1 Formalization

The original formalization has

This "completely determined" formalization requires that we define a response $\mathcal{T}(c, y)$ for each $y \in \mathcal{Y}$ and $c \in \mathcal{C} := \mathcal{Y}^* = \mathcal{U}_{n=1}^{\infty} Y^n$. This condition is stronger than necessary, as there may be contexts c in which certain observations $y \in \mathcal{Y}$ are unexpected. In such a context, any assignment $\mathcal{T}(c, y) := u$, $u \in \mathcal{U}$, would give the same policy. However, the wrong choice of $\mathcal{T}(c, y)$ can interfere with attempts at state reduction

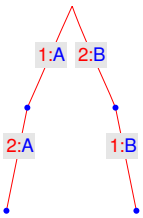
Example 1 *Suppose* $\mathcal{Y} = \{1, 2\}$ *and* $\mathcal{U} = \{A, B\}$, *and*

$$\mathcal{T}(c, y) = \begin{cases} A & (c = \emptyset, y = 1) \vee (c = 1, y = 2) \\ B & (c = \emptyset, y = 2) \vee (c = 2, y = 1) \\ \mathcal{T}'(c, y) & \textit{otherwise.} \end{cases},$$

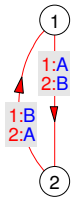
is an optimal policy, for arbitrary $\mathcal{T}' : (\mathcal{C} \times \mathcal{Y}) \to \mathcal{U}$.

Decision Table	
(1)	$\mapsto A$
(2)	$\mapsto B$
(1, 2)	$\mapsto A$
(2, 1)	$\mapsto B$

(a)



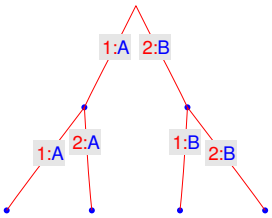
(b)



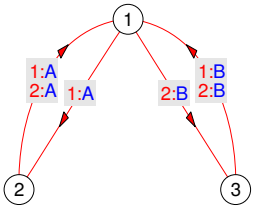
(c)

Decision Table	
(1)	$\mapsto A$
(2)	$\mapsto B$
(1, 1)	$\mapsto A$
(1, 2)	$\mapsto A$
(2, 1)	$\mapsto B$
(2, 2)	$\mapsto B$

(d)



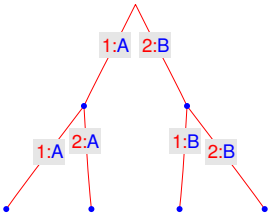
(e)



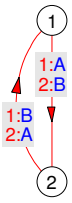
(f)

Decision Table	
(1)	$\mapsto A$
(2)	$\mapsto B$
(1, 1)	$\mapsto B$
(1, 2)	$\mapsto A$
(2, 1)	$\mapsto B$
(2, 2)	$\mapsto A$

(g)



(h)



(i)

Decision tables $\mathcal{T} : \mathcal{C} \to \mathcal{U}$, where $\mathcal{C} = \bigcup_{i=1}^{\infty} \mathcal{C}_i$, and we recursively define

$$C_0 = \{\emptyset\} \quad \text{and} \quad C_{i+1} = \{(c_1, \ldots, c_i, y) \,|\, c = (c_1, \ldots, c_i) \in \mathcal{C}_i,\ y \in \mathcal{Y}_c\}. \tag{1}$$

where $\mathcal{Y}_c$ is the set of observations that may be seen in

Alternatively we can use the equivalent to adding a nop command to

The advantage to this formalization is that it allows

2 Algorithm

2.1 Compatibility Relations

2.2 Assembling Cliques

3 Code

The main MATLAB routine is in `decision_script.m`.

