Testing Safety-critical Discrete-State Systems – Mathematical Foundations and Concrete Algorithms

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

Background

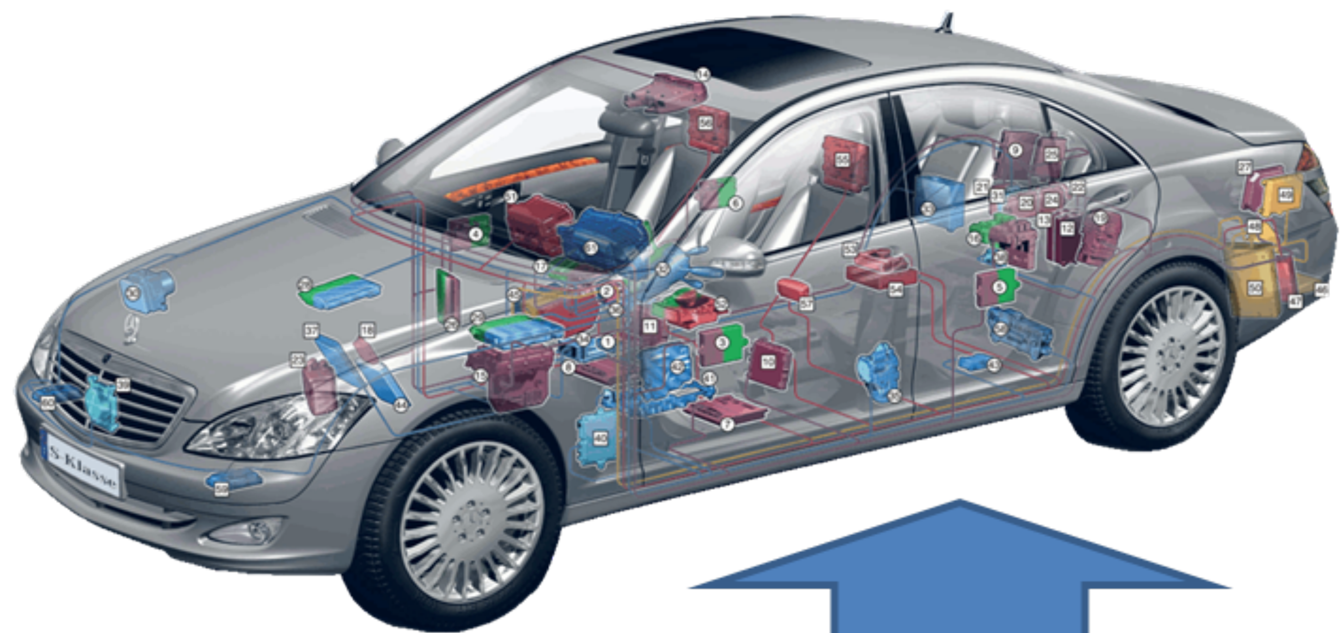
- **My research group at the University of Bremen**
  - Specialised on verification of safety-critical control systems
  - Focus on test automation
  - Collaboration with Prof. Dr. habil. Wen-ling Huang in this field

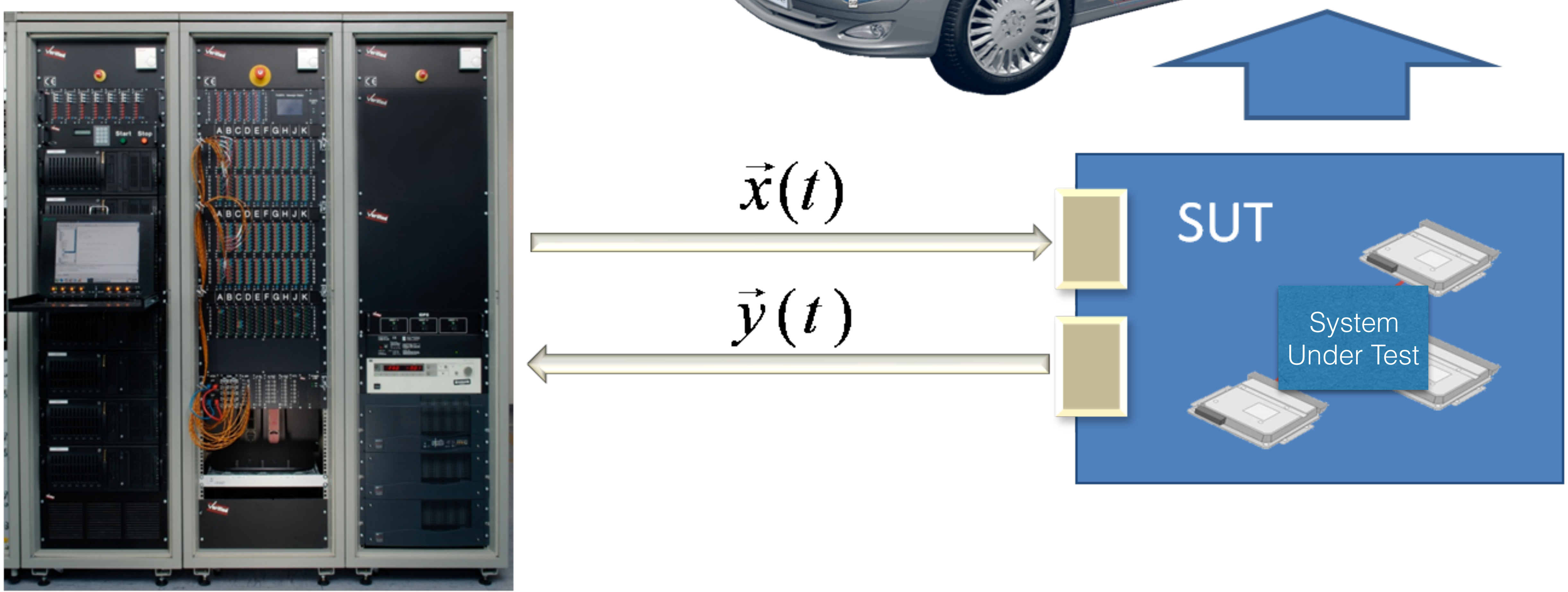
Background

- **Verified Systems International GmbH**
  - Founded 1998 as a spinoff company from the University of Bremen
  - Specialised on verification and validation of safety-critical systems – aerospace, railways, automotive
  - Tool development, hardware-in-the-loop test bench development, and service provision
  - Main customers Airbus, Siemens
  - 25 employees

Hardware-in-the-Loop Test Benches

For testing integrated HW/SW systems





An Observation . . .

- Many maths and computer science students in their first semesters seem to think that complex theory and difficult algorithms are only needed to pass exams . . .

- This is not true!

- Many of the most important innovations and products are based on highly complex mathematical foundations and on sophisticated software algorithms

Safety-critical systems – Examples

Safety-critical Systems

A **. . . safety-critical system** is a system whose failure or malfunction may result in one (or more) of the following outcomes:

- death or serious injury to people
- loss or severe damage to equipment/property
- environmental harm

Definition taken from https://en.wikipedia.org/wiki/Life-critical_system

Safety-critical Systems – Examples

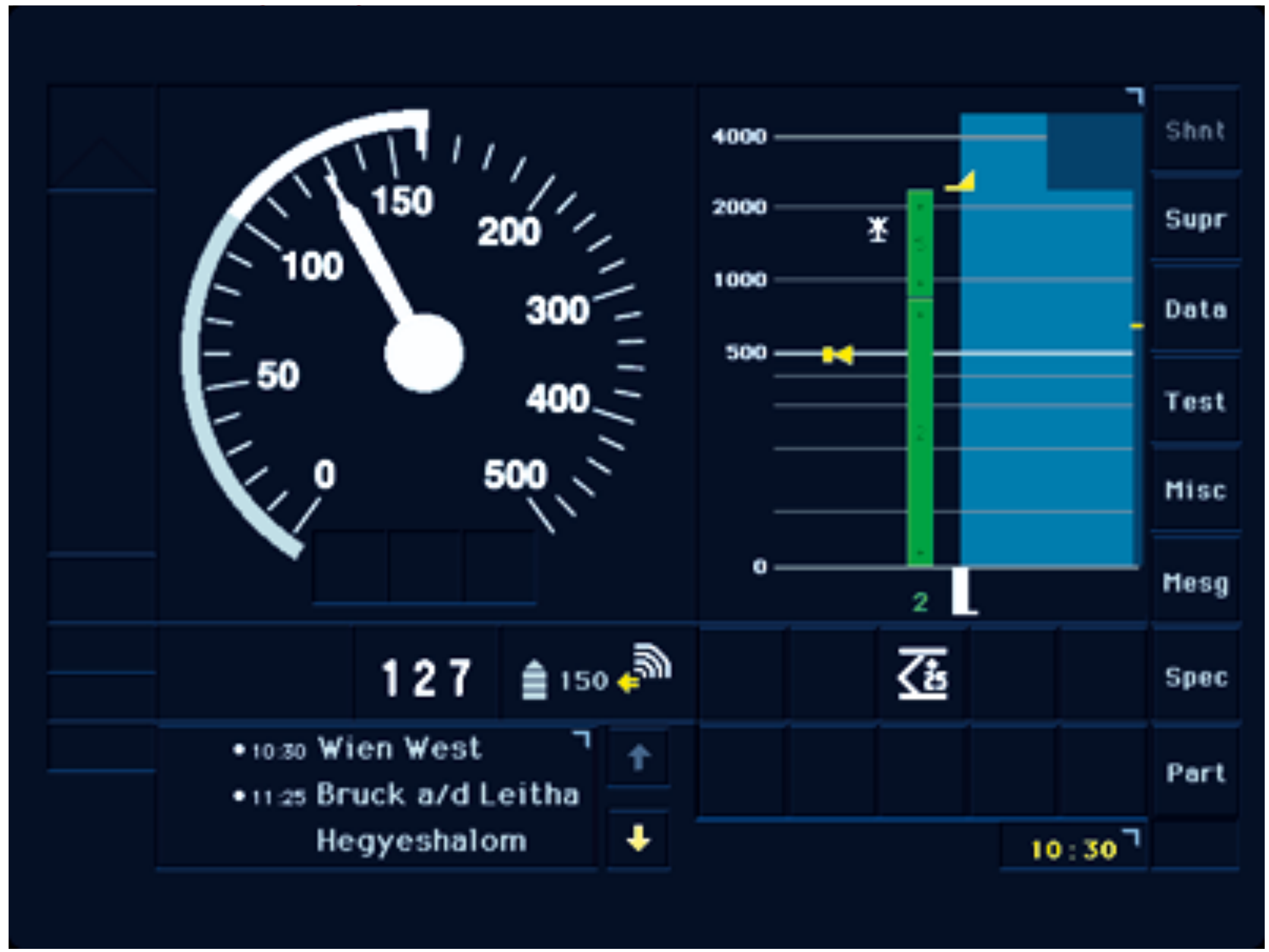
Airbag controller

Safety-critical Systems – Examples

Aircraft thrust reversal

Safety-critical Systems – Examples

Train speed supervision



Verification Requirements of International Standards

Verification Requirements of International Standards

- Safety-critical systems development and verification is controlled by laws
- These laws state that development and verification must follow the rules specified in applicable standards, such as
  - Avionic domain: **RTCA DO-178C**
  - Railway domain: **CENELEC EN50128:2011**
  - Automotive domain: **ISO 26262**

Verification Requirements of International Standards

- Safety-critical systems development and verification is controlled by laws
- These ... verification must ... able standards, such ...

Avionic System. A control computer in an aircraft

  - Avionic domain: **RTCA DO-178C**
  - Railway domain: **CENELEC EN50128:2011**
  - Automotive domain: **ISO 26262**

Verification Requirements of International Standards

- These standards differ in many details, but they contain some basic requirements
  - Development must be based on **requirements**
  - Every piece of software code and every hardware component must be **traced** back to at least one requirement

- For the most critical applications, requirements should be expressed by **formal models** with mathematical interpretation
    - **Syntax and static model semantics**: is the model well-formed?
    - **Behavioural model semantics**: how does the model state, including inputs and outputs, change over time?
- **Requirements (models) must be verified**

- **Code must be verified**
  - Does it implement the related requirements correctly?
  - Verification is preferably performed by **testing** the software integrated in the controller’s hardware
- Verification results need to be checked with respect to completeness and correctness
- Tools automating development or verification steps need to be **qualified**

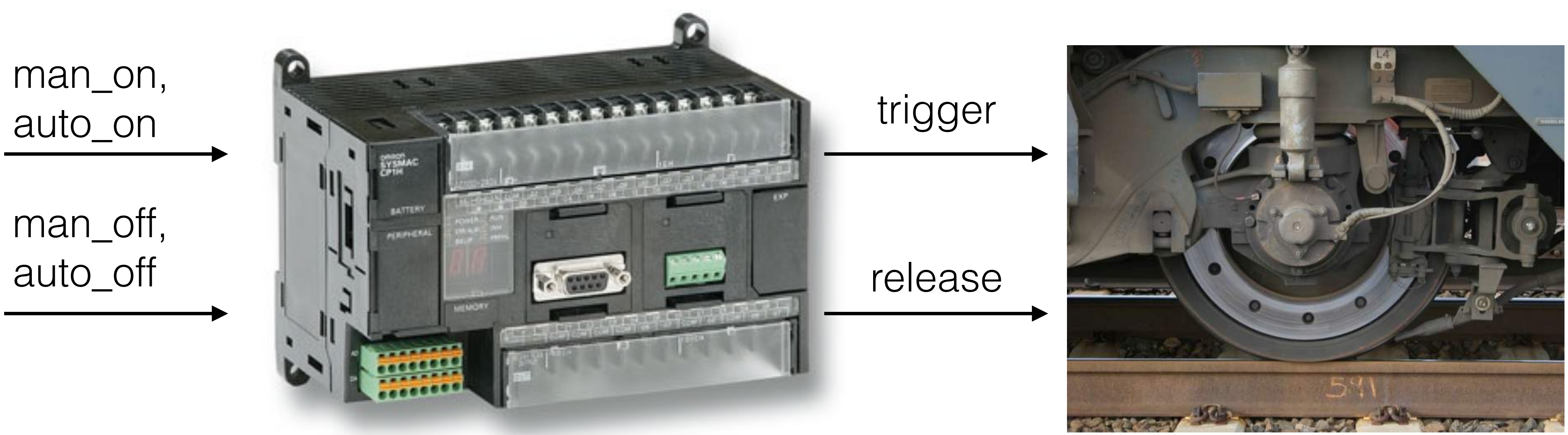
Motivation

- **Safety-critical systems can be developed by starting with a model of the required system behaviour**
- **The developed system can then be tested against the model**
  - **Input data to be exercised on the system under test (SUT) can be derived from the model**
  - **The behaviour of the SUT can be compared to the expected behaviour specified in the model**
  - **https://zh.wikipedia.org/wiki/生命攸關系統#.E9.81.8B.E8.BC.B8**

Motivation

- **Testing theories**
  - **Define methods to generate test cases from models**
    - **Test case. Sequence of input data + expected outputs to produced by the SUT when receiving these inputs**
  - **Specify which correctness properties are fulfilled if all test cases are passed by the SUT**
- **For safety-critical systems, the test strength (= the capability to uncover certain types of errors) of a testing theory must be proven**

- **Example: Model of a brake controller in a train**
  - **Controller receives commands to activate brakes**
    - **From train engine driver: *man_on*, *man_off***
    - **From a speed supervision computer: *auto_on*, *auto_off***
  - **Controller activates brakes by outputs *trigger* and *release***



- **Requirements for the brake controller**

  1. **When brakes are not activated, they can be triggered manually (*man_on*) or by the speed supervision computer (*auto_on*)**

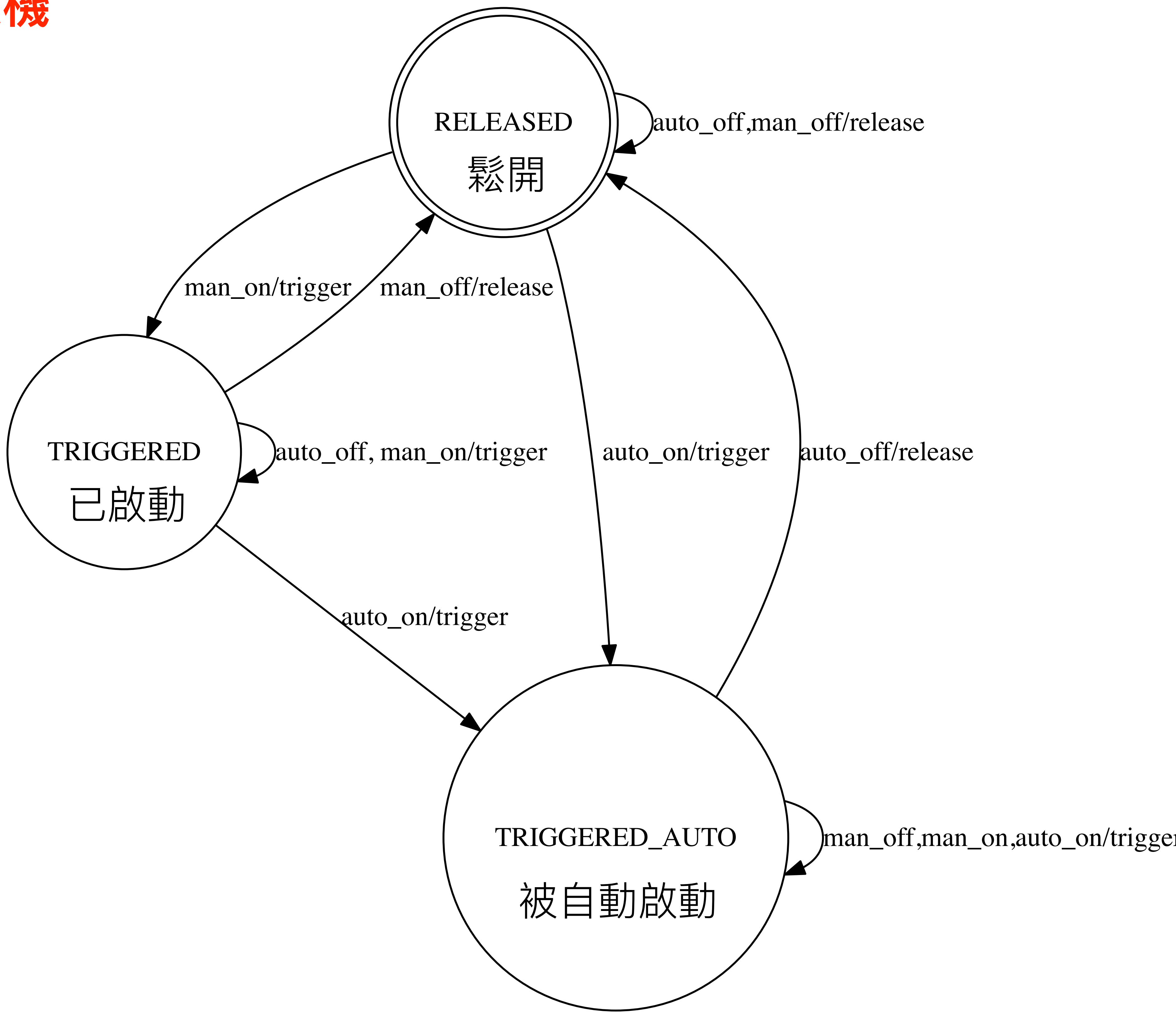
  2. **When brakes have been activated manually (*man_on*), they can only be released manually (*man_off*)**

  3. **When brakes have been activated via speed supervision computer (*auto_on*), they can be only released via command *auto_off* from the computer**

  4. **When the supervision computer sends *auto_on* while already braking, the brakes can only be released by *auto_off***

