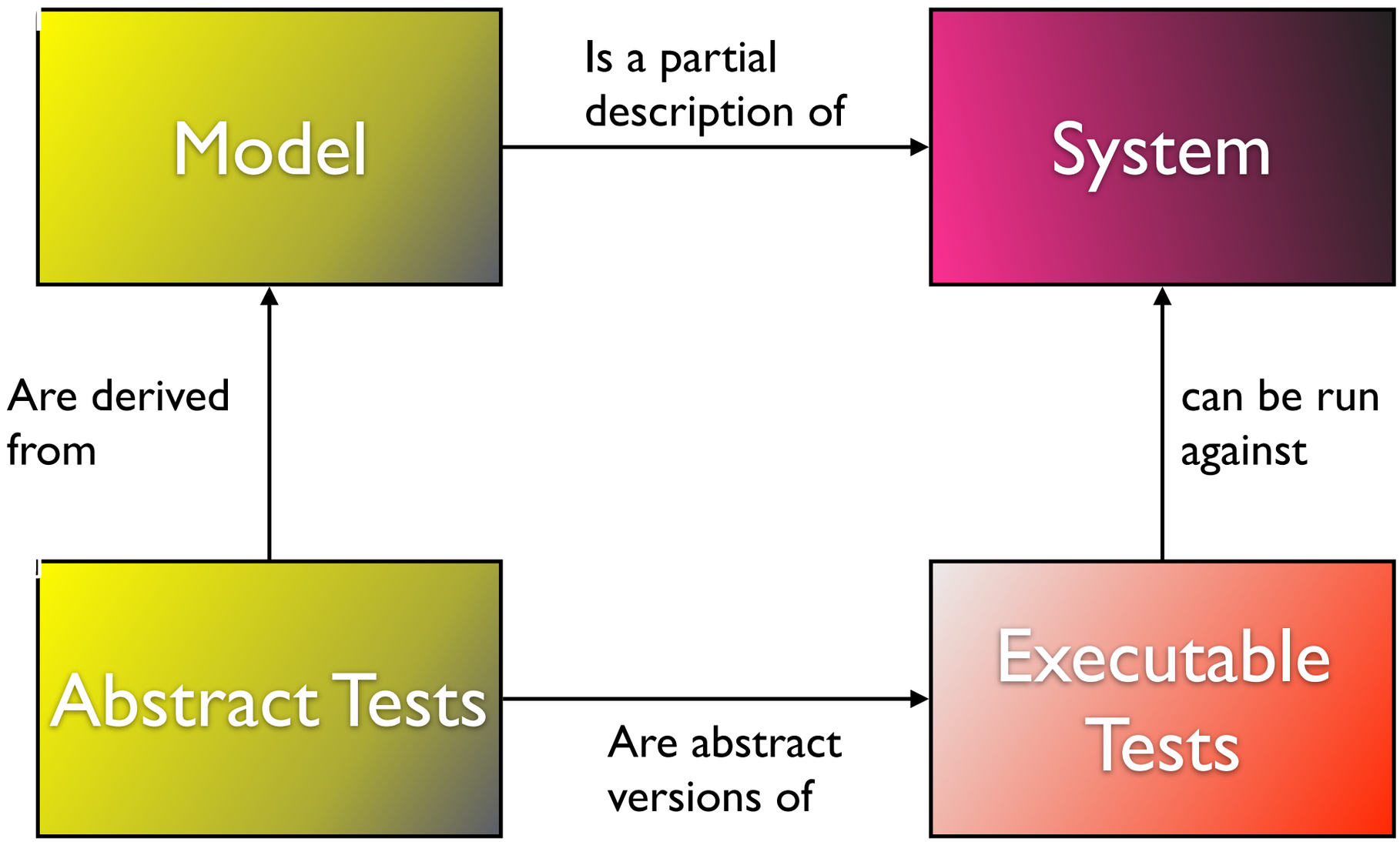
Testing Infinite State Systems – Mathematical Foundations and Concrete Algorithms

Wen-ling Huang and Jan Peleska
University of Bremen
{huang,jp}@cs.uni-bremen.de

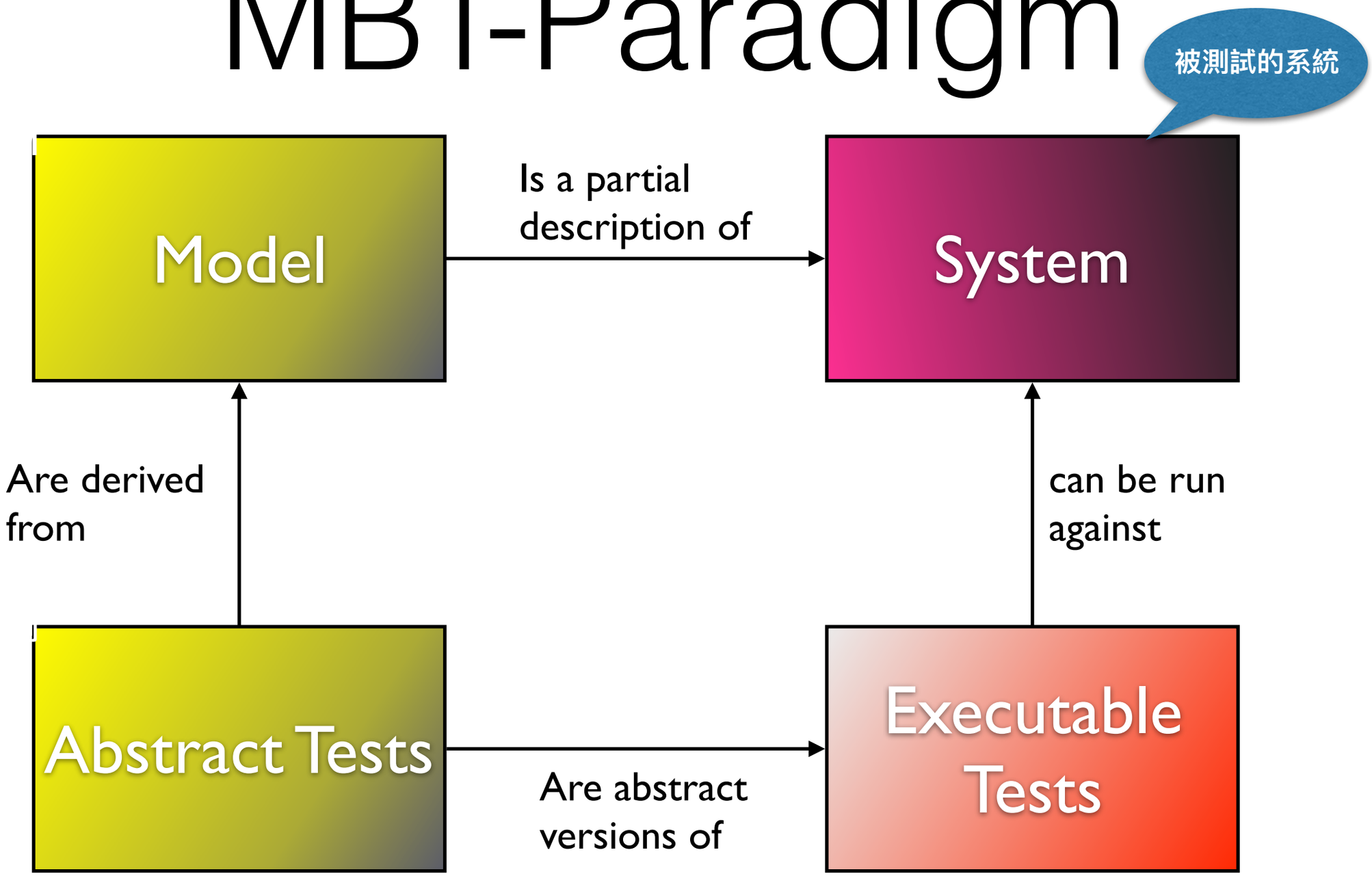
Model-Based Testing

- Model-based testing (MBT) as defined in Wikipedia
  - “**Model-based testing** is an application of **model-based** design for designing and optionally also executing artifacts to perform software **testing** or system **testing**. **Models** can be used to represent the desired behavior of a System Under Test (SUT), or to represent **testing** strategies and a test environment.”

