No.

By the way...

Is non-determinism a bad thing in general? Just the opposite: non-determinism is a very, very powerful modelling tool.

Read table \( T_{\text{abstr}} \) as:

- the button may switch the ventilation on under certain conditions (which I will specify later), and
- the button may switch the ventilation off under certain conditions (which I will specify later).

We in particular state that we do not (under any condition) want to see on and off executed together, and that we do not (under any condition) see go or stop without button pressed.

On the other hand: non-determinism may not be intended by the customer.
Hydroplaning wheels didn't turn fast due to:
• on landing gear too little weight

Anti-crosswind manoeuvre puts:
• wind conditions not as announced from tower, tail- and crosswinds,

14 Sep. 1993 confl means we don't exclude any states from consideration.

\[ \text{false} = \phi \]

\[ \text{T} \] does not tell us what to do in that case!

\[ \text{T} \] Decision table

\[ \text{Relative Completeness} \]

\[ \text{Completeness wrt. Conflict Axiom} \]

\[ \text{Definition.} \]

\[ \text{Intuition} \]

\[ \text{Pitfall} \]

\[ \text{Enabling one of those while in the air, can have fatal consequences.} \]

\[ \text{Example} \]

\[ \text{Domain Modelling for Decision Tables} \]

\[ \text{Pitfalls in Domain Modelling} \]

\[ \text{Domain Modelling} \]
Collecting semantics and consistency rules may do harm in operation! Rules that are inconsistent with conflict relation are not supposed to be executed at the same time— in particular in (multiple) decision tables, created and edited by multiple people, Inconsistencies may go and stop ventilation on? off ventilation on? stop ventilation off? ventilation off?

A decision table with is inconsistent with conflict relation .

A rule of decision table is called \( T^\square (i) \) .

A rule of decision table is called \( T^\square (ii) \) .

Let \( F = \bigwedge \). Let \( \square \) be a rule of decision table .

Let \( \square \) be the transitive, symmetric closure of .

Collecting semantics and consistency rules may do harm in operation! Rules that are inconsistent with conflict relation are not supposed to be executed at the same time— in particular in (multiple) decision tables, created and edited by multiple people, Inconsistencies may go and stop ventilation on? off ventilation on? stop ventilation off? ventilation off?

A decision table with is inconsistent with conflict relation .

A rule of decision table is called \( T^\square (i) \) .

Let \( F = \bigwedge \). Let \( \square \) be a rule of decision table .

Let \( \square \) be the transitive, symmetric closure of .

Collecting semantics and consistency rules may do harm in operation! Rules that are inconsistent with conflict relation are not supposed to be executed at the same time— in particular in (multiple) decision tables, created and edited by multiple people, Inconsistencies may go and stop ventilation on? off ventilation on? stop ventilation off? ventilation off?

A decision table with is inconsistent with conflict relation .

A rule of decision table is called \( T^\square (i) \) .

Let \( F = \bigwedge \). Let \( \square \) be a rule of decision table .

Let \( \square \) be the transitive, symmetric closure of .

Collecting semantics and consistency rules may do harm in operation! Rules that are inconsistent with conflict relation are not supposed to be executed at the same time— in particular in (multiple) decision tables, created and edited by multiple people, Inconsistencies may go and stop ventilation on? off ventilation on? stop ventilation off? ventilation off?

A decision table with is inconsistent with conflict relation .

A rule of decision table is called \( T^\square (i) \) .

Let \( F = \bigwedge \). Let \( \square \) be a rule of decision table .

Let \( \square \) be the transitive, symmetric closure of .

Collecting semantics and consistency rules may do harm in operation! Rules that are inconsistent with conflict relation are not supposed to be executed at the same time— in particular in (multiple) decision tables, created and edited by multiple people, Inconsistencies may go and stop ventilation on? off ventilation on? stop ventilation off? ventilation off?

A decision table with is inconsistent with conflict relation .

A rule of decision table is called \( T^\square (i) \) .

Let \( F = \bigwedge \). Let \( \square \) be a rule of decision table .

Let \( \square \) be the transitive, symmetric closure of .
Let $T$ be a decision table over $C$ and $A$ and $\sigma$ be a model of an observation of $C$ and $A$. Then $\text{coll}(T) := \bigwedge_{a \in A} a \leftrightarrow \bigvee_{r \in T, r(a) = \times} \text{pre}(r)$ is called the collecting semantics of $T$.

We say, $\sigma$ is allowed by $T$ in the collecting semantics if and only if $\| \sigma \| = \text{coll}(T)$. That is, if exactly all actions of all enabled rules are planned/executed.

Example:

$T$: room ventilation

<table>
<thead>
<tr>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>×</td>
<td>×</td>
<td>−</td>
</tr>
<tr>
<td>×</td>
<td>×</td>
<td>−</td>
<td>×</td>
</tr>
<tr>
<td>×</td>
<td>−</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>−</td>
<td>−</td>
<td>−</td>
<td>−</td>
</tr>
<tr>
<td>blnk</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

¬[(on ∧ off) ∨ (¬on ∧ ¬off)]

"Whenever the button is pressed, let it blink (in addition to go/stop action.

Consistency in the Collecting Semantics

Definition. [Consistency in the Collecting Semantics]

Decision table $T$ is called consistent with conflict relation $\Rightarrow$ in the collecting semantics (under conflict axiom $\phi_{\text{confl}}$) if and only if there are no conflicting actions in the effect of jointly enabled transitions, i.e. if $\| \sigma \| = \text{coll}(T) \land \neg \phi_{\text{confl}} \rightarrow \bigwedge_{(a_1, a_2) \in \Rightarrow} \neg (a_1 \land a_2)$.

Discussion

Two broad directions:

- Option 1: teach formalism (usually not economic).
- Option 2: serve as translator / mediator.
Two broad directions:

- **Option 1**: teach formalism (usually not economic).
- **Option 2**: serve as translator / mediator.

Customer FM expert ▶️

FM expert tells system scenario $S$ (maybe keep back, whether allowed / forbidden),

FM expert translates system scenario to valuation $\sigma$,

FM expert evaluates DT on $\sigma$,

FM expert translates outcome to "allowed / forbidden by DT",

compare expected outcome and real outcome.

**Recommendation**:

- use formal methods for the most important/intricate requirements (formalising all requirements is in most cases not possible),
- use the most appropriate formalism for a given task,
- use formalisms that you know (really) well.

**References**