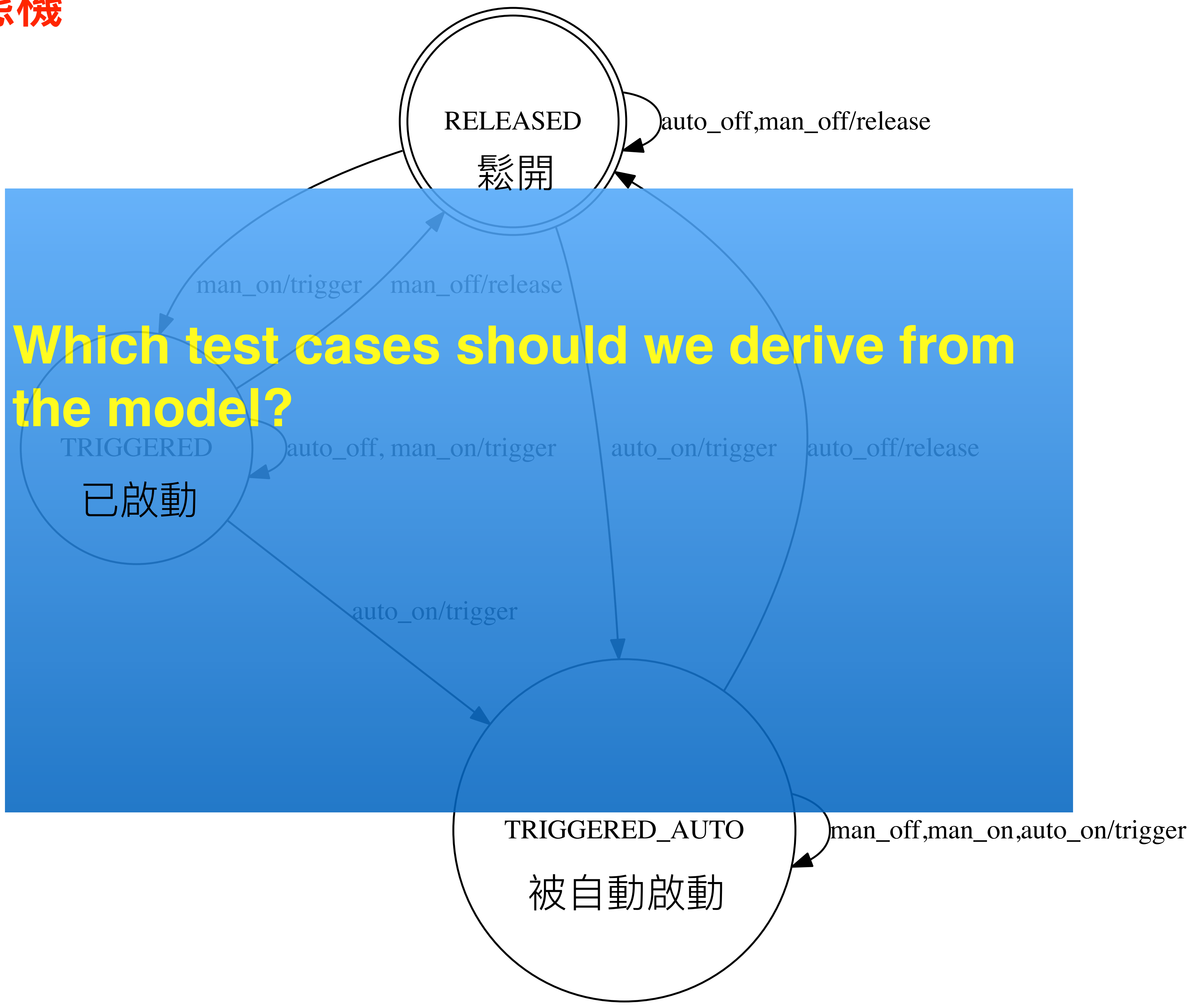
Finite State Machine modelling the behaviour of the brake controller

有限狀態機



Finite State Machine modelling the behaviour of the brake controller

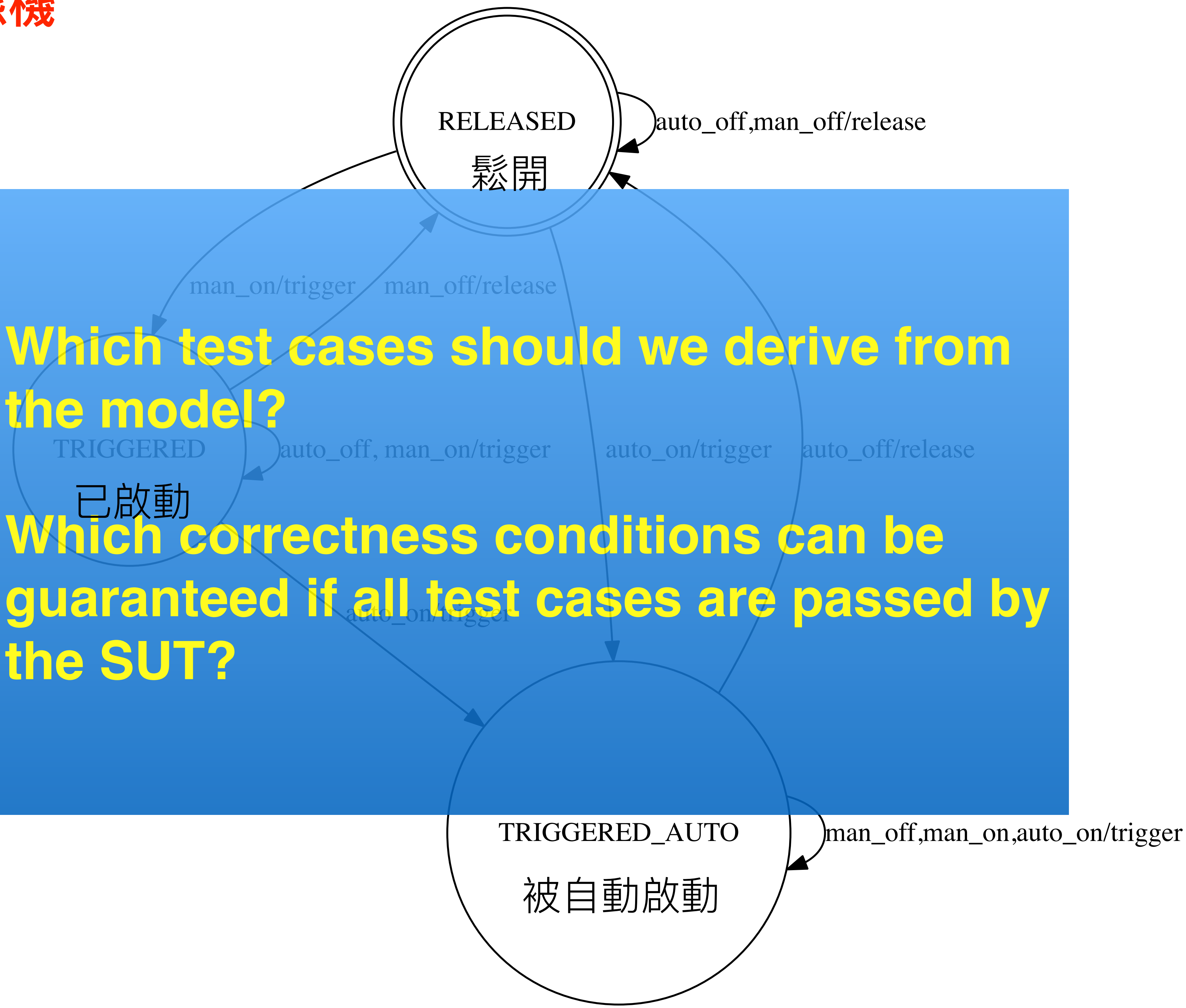
有限狀態機



Which test cases should we derive from the model?

Finite State Machine modelling the behaviour of the brake controller

有限狀態機



Which test cases should we derive from the model?

Which correctness conditions can be guaranteed if all test cases are passed by the SUT?

Overview

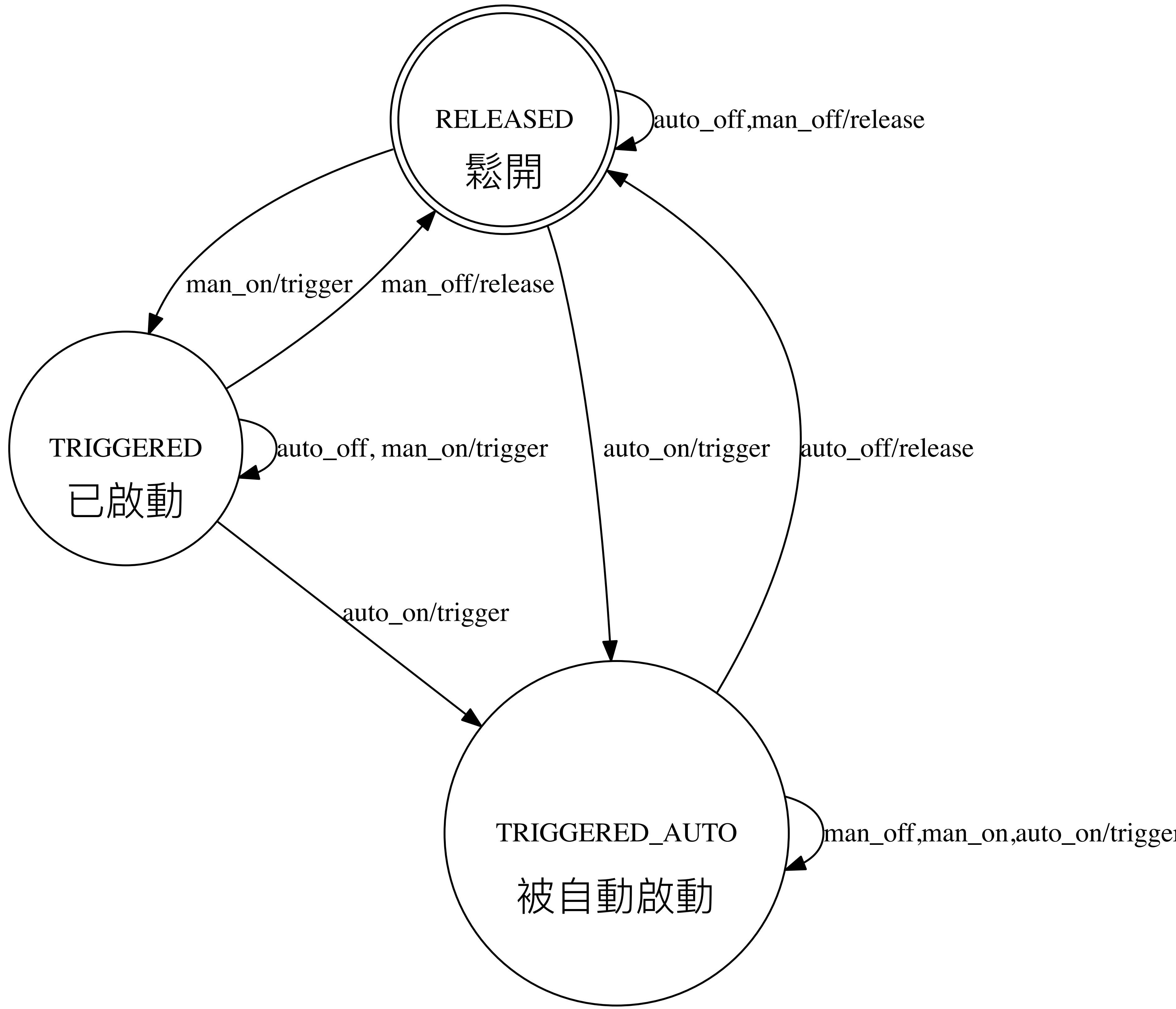
- **Mathematical definition of Finite State Machines**
- **Languages and Conformance Relations**
- **FSM properties: observability and minimality**
- **Fault Models**
- **Testing Theories**
  - **The W-Method for deterministic FSMs**
  - **The Wp-Method for nondeterministic FSMs**

Finite State Machines

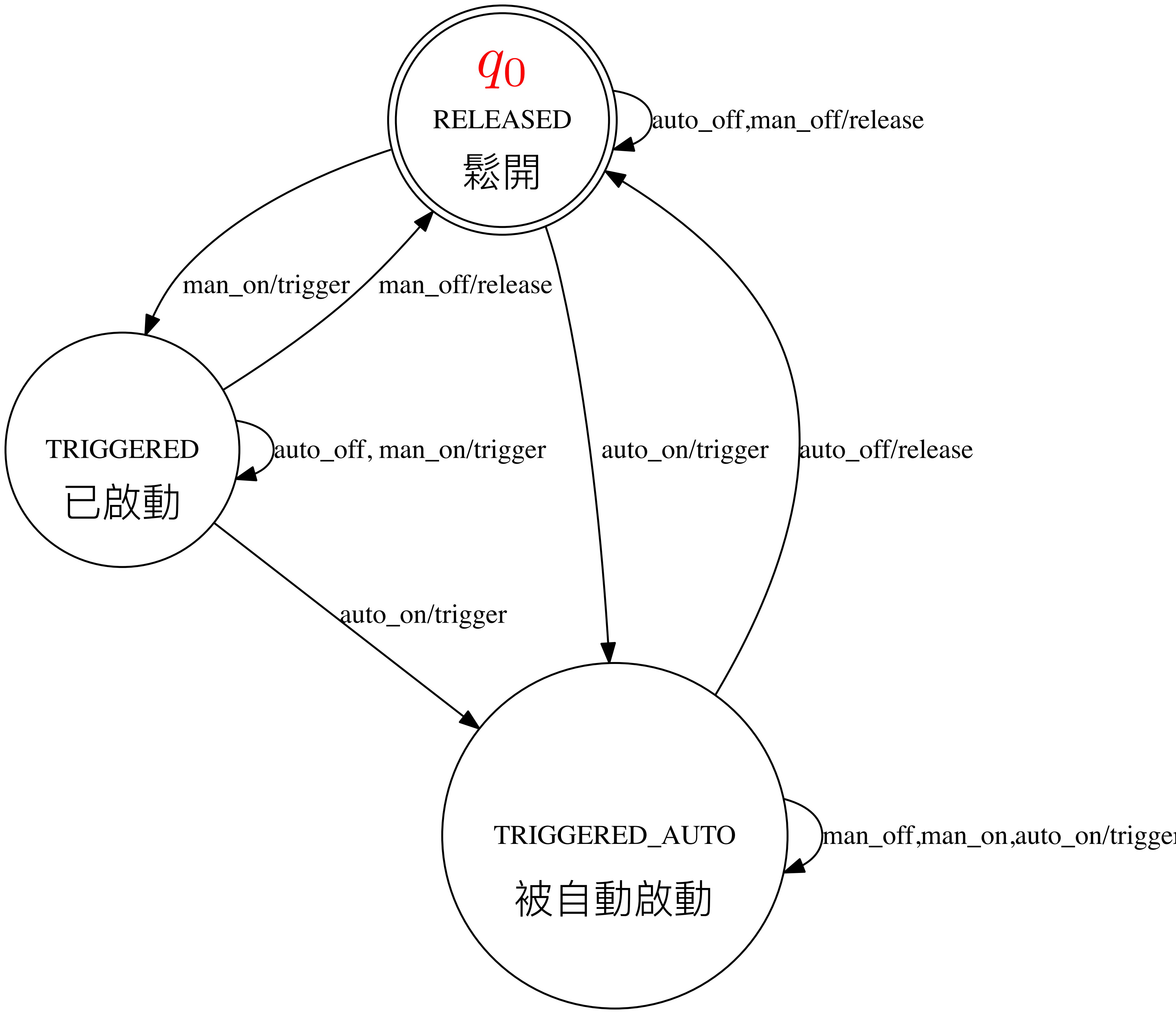
$M = (Q, q_0, I, O, h)$

- $Q \neq \emptyset$: finite set of states 狀態集
- $q_0 \in Q$: initial state 初始狀態
- $I \neq \emptyset$: finite set of input alphabet 輸入字母表
- $O \neq \emptyset$: 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

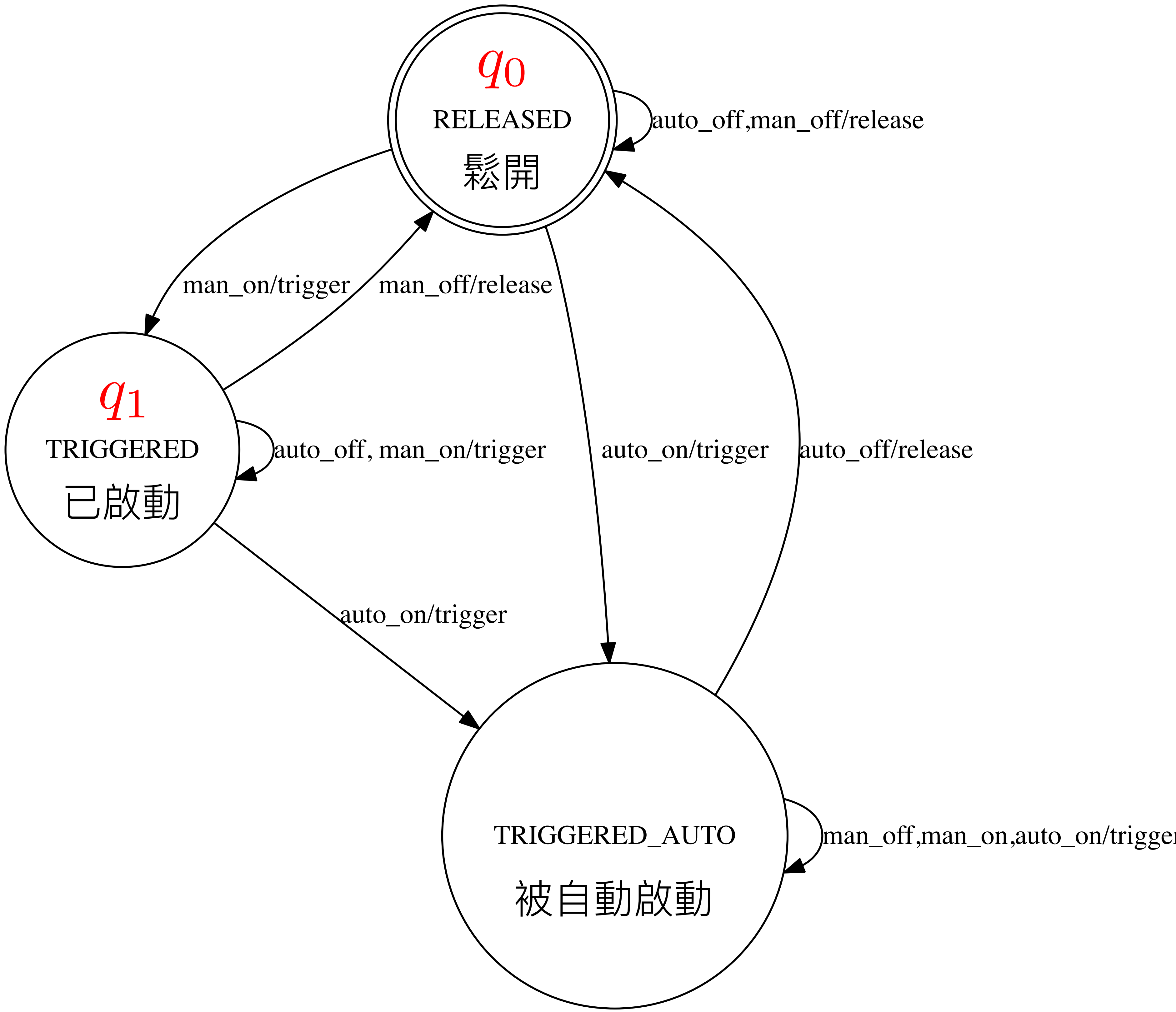
Finite State Machine modelling the behaviour of the brake controller