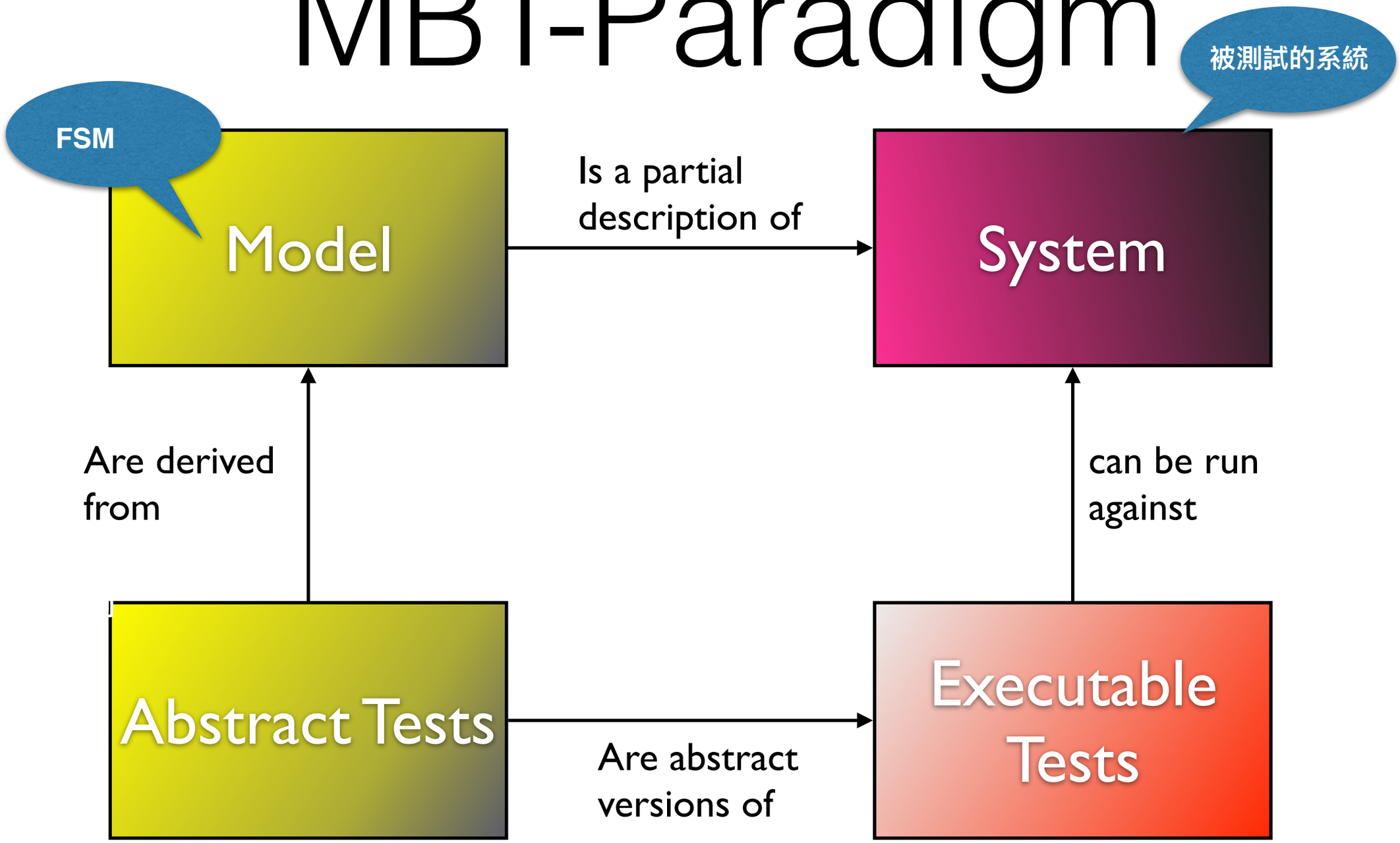
MBT-Paradigm



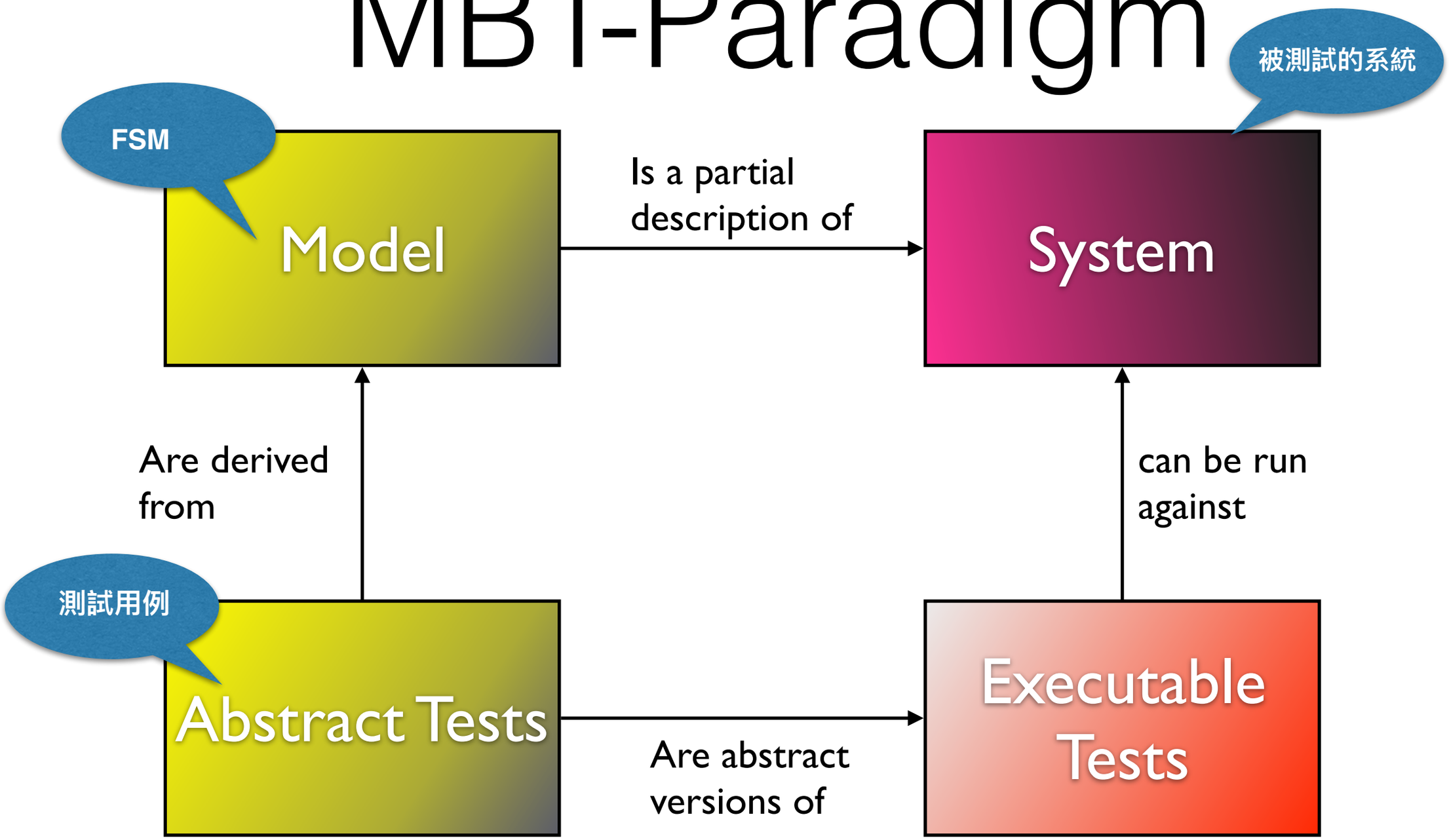
MBT-Paradigm



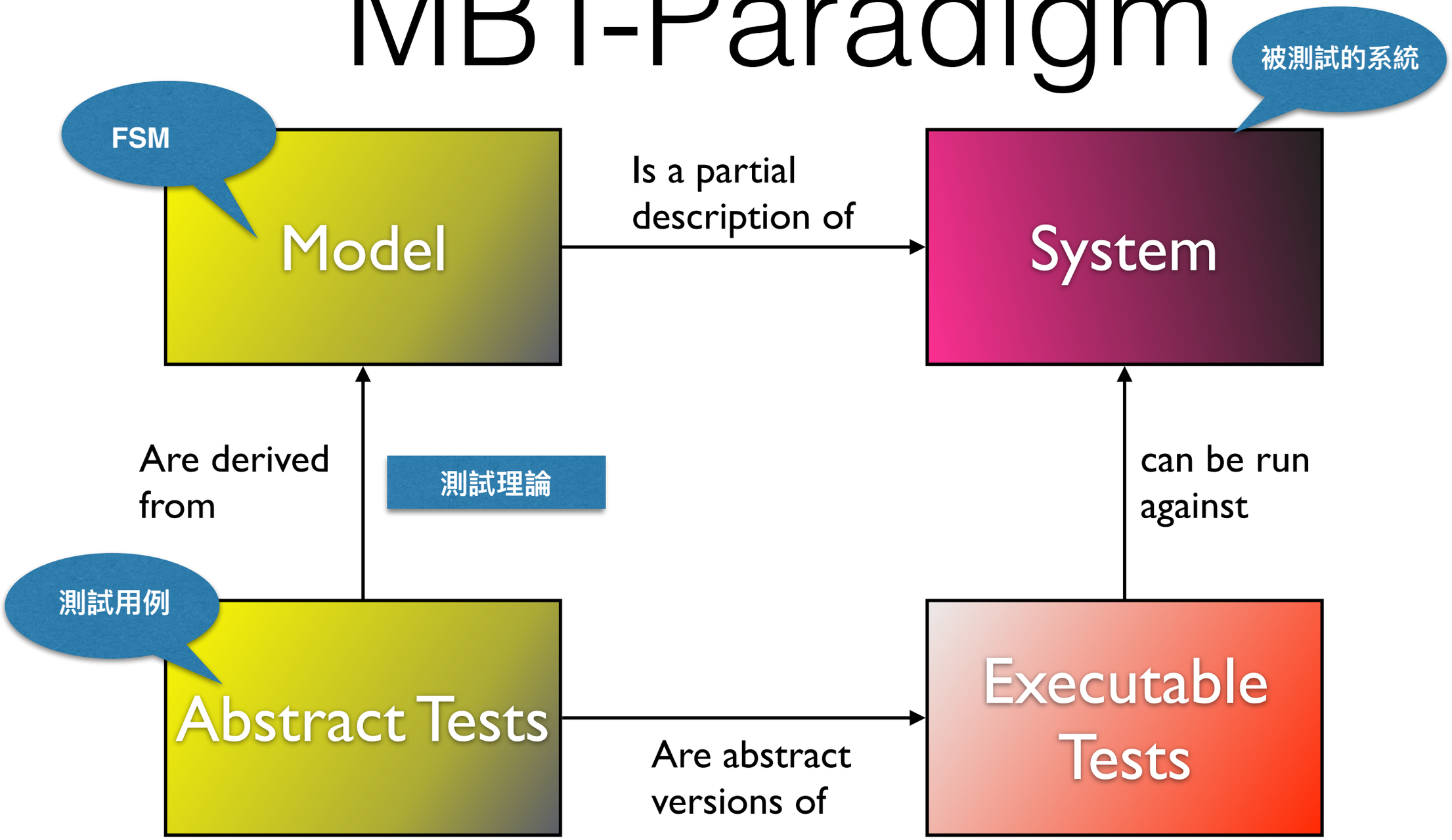
MBT-Paradigm



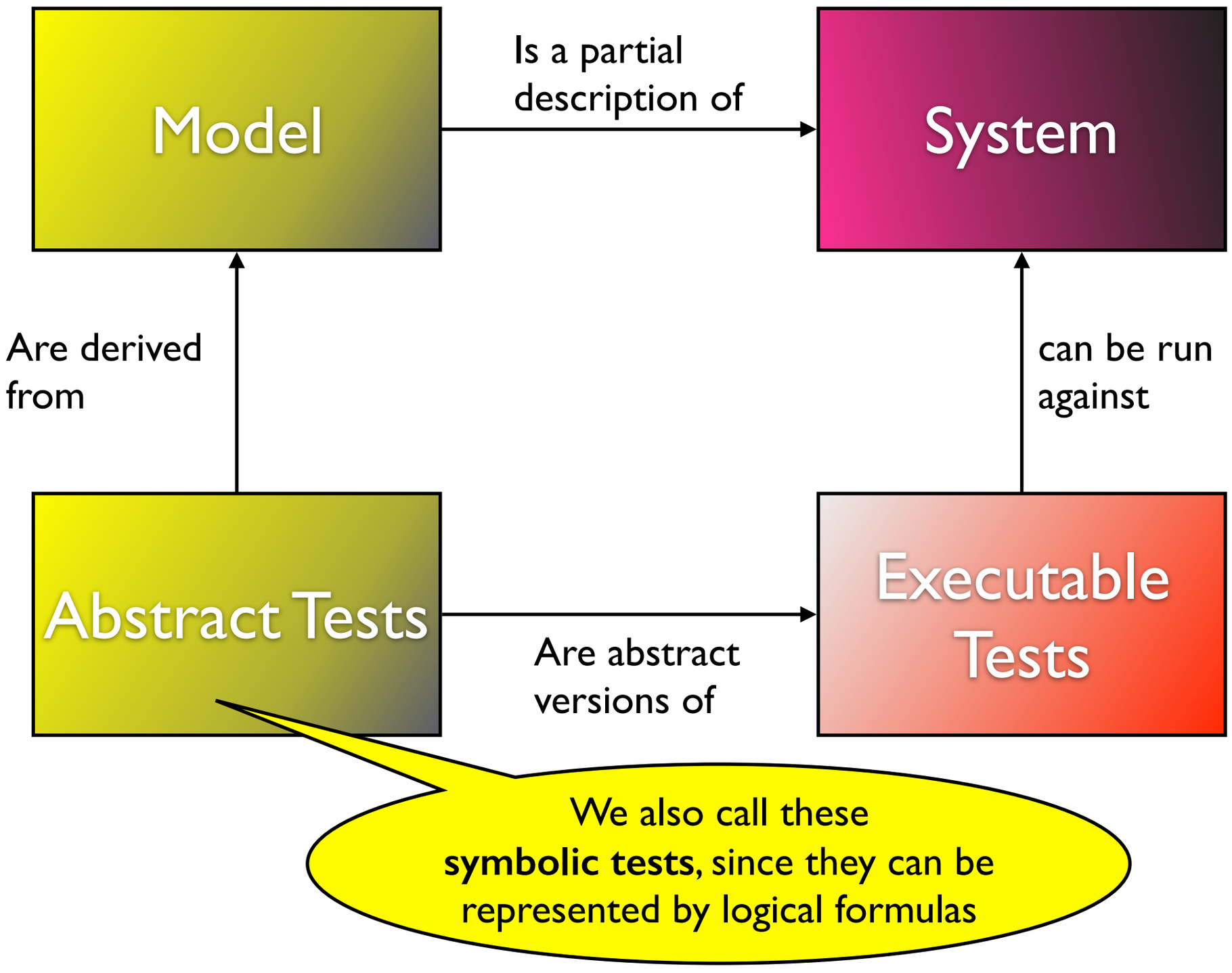
MBT-Paradigm



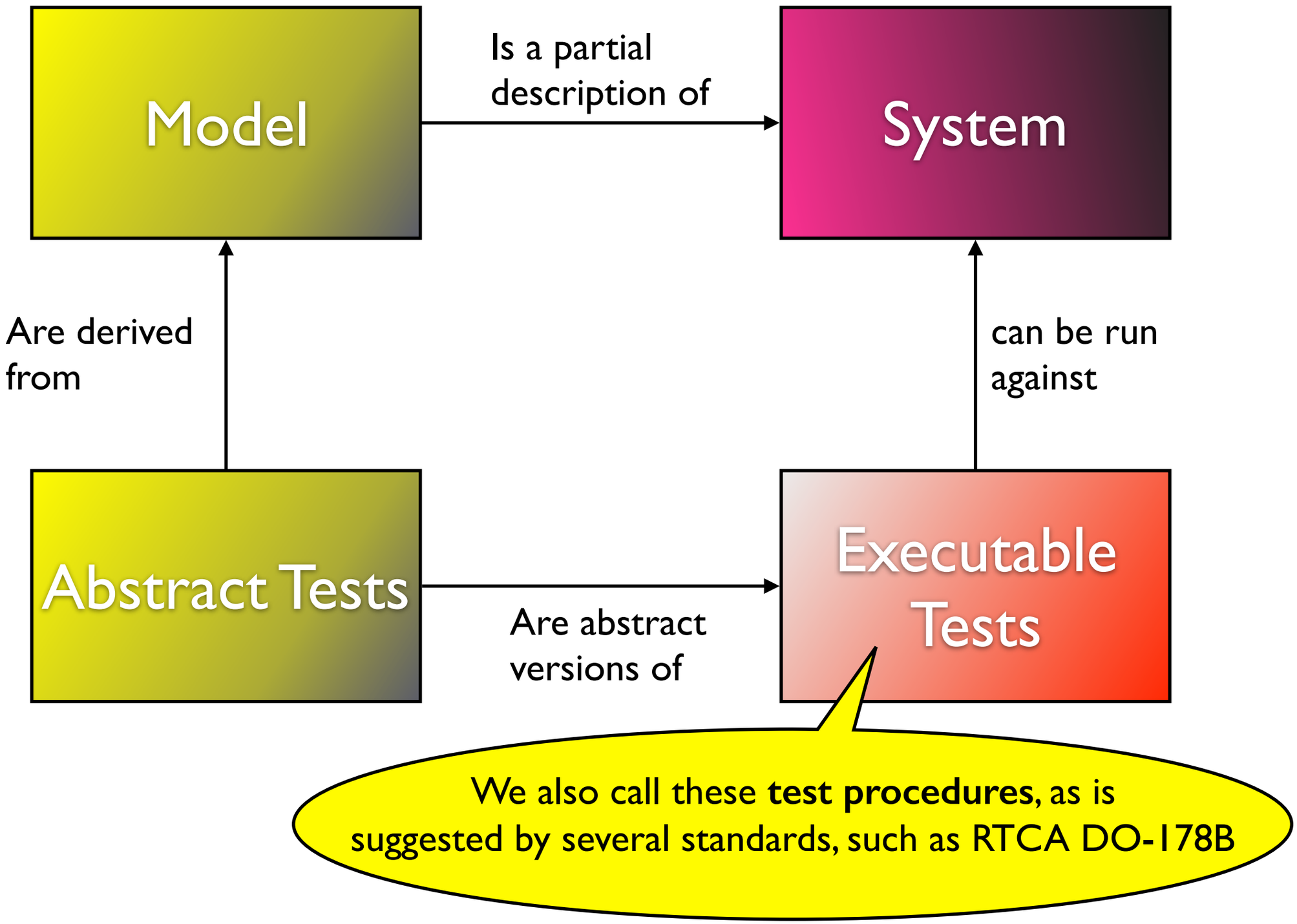
MBT-Paradigm



MBT-Paradigm



MBT-Paradigm



Testing Theories

$\mathcal{F}(M, I, O, \leq, \mathcal{D})$ **fault model**

- $M \in Sig$, **reference model** **参照模型**
- $\leq \subseteq Sig \times Sig$, **conformance relation**

  (I/O equivalence or I/O reduction)
- $\mathcal{D} \subseteq Sig$, **fault domain** **錯誤域**

Test Cases, Test Suite

Test case of deterministic Sig:
I/O sequence $\pi = x_1/y_1 \ldots x_k/y_k \in \Sigma^*$

- M **passes** π, if $\pi \in L(M)$
- M **fails** π, if $\pi \notin L(M)$

Test suite TS: a collection of test cases.

- M **passes** TS, if
  $\forall \pi \in TS,\ M$ passes π.
- M **fails** TS, if
  $\exists \pi \in TS,\ M$ fails π.

Complete Test Suites

$\mathcal{F}(M, I, O, \leq, \mathcal{D})$, fault model
$\mathbf{TS}$, test suite

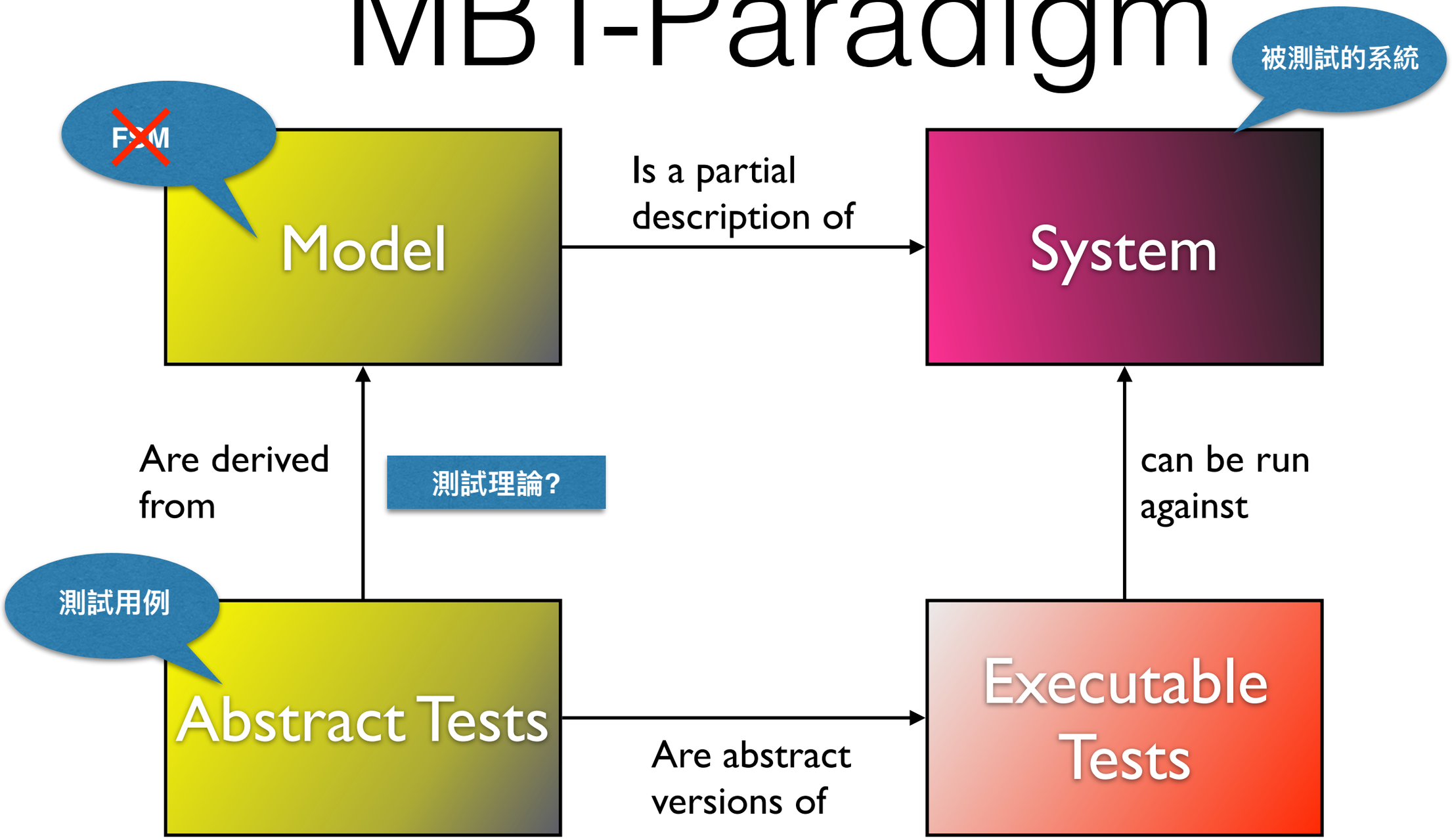
- **Soundness**: $\forall M' \in \mathcal{D} : M' \leq M \Rightarrow M' \underline{\text{pass}}\, \mathbf{TS}$
- **Exhaustiveness**: $\forall M' \in \mathcal{D} : M' \underline{\text{pass}}\, \mathbf{TS} \Rightarrow M' \leq M$
- **Completeness**: Soundness + Exhaustiveness

$$\forall M' \in \mathcal{D} : M' \leq M \Leftrightarrow M' \underline{\text{pass}}\, \mathbf{TS}$$

Testing Theories

- **Deterministic FSM:**
- **T-Method** $\mathcal{F}(M, \leq, \mathcal{D}_O)$
- **W-Method, Wp-Method** $\mathcal{F}(M, \leq, \mathcal{D}_m)$

MBT-Paradigm



Motivation

- Many physical systems have **infinite – even uncountable – state spaces**, because they involve real-valued observables like
  - time
  - speed
  - thrust
  - temperature . . .

Motivation

Which types of system fall into this category?

Where is our innovation useful?

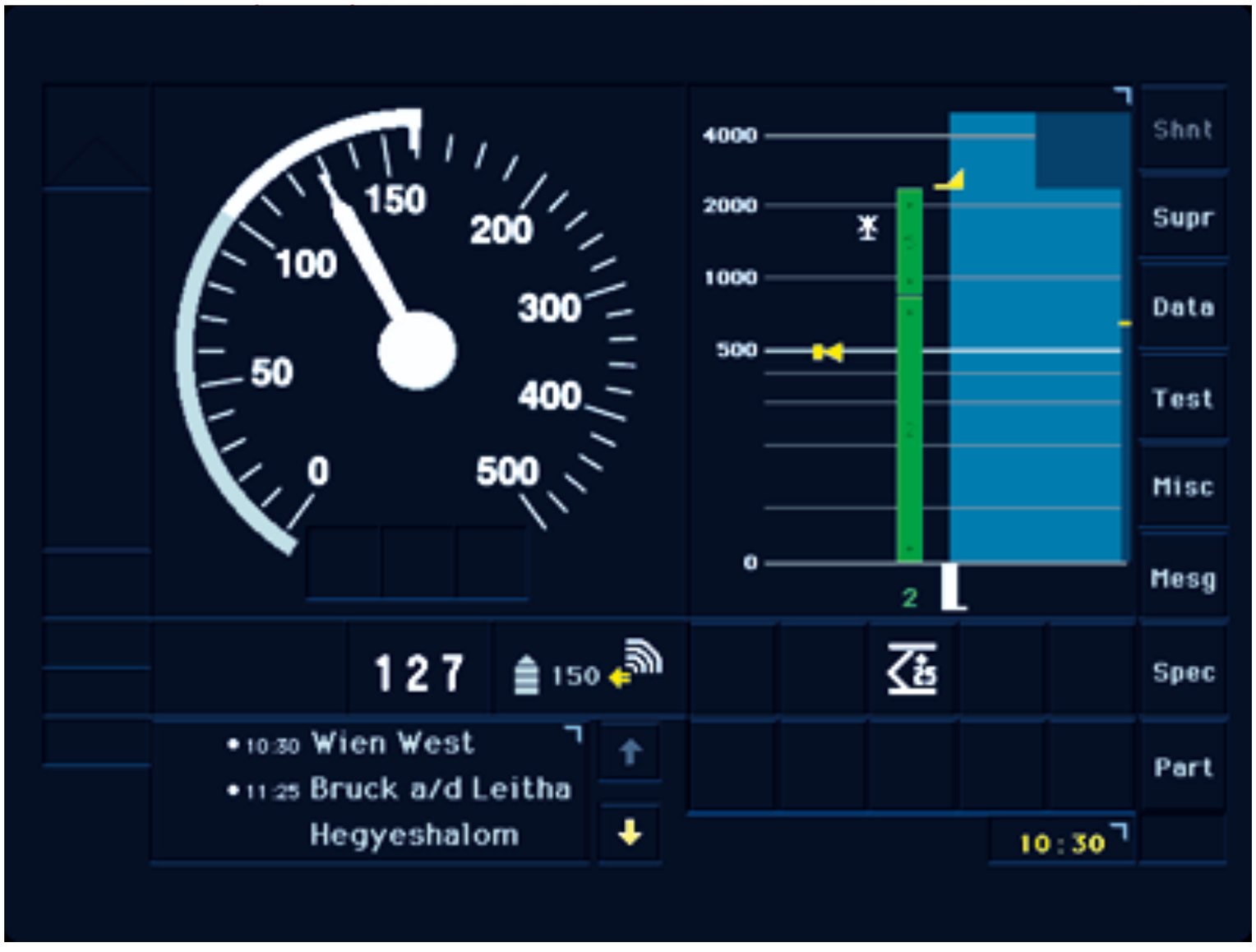
Airbag controller

Where is our innovation useful?

Aircraft thrust reversal

Where is our innovation useful?

Train speed supervision



Translation of Testing Theories

HW+SW ⟶

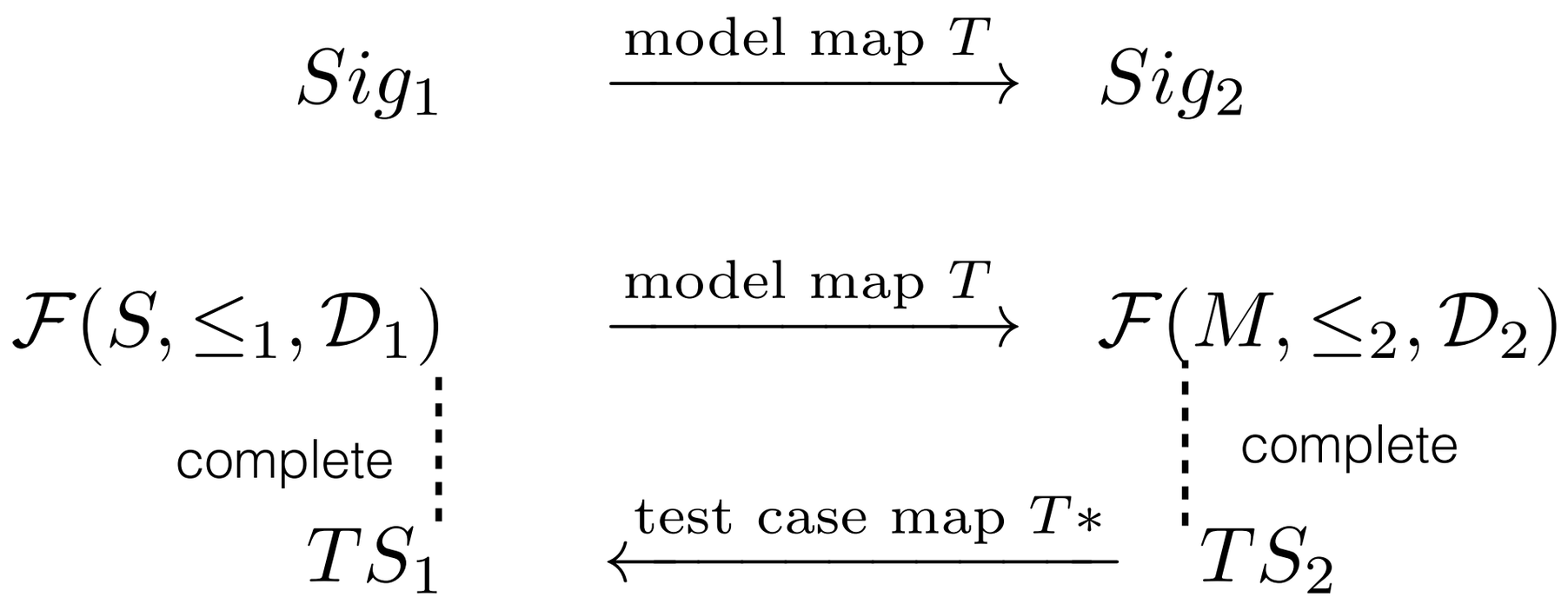
Mathematical models:

- Finite State Machines (FSM)(Sig_2)

finite directed graphs with labels

- RIOSTS(Sig_1)

infinite directed graphs



Satisfaction Condition (1)

$$(T, T^*) : \mathcal{F}(Sig_1) \times \mathbf{tc}(Sig_2) \not\to \mathcal{F}(Sig_2) \times \mathbf{tc}(Sig_1)$$

$$\mathcal{F}(S, \leq_1, \mathcal{D}_1) \xrightarrow{T} \mathcal{F}(T(S), \leq_2, \mathcal{D}_2)$$

- $\forall S' \in \mathcal{D}_1, T(S') \in \mathcal{D}_2$
- $S' \leq_1 S \Leftrightarrow T(S') \leq_2 T(S)$

$$\begin{array}{ccc} \mathcal{S} & \longrightarrow & T(\mathcal{S}) \\ \leq_1 \big| & & \big| \leq_2 \\ S' & \longrightarrow & T(S') \end{array}$$

Satisfaction Condition (2)

$$(T, T^*) : \mathcal{F}(Sig_1) \times \mathbf{tc}(Sig_2) \quad \not\to \quad \mathcal{F}(Sig_2) \times \mathbf{tc}(Sig_1)$$

$$\mathcal{F}(S, \leq_1, \mathcal{D}_1) \quad \xrightarrow{T} \quad \mathcal{F}(T(S), \leq_2, \mathcal{D}_2)$$

- $\forall S' \in \mathcal{D}_1, U \in \mathbf{tc}(Sig_2) : T^*(U) \in \mathbf{tc}(Sig_1)$
- $T(S') \underline{\mathbf{pass}}_2 U \Leftrightarrow S' \underline{\mathbf{pass}}_1 T^*(U)$

$$\begin{array}{ccc} S' & \longrightarrow & T(S') \\ \Big| & & \Big| \\ \underline{\mathbf{pass}}_1 & & \underline{\mathbf{pass}}_2 \\ \Big| & & \Big| \\ T^*(\mathcal{U}) & \longleftarrow & \mathcal{U} \end{array}$$

Theorem Let (T, T^*) satisfy the satisfaction conditions. Then for any **ts** complete test suite of $\mathcal{F}(T(S), \leq_2, \mathcal{D}_2)$, $T^*(\mathbf{ts})$ is a complete test suite of $\mathcal{F}(S, \leq_1, \mathcal{D}_1)$.

$$
\begin{array}{lcll}
\mathcal{S}' \leq_1 \mathcal{S} & \Leftrightarrow & T(\mathcal{S}') \leq_2 T(\mathcal{S}) & [\text{ satisfaction condition (1)}] \\
& \Leftrightarrow & \forall U \in \mathbf{ts}, T(\mathcal{S}') \underline{\mathbf{pass}}_2 U & [\mathbf{ts} \text{ is a complete test suite of } \mathcal{F}(T(S), \leq_2, \mathcal{D}_2)] \\
& \Leftrightarrow & \forall U \in \mathbf{ts}, \mathcal{S}' \underline{\mathbf{pass}}_1 T^*(U) & [\text{ satisfaction condition (2)}]
\end{array}
$$

Testing Theories of Finite State Machines (Sig_2) — Recall

Finite State Machines

$$M = (Q, q_0, I, O, h)$$

- $Q \neq \varnothing$: finite set of states　状態集
- $q_0 \in Q$: initial state　初始狀態
- $I \neq \varnothing$: finite set of input alphabet　輸入字母表
- $O \neq \varnothing$: finite set of output alphabet　輸出字母表
- $h \subseteq Q \times I \times O \times Q$: transition relation狀態遷移關係

Language of FSM

$$M = (Q, q_0, I, O, h)$$

- $L(q) := \{\pi \in \Sigma^* \mid \exists q' \in Q, q \xrightarrow{\pi} q'\}$ **language of** q
- $q \sim q' :\Leftrightarrow L(q) = L(q')$ q q' **are equivalent**
- $L(M) := L(q_0)$ **language of** M

Conformance Relations

$M = (Q, q_0, I, O, h)$, $M' = (Q', q'_0, I, O, h')$ two FSM.