3.1 Input Format

Given a decision table $\mathcal{T}$: (

```
fsm = {[8, 5, 2],                          ... % 1
       [2, 5, 3; 6, 3, 4; 10, 1, 5],       ... % 2
       [7, 4, 6; 8, 7, 7],                 ... % 3
       [],                                 ... % 4
       [5, 10, 8; 7, 6, 9],                ... % 5
       [4, 9, 10; 5, 5, 11; 7, 9, 12],     ... % 6
       [],                                 ... % 7
       [9, 7, 13],                         ... % 8
       [3, 8, 14],                         ... % 9
       [],                                 ... % 10
       [],                                 ... % 11
       [4, 1, 15; 6, 6, 16],               ... % 12
       [1, 3, 17],                         ... % 13
       [7, 1, 18],                         ... % 14
       [],                                 ... % 15
       [],                                 ... % 16
       []};                                    % 17
```

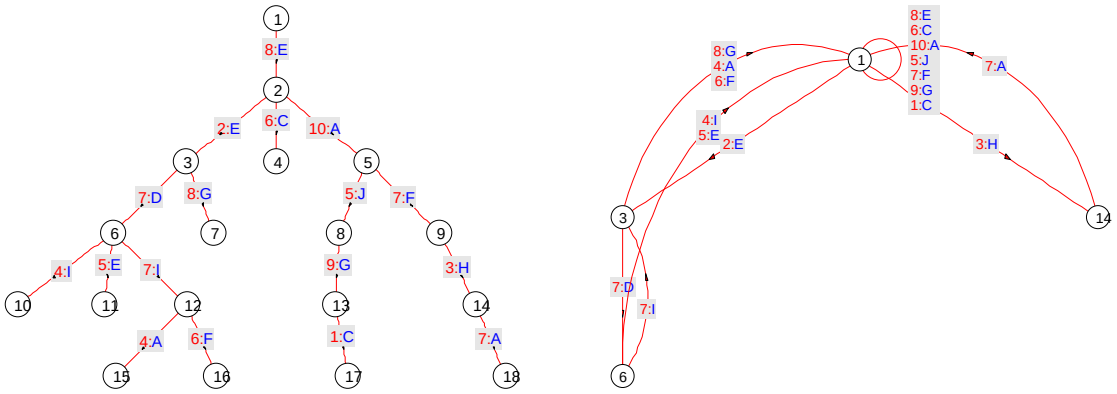
3.2 Console Output

```
Initial FSM:
--------------------------------------------------
01:  08-->(05, 02),
02:  02-->(05, 03),  06-->(03, 04),  10-->(01, 05),
03:  07-->(04, 06),  08-->(07, 07),
04:
05:  05-->(10, 08),  07-->(06, 09),
06:  04-->(09, 10),  05-->(05, 11),  07-->(09, 12),
07:
08:  09-->(07, 13),
09:  03-->(08, 14),
10:
11:
12:  04-->(01, 15),  06-->(06, 16),
13:  01-->(03, 17),
14:  07-->(01, 18),
15:
16:
17:
18:

Final equivalence classes: {1, 2, 4, 5, 7, 8, 9, 10, 11, 13, 15, 16, 17, 18}  {3, 12}  {6}  {14}
```

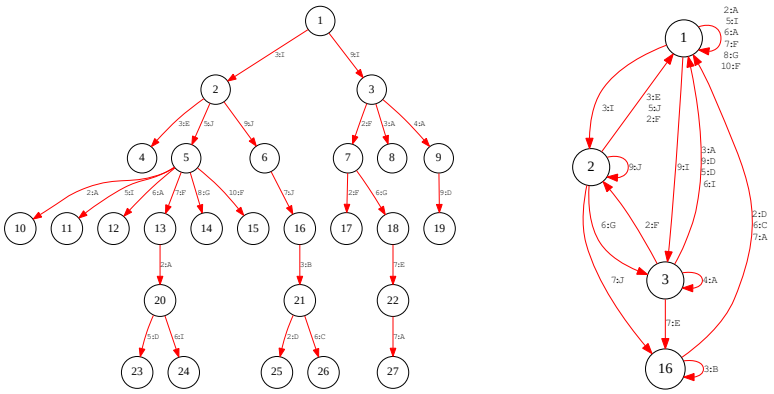
3.3 Visualizations

An optional section of this script uses Graphviz to produce high-quality FSM visualizations



(j) Original Policy (k) Reduced Policy

Figure 1: MATLAB Visualizations



(a) Original Policy (b) Reduced Policy

Figure 2: Graphviz Visualizations

This makes a system call to the script `drawgraph.sh`, which runs in the bash shell, and uses a few non-standard graphics utilities. In Ubuntu, these dependencies should be resolved by the command

```
$ sudo apt-get install ghostscript texlive-extra-utils graphviz evince
```