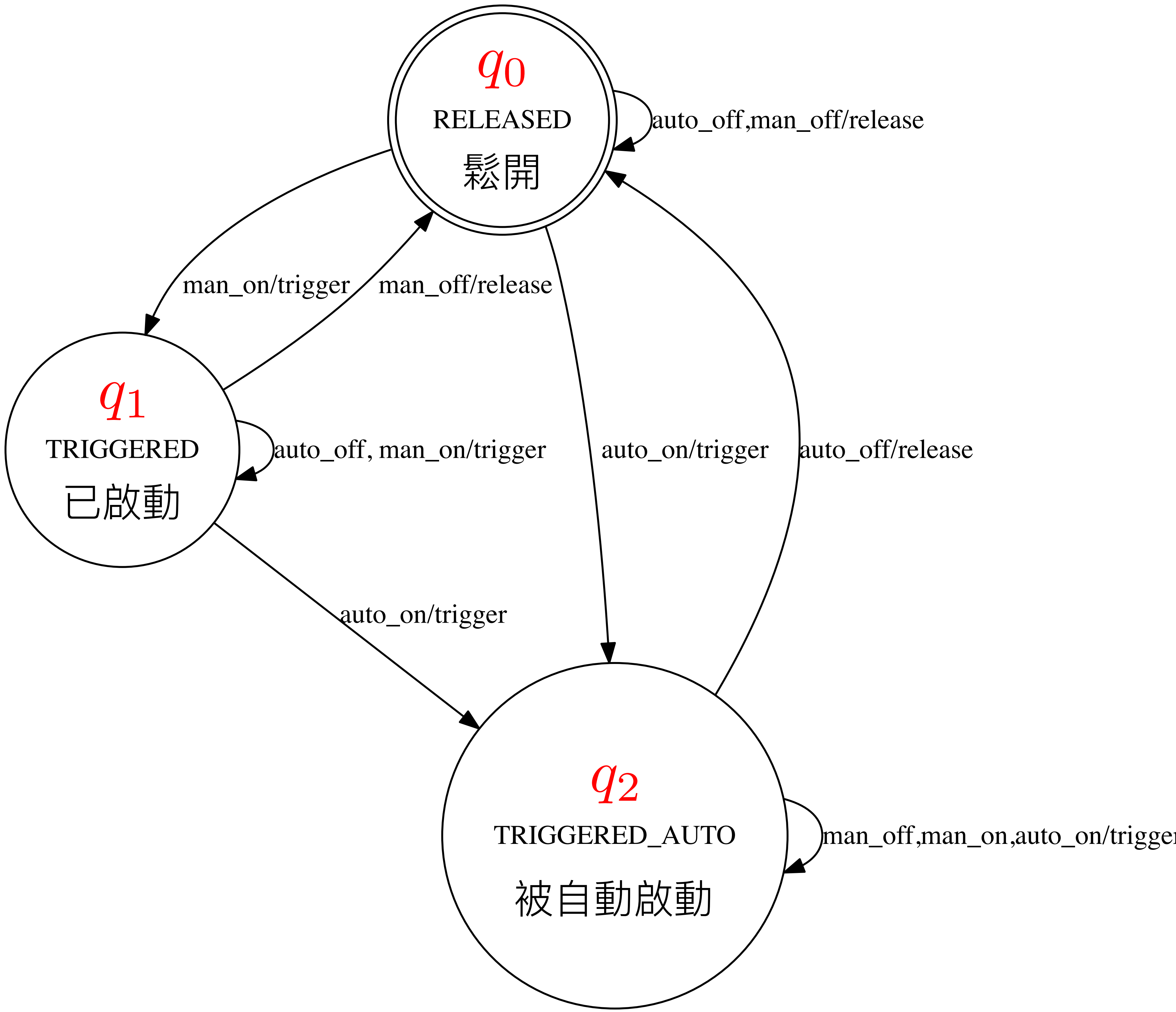
Finite State Machine modelling the behaviour of the brake controller