- M and M' are **I/O equivalent** : $L(M') = L(M)$

- M' is an **I/O reduction** of M : $L(M') \subseteq L(M)$

Testing Theories

- **Deterministic FSM:**
- **T-Method** $\mathcal{F}(M, \leq, \mathcal{D}_O)$
- **W-Method, Wp-Method** $\mathcal{F}(M, \leq, \mathcal{D}_m)$

State Cover

State cover V of $M = (Q, q_0, I, O, h)$

- $V \subseteq L(M)$
- $\varepsilon \in V$
- $\forall q \in Q, \exists \pi \in V : q_0 \xrightarrow{\pi} q.$

q_0-after-$\pi = q$

Transition Cover

Concatenation
For any $A, B_{\neq\varnothing} \subseteq \Sigma^*$.
$A.B := \{\pi.\iota \mid \pi \in A, \iota \in B\}$
Example: $\Sigma = \{a, b, c\}$
$A = \{\varepsilon\}, B = \{a.b\}, C = \{a, c\}$
$A.B = \{\varepsilon.a.b\} = \{a.b\} = B$
$B.C = \{a.b.a, a.b.c\}$

Transition cover P of $M = (Q, q_0, I, O, h)$

- $P \subset L(M)$
- $\varepsilon \in P$
- $\forall q \in Q, \sigma = x/y \in L(q), \exists \pi \in P : q_0 \xrightarrow{\pi} q \wedge \pi.\sigma \in P$

V is a state cover
$\Rightarrow V \bigoplus_M (\{\varepsilon\} \cup \Sigma) = \big(V.(\{\varepsilon\} \cup \Sigma)\big) \cap L(M)$
$= V \cup \{\pi.\sigma \in L(M) \mid \pi \in V, \sigma \in \Sigma\}$
is a transition cover

W-Method

M. P. Vasilevskii 1973 and Tsun S. Chow 1978

$$\mathcal{F}(M, I, O, \leq, \mathcal{D}_m), \text{ fault model}$$

$$\mathcal{D}_m = \{M' = (Q', q'_0, I, O, h') \mid |Q'| \leq m\}$$

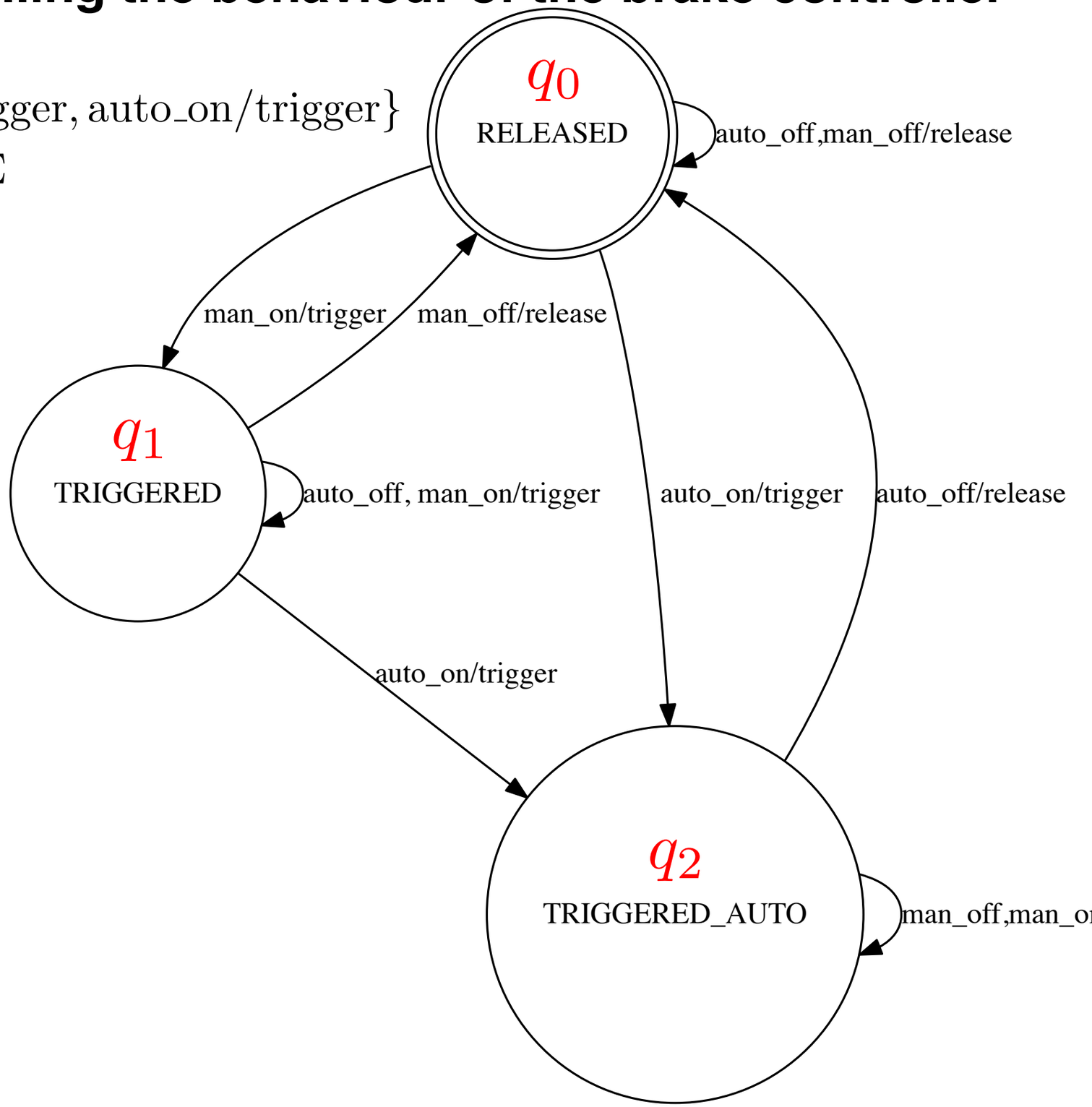
Characterization Set

Characterization set W of $M = (Q, q_0, I, O, h)$

- $W \subseteq \Sigma^*$ is a set of I/O sequences
- $\forall q_1 \neq q_2 \in Q, \exists \tau_1 \neq \tau_2 \in W : {\tau_1}_I = {\tau_2}_I \wedge \tau_i \in L(q_i), i = 1, 2$

Finite State Machine modelling the behaviour of the brake controller

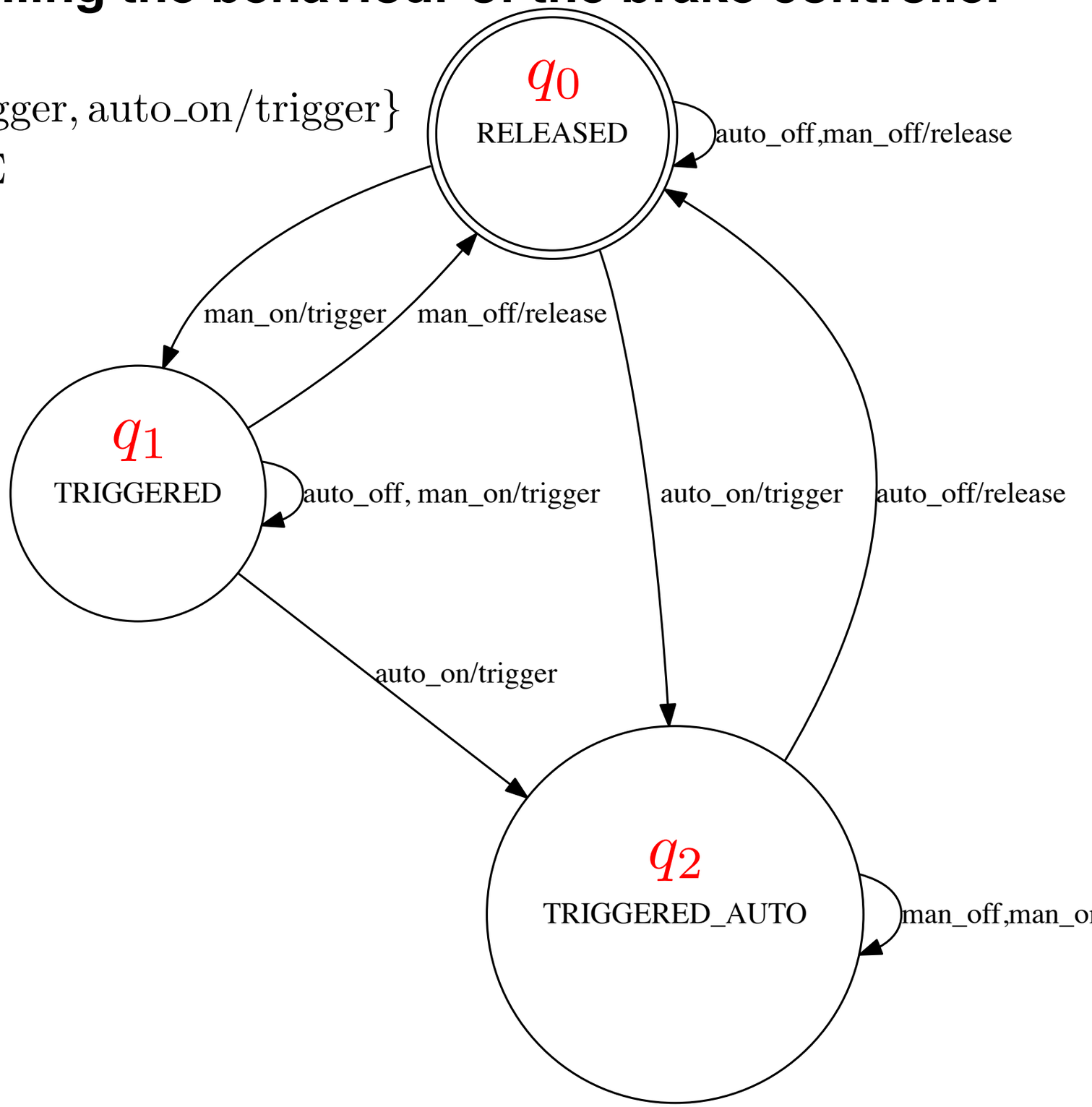
- state cover $V = \{\varepsilon, \text{man\_on/trigger}, \text{auto\_on/trigger}\}$
- transition cover $P = V \cup V \oplus \Sigma$



Finite State Machine modelling the behaviour of the brake controller

- state cover $V = \{\varepsilon, \text{man\_on/trigger}, \text{auto\_on/trigger}\}$
- transition cover $P = V \cup V \oplus \Sigma$

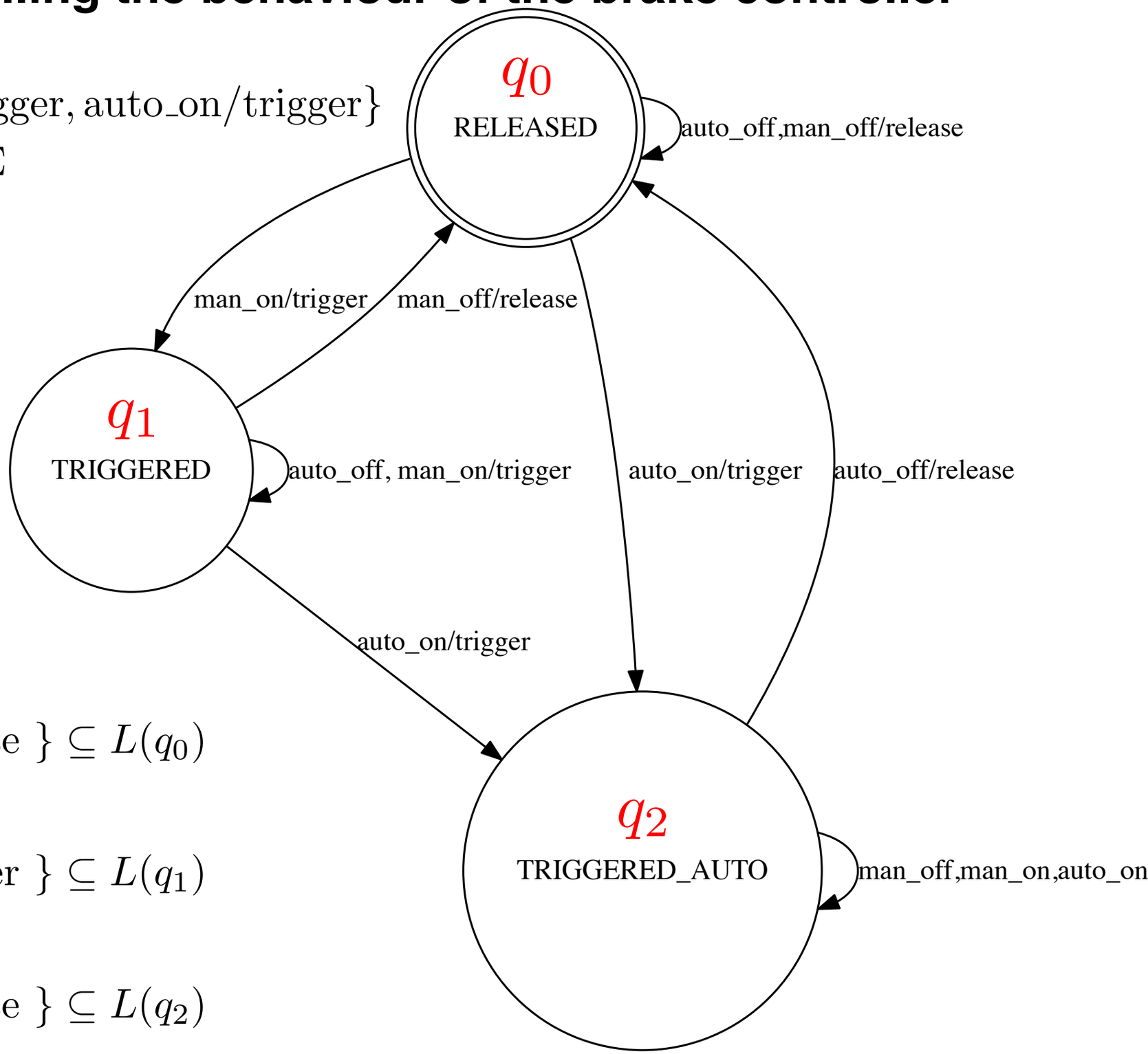
$$
\begin{aligned}
q_0 &\xrightarrow{\text{auto\_off/release}} q_0 \\
q_1 &\xrightarrow{\text{auto\_off/trigger}} q_1 \\
q_2 &\xrightarrow{\text{auto\_off/release}} q_0 \\
q_0 &\xrightarrow{\text{man\_off/release}} q_0 \\
q_2 &\xrightarrow{\text{man\_off/trigger}} q_2
\end{aligned}
$$



Finite State Machine modelling the behaviour of the brake controller

- state cover $V = \{\varepsilon, \text{man\_on/trigger}, \text{auto\_on/trigger}\}$
- transition cover $P = V \cup V \oplus \Sigma$

$$q_0 \xrightarrow{\text{auto\_off/release}} q_0$$
$$q_1 \xrightarrow{\text{auto\_off/trigger}} q_1$$
$$q_2 \xrightarrow{\text{auto\_off/release}} q_0$$
$$q_0 \xrightarrow{\text{man\_off/release}} q_0$$
$$q_2 \xrightarrow{\text{man\_off/trigger}} q_2$$

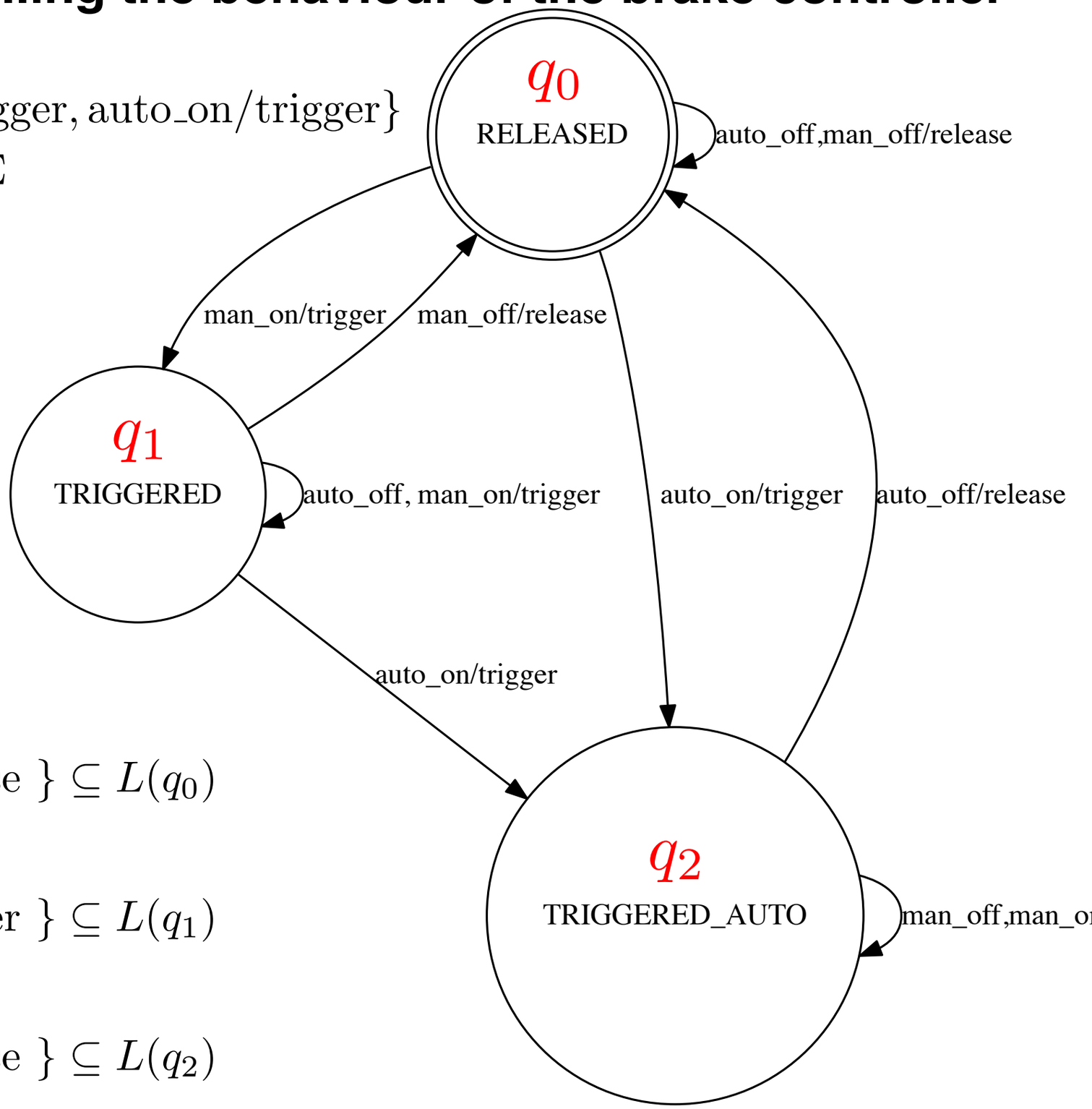


- $\{\text{man\_off/release}, \text{auto\_off/release}\ \} \subseteq L(q_0)$
- $\{\text{man\_off/release}, \text{auto\_off/trigger}\ \} \subseteq L(q_1)$
- $\{\text{man\_off/trigger}, \text{auto\_off/release}\ \} \subseteq L(q_2)$

Finite State Machine modelling the behaviour of the brake controller

- state cover $V = \{\varepsilon, \text{man\_on/trigger}, \text{auto\_on/trigger}\}$
- transition cover $P = V \cup V \oplus \Sigma$

$$q_0 \xrightarrow{\text{auto\_off/release}} q_0$$
$$q_1 \xrightarrow{\text{auto\_off/trigger}} q_1$$
$$q_2 \xrightarrow{\text{auto\_off/release}} q_0$$
$$q_0 \xrightarrow{\text{man\_off/release}} q_0$$
$$q_2 \xrightarrow{\text{man\_off/trigger}} q_2$$



- $\{\text{man\_off/release}, \text{auto\_off/release}\ \} \subseteq L(q_0)$
- $\{\text{man\_off/release}, \text{auto\_off/trigger}\ \} \subseteq L(q_1)$
- $\{\text{man\_off/trigger}, \text{auto\_off/release}\ \} \subseteq L(q_2)$

- characterization set $W = \{\text{man\_off/trigger}, \text{man\_off/release}, \text{auto\_off/trigger}, \text{auto\_off/release}\ \}$

W-Method

Every $\mathbf{TS} = V \oplus \left(\bigcup_{i=0}^{m-n+1} \Sigma^i\right) \oplus W$
is a complete test suite of $\mathcal{F}(M, I, O, \sim, \mathcal{D}_m), n = |Q|$

W-Method

Every $\mathbf{TS} = V \oplus \left(\bigcup_{i=0}^{m-n+1} \Sigma^i \right) \oplus W$
is a complete test suite of $\mathcal{F}(M, I, O, \sim, \mathcal{D}_m), n = |Q|$

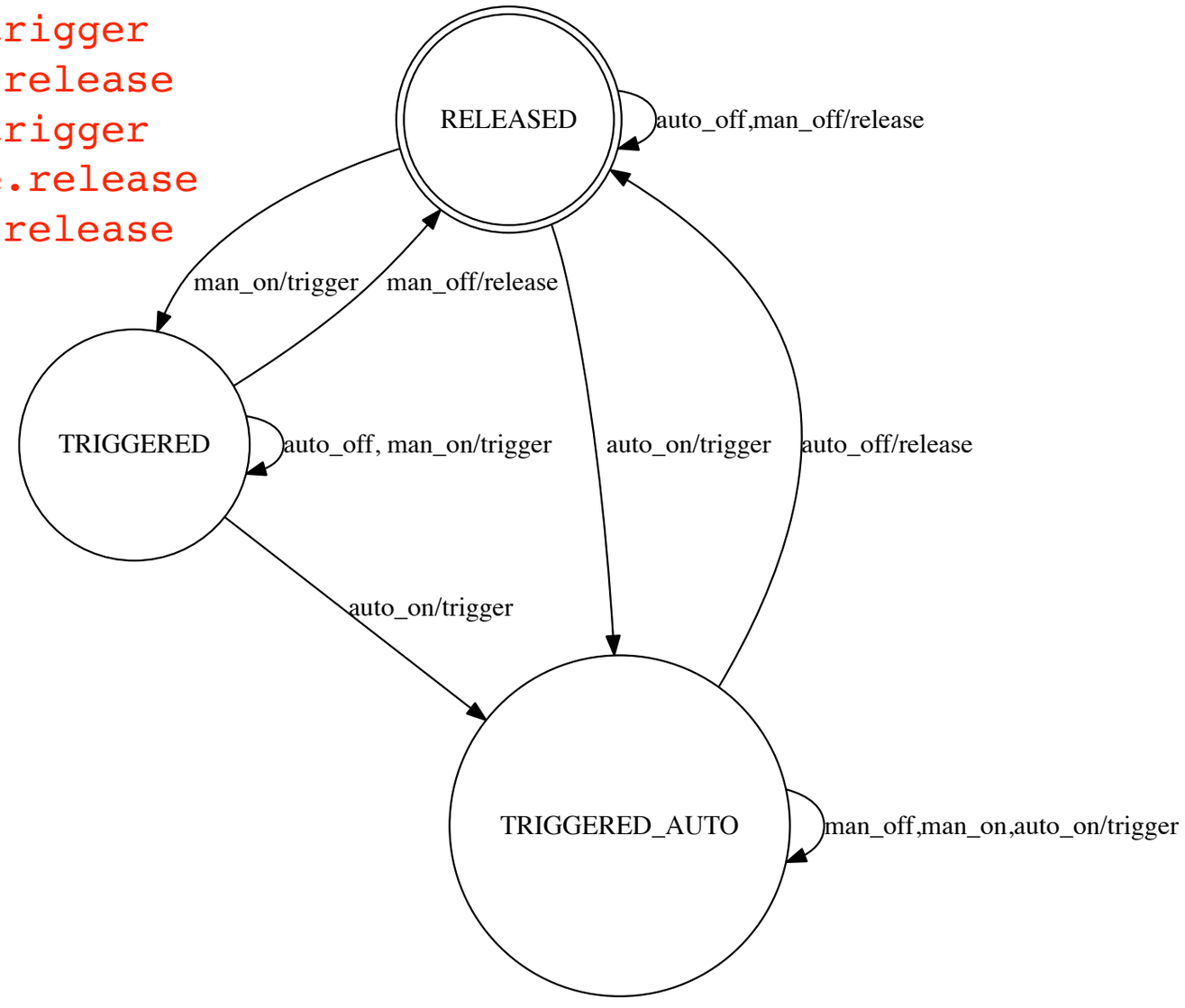
$$\mathbf{TS}_I = V_I.\left(\bigcup_{i=0}^{m-n+1} I^i \right).W_I$$

Test cases for hypothesis m = 3

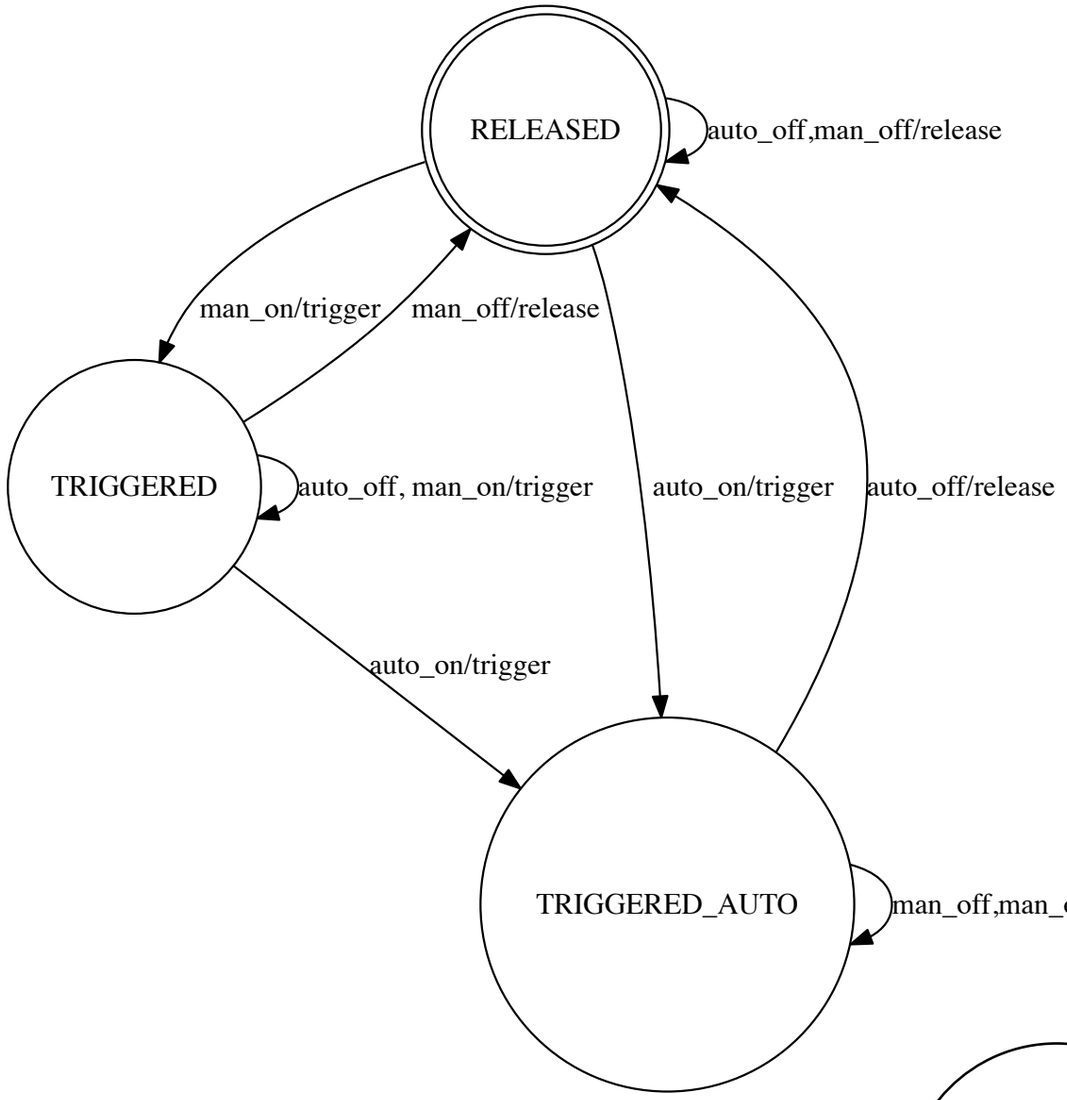
1. man_on.man_on.auto_off/trigger.trigger.trigger
2. man_on.man_on.man_off/trigger.trigger.release
3. man_on.auto_on.auto_off/trigger.trigger.release
4. man_on.auto_on.man_off/trigger.trigger.trigger
5. man_on.man_off.auto_off/trigger.release.release
6. man_on.man_off.man_off/trigger.release.release
7. man_on.auto_off.auto_off/trigger.trigger.trigger
8. man_on.auto_off.man_off/trigger.trigger.release
9. auto_on.man_on.auto_off/trigger.trigger.release
10. auto_on.man_on.man_off/trigger.trigger.trigger
11. auto_on.auto_on.auto_off/trigger.trigger.release
12. auto_on.auto_on.man_off/trigger.trigger.trigger
13. auto_on.man_off.auto_off/trigger.trigger.release
14. auto_on.man_off.man_off/trigger.trigger.trigger
15. auto_on.auto_off.auto_off/trigger.release.release
16. auto_on.auto_off.man_off/trigger.release.release
17. man_off.auto_off/release.release
18. man_off.man_off/release.release
19. auto_off.auto_off/release.release
20. auto_off.man_off/release.release

21. man_off/release is a prefix of 17.
22. auto_off/release is a prefix of 19.

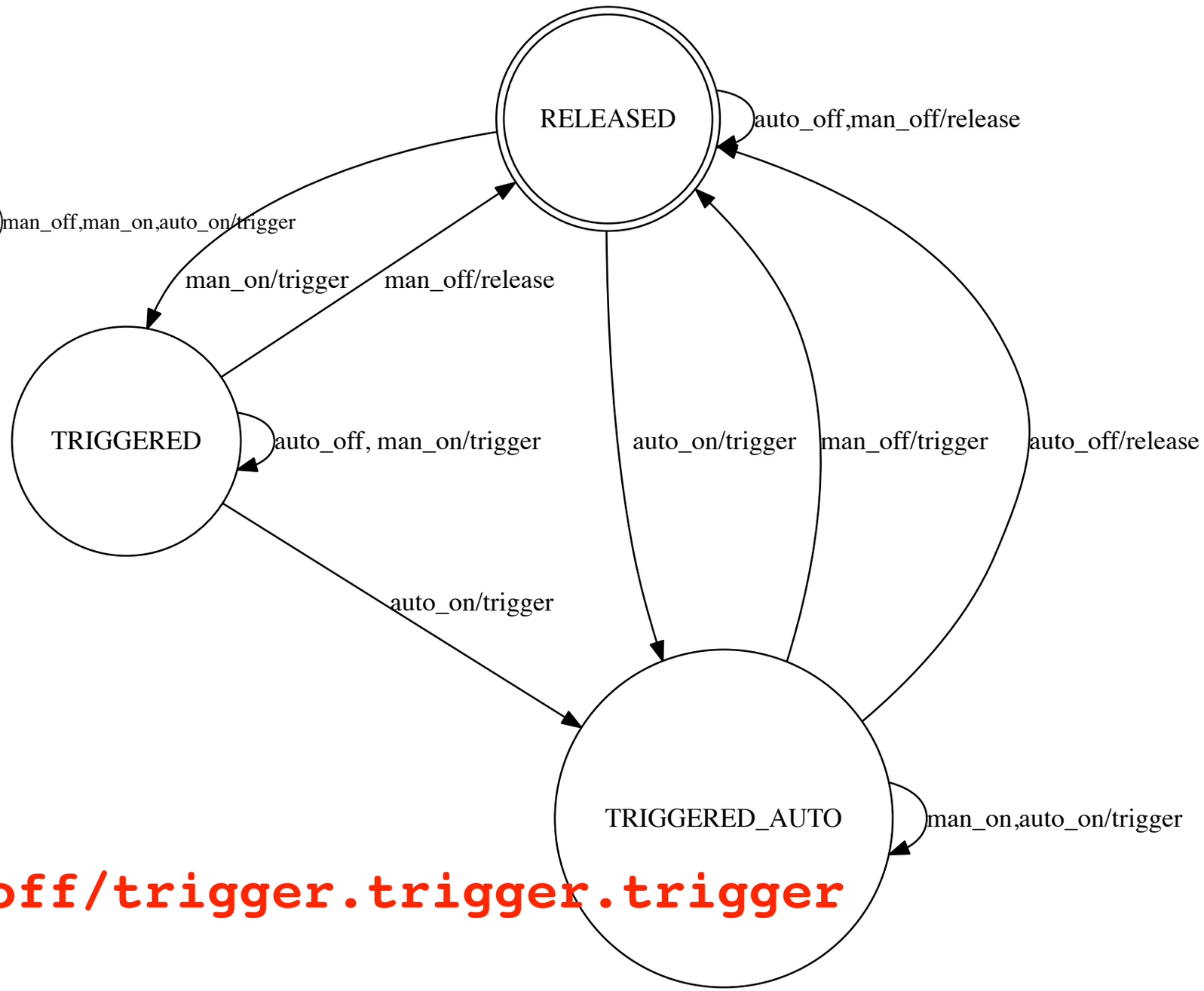
This test suite uncovers every error, provided that the implementation has at most 3 states