Finite State Machine modelling the behaviour of the brake controller

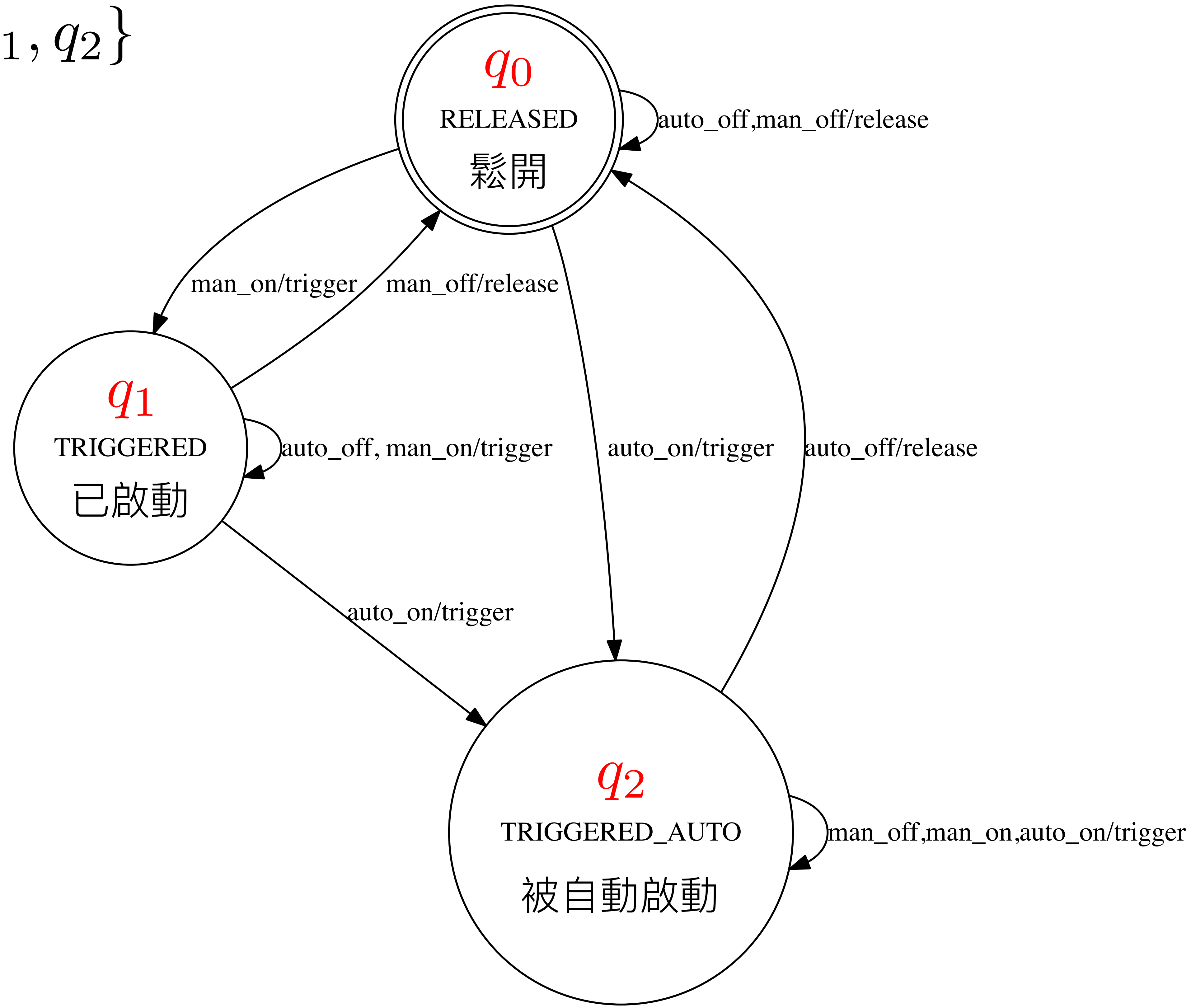


Finite State Machine modelling the behaviour of the brake controller



Finite State Machine modelling the behaviour of the brake controller

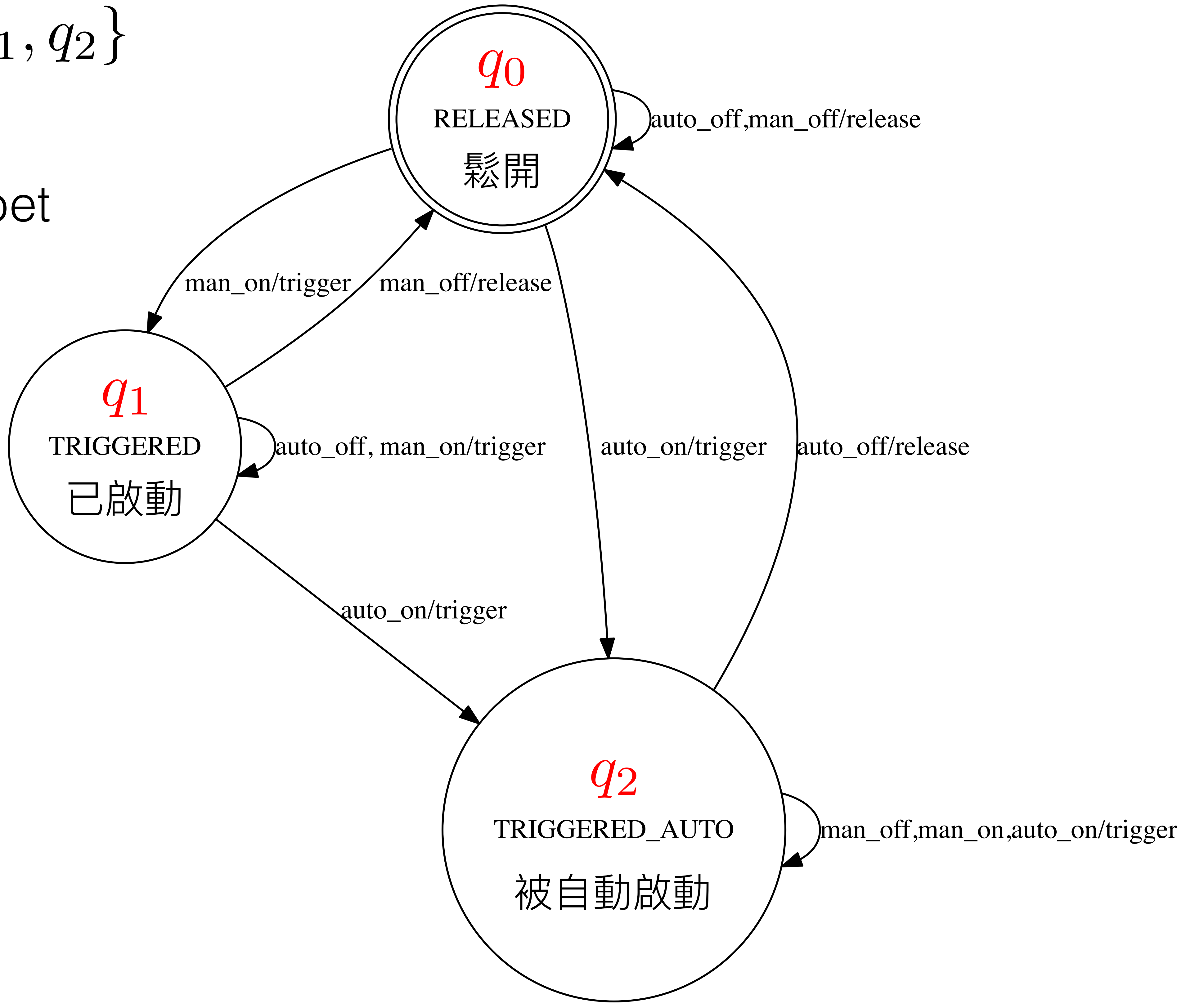
$$Q = \{q_0, q_1, q_2\}$$



Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

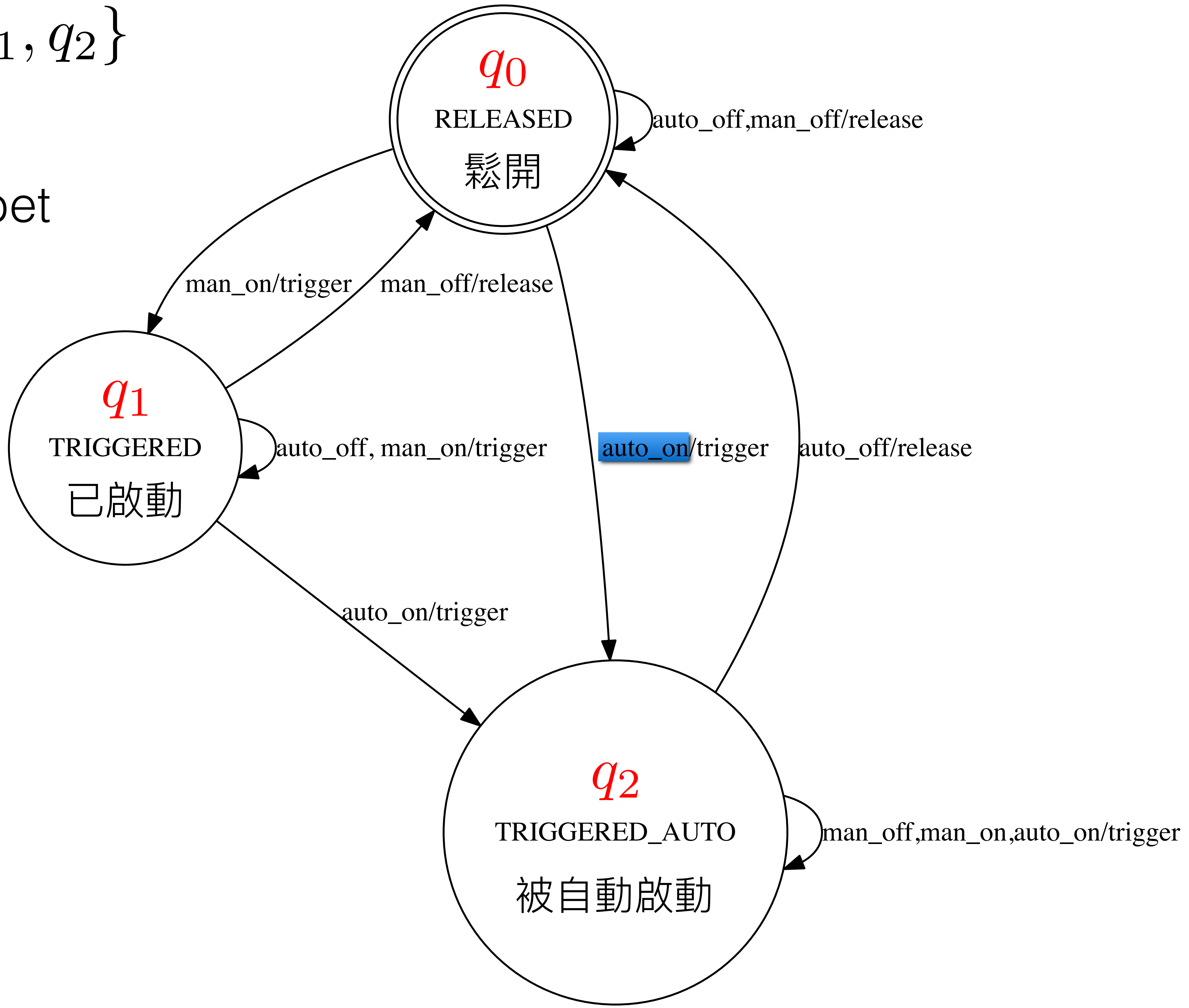
Input Alphabet



Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

Input Alphabet

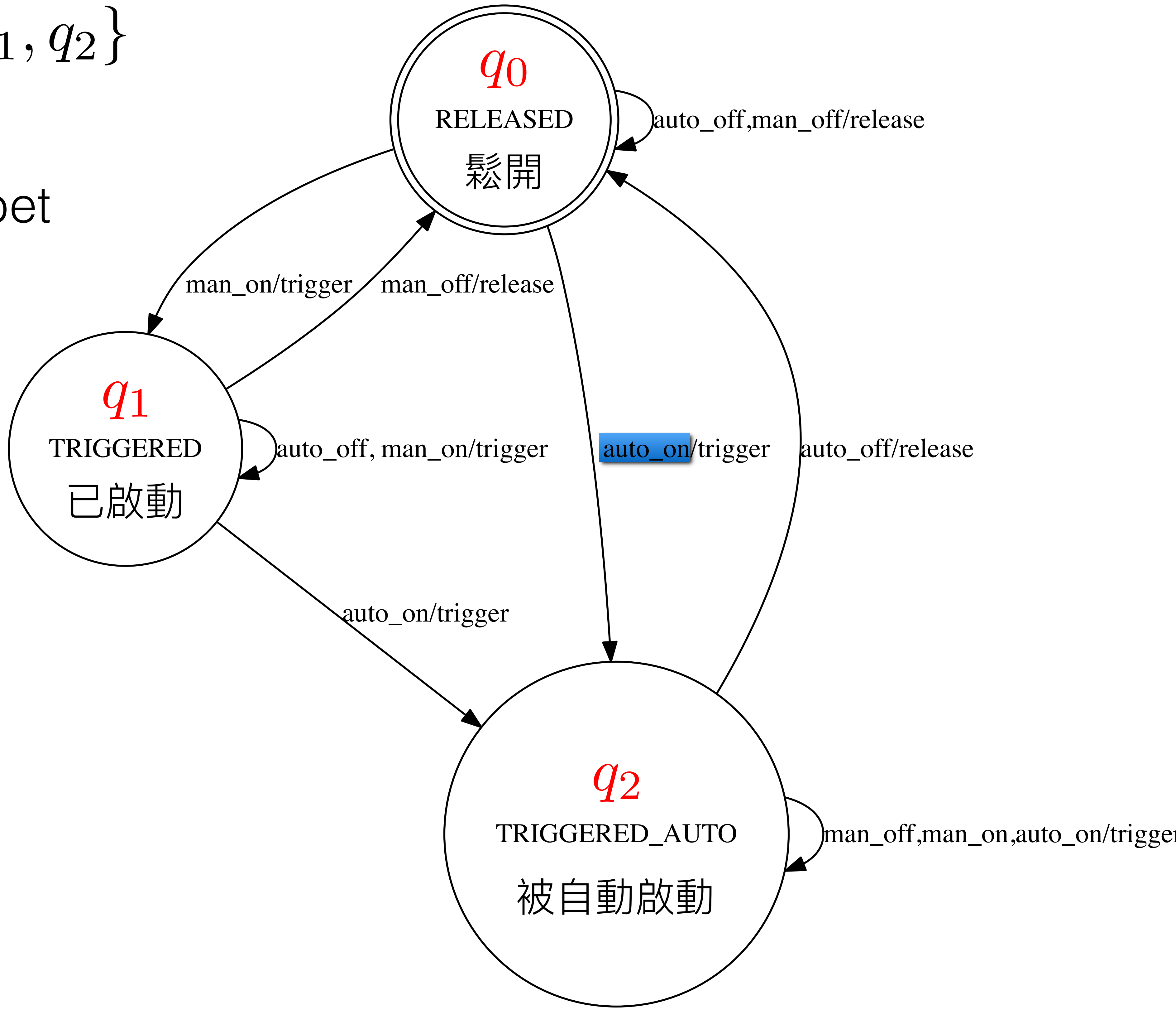


Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

Input Alphabet

man_on,
auto_on

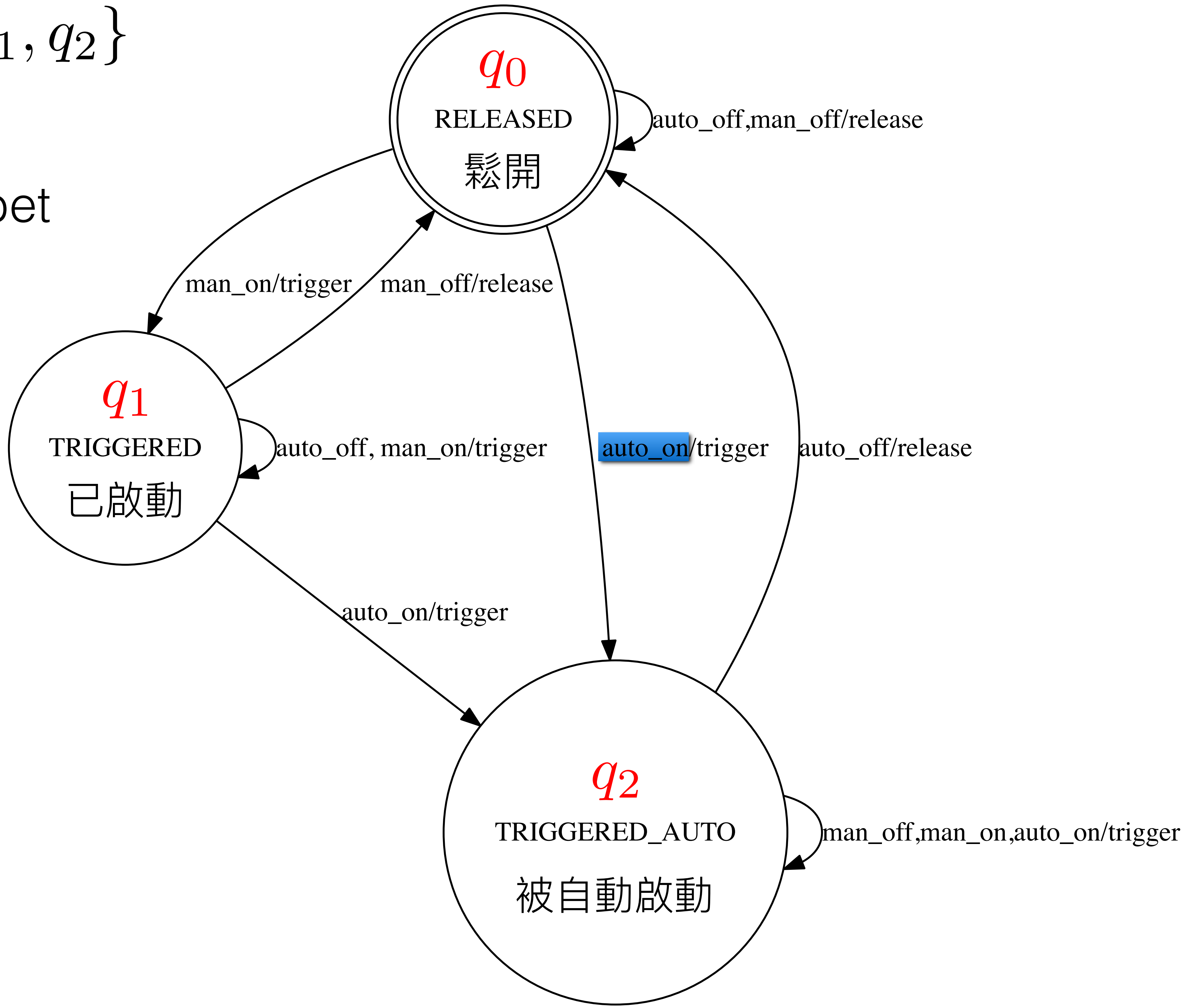


Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

Input Alphabet

man_on,
auto_on
man_off,
auto_off



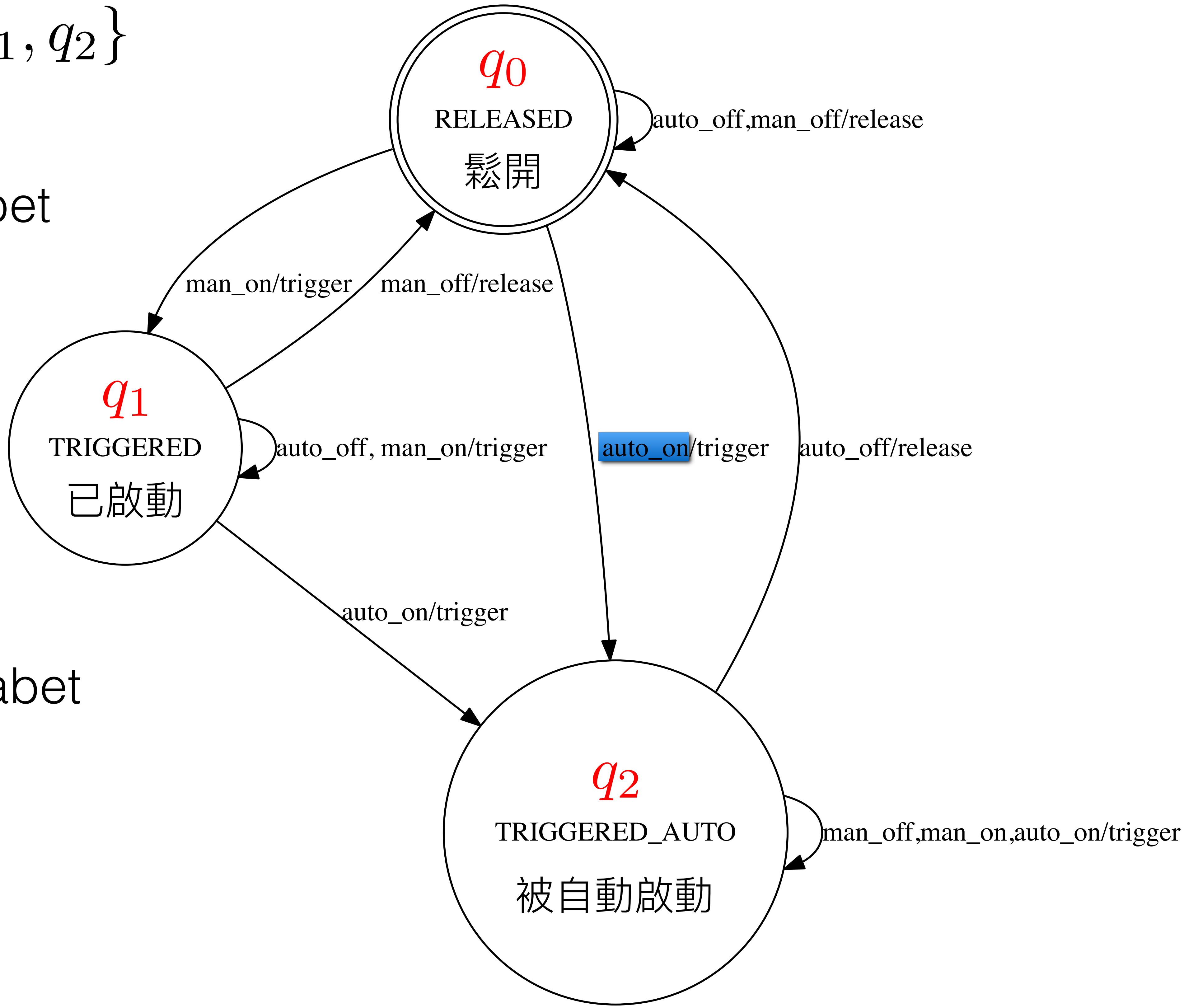
Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

Input Alphabet

man_on,
auto_on
man_off,
auto_off

Output Alphabet



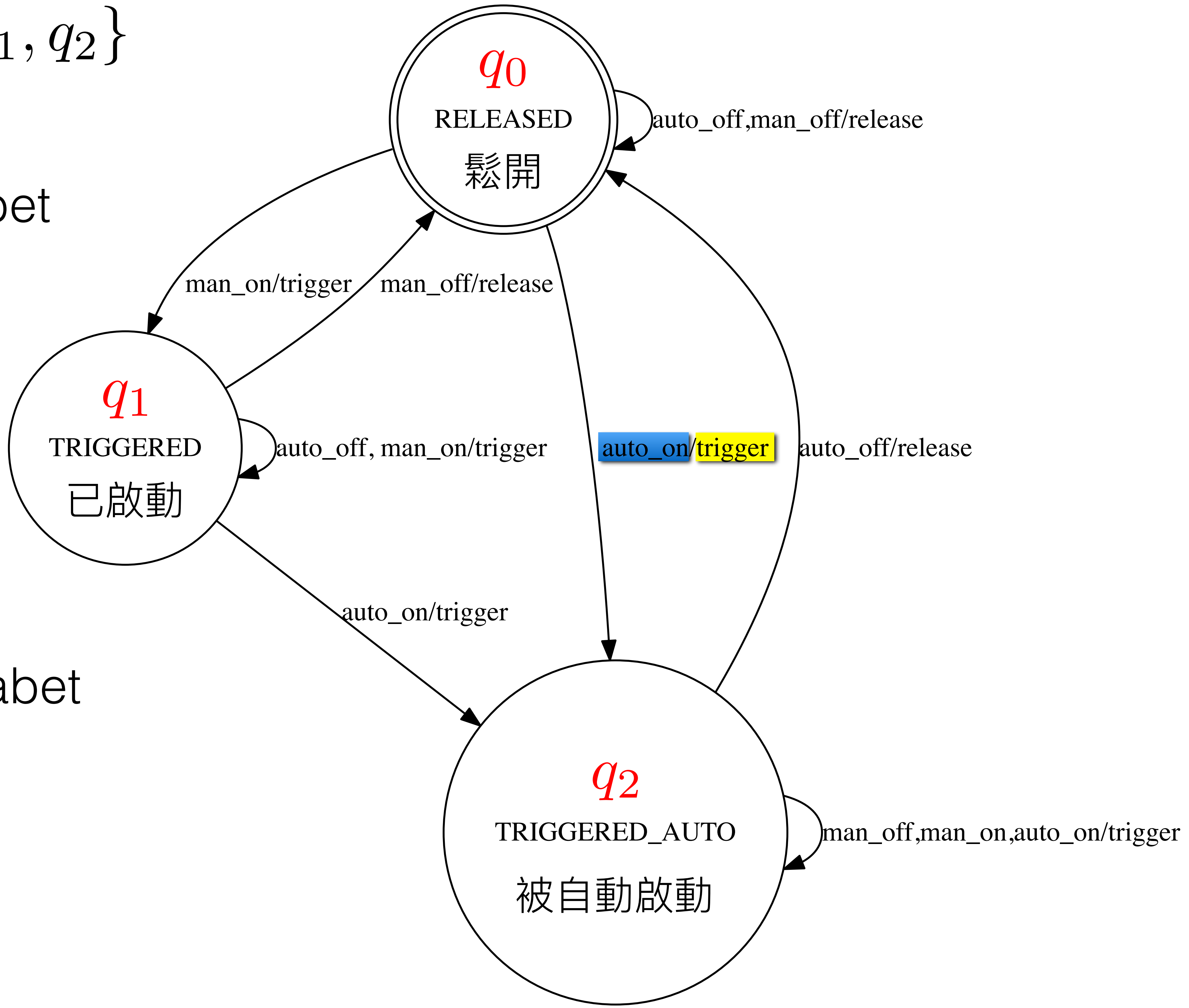
Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