Reference model

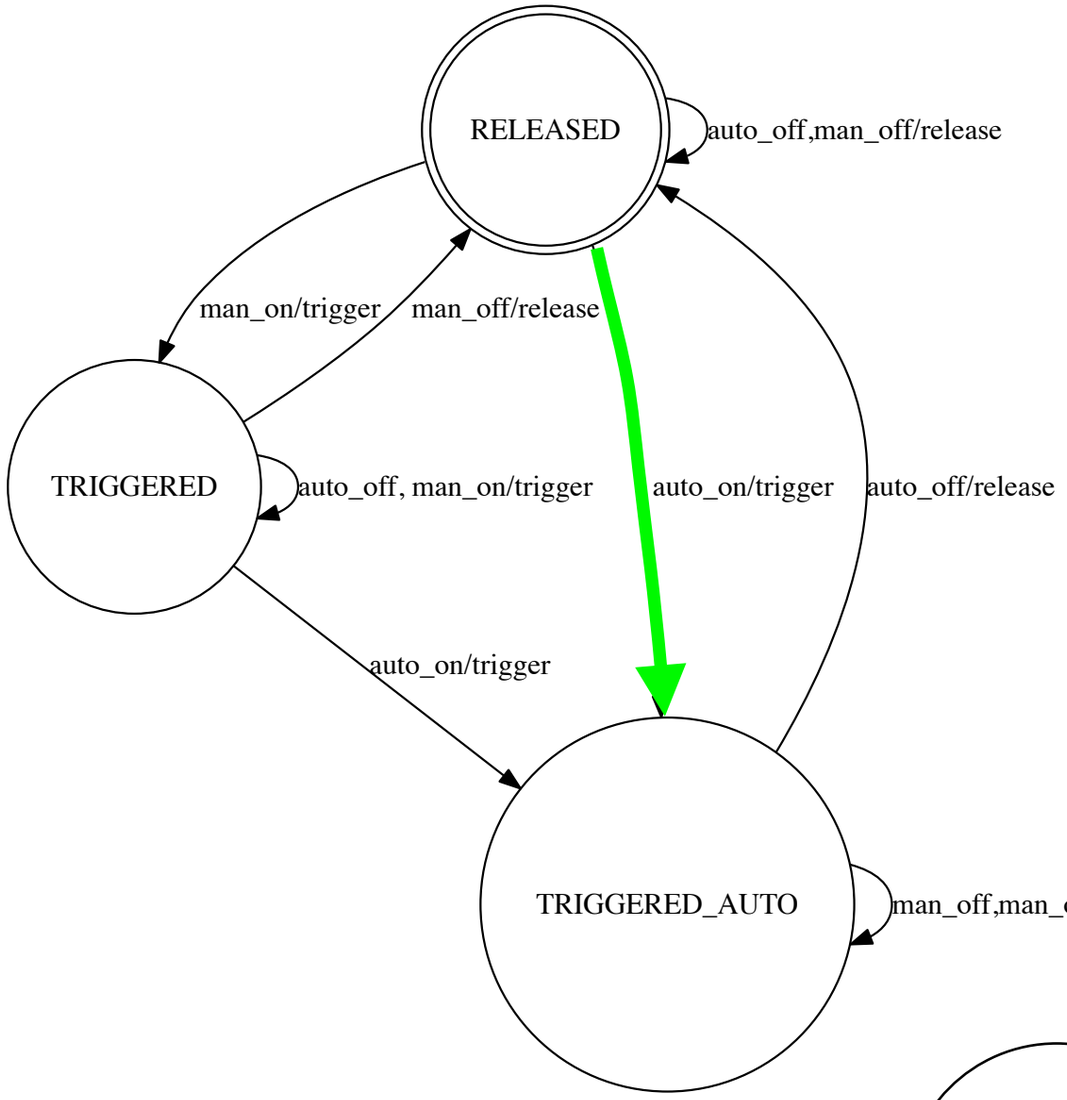


Faulty implementation

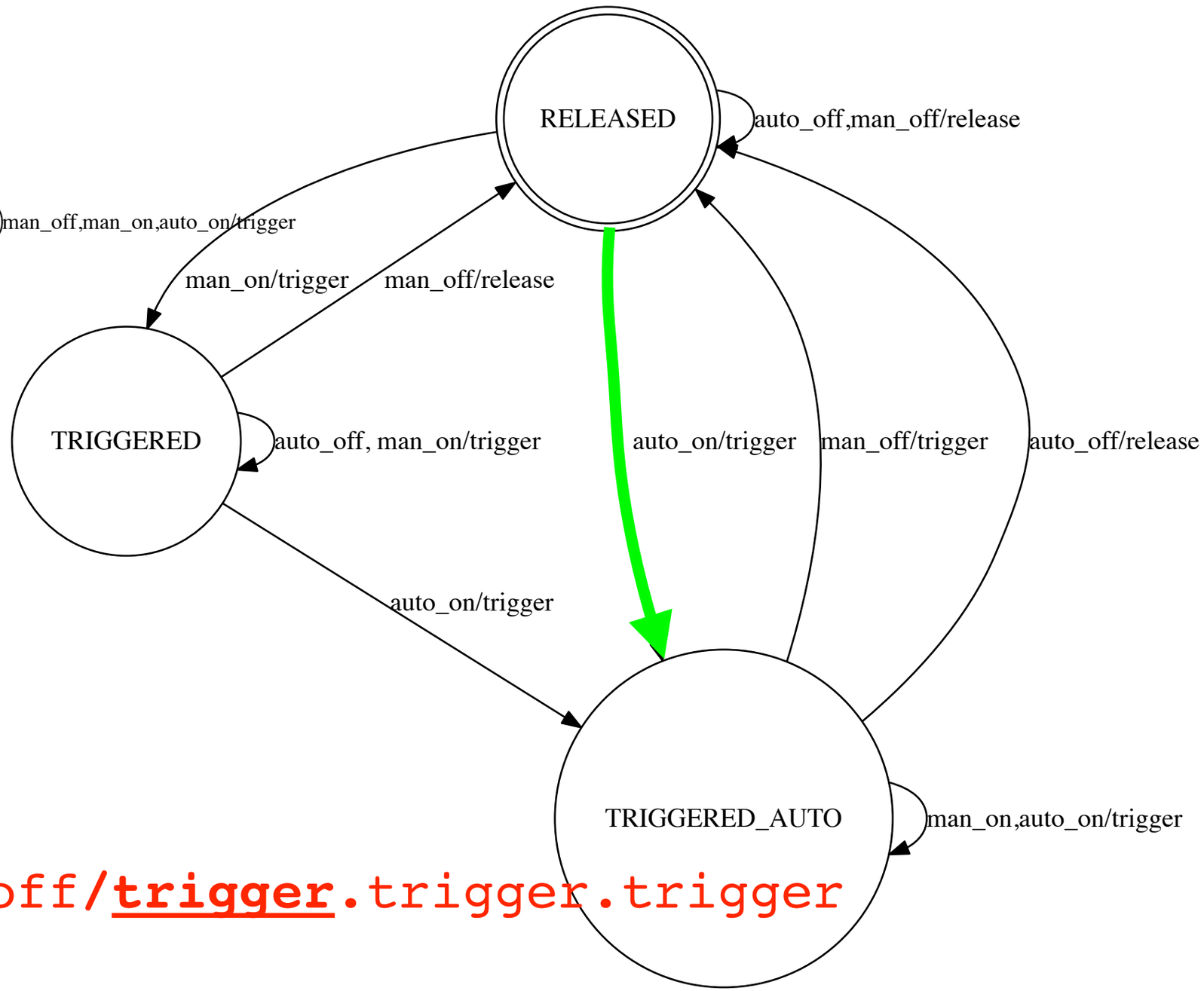


Test case 14.
auto_on.man_off.man_off/trigger.trigger.trigger

Reference model



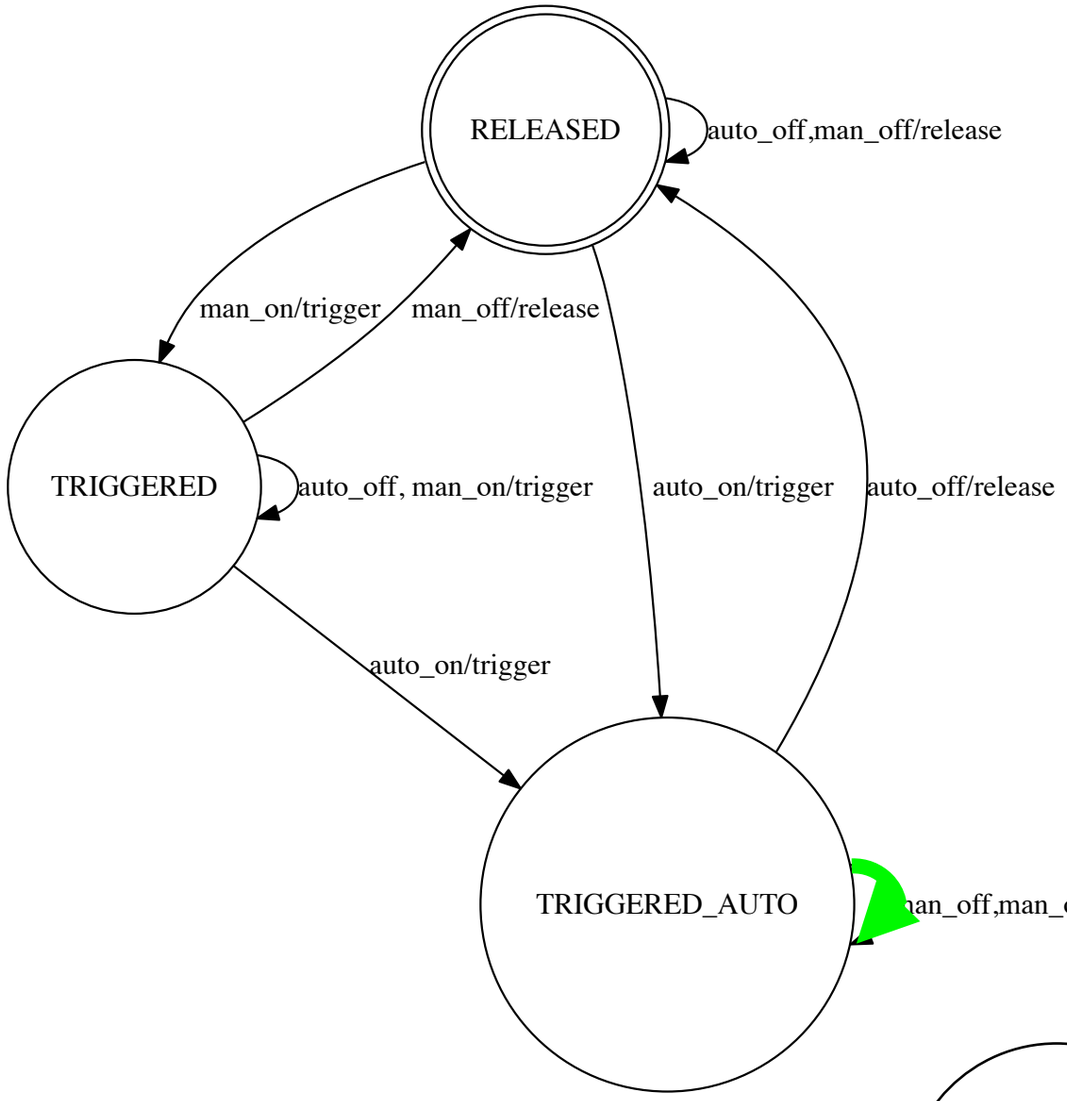
Faulty implementation



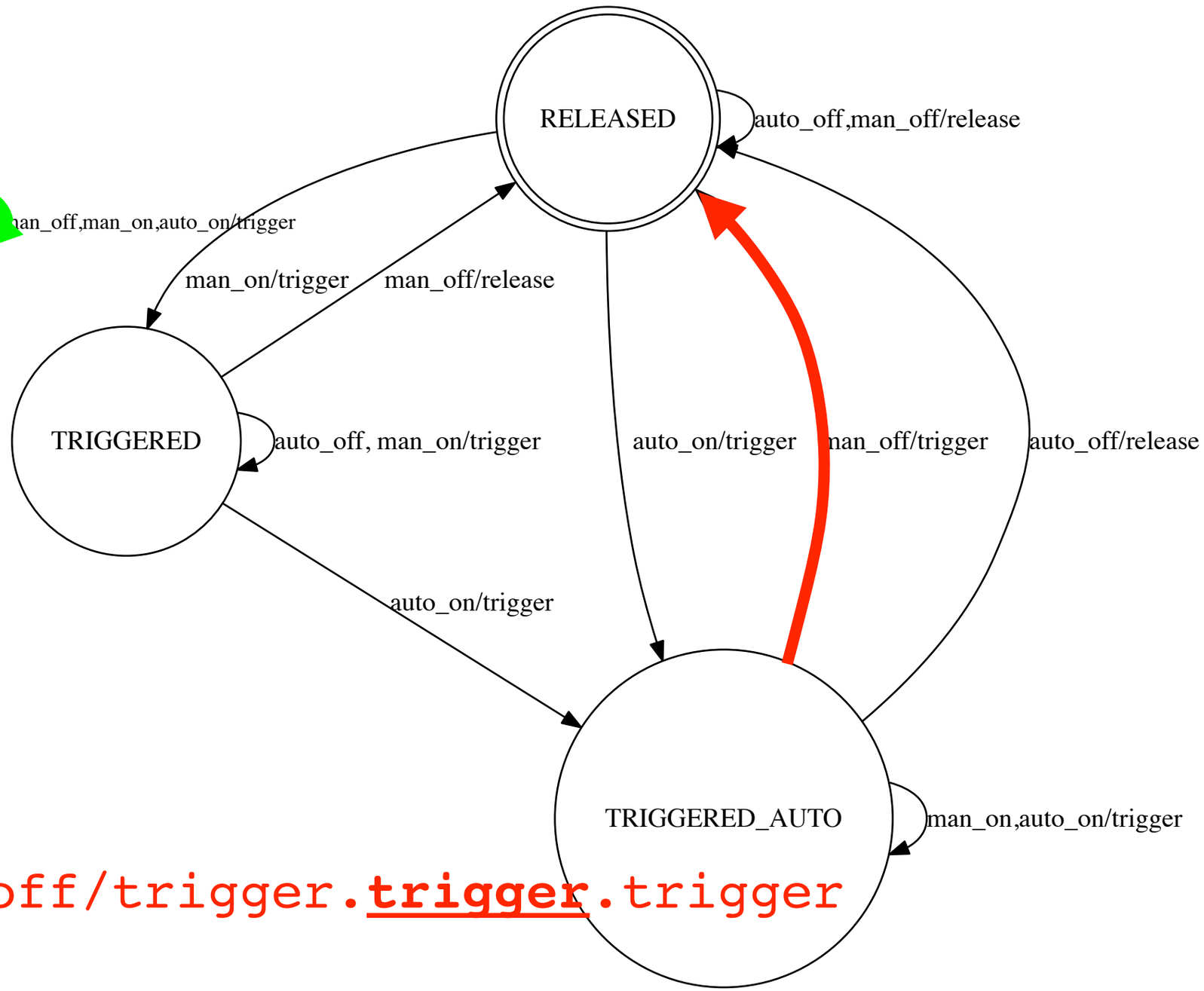
Test case 14.

auto_on.man_off.man_off/**trigger**.trigger.trigger

Reference model



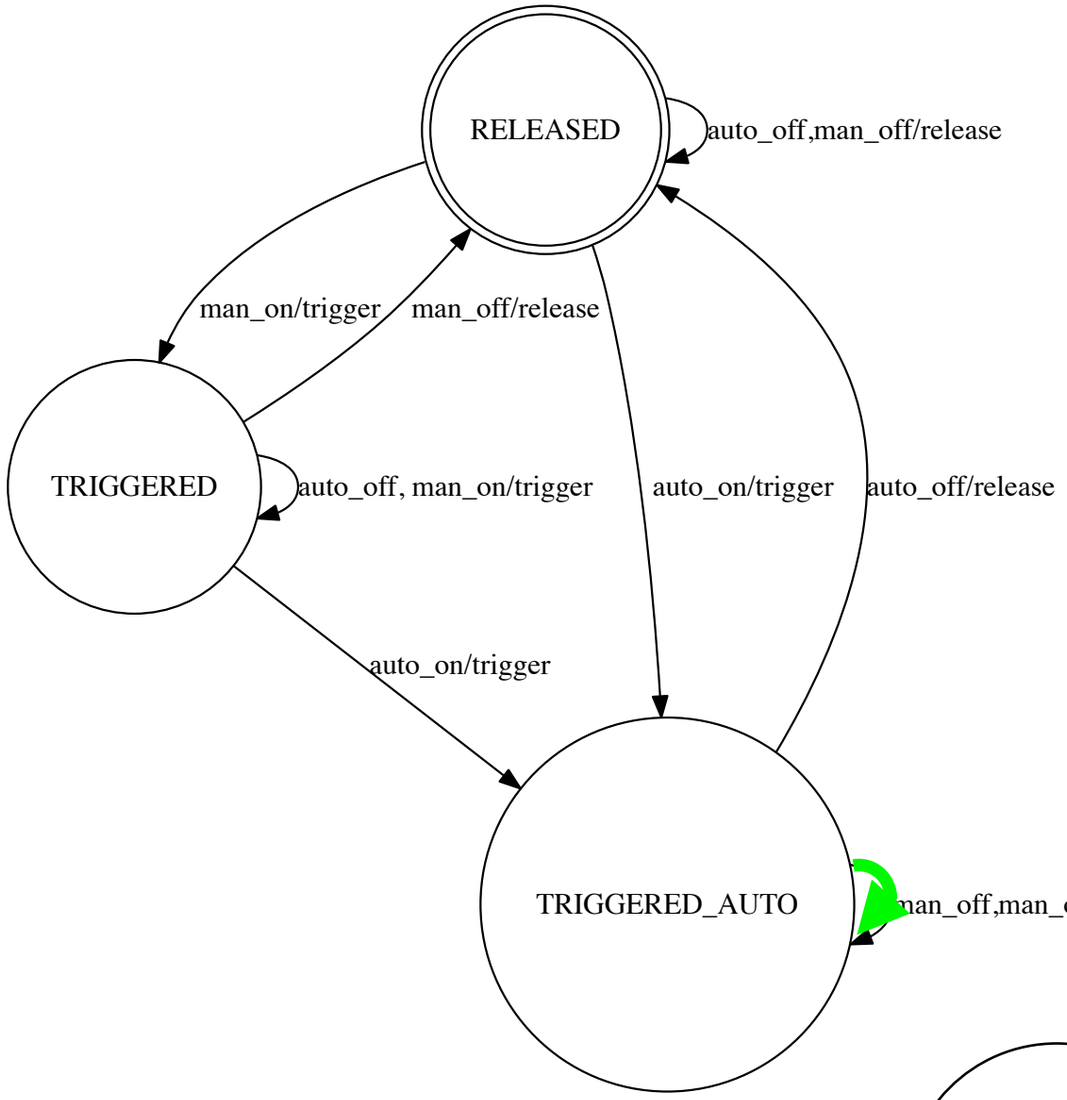
Faulty implementation



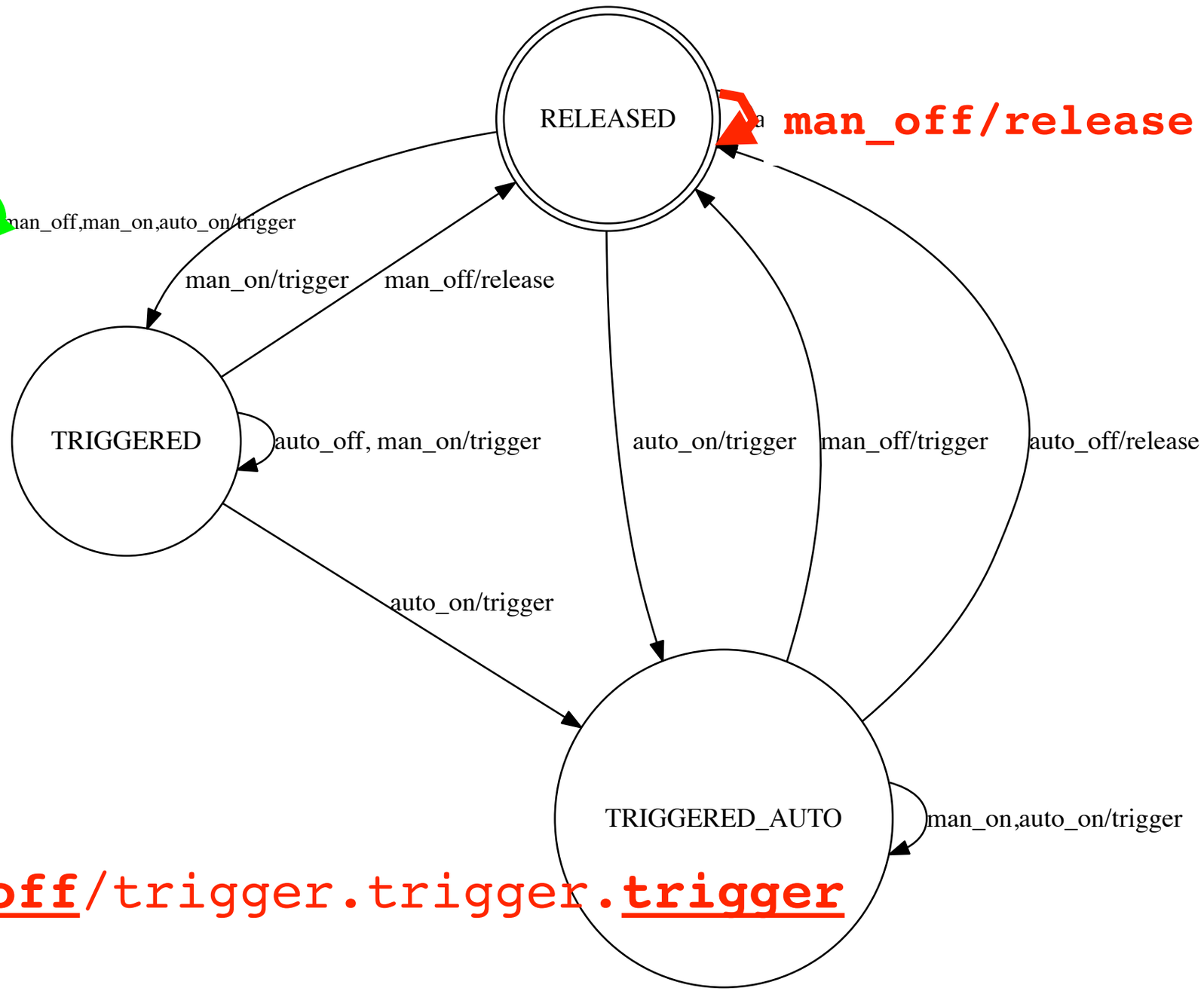
Test case 14.

auto_on.**man_off**.man_off/trigger.**trigger**.trigger

Reference model



Faulty implementation



Test case 14.

auto_on.man_off.**man_off**/trigger.trigger.**trigger**

Wp-Method

$\mathcal{F}(M, I, O, \leq, \mathcal{D}_m)$, fault model

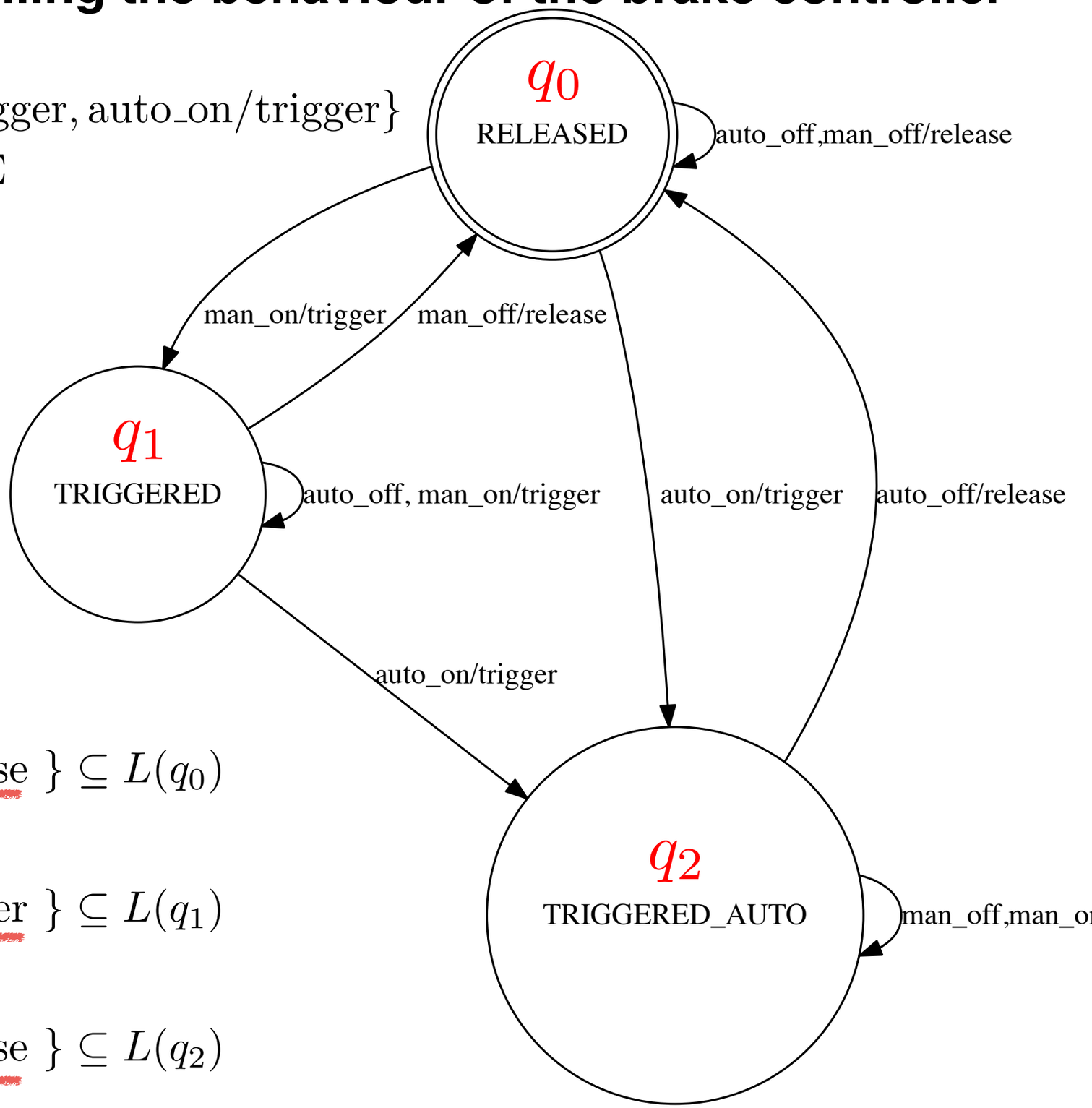
$\mathcal{D}_m = \{M' = (Q', q'_0, I, O, h') \in Sig_1 \mid |Q'| \leq m\}$

1. V a state cover of M.
2. $P = V \oplus (\{\varepsilon\} \cup \Sigma)$ a transition cover of M
3. $R = P \setminus V$.
4. W a characterisation set of M.
5. $\{W_0, \ldots, W_{n-1}\}$ *state identification sets* of M, such that
   - $W_i \subseteq pref(W)$ for $i = 0, \ldots, n-1$.
   - W_i distinguishes q_i from all other states in Q.

Finite State Machine modelling the behaviour of the brake controller

- state cover $V = \{\varepsilon, \text{man\_on/trigger}, \text{auto\_on/trigger}\}$
- transition cover $P = V \cup V \oplus \Sigma$

$$q_0 \xrightarrow{\text{auto\_off/release}} q_0$$
$$q_1 \xrightarrow{\text{auto\_off/trigger}} q_1$$
$$q_2 \xrightarrow{\text{auto\_off/release}} q_0$$
$$q_0 \xrightarrow{\text{man\_off/release}} q_0$$
$$q_2 \xrightarrow{\text{man\_off/trigger}} q_2$$



- $\{\text{man\_off/release}, \text{auto\_off/release}\} \subseteq L(q_0)$ — W_0
- $\{\text{man\_off/release}, \text{auto\_off/trigger}\} \subseteq L(q_1)$ — W_1
- $\{\text{man\_off/trigger}, \text{auto\_off/release}\} \subseteq L(q_2)$ — W_2

- characterization set $W = \{\text{man\_off/trigger}, \text{man\_off/release}, \text{auto\_off/trigger}, \text{auto\_off/release}\}$

Wp-Method

- $Wp_1 = V \oplus \left(\bigcup_{i=0}^{m-n} \Sigma^i \right) \oplus W$
- $Wp_2 = R \oplus \Sigma^{m-n} \oplus \{W_0, \ldots, W_{n-1}\}$
- $\mathbf{TS} = Wp_1 \cup Wp_2$
  is a complete test suite of $\mathcal{F} = (M, I, O, \sim, \mathcal{D}_m)$.

$$U \oplus \{W_0, \ldots, W_{n-1}\} =$$

$$\bigcup_{\pi \in U \wedge q_i = q_0\text{-after-}\pi} \{\pi\}.W_i$$

Wp-Method

- $W p_1 = V \oplus \left(\bigcup_{i=0}^{m-n} \Sigma^i \right) \oplus W$
- $W p_2 = R \oplus \Sigma^{m-n} \oplus \{W_0, \ldots, W_{n-1}\}$
- $\mathbf{TS} = W p_1 \cup W p_2$
  is a complete test suite of $\mathcal{F} = (M, I, O, \sim, \mathcal{D}_m)$.

$$U \oplus \{W_0, \ldots, W_{n-1}\} =$$

$$\bigcup_{\pi \in U \wedge q_i = q_0\text{-after-}\pi} \{\pi\}.W_i$$

Reactive I/O Transition System (Sig_1)

Reactive I/O Transition System

$\mathcal{S} = (S, s_0, R)$ reactive I/O transition system:

- $S = I \times M \times O$: state space $S = \{(x, m, y) \mid x \in I, m \in M, y \in O\}$
- $s_0 \in S$: initial state
- $I \neq \varnothing$: input alphabet
- $M \neq \varnothing$: internal state values
- $O \neq \varnothing$: output alphabet
- $R \subseteq S \times S$: transition relation

$$S = S_Q \cup S_T,\ S_Q \cap S_T = \varnothing$$

- S_Q: quiescent states **靜態狀態**
- S_T: transient states **瞬變狀態**

$R \subseteq S \times S$:

- s_1 quiescent: $(s_1, s_2) \in R \Leftrightarrow s_2(m, y) = s_1(m, y)$
- s_1 transient : $(s_1, s_2) \in R \Rightarrow s_2(x) = s_1(x) \wedge s_2 \in S_Q$

R ⋯⋯⋯⋯⋯▶

$(x, m, y) \xrightarrow{x'} (x', m, y) \xrightarrow{y'} (x', m', y')$

quiescent state transient state quiescent state

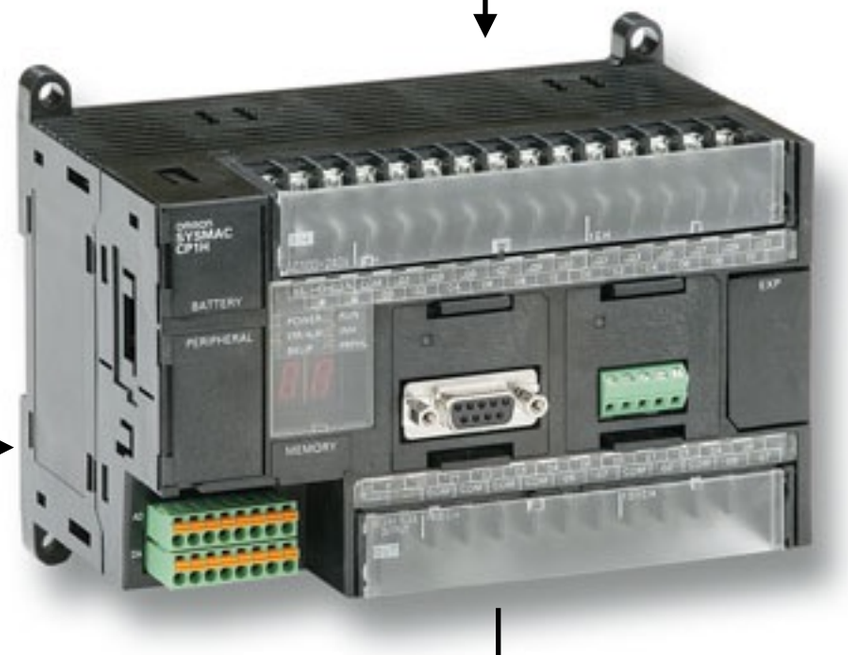
Recall: Application Scenario – Train Onboard Speed Control

Train engine driver brake command

man_on, man_off

Onboard main controller

Brake controller



Automated braking command

auto_on, auto_off

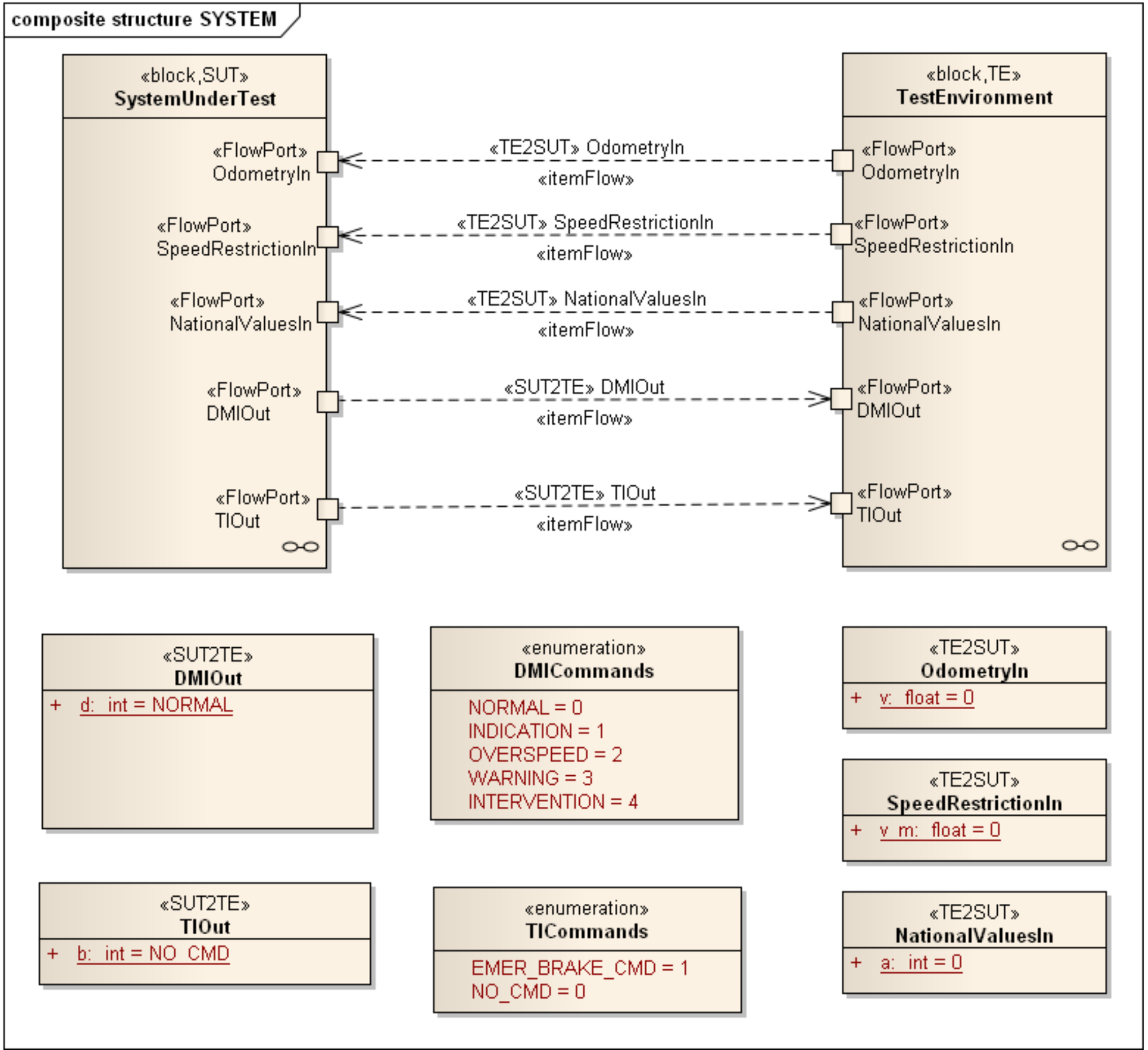
trigger, release

Current speed v

Current maximum speed v_m

Emergency brake

最高速限監測器


composite structure SYSTEM
«block,SUT»
SystemUnderTest
«block,TE»
TestEnvironment
«FlowPort»
OdometryIn
«TE2SUT» OdometryIn
«itemFlow»
«FlowPort»
OdometryIn
«FlowPort»
SpeedRestrictionIn
«TE2SUT» SpeedRestrictionIn
«itemFlow»
«FlowPort»
SpeedRestrictionIn
«FlowPort»
NationalValuesIn
«TE2SUT» NationalValuesIn
«itemFlow»
«FlowPort»
NationalValuesIn
«FlowPort»
DMIOut
«SUT2TE» DMIOut
«itemFlow»
«FlowPort»
DMIOut
«FlowPort»
TIOut
«SUT2TE» TIOut
«itemFlow»
«FlowPort»
TIOut
«SUT2TE»
DMIOut
+ d: int = NORMAL
«enumeration»
DMICommands
NORMAL = 0
INDICATION = 1
OVERSPEED = 2
WARNING = 3
INTERVENTION = 4
«TE2SUT»
OdometryIn
+ v: float = 0
«TE2SUT»
SpeedRestrictionIn
+ v_m: float = 0
«SUT2TE»
TIOut
+ b: int = NO_CMD
«enumeration»
TICommands
EMER_BRAKE_CMD = 1
NO_CMD = 0
«TE2SUT»
NationalValuesIn
+ a: int = 0

最高速限監測器

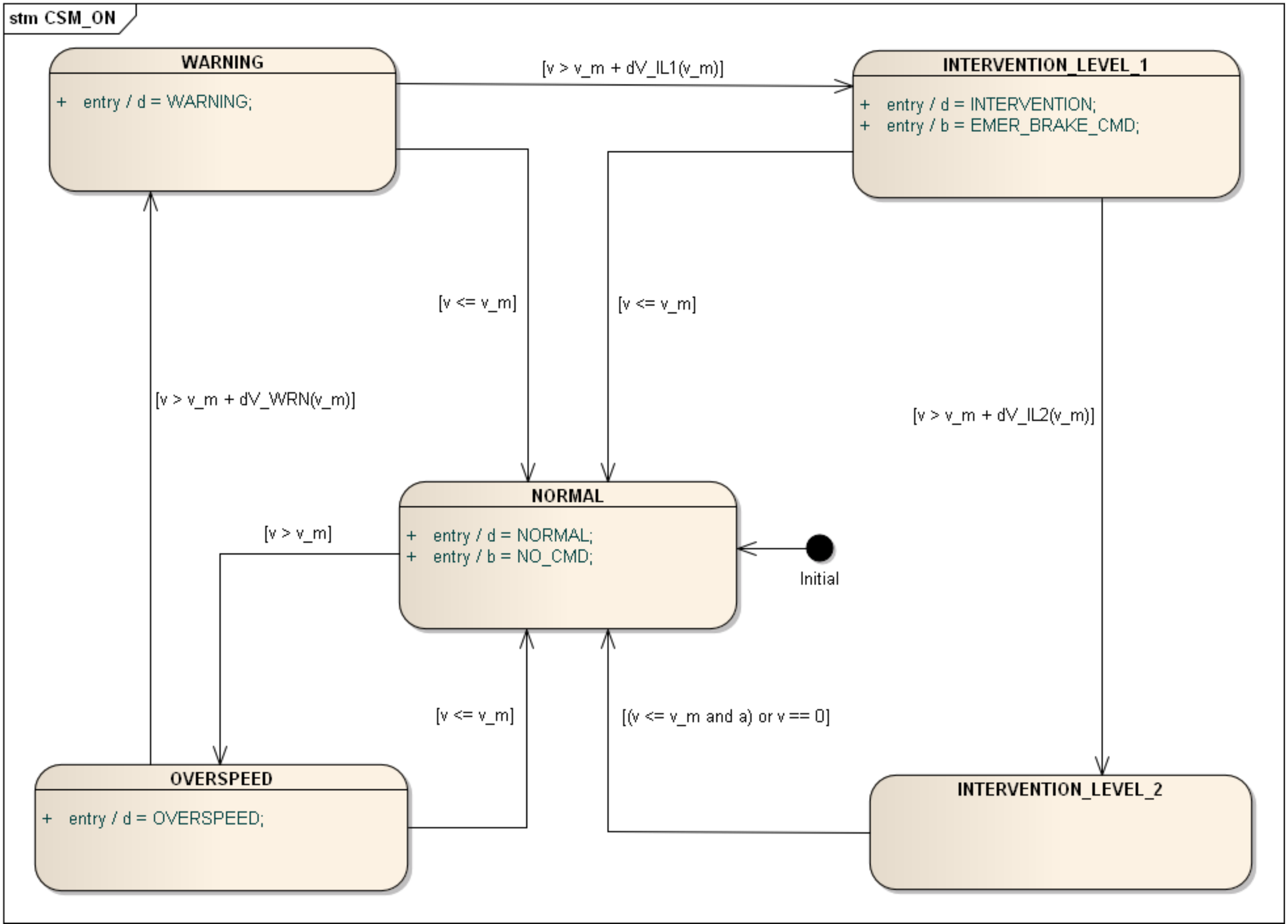

stm CSM_ON
WARNING
+ entry / d = WARNING;
[v > v_m + dV_IL1(v_m)]
INTERVENTION_LEVEL_1
+ entry / d = INTERVENTION;
+ entry / b = EMER_BRAKE_CMD;
[v <= v_m]
[v <= v_m]
[v > v_m + dV_WRN(v_m)]
[v > v_m + dV_IL2(v_m)]
NORMAL
+ entry / d = NORMAL;
+ entry / b = NO_CMD;
[v > v_m]
Initial
[v <= v_m]
[(v <= v_m and a) or v == 0]
OVERSPEED
+ entry / d = OVERSPEED;
INTERVENTION_LEVEL_2

Table 1: Identification of basic states in machine CSM_ON

State Machine in Basic State	d	m
NORMAL	0	1
OVERSPEED	2	2
WARNING	3	3
INTERVENTION_LEVEL_1	4	4
INTERVENTION_LEVEL_2	4	5

stm CSM_ON

m=3

WARNING

+ entry / d = WARNING;

[v > v_m + dV_IL1(v_m)]

m=4

INTERVENTION_LEVEL_1

+ entry / d = INTERVENTION;
+ entry / b = EMER_BRAKE_CMD;

[v <= v_m]

[v > v_m + dV_WRN(v_m)]

[v <= v_m]

[v > v_m + dV_IL2(v_m)]

m=1

NORMAL

+ entry / d = NORMAL;
+ entry / b = NO_CMD;

Initial

[v > v_m]

[v <= v_m]

[(v <= v_m and a) or v == 0]

m=2

OVERSPEED

+ entry / d = OVERSPEED;

m=5

INTERVENTION_LEVEL_2

- $x = (v, v_m, a) \in I,\ I = [0, 350] \times [0, 350] \times \{0, 1\}$
- $m \in \{1, 2, 3, 4, 5\}$
- $y = (d, b) \in O,\ O = \{0, 2, 3, 4\} \times \{0, 1\}$

$$\mathrm{dV_{WRN}}(v_m) = \begin{cases} 4 & \text{if} \quad v_m \leq 110 \\ \frac{1}{3} + \frac{1}{30} \cdot v_m & \text{if} \quad 110 < v_m \leq 140 \\ 5 & \text{if} \quad 140 < v_m \end{cases} \tag{1}$$

$$\mathrm{dV_{IL1}}(v_m) = \begin{cases} 5.5 & \text{if} \quad v_m \leq 110 \\ 0.55 + 0.045 \cdot v_m & \text{if} \quad 110 < v_m \leq 210 \\ 10 & \text{if} \quad 210 < v_m \end{cases} \tag{2}$$

$$\mathrm{dV_{IL2}}(v_m) = \begin{cases} 7.5 & \text{if} \quad v_m \leq 110 \\ -0.75 + 0.075 \cdot v_m & \text{if} \quad 110 < v_m \leq 210 \\ 15 & \text{if} \quad 210 < v_m \end{cases} \tag{3}$$

quiescent pre-state

$$\mathcal{R} \equiv \bigvee_{i \in \mathrm{IDX}} \left(\alpha_i \wedge (m, y) = (i, \mathbf{e}_i) \wedge (m', y') = (i, \mathbf{e}_i)\right)$$

$$\vee \bigvee_{(i,j) \in J} \left(g_{i,j} \wedge (m, y) = (i, \mathbf{e}_i) \wedge (m', y') = (j, \mathbf{e}_j)\right)$$

transient pre-state

quiescent post-state

$$\mathrm{IDX} = \{1, 2, 3, 4, 5\}$$

$$J = \{(1,2),(1,3),(1,4),(1,5),(2,1),(2,3),(2,4),(2,5),$$
$$(3,1),(3,4),(3,5),(4,1),(4,5),(5,1)\}$$

$$\mathbf{e}_1 = (d = 0, b = 0), \mathbf{e}_2 = (2, 0), \mathbf{e}_3 = (3, 0), \mathbf{e}_4 = \mathbf{e}_5 = (4, 1)$$

$$
\begin{aligned}
\alpha_1 &\equiv v \leq v_m \\
\alpha_2 &\equiv v_m < v \wedge v \leq v_m + \mathrm{dV_{WRN}}(v_m) \\
\alpha_3 &\equiv v_m < v \wedge v \leq v_m + \mathrm{dV_{IL1}}(v_m) \\
\alpha_4 &\equiv v_m < v \wedge v \leq v_m + \mathrm{dV_{IL2}}(v_m) \\
\alpha_5 &\equiv (0 < v \wedge a = 0) \vee (v_m < v \wedge a = 1)
\end{aligned}
$$

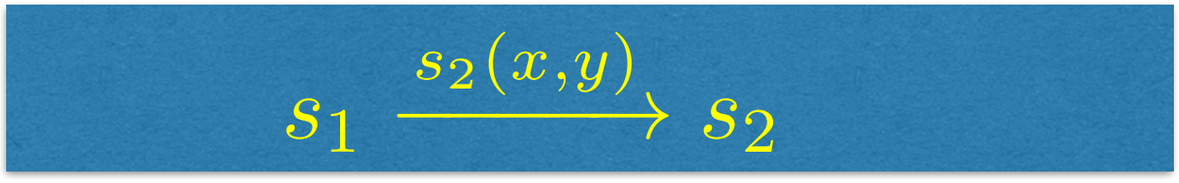
$$
\begin{aligned}
g_{1,2} &\equiv v_m < v \wedge v \leq v_m + \mathrm{dV_{WRN}}(v_m) \\
g_{1,3} &\equiv v_m + \mathrm{dV_{WRN}}(v_m) < v \leq v_m + \mathrm{dV_{IL1}}(v_m) \\
g_{1,4} &\equiv v_m + \mathrm{dV_{IL1}}(v_m) < v \leq v_m + \mathrm{dV_{IL2}}(v_m) \\
g_{1,5} &\equiv v_m + \mathrm{dV_{IL2}}(v_m) < v
\end{aligned}
$$

$$
\begin{aligned}
g_{2,3} &\equiv g_{1,3} \\
g_{2,4} &\equiv g_{3,4} \equiv g_{1,4} \\
g_{2,5} &\equiv g_{3,5} \equiv g_{4,5} \equiv g_{1,5} \\
g_{2,1} &\equiv v \leq v_m \\
g_{3,1} &\equiv g_{4,1} \equiv g_{2,1} \\
g_{5,1} &\equiv v = 0 \vee (v \leq v_m \wedge a = 1)
\end{aligned}
$$

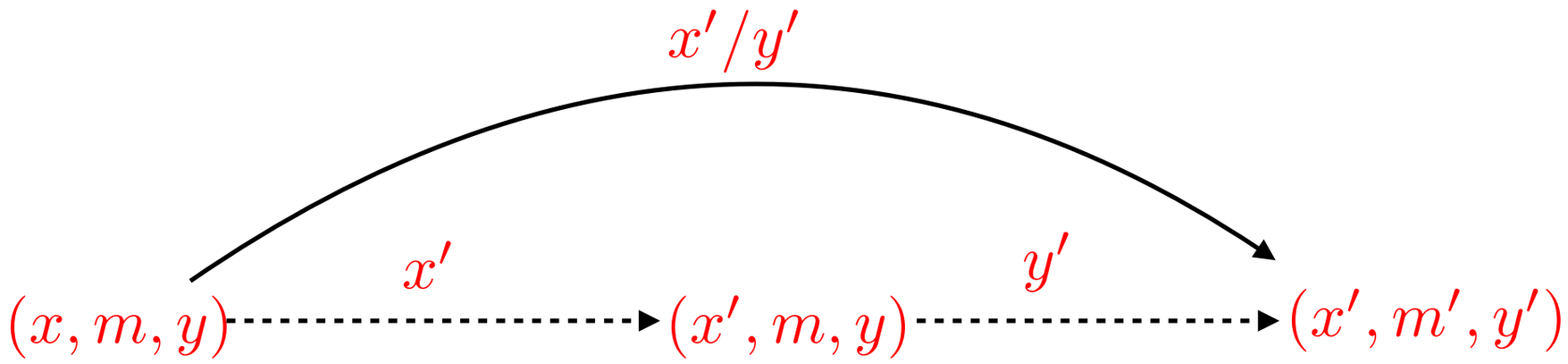
Quiescent Reduction

(S_Q, s_0, R_Q) is called quiescent reduction of (S, s_0, R):

- $S_Q \subseteq S$ set of quiescent states
- $R_Q \subseteq S_Q \times S_Q, \forall s_1, s_2 \in S_Q, (s_1, s_2) \in R_Q :\Leftrightarrow$
  - $(s_1, s_2) \in R$ or
  - $\exists s \in S_T, (s_1, s), (s, s_2) \in R$

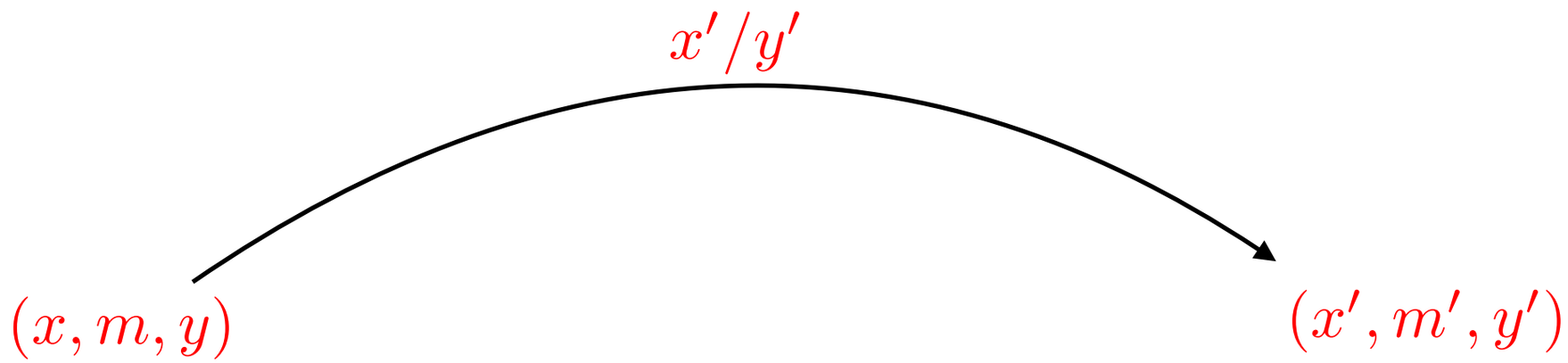


R_Q ⟶ R ⇢



quiescent state transient state quiescent state

R_Q ⟶



quiescent state

quiescent state

$s \xrightarrow{(x_1,y_1)\ldots(x_k,y_k)} s'$, if and only if
$\exists s, s_1, \ldots, s_k = s' \in S_Q$, such that

1. $s_i(x, y) = (x_i, y_i), i = 1, \ldots, k$

2. $(s, s_1), (s_i, s_{i+1}) \in R_Q, i = 1, \ldots, k-1$

Language

$\mathcal{S} = (S, s_0, R)$
For any $s \in S_Q$ quiescent state

- $L(s) := \{\pi \in \Sigma^* \mid \exists s' \in S_Q, s \xrightarrow{\pi} s'\}$ **language of** s
- $s \sim s' :\Leftrightarrow L(s) = L(s')$ s s' **are equivalent**
- $L(\mathcal{S}) := L(s_0)$ **language of** $\mathcal{S}$

RIOSTS Properties

$\mathcal{S}_Q = (S_Q, s_0, R_Q)$

- $\mathcal{S}_Q$ is **input-complete** : always holds

  $\forall s \in S_Q \wedge x \in I, \exists y \in O \wedge s' \in S_Q : s \xrightarrow{x/y} s'$

- $\mathcal{S}_Q$ is **deterministic** :

  $\forall s \overset{x/y_1}{\rightarrow} s_1, s \overset{x/y_2}{\rightarrow} s_2 \in R_Q \Rightarrow y_1 = y_2 \wedge s_1 = s_2$

RIOSTS Properties

- $\mathcal{S}_Q$ is **observable**, if
  $\forall s \xrightarrow{x/y} s_1, s \xrightarrow{x/y} s_2 \in R_Q \Rightarrow s_1 = s_2$

- $\mathcal{S}_Q$ is **minimal**, if always fails,if I contains more than two inputs
  - $\forall s \in S_Q, \exists \pi \in L(s_0) : s_0 \xrightarrow{\pi} s$
  - $\forall s_1 \neq s_2 \in S_Q \Rightarrow L(s_1) \neq L(s_2)$

State Equivalence

等価類

$$\mathcal{S} = (S, s_0, I, O, R),\ s, s' \in S_Q$$

$$s \sim s' :\Leftrightarrow L(s) = L(s') \quad s\ s' \textbf{ are equivalent}$$

$$[s] = \{s' \in S_Q \mid s \sim s'\} \quad \textbf{state equivalence class}$$

$$S_Q/\sim = \{[s] \mid s \in S_Q\} \quad \textbf{equivalence class partition}$$

finite ? or infinite ?