Input Alphabet

man_on,
auto_on
man_off,
auto_off

Output Alphabet



Finite State Machine modelling the behaviour of the brake controller

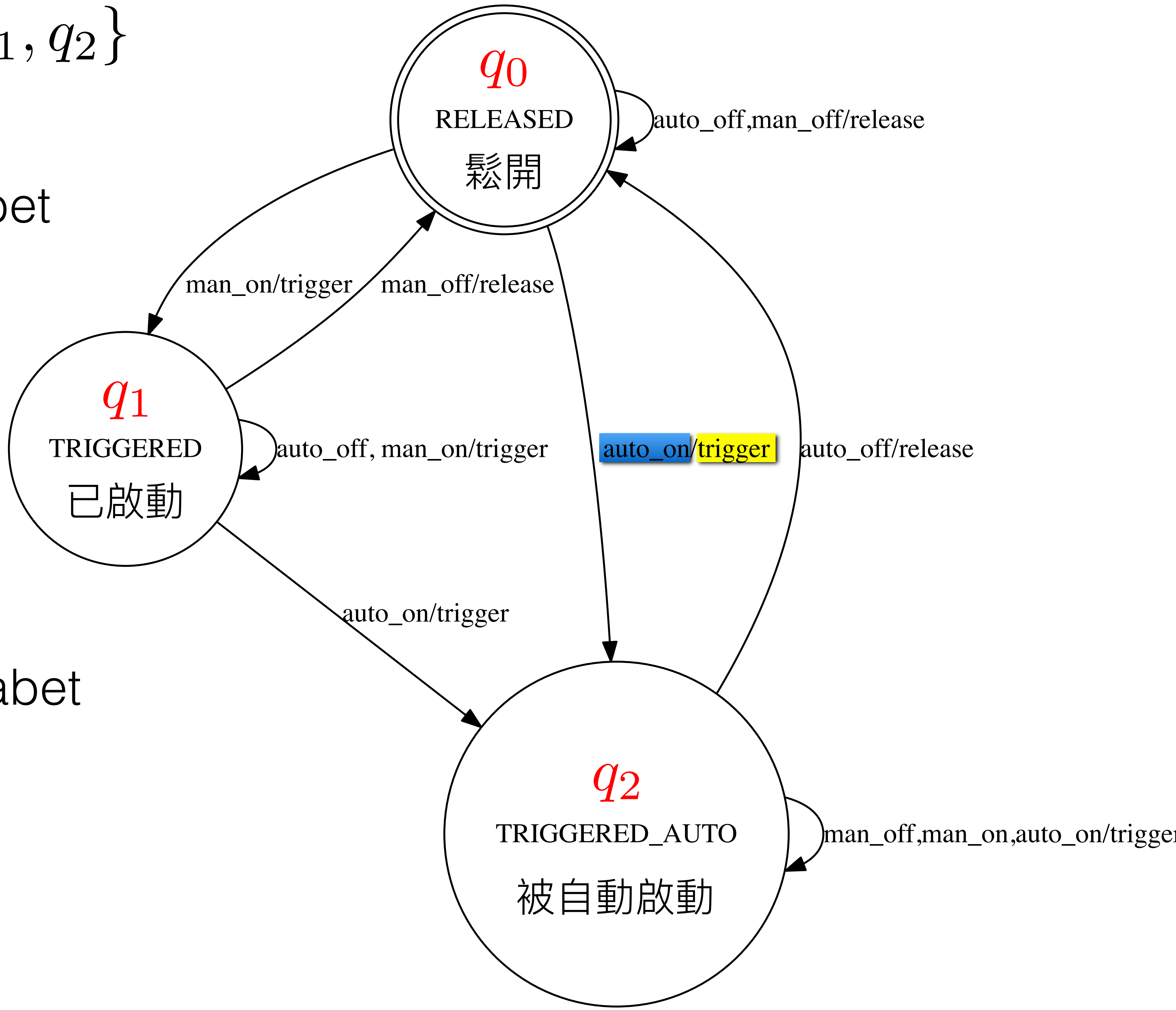
$$Q = \{q_0, q_1, q_2\}$$

Input Alphabet

man_on,
auto_on
man_off,
auto_off

Output Alphabet

trigger



Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

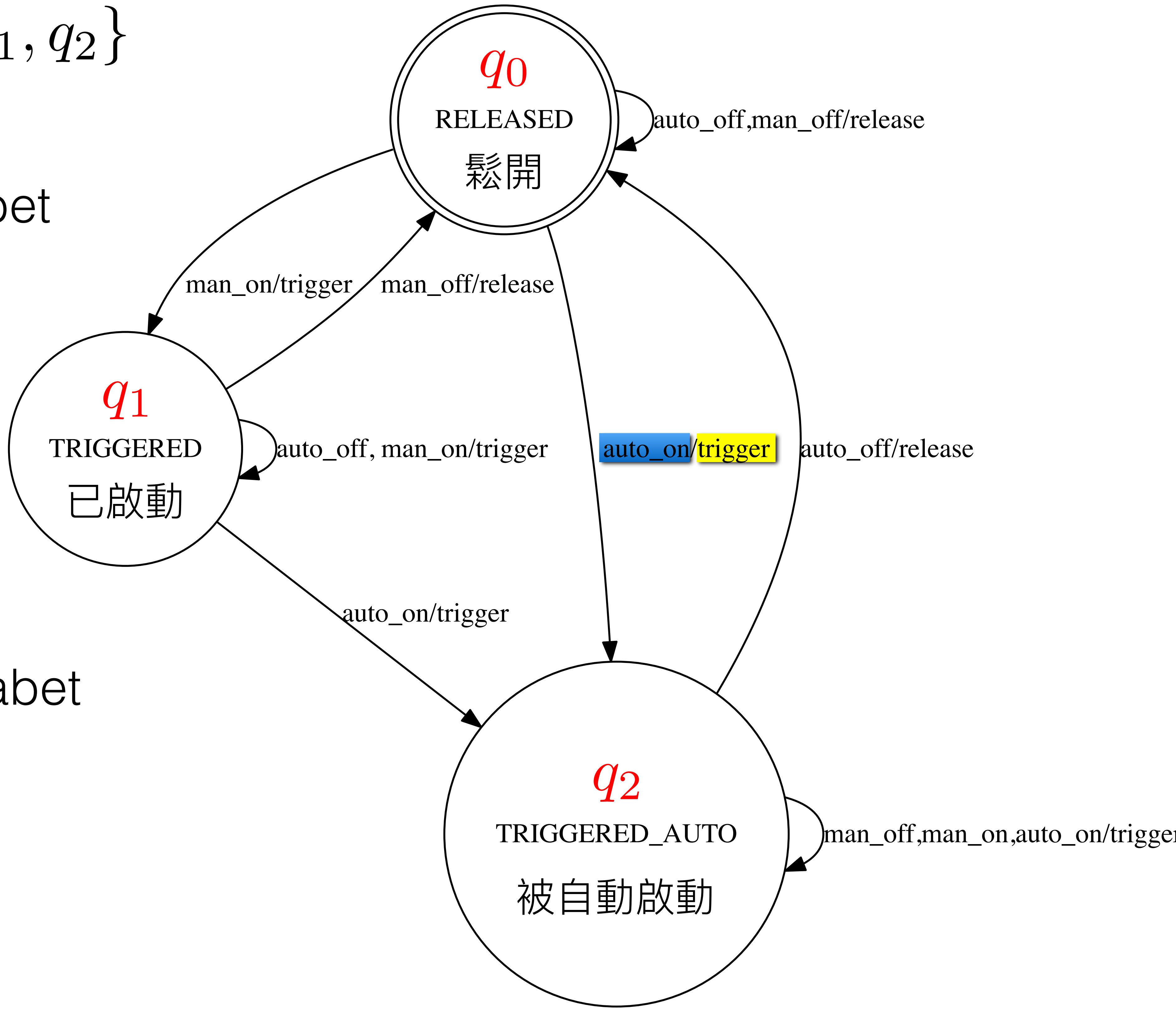
Input Alphabet

man_on,
auto_on
man_off,
auto_off

Output Alphabet

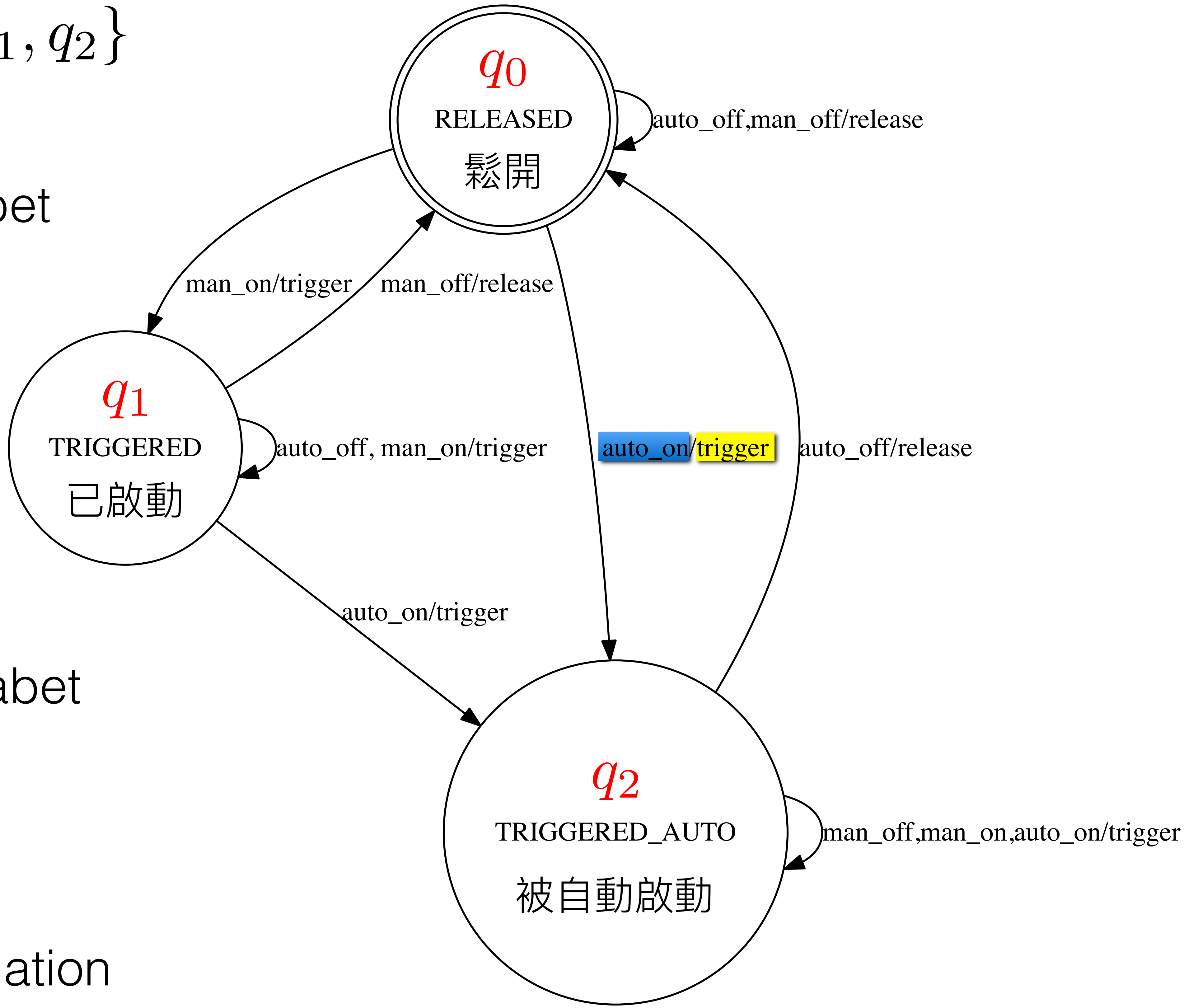
trigger

release



Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$



Input Alphabet

man_on,
auto_on
man_off,
auto_off

Output Alphabet

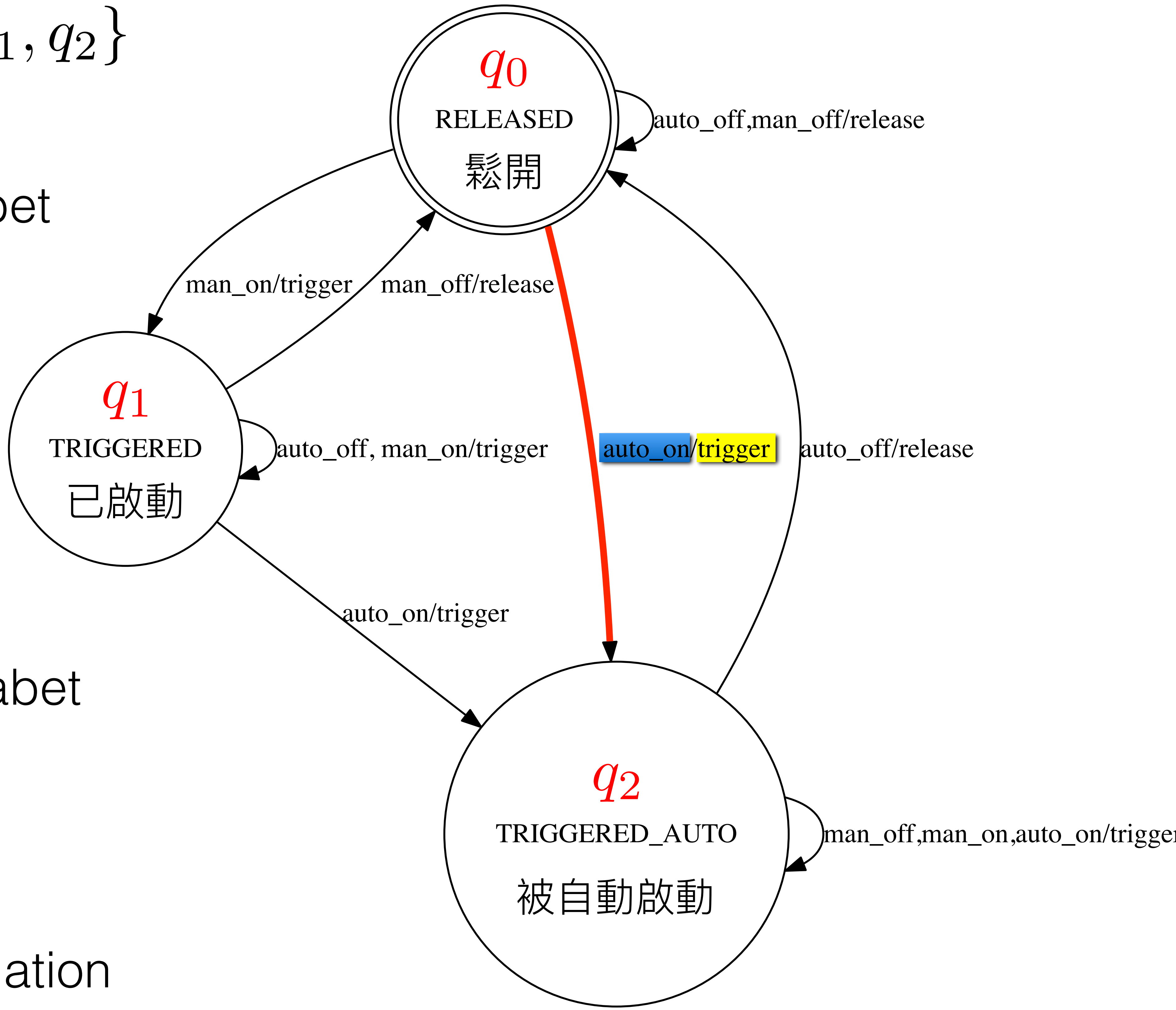
trigger

release

Transition Relation

Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$



Input Alphabet

man_on,
auto_on
man_off,
auto_off

Output Alphabet

trigger

release

Transition Relation

Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

Input Alphabet

man_on,
auto_on
man_off,
auto_off

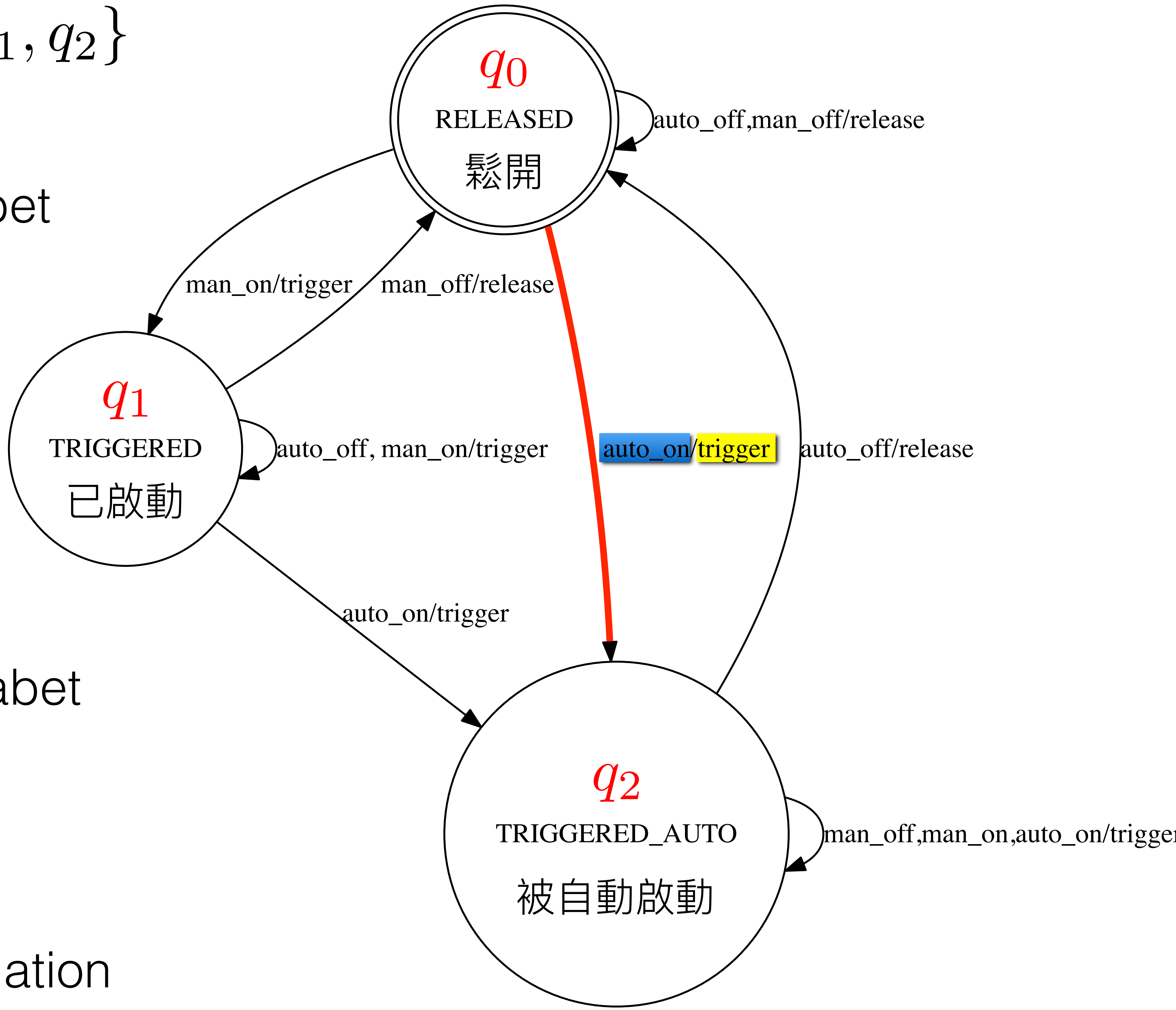
Output Alphabet

trigger

release

Transition Relation

$$(q_0, \text{auto\_on}, \text{trigger}, q_2), \ldots$$

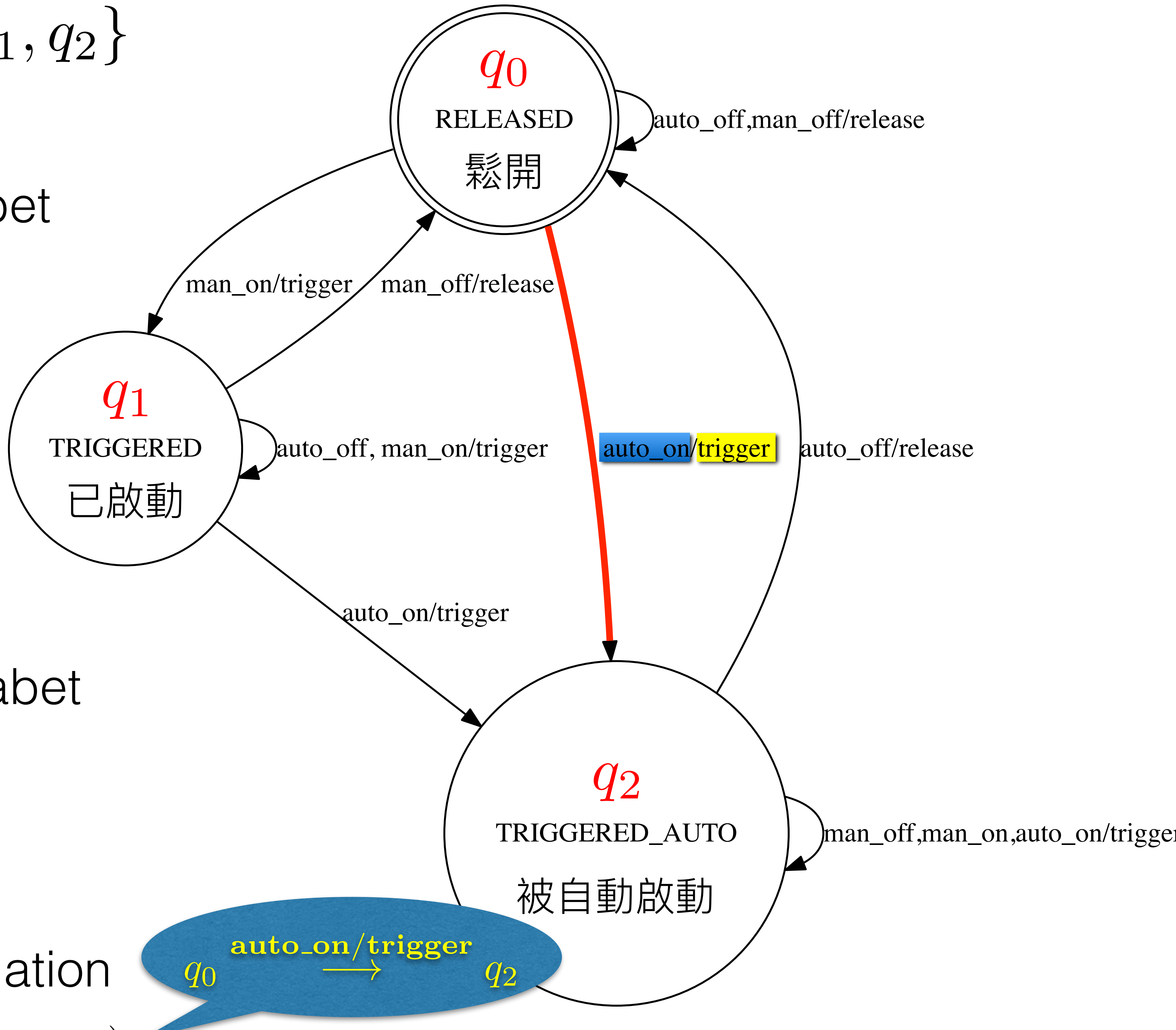


Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$

Input Alphabet

man_on,
auto_on
man_off,
auto_off



Output Alphabet

trigger

release

Transition Relation

$(q_0, \text{auto\_on}, \text{trigger}, q_2), \ldots$

Language

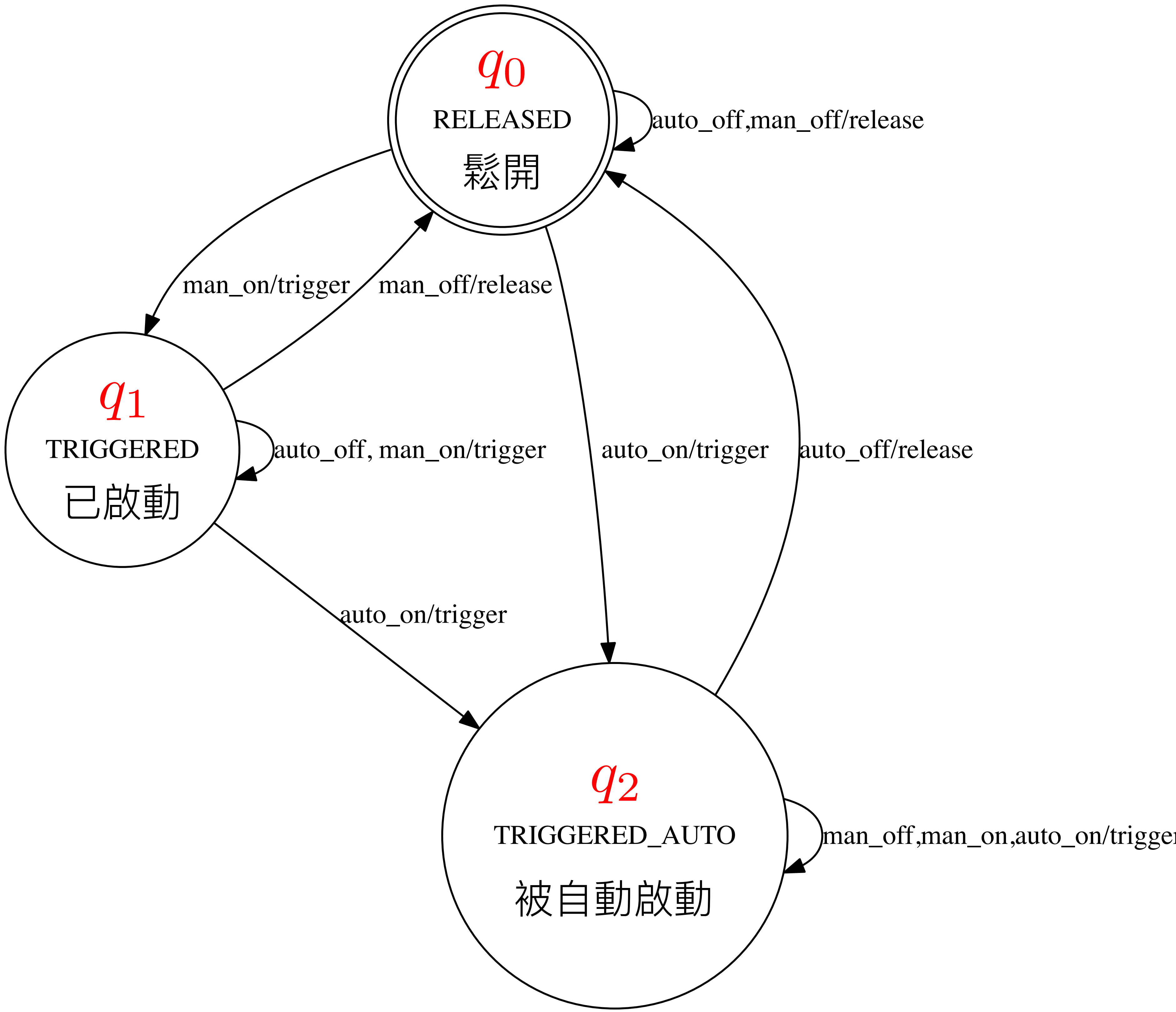
- $\Sigma := I \times O$ **alphabet** 字母表
- $\Sigma^* := \bigcup_{k \in \mathbf{N}_0} \{\sigma_1 \ldots \sigma_k \mid \sigma_i \in \Sigma, \forall i = 1, \ldots, k\}$ 有限字符序列集
- $\mathcal{L} \subseteq \Sigma^*$ **language over Σ** $:\Leftrightarrow$
  - $\varepsilon \in \mathcal{L}$ 空序列
  - $\mathcal{L}$ is prefix-closed 前置序列

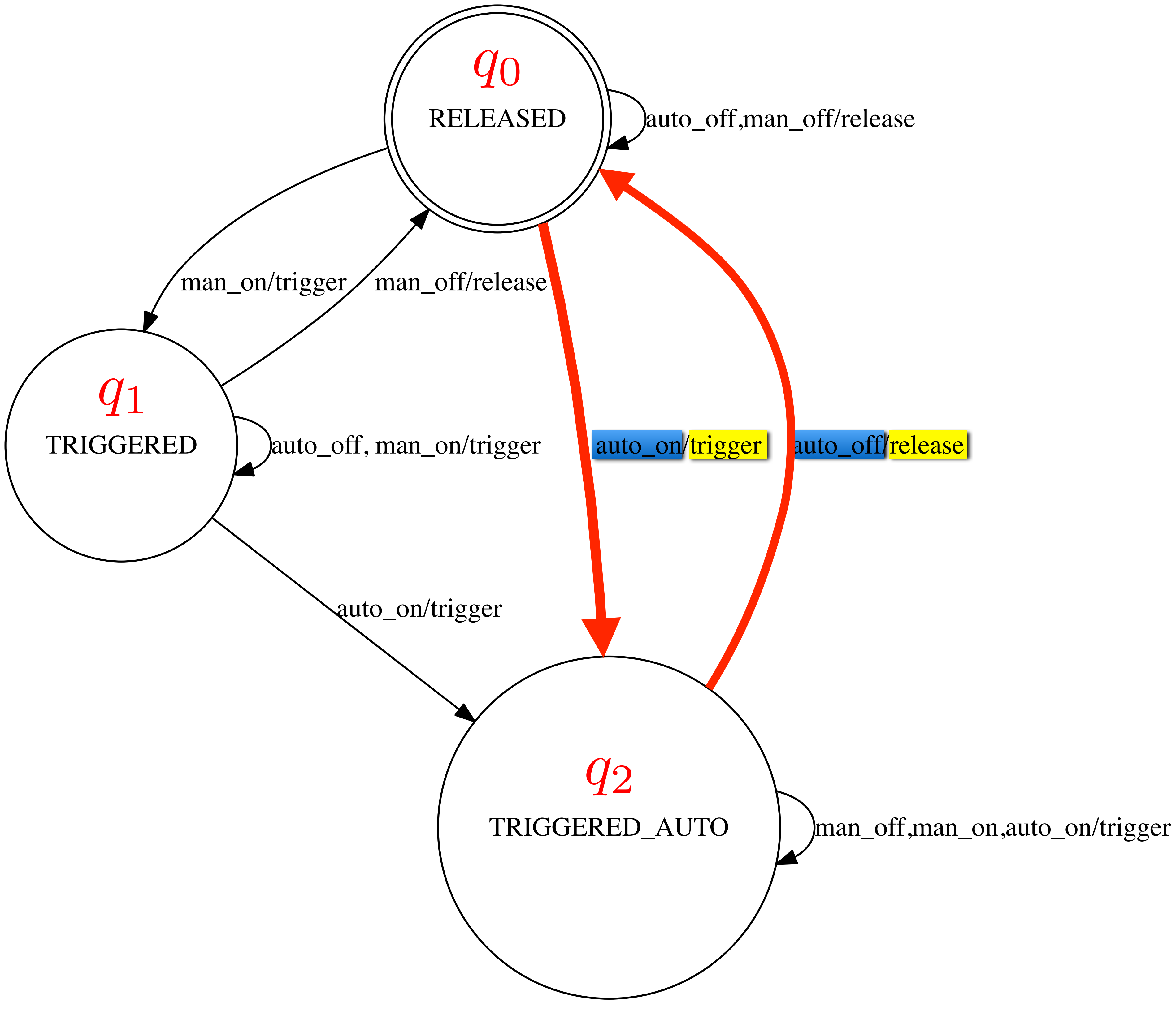
Language

- $\Sigma := I \times O$ **alphabet** 字母表
- $\Sigma^* := \bigcup_{k \in \mathbf{N}_0} \{\sigma_1 \ldots \sigma_k \mid \sigma_i \in \Sigma, \forall i = 1, \ldots, k\}$ 有限字符序列集
- $\mathcal{L} \subseteq \Sigma^*$ **language over** Σ $:\Leftrightarrow$
  - $\varepsilon \in \mathcal{L}$ 空序列
  - $\mathcal{L}$ is prefix-closed 前置序列

$$\forall \pi = \sigma_1 \ldots \sigma_k \in \mathcal{L} : \sigma_1 \ldots \sigma_i \in \mathcal{L}, \forall i \leq k$$