When is $S_Q/\sim$ finite?

sufficient condition : M and O are finite.

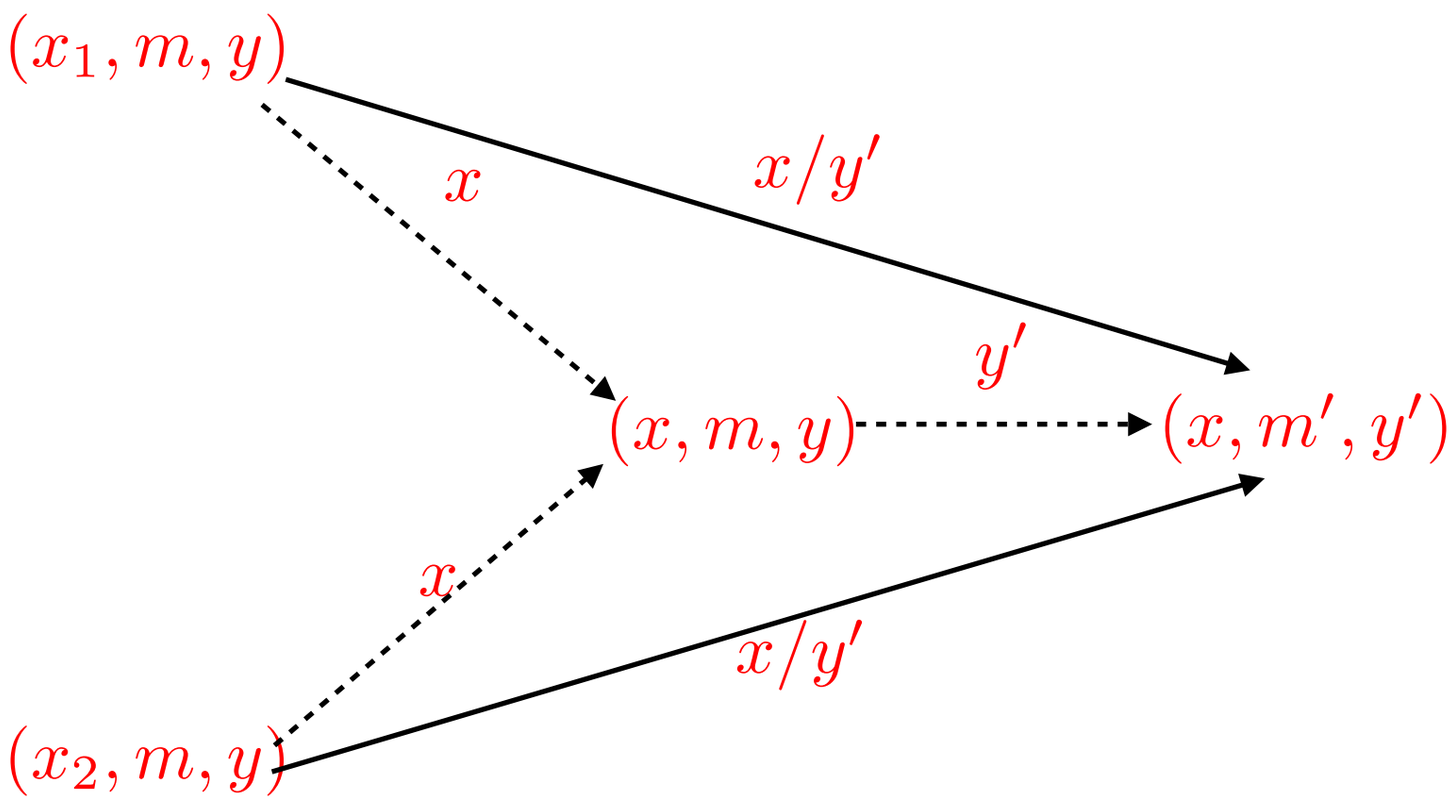
$$s_1 = (x_1, m_1, y_1), s_2 = (x_2, m_2, y_2) \in S_Q \wedge (m_1, y_1) = (m_2, y_2)$$
$$\Rightarrow \forall x \in I, s = (x, m_1, y_1) \in S : (s_1, s), (s_2, s) \in R$$
$$\Rightarrow \forall x \in I, s' = (x, m', y') \in S_Q : ((s_1, s') \in R_Q \Leftrightarrow (s_2, s') \in R_Q)$$
$$\Rightarrow L(s_1) = L(s_2)$$

$M \times O =: \{(m_1, y_1), \ldots, (m_n, y_n)\}$, $n \leq |M| \cdot |O|$

$A_i := \{s \in S_Q \mid s(m, y) = (m_i, y_i)\}, i = 1, \ldots, n$

$\{A_i \mid i = 1, \ldots, n\}$ is a refinement of $S_Q/\sim$

R_Q ⟶ R ⇢



$$\mathcal{R} \equiv \bigvee_{i \in \mathrm{IDX}} \left(\alpha_i \wedge (m, y) = (m_i, y_i) \wedge (m', y') = (m_i, y_i)\right)$$

$$\vee \bigvee_{(i,j) \in J} \left(g_{i,j} \wedge (m, y) = (m_i, y_i) \wedge (m', y') = (m_j, y_j)\right)$$

$$A_i = \{s \in S \mid s(\alpha_i) \wedge s(m) = m_i \wedge s(y) = y_i\}$$

Signature Sig_1 is a set of

deterministic RIOSTS with

- finite outputs
- finite internal state values

Let $S_Q/\sim = \{[s_1], \ldots, [s_n]\}$ and $O = \{y_1, \ldots, y_\ell\}$

The **transition index function**

$$\delta : I \to \{1, \ldots, n\}^n; \quad x \mapsto (\delta_1(x), \ldots, \delta_n(x))$$

$$\delta_i(x) = j :\Leftrightarrow \forall s \in [s_i], \exists s' \in [s_j], s \xrightarrow{x/y} s'$$

The **output index function**

$$\omega : I \to \{1, \ldots, \ell\}^n; \quad x \mapsto (\omega_1(x), \ldots, \omega_n(x))$$

$$\omega_i(x) = k :\Leftrightarrow \forall s \in [s_i], \exists s' \in S_Q, s \xrightarrow{x/y_k} s'$$

The **transition index function**

$$\delta : I \to \{1, \ldots, n\}^n; \quad x \mapsto (\delta_1(x), \ldots, \delta_n(x))$$

$$\delta_i(x) = j :\Leftrightarrow \forall s \in [s_i], \exists s' \in [s_j], s \xrightarrow{x/y} s'$$

and the **output index function**

$$\omega : I \to \{1, \ldots, \ell\}^n; \quad x \mapsto (\omega_1(x), \ldots, \omega_n(x))$$

$$\omega_i(x) = k :\Leftrightarrow \forall s \in [s_i],\ x/y_k \in L(s)$$

are well-defined:

$$s \sim u \Leftrightarrow L(s) = L(u) \Leftrightarrow \forall x \in I:$$
$$(s \overset{x/y}{\to} s', u \overset{x/y'}{\to} u' \Rightarrow s' \sim u' \wedge y = y')$$

Let $S_Q/\sim = \{[s_1], \ldots, [s_n]\}$ and $O = \{y_1, \ldots, y_\ell\}$

$$\delta_i(x) = j \wedge \omega_i(x) = k$$

$$x/y_k$$

$$[s_i] \longrightarrow [s_j]$$

Input Equivalence

$\mathcal{S} = (S, s_0, I, O, R),\ x, x' \in I$

$x \sim x' :\Leftrightarrow \delta(x) = \delta(x') \wedge \omega(x) = \omega(x')$ $x\ x'$ **are equivalent**

$[x] = \{x' \in I \mid x \sim x'\}$ **input equivalence class**

$I/_\sim = \{[x] \mid x \in I\}$ **input equivalence class partition**

finite ? or infinite ?

$$[x] = \{x' \in I \mid x \sim x'\} = \{x' \in I \mid \delta(x') = \delta(x) \wedge \omega(x') = \omega(x)\}$$

$$\delta(x) \in \{1, \ldots, n\}^n,\ \omega(x) \in \{1, \ldots, \ell\}^n$$

$$\{(\delta(x), \omega(x)) \mid x \in I\} \subseteq \{1, \ldots, n\}^n \times \{1, \ldots, \ell\}^n$$

$$|\{(\delta(x), \omega(x)) \mid x \in I\}| \leq |\{1, \ldots, n\}^n \times \{1, \ldots, \ell\}^n\}| = n^n \cdot \ell^n$$

$|I/_\sim| = |\{(\delta(x), \omega(x)) \mid x \in I\}| \leq n^n \cdot \ell^n$ is finite !

Let $S_Q/\sim = \{[s_1], \ldots, [s_n]\}$ and $O = \{y_1, \ldots, y_\ell\}$

$$\delta_i(x) = j \wedge \omega_i(x) = k$$

$$[x]/y_k$$

$$[s_i] \longrightarrow [s_j]$$

Model Map

Sig_1: deterministic RIOSTS with $|O|$, $|M|$ are finite.
Sig_2: input completed, minimal and deterministic FSMs.
$\mathcal{I}$: any refinement of $I/\sim$.

$$
\begin{array}{rcl}
T : Sig_1 & \rightarrow & Sig_2 \\
(S, s_0, R) & \mapsto & (S_Q/\sim, [s_0], \mathcal{I}, O, h)
\end{array}
$$

$$[s_1] \xrightarrow{[x]/y} [s_2] \in h$$

$$\Leftrightarrow \exists s \in [s_1], s' \in [s_2], x_1 \in [x]_{\mathcal{I}} : s \xrightarrow{x_1/y} s' \in R_Q$$

Satisfaction Condition (1)

$$(T, T^*) : \mathcal{F}(Sig_1) \times \mathbf{tc}(Sig_2) \not\to \mathcal{F}(Sig_2) \times \mathbf{tc}(Sig_1)$$

$$\mathcal{F}(S, \leq_1, \mathcal{D}_1) \xrightarrow{T} \mathcal{F}(T(S), \leq_2, \mathcal{D}_2)$$

$$\mathcal{D}_1 = \{\mathcal{S}' \in Sig_1 \mid \mathcal{I} \text{ is a refinement of } I/\sim'\}$$

$$\begin{array}{ccc} \mathcal{S} & \longrightarrow & T(\mathcal{S}) \\ \Big| & & \Big| \\ \leq_1 & & \leq_2 \\ \Big| & & \Big| \\ S' & \longrightarrow & T(S') \end{array}$$

stm CSM_ON

m=3

WARNING

+ entry / d = WARNING;

[v > v_m + dV_IL1(v_m)]

m=4

INTERVENTION_LEVEL_1

+ entry / d = INTERVENTION;
+ entry / b = EMER_BRAKE_CMD;

[v <= v_m]

[v > v_m + dV_WRN(v_m)]

[v <= v_m]

[v > v_m + dV_IL2(v_m)]

m=1

NORMAL

+ entry / d = NORMAL;
+ entry / b = NO_CMD;

Initial

[v > v_m]

[v <= v_m]

[(v <= v_m and a) or v == 0]

m=2

OVERSPEED

+ entry / d = OVERSPEED;

m=5

INTERVENTION_LEVEL_2

- $x = (v, v_m, a) \in I,\ I = [0, 350] \times [0, 350] \times \{0, 1\}$
- $m \in \{1, 2, 3, 4, 5\}$
- $y = (d, b) \in O,\ O = \{0, 2, 3, 4\} \times \{0, 1\}$

$$y_1 = (d = 0, b = 0), y_2 = (2, 0),$$
$$y_3 = (3, 0), y_4 = y_5 = (4, 1)$$

$$A_i = \{s \in S \mid s(\alpha_i) \wedge s(m) = m_i \wedge s(y) = y_i\}$$

$$
\begin{aligned}
\Phi_1 &\equiv g_{1,1} \wedge g_{2,1} \wedge g_{3,1} \wedge g_{4,1} \wedge g_{5,5} \\
&\equiv 0 < v \leq v_m \wedge a = 0 \\
\Phi_2 &\equiv g_{1,1} \wedge g_{2,1} \wedge g_{3,1} \wedge g_{4,1} \wedge g_{5,1} \\
&\equiv v = 0 \vee (v \leq v_m \wedge a = 1) \\
\Phi_3 &\equiv g_{1,2} \wedge g_{2,2} \wedge g_{3,3} \wedge g_{4,4} \wedge g_{5,5} \\
&\equiv v_m < v \leq v_m + \mathrm{dV}_{\mathrm{WRN}}(v_m) \\
\Phi_4 &\equiv g_{1,3} \wedge g_{2,3} \wedge g_{3,3} \wedge g_{4,4} \wedge g_{5,5} \\
&\equiv v_m + \mathrm{dV}_{\mathrm{WRN}}(v_m) < v \leq v_m + \mathrm{dV}_{\mathrm{IL1}}(v_m) \\
\Phi_5 &\equiv g_{1,4} \wedge g_{2,4} \wedge g_{3,4} \wedge g_{4,4} \wedge g_{5,5} \\
&\equiv v_m + \mathrm{dV}_{\mathrm{IL1}}(v_m) < v \leq v_m + \mathrm{dV}_{\mathrm{IL2}}(v_m) \\
\Phi_6 &\equiv g_{1,5} \wedge g_{2,5} \wedge g_{3,5} \wedge g_{4,5} \wedge g_{5,5} \\
&\equiv v_m + \mathrm{dV}_{\mathrm{IL2}}(v_m) < v
\end{aligned}
$$

Define

$$X_i = \{(v, v_m, a) \in I \mid (v, v_m, a) \models \Phi_i\} \quad i = 1, \ldots, 6$$

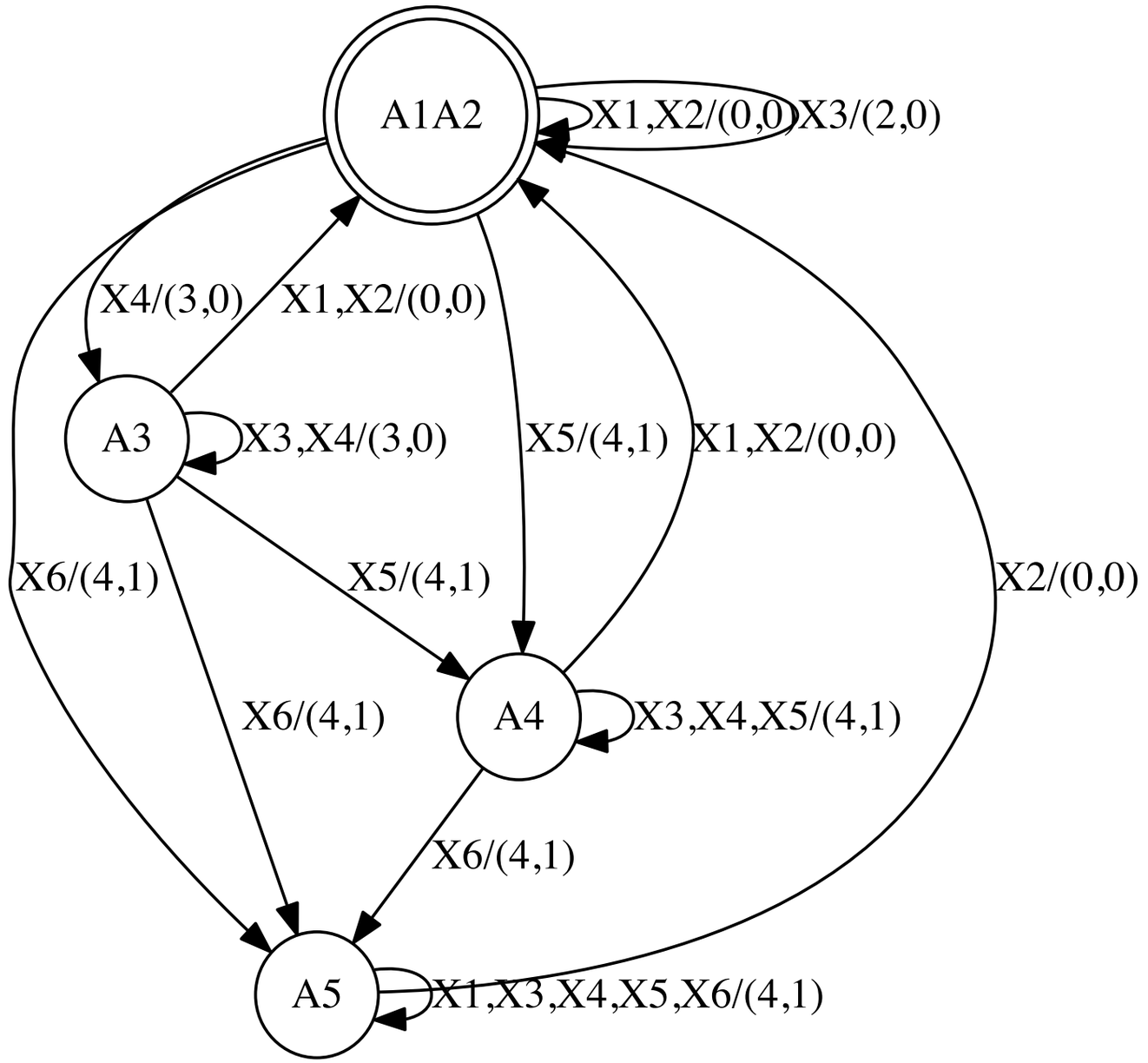
$\mathcal{I} = \{X_1, X_2, \ldots, X_6\}$. Then $\mathcal{I}$ is an IECP for the CSM

Table 1: Input Representatives of X_i.

$\mathbf{c}_i$	v	v_m	a	X_i
$\mathbf{c}_1$	60	90	0	X_1
$\mathbf{c}_2$	60	90	1	X_2
$\mathbf{c}_3$	152	150	0	X_3
$\mathbf{c}_4$	125	120	1	X_4
$\mathbf{c}_5$	66	60	0	X_5
$\mathbf{c}_6$	260	230	0	X_6

Table 1: DFSM Transition Table.

Source	**Input**	**Target**	**d**	**b**
A_1	$X_1 \cup X_2$	A_1	0	0
A_1	X_3	A_2	2	0
A_1	X_4	A_3	3	0
A_1	X_5	A_4	4	1
A_1	X_6	A_5	4	1
A_2	$X_1 \cup X_2$	A_1	0	0
A_2	X_3	A_2	2	0
A_2	X_4	A_3	3	0
A_2	X_5	A_4	4	1
A_2	X_6	A_5	4	1
A_3	$X_1 \cup X_2$	A_1	0	0
A_3	$X_3 \cup X_4$	A_3	3	0
A_3	X_5	A_4	4	1
A_3	X_6	A_5	4	1
A_4	$X_1 \cup X_2$	A_1	0	0
A_4	$X_3 \cup X_4 \cup X_5$	A_4	4	1
A_4	X_6	A_5	4	1
A_5	X_2	A_1	0	0
A_5	$\bigcup_{i \in \{1,3,4,5,6\}} X_i$	A_5	4	1



$$L(S) = \{\pi \in \Sigma^* \mid [\pi] \in L(T(S))\}$$

Test Case Map

$$T^* : \mathbf{tc} \text{ of } Sig_2 \rightarrow \mathbf{tc} \text{ of } Sig_1$$

- A test case for deterministic FSM
  is a finite input/output sequence.

- A test case for nondeterministic FSM
  is considered as a FSM,
  —"preset" or "adaptive" test case.

- A test case for deterministic RIOSTS
  is a finite input/output sequence.