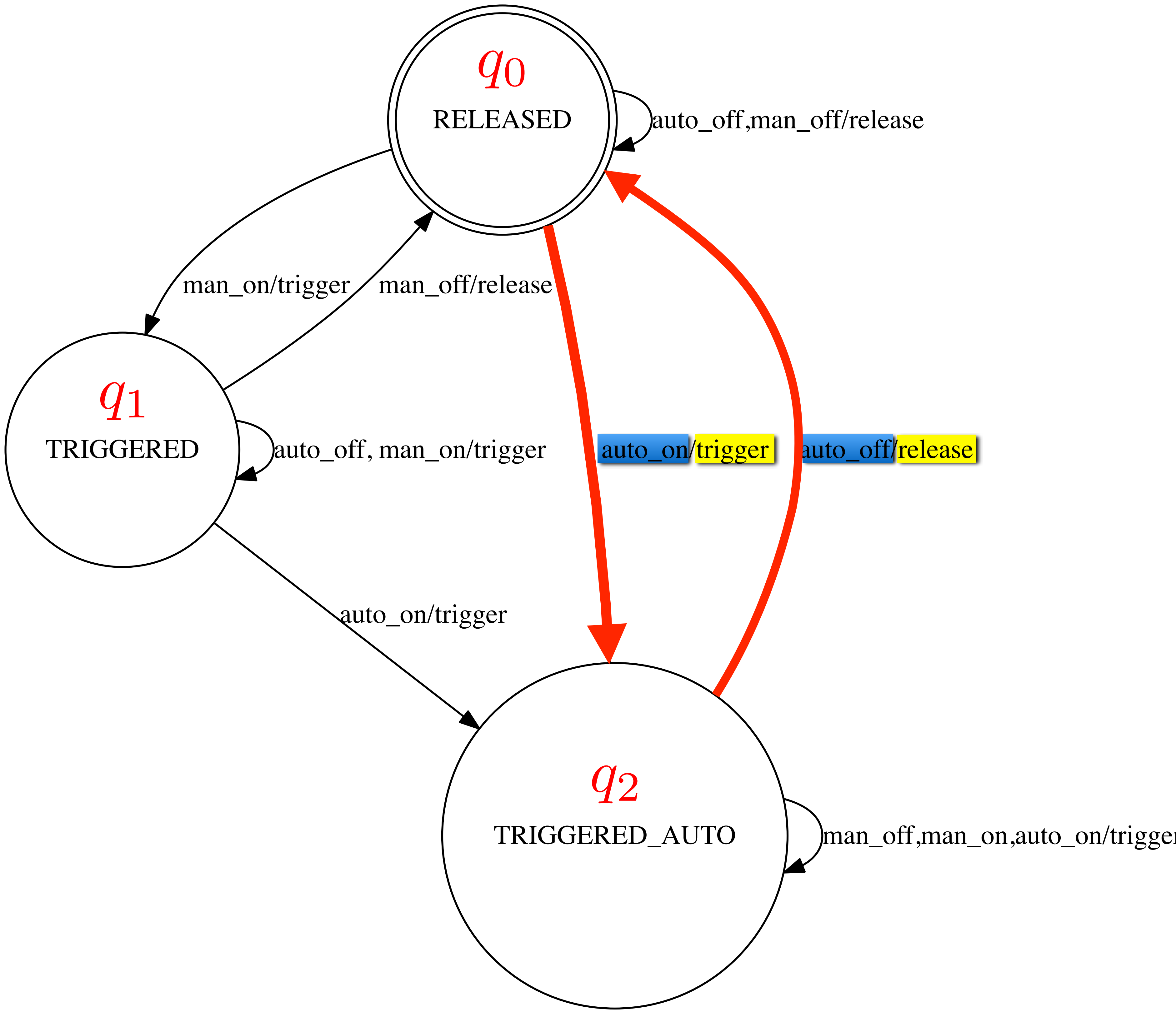
Finite State Machine modelling the behaviour of the brake controller



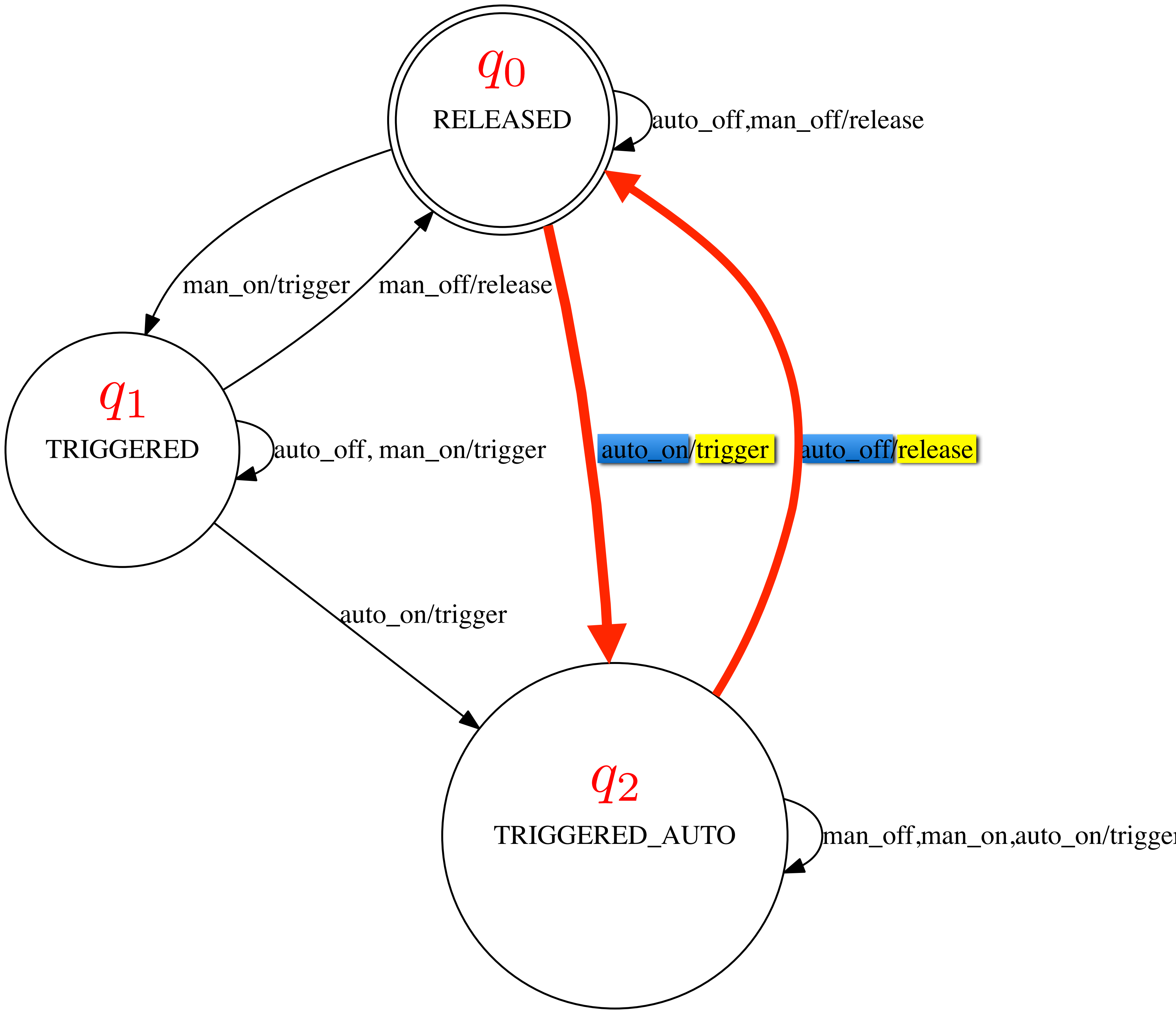


q_0
RELEASED
auto_off,man_off/release
man_on/trigger
man_off/release
q_1
TRIGGERED
auto_off, man_on/trigger
auto_on/trigger
auto_off/release
auto_on/trigger
q_2
TRIGGERED_AUTO
man_off,man_on,auto_on/trigger

$q_0 \xrightarrow{\textbf{auto\_on}/\text{trigger}} q_2 \xrightarrow{\textbf{auto\_off}/\text{release}} q_0$



$$q_0 \xrightarrow{\textbf{auto\_on}/\text{trigger}} q_2 \xrightarrow{\textbf{auto\_off}/\text{release}} q_0 \qquad (\textbf{auto\_on}, \textbf{trigger}).(\textbf{auto\_off}, \textbf{release}) \in L(q_0)$$



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 : $L(M') \subseteq L(M)$

FSM Properties

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

- M is **input-complete** : 輸入完備性

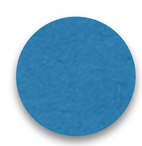
  $\forall q \in Q \wedge x \in I, \exists y \in O \wedge q' \in Q : q \xrightarrow{x/y} q'$

FSM Properties

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

- M is **input-complete** ： **輸入完備性**

$$\forall q \in Q \land x \in I, \exists y \in O \land q' \in Q : q \xrightarrow{x/y} q'$$



FSM Properties

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

- M is **input-complete**： **輸入完備性**

  $\forall q \in Q \land x \in I, \exists y \in O \land q' \in Q : q \xrightarrow{x/y} q'$

FSM Properties

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

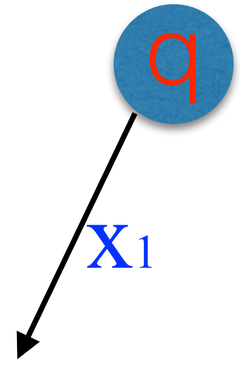
- M is **input-complete**： **輸入完備性**

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

FSM Properties

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

- M is **input-complete** : 輸入完備性
  $\forall q \in Q \wedge x \in I, \exists y \in O \wedge q' \in Q : q \xrightarrow{x/y} q'$

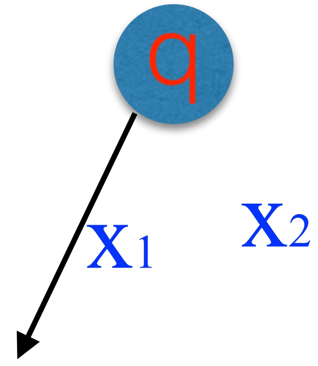


FSM Properties

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

- M is **input-complete** : 輸入完備性

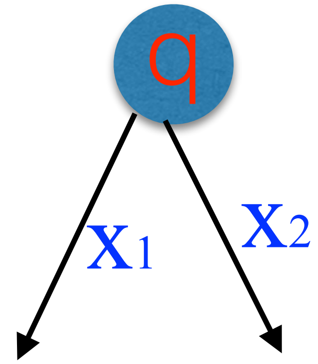
$$\forall q \in Q \land x \in I, \exists y \in O \land q' \in Q : q \xrightarrow{x/y} q'$$



FSM Properties

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

- M is **input-complete**： **輸入完備性**
  $\forall q \in Q \wedge x \in I, \exists y \in O \wedge q' \in Q : q \xrightarrow{x/y} q'$

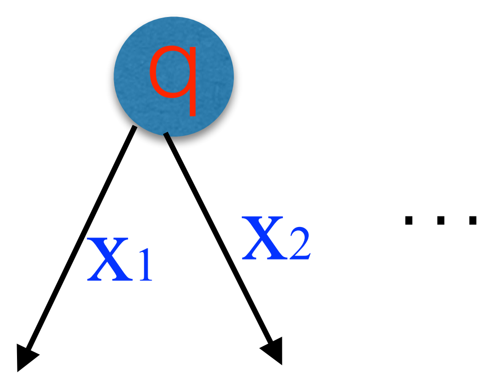


FSM Properties

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

- M is **input-complete**： 輸入完備性

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

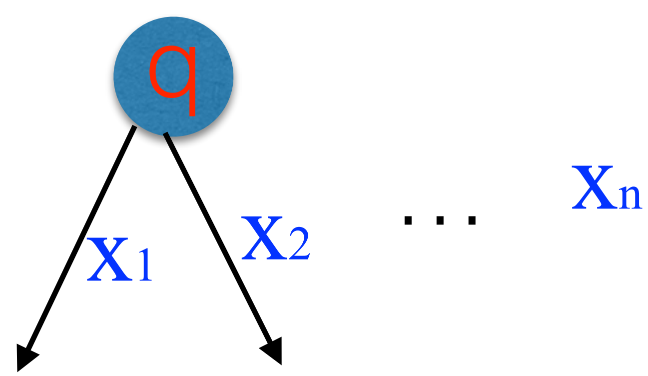


FSM Properties

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

- M is **input-complete** : **輸入完備性**

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

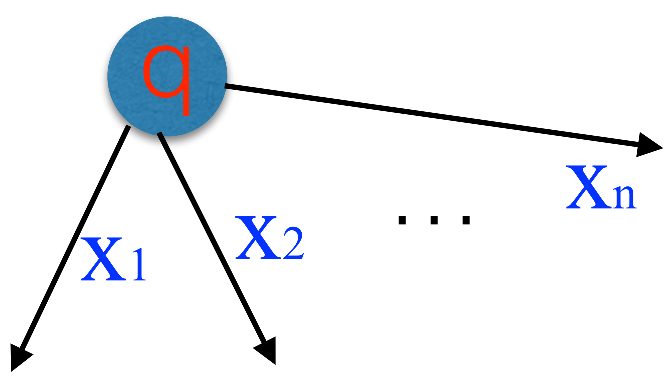


FSM Properties

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

- M is **input-complete** : 輸入完備性

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



FSM Properties

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

- M is **input-complete** : 輸入完備性

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

- M is **deterministic** : 確定性

  $$\forall q \overset{x/y_1}{\to} q_1, q \overset{x/y_2}{\to} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$$

FSM Properties

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

- M is **input-complete** ： **輸入完備性**

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

- M is **deterministic** ： **確定性**

  $\forall q \stackrel{x/y_1}{\rightarrow} q_1, q \stackrel{x/y_2}{\rightarrow} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$

FSM Properties

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

- M is **input-complete** : **輸入完備性**

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

- M is **deterministic** : **確定性**

  $\forall q \overset{x/y_1}{\rightarrow} q_1, q \overset{x/y_2}{\rightarrow} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$

FSM Properties

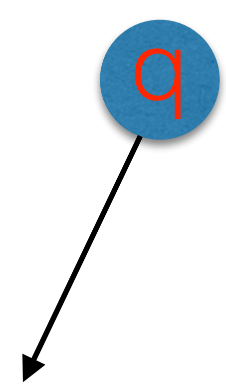
$$M = (Q, q_0, I, O, h)$$

- M is **input-complete** ： **輸入完備性**

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

- M is **deterministic** ： **確定性**

  $\forall q \xrightarrow{x/y_1} q_1, q \xrightarrow{x/y_2} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$



FSM Properties

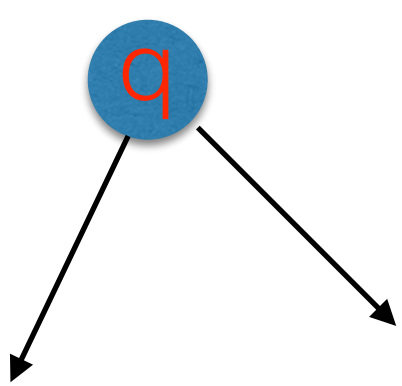
$$M = (Q, q_0, I, O, h)$$

- M is **input-complete** ： 輸入完備性

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

- M is **deterministic** ：確定性

  $\forall q \overset{x/y_1}{\to} q_1, q \overset{x/y_2}{\to} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$



FSM Properties

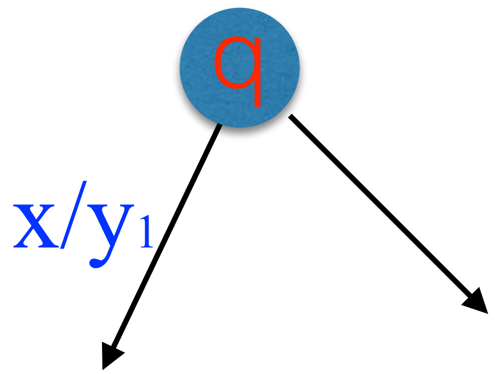
$$M = (Q, q_0, I, O, h)$$

- M is **input-complete**： **輸入完備性**

  $$\forall q \in Q \land x \in I, \exists y \in O \land q' \in Q : q \xrightarrow{x/y} q'$$

- M is **deterministic**：**確定性**

  $$\forall q \overset{x/y_1}{\rightarrow} q_1, q \overset{x/y_2}{\rightarrow} q_2 \in h \Rightarrow y_1 = y_2 \land q_1 = q_2$$



FSM Properties

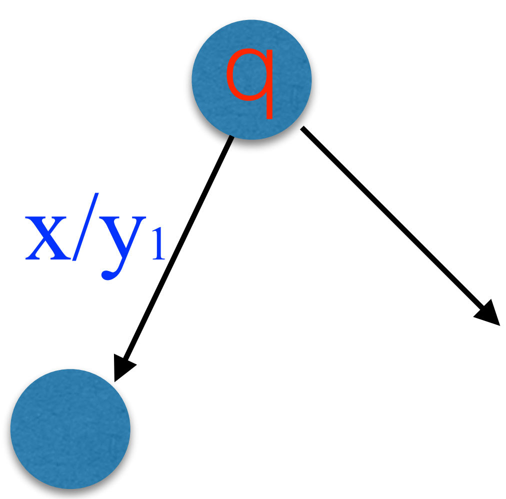
$$M = (Q, q_0, I, O, h)$$

- M is **input-complete** ： 輸入完備性

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

- M is **deterministic** ：確定性

$$\forall q \overset{x/y_1}{\to} q_1, q \overset{x/y_2}{\to} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$$



FSM Properties

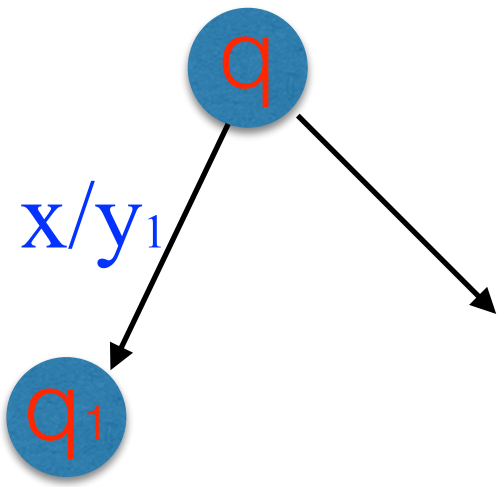
$$M = (Q, q_0, I, O, h)$$

- M is **input-complete** : 輸入完備性

  $\forall q \in Q \land x \in I, \exists y \in O \land q' \in Q : q \xrightarrow{x/y} q'$

- M is **deterministic** : 確定性

  $\forall q \overset{x/y_1}{\rightarrow} q_1, q \overset{x/y_2}{\rightarrow} q_2 \in h \Rightarrow y_1 = y_2 \land q_1 = q_2$



FSM Properties

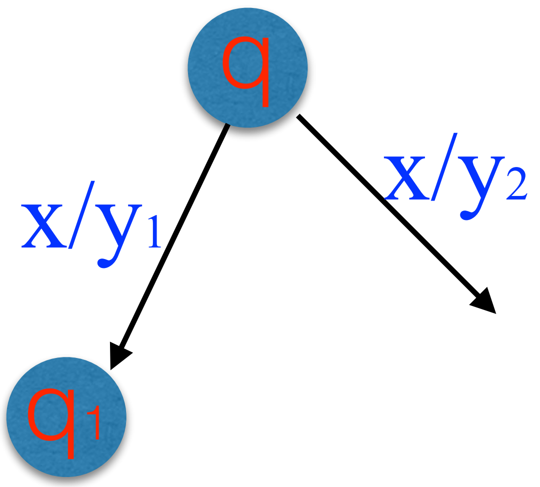
$$M = (Q, q_0, I, O, h)$$

- M is **input-complete**： **輸入完備性**

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

- M is **deterministic**：**確定性**

  $\forall q \overset{x/y_1}{\rightarrow} q_1, q \overset{x/y_2}{\rightarrow} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$



FSM Properties

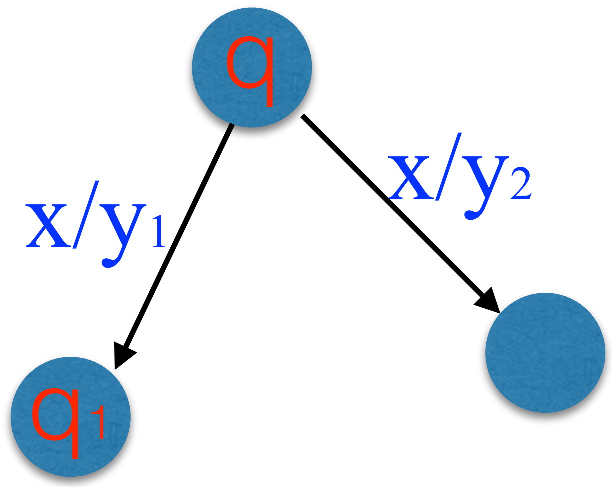
$$M = (Q, q_0, I, O, h)$$

- M is **input-complete**： **輸入完備性**

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

- M is **deterministic**：**確定性**

  $\forall q \overset{x/y_1}{\rightarrow} q_1, q \overset{x/y_2}{\rightarrow} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$

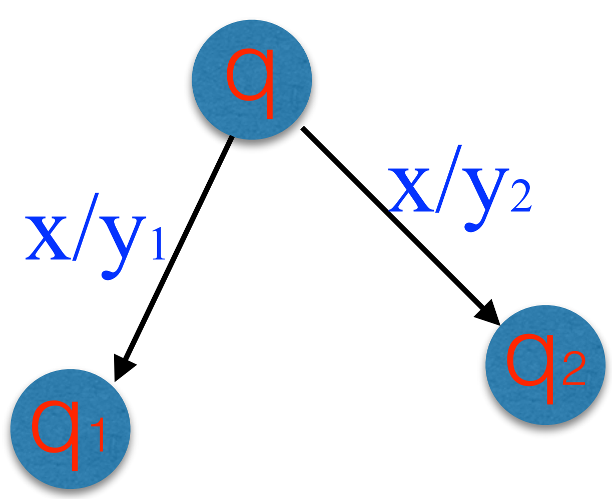


FSM Properties

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

- M is **input-complete** ： **輸入完備性**
  $\forall q \in Q \wedge x \in I, \exists y \in O \wedge q' \in Q : q \xrightarrow{x/y} q'$

- M is **deterministic** ： **確定性**
  $\forall q \overset{x/y_1}{\rightarrow} q_1, q \overset{x/y_2}{\rightarrow} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$



FSM Properties

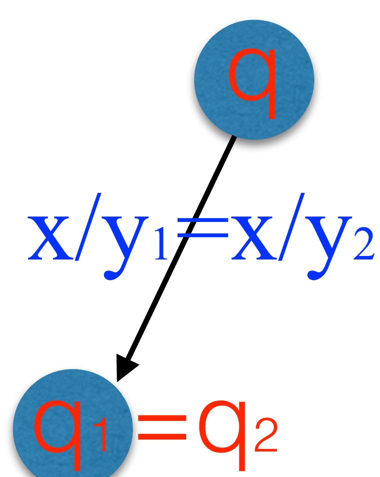
$$M = (Q, q_0, I, O, h)$$

- M is **input-complete** ： 輸入完備性

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

- M is **deterministic** ：確定性

  $\forall q \overset{x/y_1}{\rightarrow} q_1, q \overset{x/y_2}{\rightarrow} q_2 \in h \Rightarrow y_1 = y_2 \wedge q_1 = q_2$



FSM Properties

可觀察的

- M is **observable**, if
  $\forall q \stackrel{x/y}{\rightarrow} q_1, q \stackrel{x/y}{\rightarrow} q_2 \in h \Rightarrow q_1 = q_2$

FSM Properties

可觀察的

- M is **observable**, if

$$\forall q \xrightarrow{x/y} q_1, q \xrightarrow{x/y} q_2 \in h \Rightarrow q_1 = q_2$$

M is deterministic $\Rightarrow$ M is observable

FSM Properties

可觀察的

- M is **observable**, if
  $\forall q \stackrel{x/y}{\rightarrow} q_1, q \stackrel{x/y}{\rightarrow} q_2 \in h \Rightarrow q_1 = q_2$

M is deterministic $\Rightarrow$ M is observable

極小系統

FSM Properties

可觀察的

- M is **observable**, if

  $$\forall q \xrightarrow{x/y} q_1, q \xrightarrow{x/y} q_2 \in h \Rightarrow q_1 = q_2$$

M is deterministic $\Rightarrow$ M is observable

極小系統

- M is **minimal**, if
  - $\forall q \in Q, \exists \pi \in L(q_0) : q_0 \xrightarrow{\pi} q$
  - $\forall q_1 \neq q_2 \in Q \Rightarrow L(q_1) \neq L(q_2)$

FSM Properties

可觀察的

- M is **observable**, if

$$\forall q \xrightarrow{x/y} q_1, q \xrightarrow{x/y} q_2 \in h \Rightarrow q_1 = q_2$$

M is deterministic $\Rightarrow$ M is observable

極小系統

- M is **minimal**, if
  - $\forall q \in Q, \exists \pi \in L(q_0) : q_0 \xrightarrow{\pi} q$ — initial connected
  - $\forall q_1 \neq q_2 \in Q \Rightarrow L(q_1) \neq L(q_2)$

Signature Sig is a set of

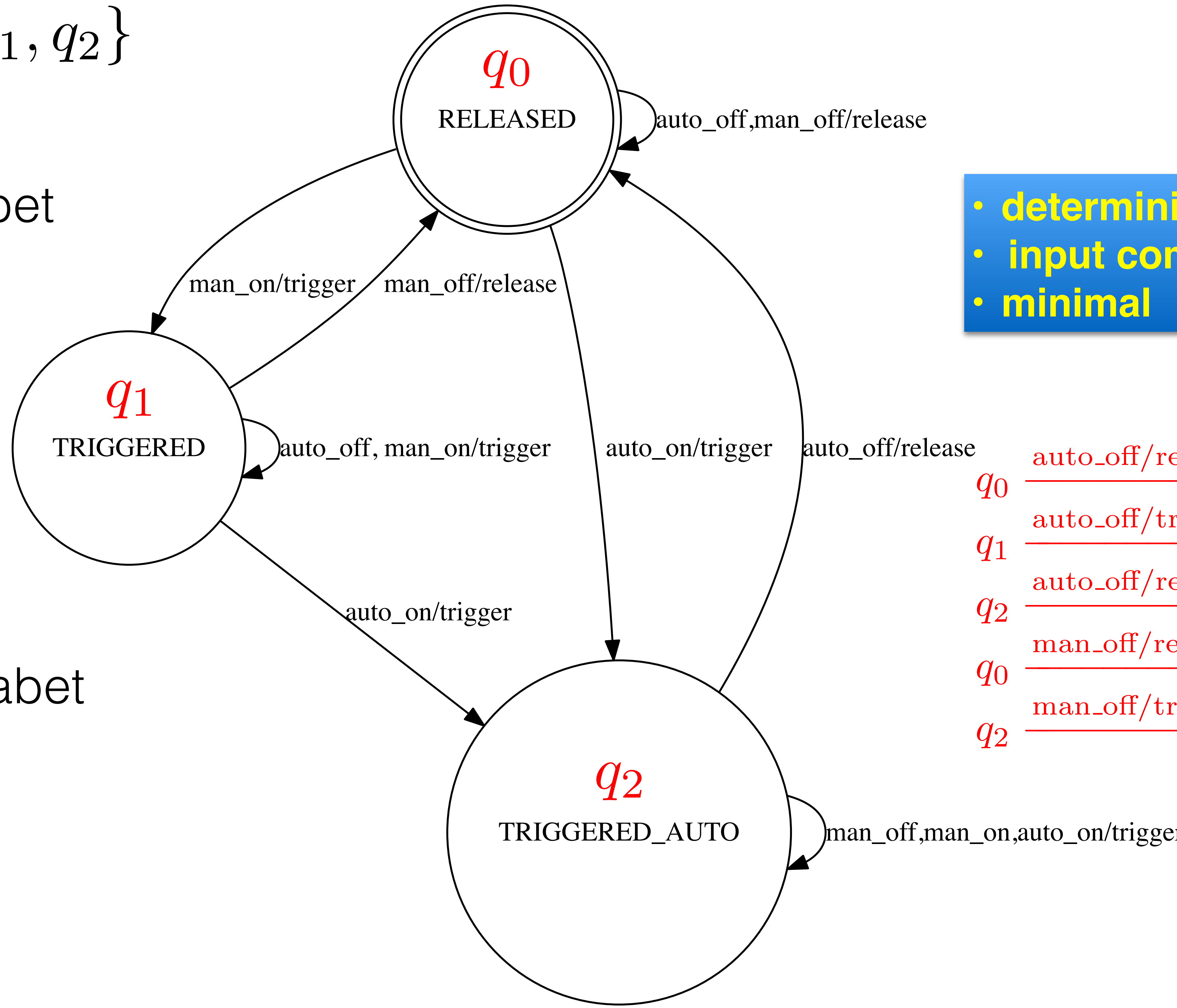
- input-complete
- minimal
- observable

finite state machines over $\Sigma = I \times O$

$$\forall M \;\text{ over } \Sigma,\; \exists M' \in \mathbf{Sig} : L(M) = L(M')$$

Finite State Machine modelling the behaviour of the brake controller

$$Q = \{q_0, q_1, q_2\}$$



Input Alphabet

man_on,
auto_on
man_off,
auto_off

Output Alphabet

trigger

release

- **deterministic**
- **input complete**
- **minimal**

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

$$M = (Q, q_0, I, O, h), M' = (Q', q_0', I, O, h') \in Sig$$

$L(M) = L(M') \Leftrightarrow M, M'$ **isomorphic**

$\exists f : M \to M'$:

- $Q \mapsto Q'$ bijection
- $q_0 \mapsto q_0'$
- $I \mapsto I$ identity map
- $O \mapsto O$ identity map
- $q_1 \stackrel{x/y}{\to} q_2 \in h \Leftrightarrow f(q_1) \stackrel{x/y}{\to} f(q_2) \in h'$

$$M = (Q, q_0, I, O, h), M' = (Q', q_0', I, O, h') \in Sig$$

$L(M) = L(M') \Leftrightarrow M, M'$ **isomorphic**

$\exists f : M \rightarrow M'$:

- $Q \mapsto Q'$ bijection
- $q_0 \mapsto q_0'$
- $I \mapsto I$ identity map
- $O \mapsto O$ identity map
- $q_1 \overset{x/y}{\rightarrow} q_2 \in h \Leftrightarrow f(q_1) \overset{x/y}{\rightarrow} f(q_2) \in h'$

Fault Models

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

- $M \in Sig$, **reference model** 參照模型
- $\leq \; \subseteq Sig \times Sig$, **conformance relation** 形變關係

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

Test Cases

測試範例

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

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

Test Suite

測試範例組

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

完備測試範例組

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

$\mathbf{TS}$, test suite

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

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

Testing Theories

- **Deterministic FSM:**
- **T-Method**
- **Product Automata**
- **W-Method, Wp-Method**
- **Nondeterministic FSM:**
- **W-Method, Wp-Method**
- **Adaptive Testing**

Signature Sig is the set of

- input-complete
- minimal
- deterministic

finite state machines over $\Sigma = I \times O$

T-Method

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

$M = (Q, q_0, I, O, h)$, $\forall M' \in \mathcal{D}_O \subseteq Sig$:

- $M' = (Q, q_0, I, O, h')$
- $\exists f : h \to h'$ bijective,
  $(q_1 \overset{x/y}{\longrightarrow} q_2) \mapsto (q_1 \overset{x/y'}{\longrightarrow} q_2)$

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-$\pi = q$

State Cover

exists? finite?

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-$\pi = q$

Lemma

Let $M = (Q, q_0, I, O, h) \in Sig$ and $\varepsilon \in U \subseteq L(M)$.
Then U is a state cover of M or
$\{q_0\text{-after-}\pi.\sigma \mid \pi \in U \wedge \sigma \in \Sigma \wedge \pi.\sigma \in L(M)\}$
$\not\subseteq \{q_0\text{-after-}\pi \mid \pi \in U\}$.

Transition Cover

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

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

Transition Cover

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

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

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

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

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

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

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

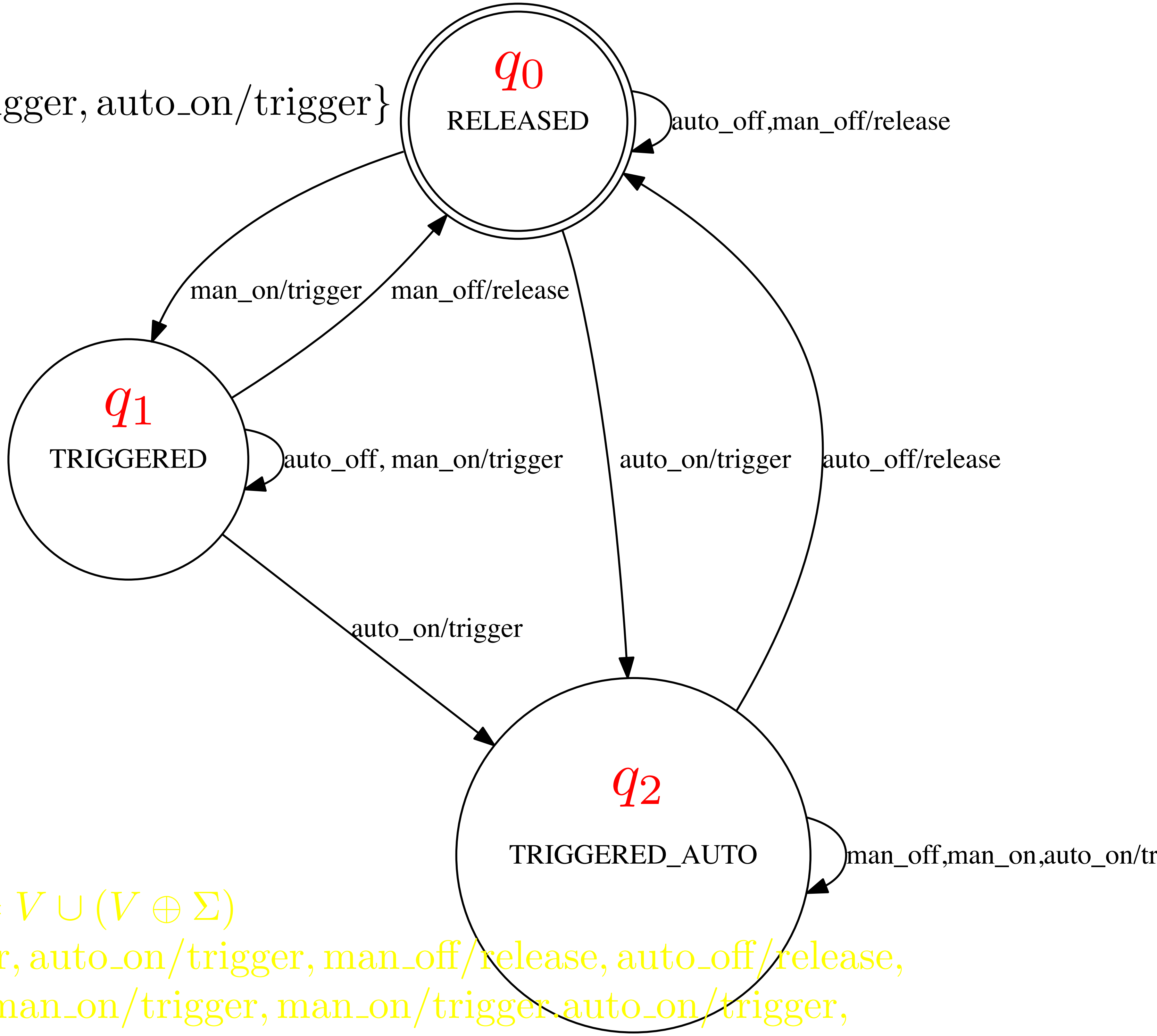
$\Rightarrow V \bigoplus_M (\{\varepsilon\} \cup \Sigma) = \big(V.(\{\varepsilon\} \cup \Sigma)\big) \cap L(M)$

$= V \cup \{\pi.\sigma \in L(M) \mid \pi \in V, \sigma \in \Sigma\}$

is a transition cover

Finite State Machine modelling the behaviour of the brake controller

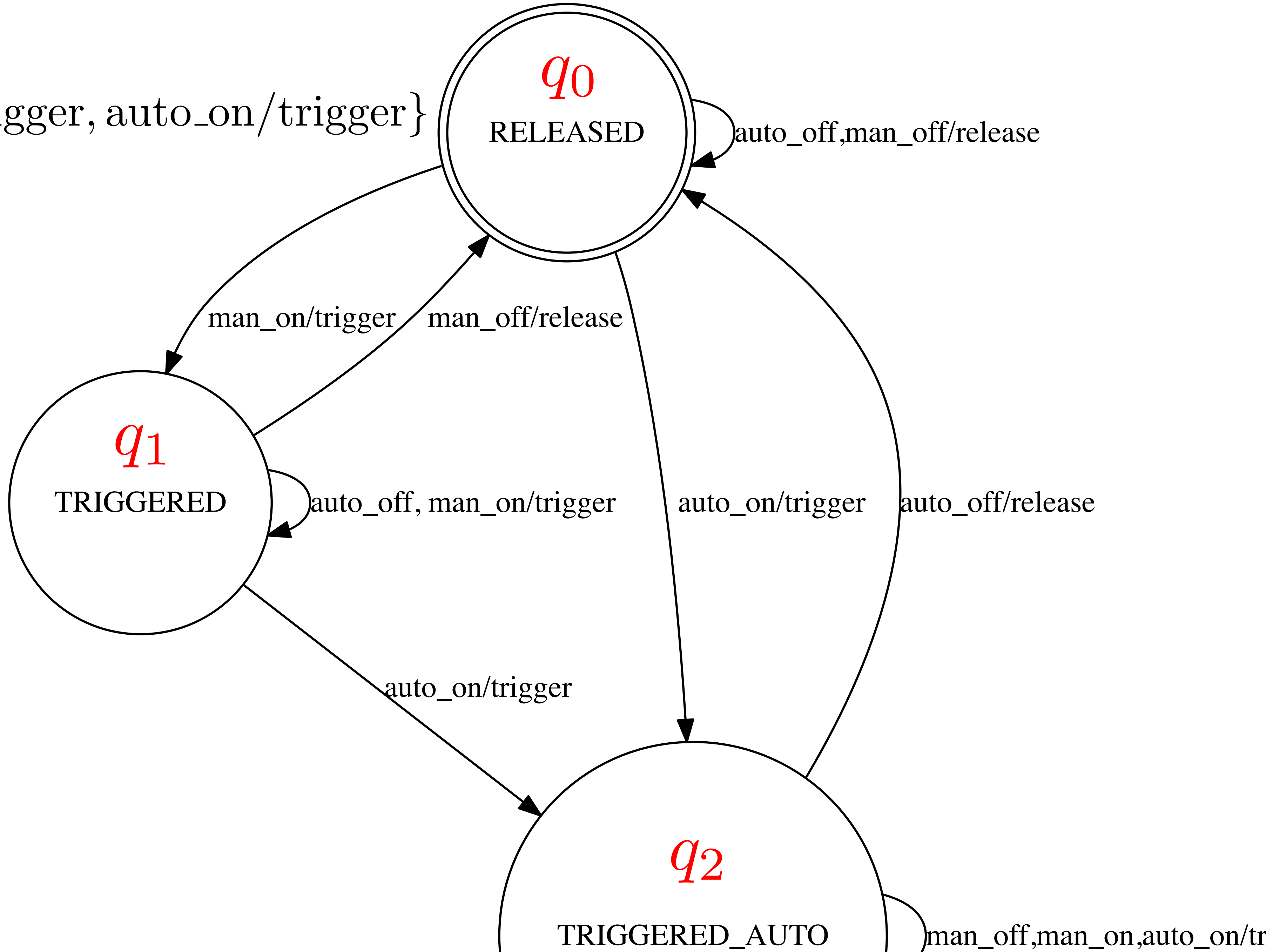
- state cover $V = \{\varepsilon, \text{man\_on}/\text{trigger}, \text{auto\_on}/\text{trigger}\}$



- transition cover $P = V \cup (V \oplus \Sigma)$
  $= \{\varepsilon, \text{man\_on}/\text{trigger}, \text{auto\_on}/\text{trigger}, \text{man\_off}/\text{release}, \text{auto\_off}/\text{release},$
  $\text{man\_on}/\text{trigger}.\text{man\_on}/\text{trigger}, \text{man\_on}/\text{trigger}.\text{auto\_on}/\text{trigger},$
  $\text{man\_on}/\text{trigger}.\text{man\_off}/\text{release}, \text{man\_on}/\text{trigger}.\text{auto\_off}/\text{trigger},$
  $\text{auto\_on}/\text{trigger}.\text{man\_on}/\text{trigger}, \text{auto\_on}/\text{trigger}.\text{auto\_on}/\text{trigger},$
  $\text{auto\_on}/\text{trigger}.\text{man\_off}/\text{trigger}, \text{auto\_on}/\text{trigger}.\text{auto\_off}/\text{release}\}$

Finite State Machine modelling the behaviour of the brake controller

- state cover $V = \{\varepsilon, \text{man\_on}/\text{trigger}, \text{auto\_on}/\text{trigger}\}$



- transition cover $P = V \cup (V \oplus \Sigma)$
  $= \{\varepsilon, \text{man\_on}/\text{trigger}, \text{auto\_on}/\text{trigger}, \text{man\_off}/\text{release}, \text{auto\_off}/\text{release},$
  $\text{man\_on}/\text{trigger}.\text{man\_on}/\text{trigger}, \text{man\_on}/\text{trigger}.\text{auto\_on}/\text{trigger},$
  $\text{man\_on}/\text{trigger}.\text{man\_off}/\text{release}, \text{man\_on}/\text{trigger}.\text{auto\_off}/\text{trigger},$
  $\text{auto\_on}/\text{trigger}.\text{man\_on}/\text{trigger}, \text{auto\_on}/\text{trigger}.\text{auto\_on}/\text{trigger},$
  $\text{auto\_on}/\text{trigger}.\text{man\_off}/\text{trigger}, \text{auto\_on}/\text{trigger}.\text{auto\_off}/\text{release}\}$

T-Method

Every $\mathbf{TS} = P$ transition cover of M
is a complete test suite of $\mathcal{F}(M, I, O, \sim, \mathcal{D}_O)$

Proof:

Suppose M', M are not I/O equivalent.