- A test case for nondeterministic RIOSTS
  is considered as an RIOSTS
  with input alphabet O and output alphabet I.

Test Case Map

$$T^* : \mathbf{tc} \text{ of } Sig_2 \rightarrow \mathbf{tc} \text{ of } Sig_1$$

$$\begin{array}{rcl} (S, s_0, I, O, R) & \xrightarrow{T} & (S_Q/\sim, [s_0], \mathcal{I}, O, h) \\ X_1/y_1 \ldots X_K/y_k & \xrightarrow{T^*} & x_1/y_1 \ldots x_k/y_k \end{array}$$

$\mathcal{I}$ is a refinement of $I/\sim$ and $[x_i]_{\mathcal{I}} = X_i, i = 1, \ldots, k$

Satisfaction Condition (2)

$$(T, T^*) : \mathcal{F}(Sig_1) \times \mathbf{tc}(Sig_2) \quad \not\to \quad \mathcal{F}(Sig_2) \times \mathbf{tc}(Sig_1)$$

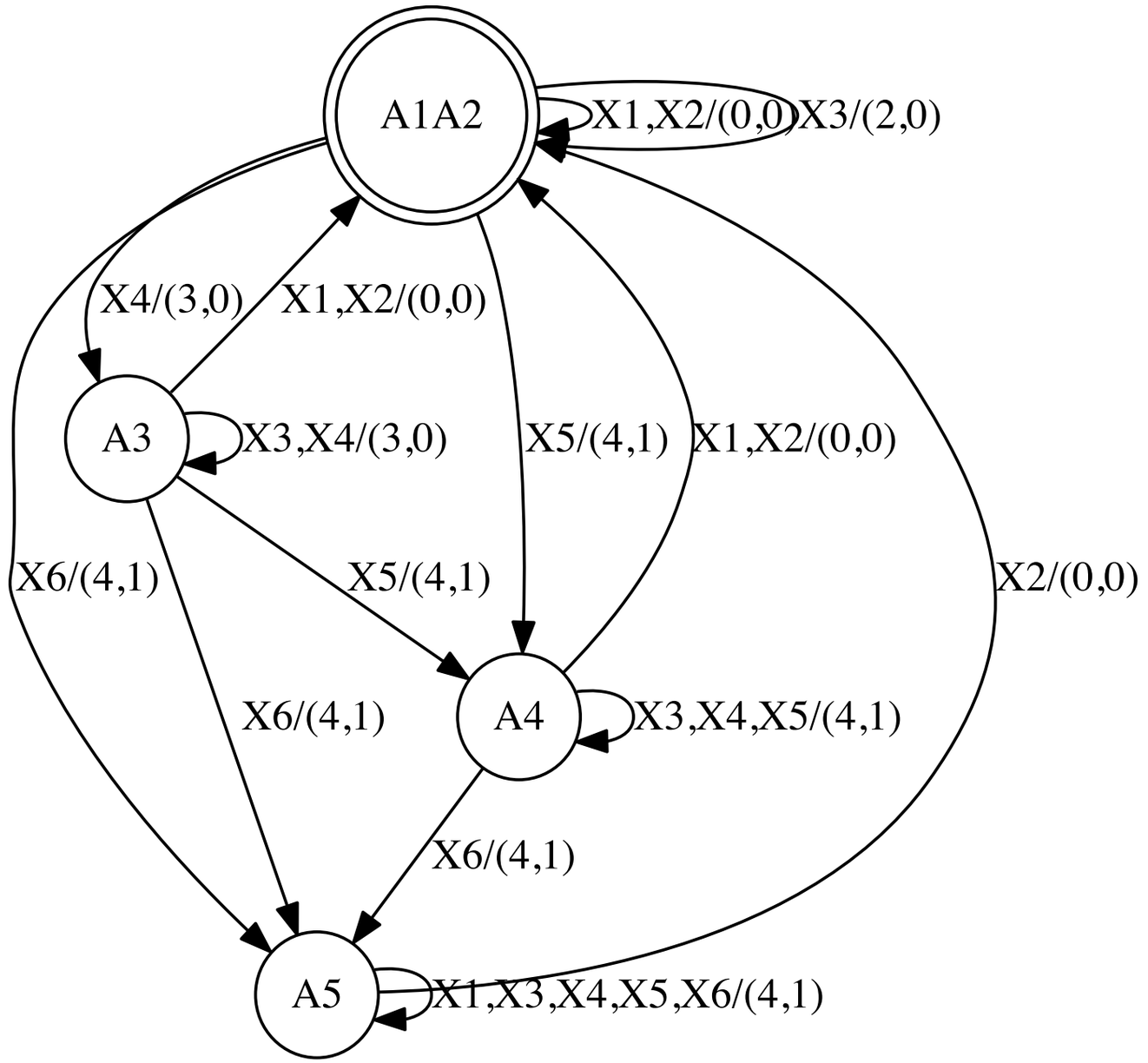
$$\mathcal{F}(S, \leq_1, \mathcal{D}_1) \quad \xrightarrow{T} \quad \mathcal{F}(T(S), \leq_2, \mathcal{D}_2)$$

- $\forall S' \in \mathcal{D}_1, U \in \mathbf{tc}(\mathcal{I}, O) : T^*(U) \in \mathbf{tc}(I, O)$
- $T(S') \; \underline{\mathbf{pass}}_2 \; U \Leftrightarrow S' \; \underline{\mathbf{pass}}_1 \; T^*(U)$

$$\begin{array}{ccc} S' & \longrightarrow & T(S') \\ \Big| & & \Big| \\ \underline{\mathbf{pass}}_1 & & \underline{\mathbf{pass}}_2 \\ \Big| & & \Big| \\ T^*(\mathcal{U}) & \longleftarrow & \mathcal{U} \end{array}$$

Application of Theorem 2

1. Calculate **input equivalence classes**
2. **Map speed monitor model to FSM** with
    1. input equivalence classes as input alphabet
    2. original discrete outputs as output alphabet
3. Use W-Method or similar method to **create complete FSM test suite**
4. **Translate FSM test suite** to concrete test suite for speed monitor


A1A2
X1,X2/(0,0)
X3/(2,0)
X4/(3,0)
X1,X2/(0,0)
A3
X3,X4/(3,0)
X5/(4,1)
X1,X2/(0,0)
X6/(4,1)
X5/(4,1)
X2/(0,0)
X6/(4,1)
A4
X3,X4,X5/(4,1)
X6/(4,1)
A5
X1,X3,X4,X5,X6/(4,1)

Symbolic test cases resulting from W-Method

1. X4.X1.X3/(3,0).(0,0).(2,0)
2. X4.X1.X1/(3,0).(0,0).(0,0)
3. X4.X2.X3/(3,0).(0,0).(2,0)
4. X4.X2.X1/(3,0).(0,0).(0,0)
5. X4.X3.X3/(3,0).(3,0).(3,0)
6. X4.X3.X1/(3,0).(3,0).(0,0)
7. X4.X4.X3/(3,0).(3,0).(3,0)
8. X4.X4.X1/(3,0).(3,0).(0,0)
9. X4.X5.X3/(3,0).(4,1).(4,1)
10. X4.X5.X1/(3,0).(4,1).(0,0)
11. X4.X6.X3/(3,0).(4,1).(4,1)
12. X4.X6.X1/(3,0).(4,1).(4,1)
13. X5.X1.X3/(4,1).(0,0).(2,0)
14. X5.X1.X1/(4,1).(0,0).(0,0)
15. X5.X2.X3/(4,1).(0,0).(2,0)
16. X5.X2.X1/(4,1).(0,0).(0,0)
17. X5.X3.X3/(4,1).(4,1).(4,1)
18. X5.X3.X1/(4,1).(4,1).(0,0)
19. X5.X4.X3/(4,1).(4,1).(4,1)
20. X5.X4.X1/(4,1).(4,1).(0,0)
21. X5.X5.X3/(4,1).(4,1).(4,1)
22. X5.X5.X1/(4,1).(4,1).(0,0)
23. X5.X6.X3/(4,1).(4,1).(4,1)
24. X5.X6.X1/(4,1).(4,1).(4,1)
25. X6.X1.X3/(4,1).(4,1).(4,1)
26. X6.X1.X1/(4,1).(4,1).(4,1)
27. X6.X2.X3/(4,1).(0,0).(2,0)
28. X6.X2.X1/(4,1).(0,0).(0,0)
29. X6.X3.X3/(4,1).(4,1).(4,1)
30. X6.X3.X1/(4,1).(4,1).(4,1)
31. X6.X4.X3/(4,1).(4,1).(4,1)
32. X6.X4.X1/(4,1).(4,1).(4,1)
33. X6.X5.X3/(4,1).(4,1).(4,1)
34. X6.X5.X1/(4,1).(4,1).(4,1)
35. X6.X6.X3/(4,1).(4,1).(4,1)
36. X6.X6.X1/(4,1).(4,1).(4,1)
37. X1.X3/(0,0).(2,0)
38. X1.X1/(0,0).(0,0)
39. X2.X3/(0,0).(2,0)
40. X2.X1/(0,0).(0,0)
41. X3.X3/(2,0).(2,0)
42. X3.X1/(2,0).(0,0)

Symbolic test cases resulting from W-Method

Symbolic means that concrete test data still has to be selected from each X_i when it is referenced in a test case

This can be done automatically using a mathematical constraint solver **(SMT-solver)**

1. X4.X[illegible]
2. X4.X1.[illegible]
3. X4.X2.[illegible]
4. X4.X2.[illegible]
5. X4.X3.[illegible]
6. X4.X3.[illegible]
7. X4.X4.[illegible]
8. X4.X4.[illegible]
9. X4.X5.[illegible]
10. X4.X5[illegible]
11. X4.X6[illegible]
12. X4.X6[illegible]
13. X5.X1[illegible]
14. X5.X1.X1/(4,1).(0,0).(0,0)
15. X5.X2.X3/(4,1).(0,0).(2,0)
16. X5.X2.X1/(4,1).(0,0).(0,0)
17. X5.X3.X3/(4,1).(4,1).(4,1)
18. X5.X3.X1/(4,1).(4,1).(0,0)
19. X5.X4.X3/(4,1).(4,1).(4,1)
20. X5.X4.X1/(4,1).(4,1).(0,0)

[illegible].X5.X5.X3/(4,1).(4,1).(4,1)
[illegible].(4,1).(4,1).(0,0)
[illegible].(4,1).(4,1).(4,1)
[illegible].(4,1).(4,1).(4,1)
[illegible].(4,1).(4,1).(4,1)
[illegible].(4,1).(4,1).(4,1)
[illegible].(4,1).(0,0).(2,0)
[illegible].(4,1).(0,0).(0,0)
[illegible].(4,1).(4,1).(4,1)
[illegible].(4,1).(4,1).(4,1)
[illegible].(4,1).(4,1).(4,1)
[illegible].(4,1).(4,1).(4,1)
34. X6.X3.X1/(4,1).(4,1).(4,1)
35. X6.X6.X3/(4,1).(4,1).(4,1)
36. X6.X6.X1/(4,1).(4,1).(4,1)
37. X1.X3/(0,0).(2,0)
38. X1.X1/(0,0).(0,0)
39. X2.X3/(0,0).(2,0)
40. X2.X1/(0,0).(0,0)
41. X3.X3/(2,0).(2,0)
42. X3.X1/(2,0).(0,0)

Combination With Random and Boundary Value Testing

- Instead of always using the same representative of each input class representative, **select a random value of this class**, whenever it is used in the test case – combine this technique with **boundary value tests**

- Completeness is still guaranteed for SUTs inside the fault domain

- **For SUTs outside the fault domain, the test strength is significantly increased**

Felix Hubner, Wen-ling Huang, and Jan Peleska:
Experimental Evaluation of a Novel Equivalence Class Partition Testing Strategy. In Blanchette and Kosmatov (eds.): Proceedings of the TAP 2015, Springer LNCS, Vol. 9154, pp. 155-173, 2015.

Summary of the Benefits

- A new complete testing strategy for systems with infinite input domains and finite internal states and finite outputs

- Effectively implementable in model-based testing tools – fully automated

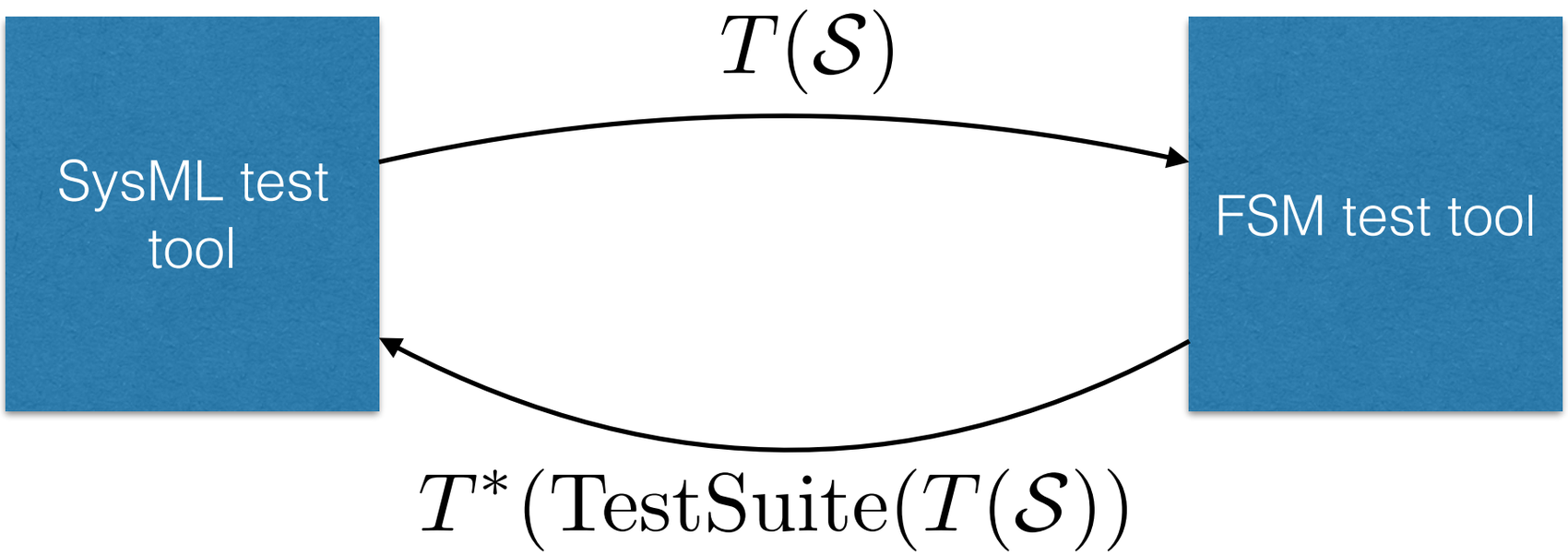
- Significantly **higher test strength** compared to heuristic test strategies

- Significant **reduction of test effort** in application domains where the testing is very costly: railway interlocking systems

Jan Peleska, Wen-ling Huang, and Felix Hubner:
A novel approach to HW/SW integration testing of route-based interlocking system controllers. To appear in proceedings of the RSSR 2016, Springer LNCS, 2016.

Summary of the Benefits

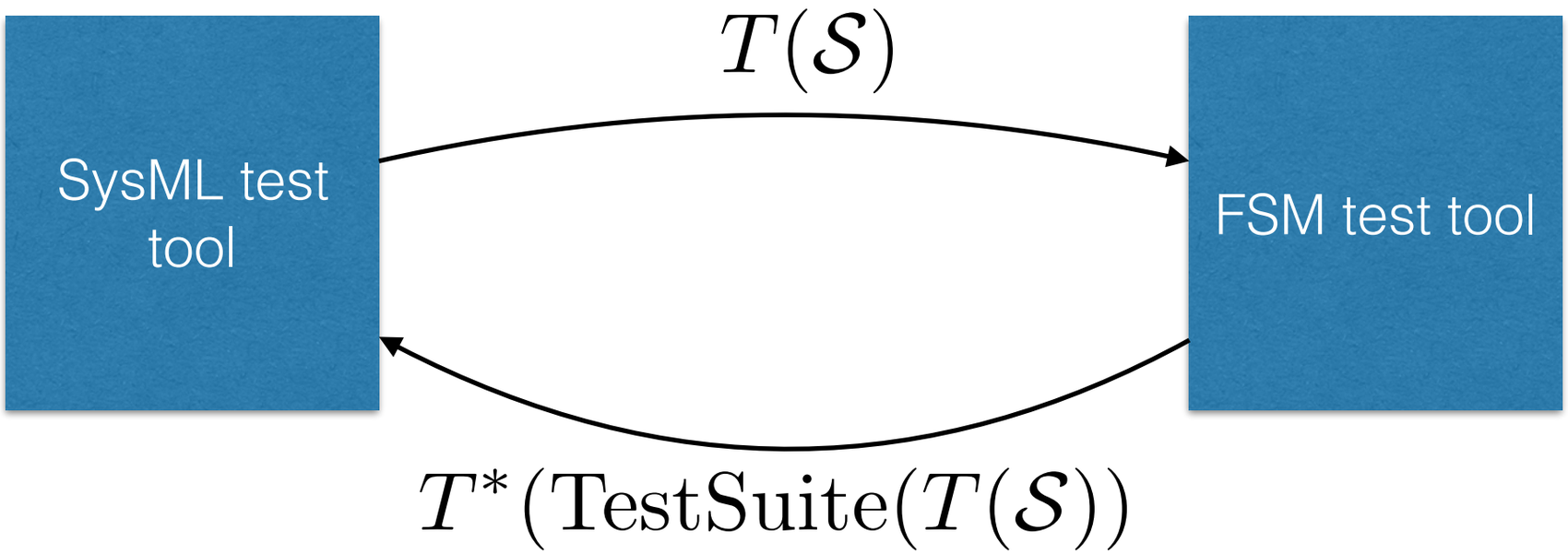
- When building a new tool for model-based testing of SysML state machines (with infinite input domains), the test case generation can be performed by an existing tool implementing these algorithms for FSMs



Summary of the Benefits

- When building a new tool for model-based testing of SysML state machines (with conceptually infinite input domains), the test
performed by an existin
algorithms for FSMs

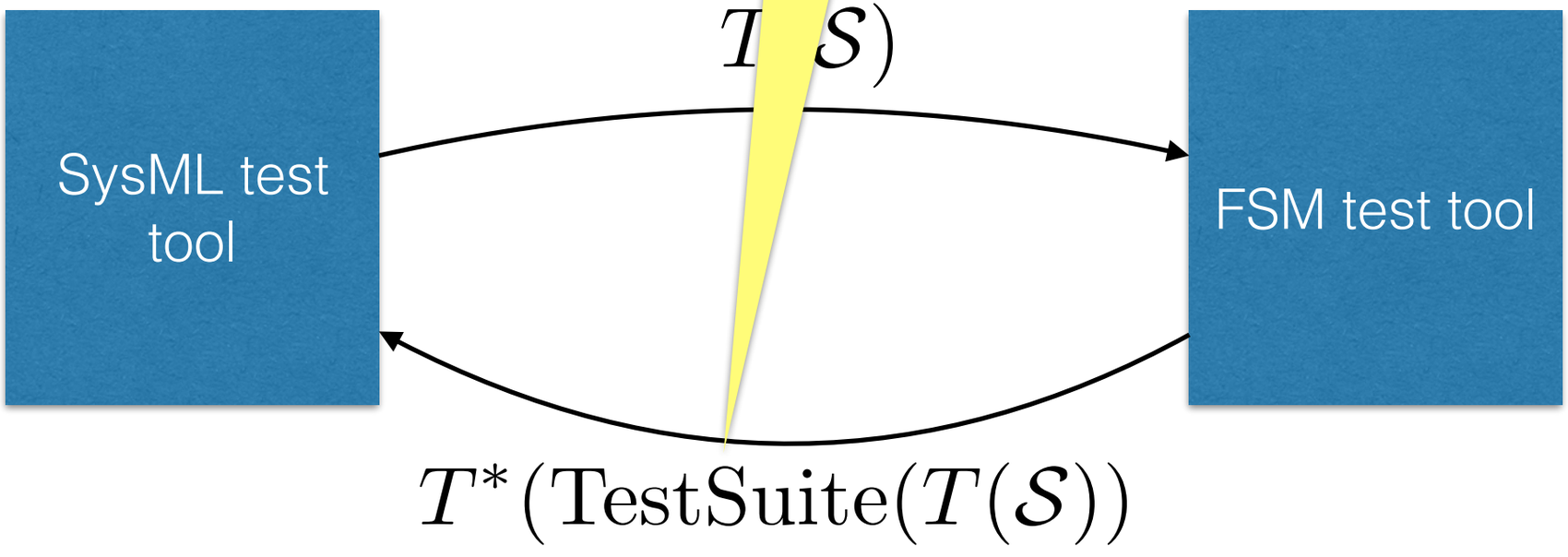
SysML test tool performs FSM abstraction and sends it to FSM test tool



Summary of the Benefits

- When building a new tool for model-based testing of SysML state machines (with conceptually infinite input domains), the test
performed by an existin
algorithms for FSMs

FSM tool generates complete test suite and sends the translated result to SysML test tool



Further Reading

1. Publications of Jan Peleska, Wen-ling Huang, and their co-authors. http://www.informatik.uni-bremen.de/agbs/jp/jp_papers_e.html

2. ERTMS/ETCS SystemRequirements Specification, Chapter 3, Principles, volume Subset-026-3, Issue 3.4.0 (2015), available under http://www.era.europa.eu/Document-Register/Pages/Set-2-System-Requirements-Specification.aspx

3. Nancy Leveson. SafeWare: System Safety and Computers. Addison Wesley 1995.

4. Neil Storey. Safety-critical Computer Systems. Addison-Welly, 1996.