Then there exists $\pi \in \Sigma^*$, $\sigma \neq \sigma' \in \Sigma$ such that $\pi.\sigma \in L(M)$ and $\pi.\sigma' \in L(M')$ with $\sigma_I = \sigma'_I$.

Let $q_0 \xrightarrow{\pi} q$, $q \xrightarrow{\sigma} q_1 \in h$, and $q \xrightarrow{\sigma'} q_1 \in h'$.

Since **TS** is a transition cover, there is $\tau \in \mathbf{TS}$ such that $q_0 \xrightarrow{\tau} q \in h$ and $\tau.\sigma \in \mathbf{TS}$.

Let $q_0 \xrightarrow{\tau'} q \in h'$ with $\tau_I = \tau'_I$.

Then $q_0 \xrightarrow{\tau'} q \xrightarrow{\sigma'} q_1 \in h'$ and $q_0 \xrightarrow{\tau} q \xrightarrow{\sigma} q_1 \in h$.

Since $\tau'_I.\sigma'_I = \tau_I.\sigma_I$ and $\tau'_O.\sigma'_O \neq \tau_O.\sigma_O$, we have $\tau.\sigma \notin L(M')$

Hence M' fails the test case $\tau.\sigma \in \mathbf{TS}$.

W-Method

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

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

$$\mathcal{D}_m = \{M' = (Q', q_0', I, O, h') \in Sig \mid |Q'| \leq m\}$$

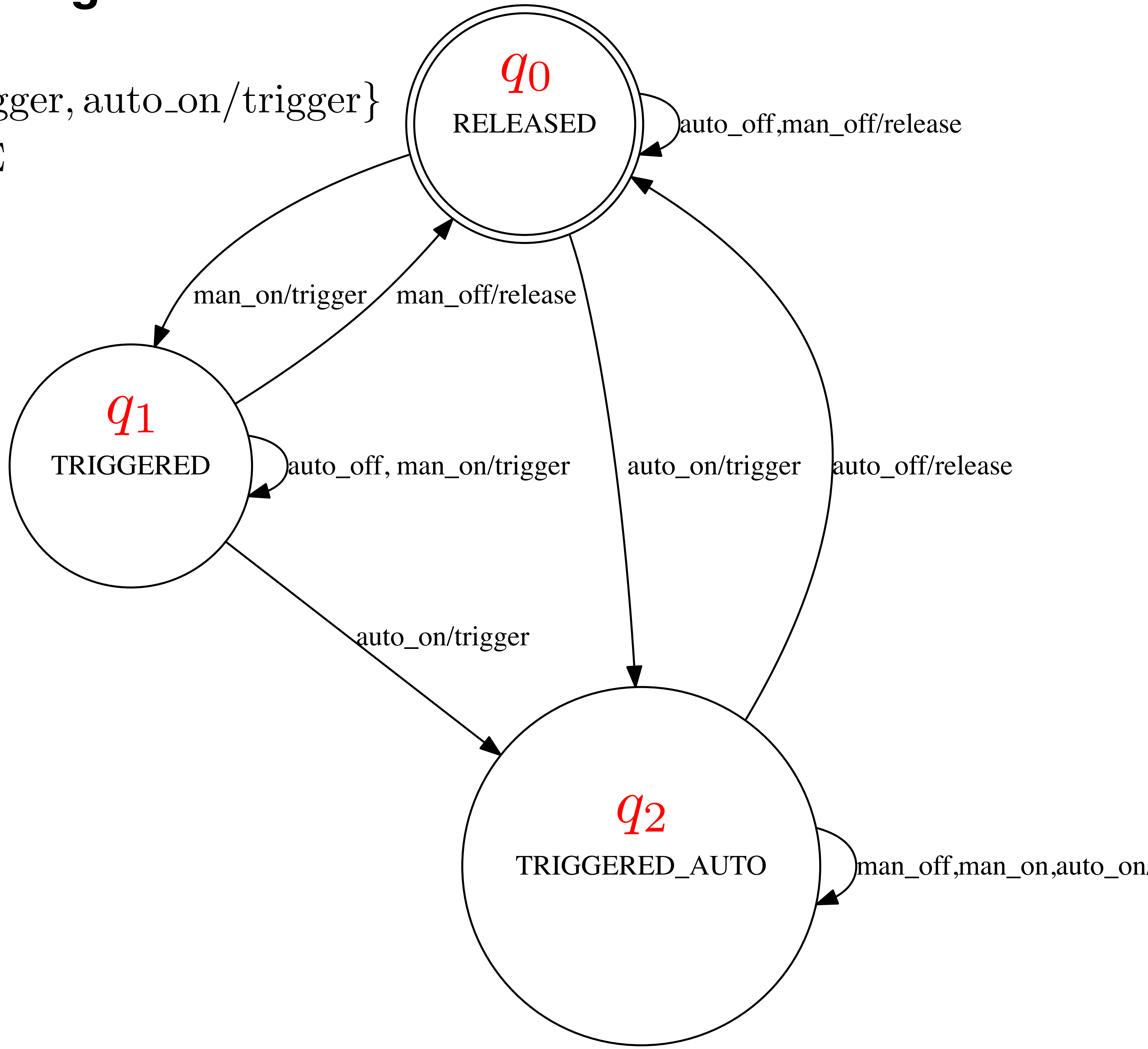
Characterization Set

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

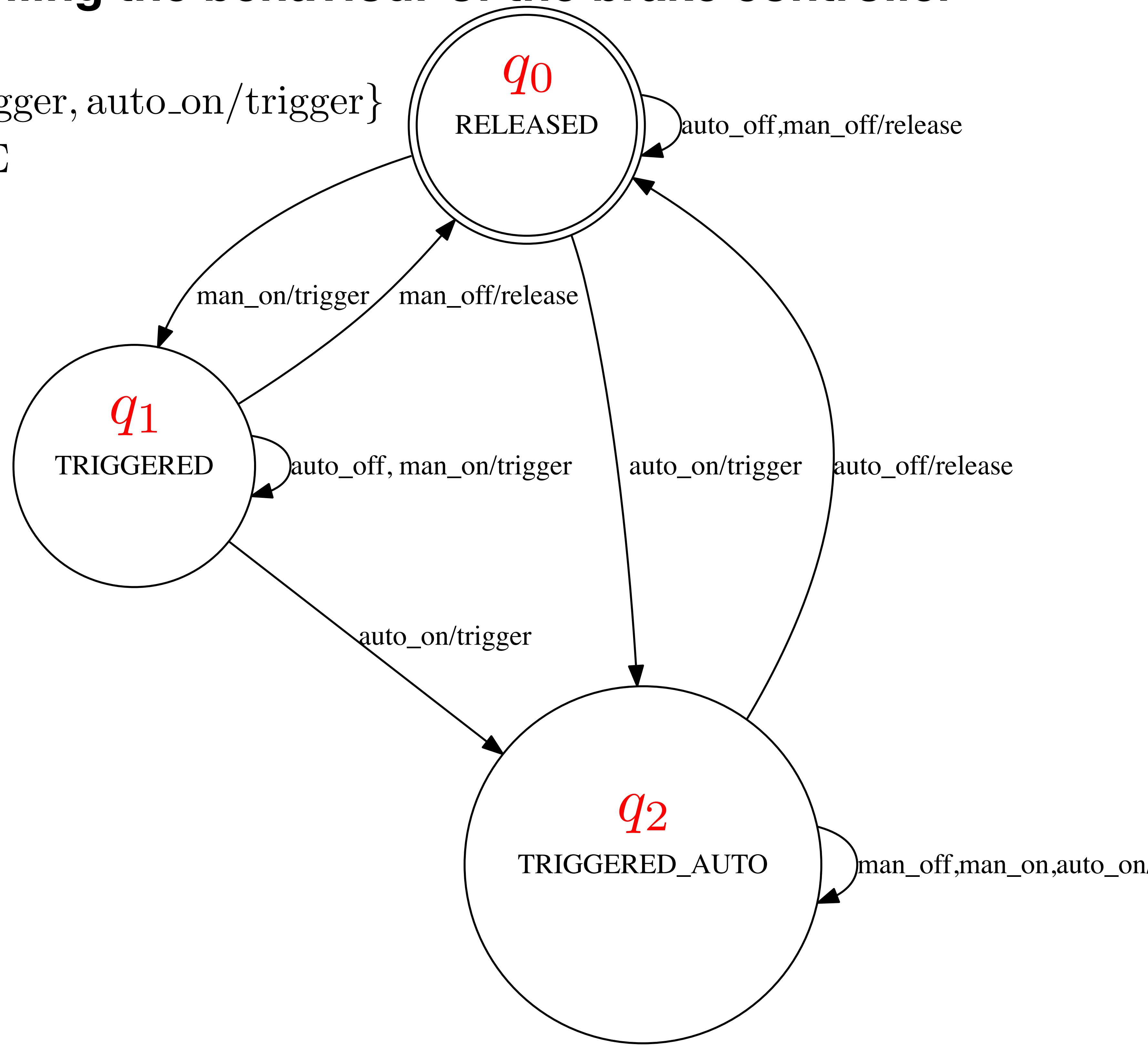
- state cover $V = \{\varepsilon, \text{man\_on/trigger}, \text{auto\_on/trigger}\}$
- transition cover $P = V \cup V \oplus \Sigma$



Finite State Machine modelling the behaviour of the brake controller

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

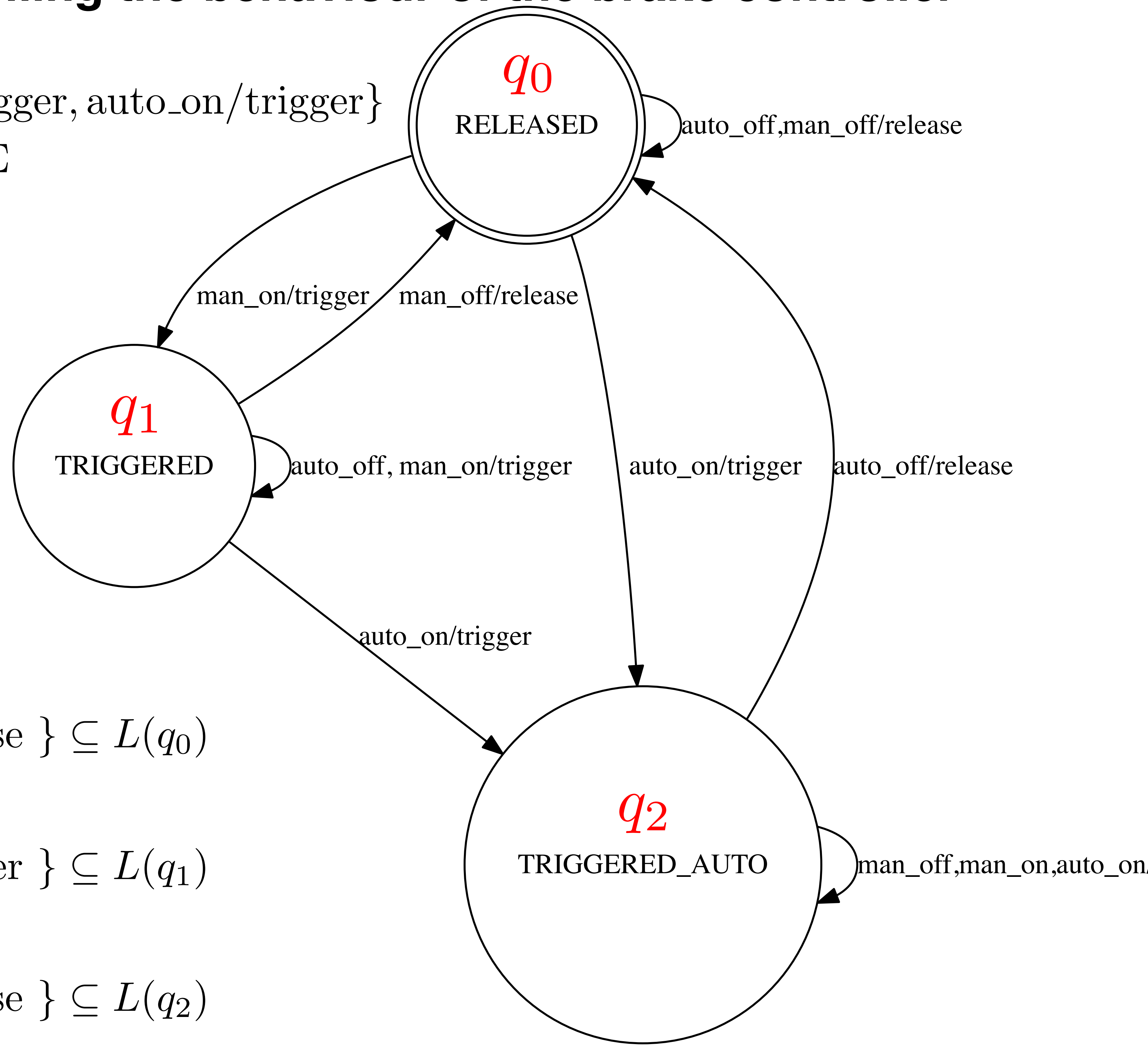
$$q_0 \xrightarrow{\text{auto\_off/release}} q_0$$
$$q_1 \xrightarrow{\text{auto\_off/trigger}} q_1$$
$$q_2 \xrightarrow{\text{auto\_off/release}} q_0$$
$$q_0 \xrightarrow{\text{man\_off/release}} q_0$$
$$q_2 \xrightarrow{\text{man\_off/trigger}} q_2$$



Finite State Machine modelling the behaviour of the brake controller

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

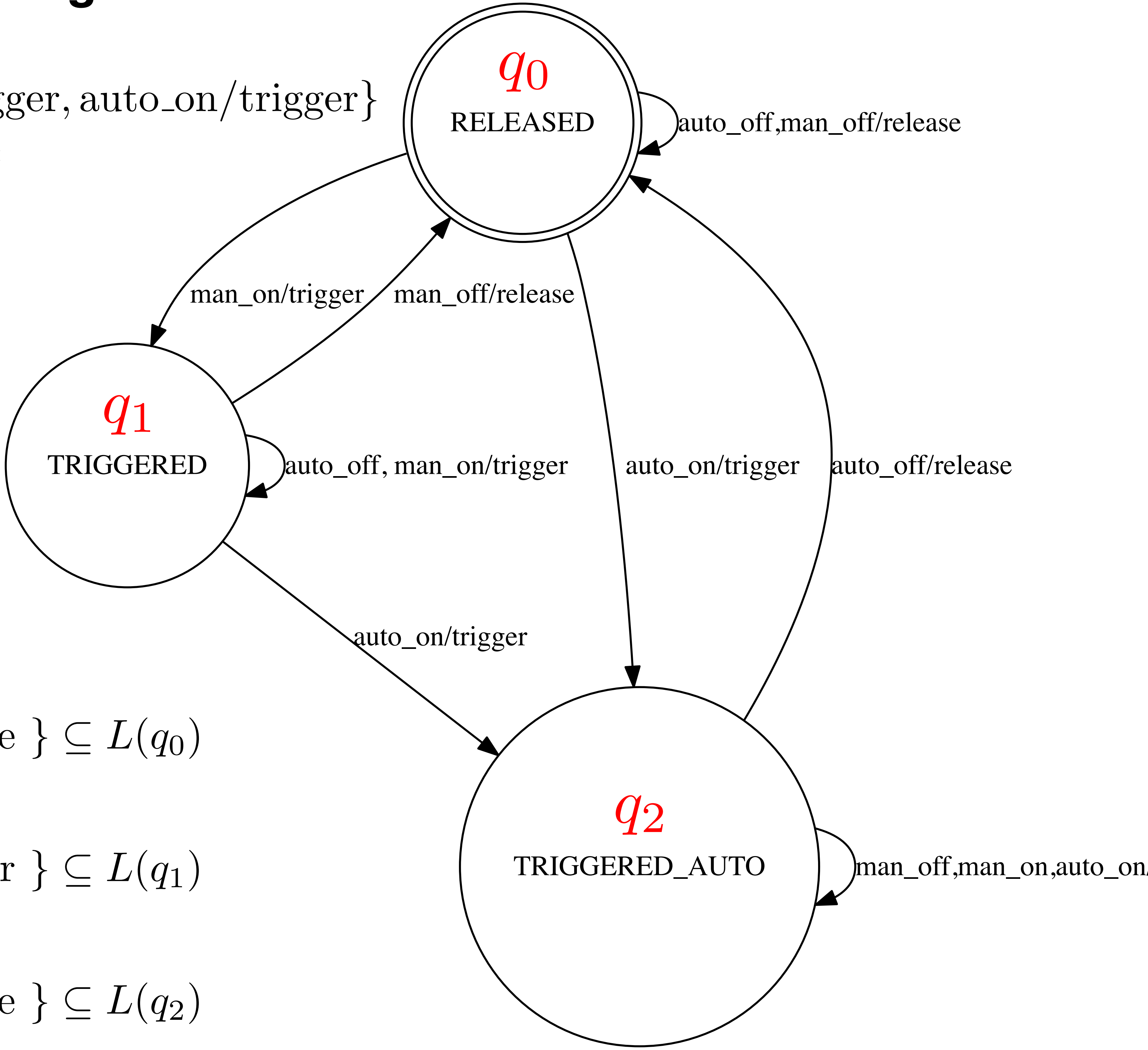
$$q_0 \xrightarrow{\text{auto\_off}/\text{release}} q_0$$
$$q_1 \xrightarrow{\text{auto\_off}/\text{trigger}} q_1$$
$$q_2 \xrightarrow{\text{auto\_off}/\text{release}} q_0$$
$$q_0 \xrightarrow{\text{man\_off}/\text{release}} q_0$$
$$q_2 \xrightarrow{\text{man\_off}/\text{trigger}} q_2$$



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

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 $\mathcal{F}(M, I, O, \sim, \mathcal{D}_m), n = |Q|$

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

Step 1.
$M = (Q, q_0, I, O, h), M' = (Q', q_0', I, O, h') \in Sig.$
Suppose $\exists f : Q \to Q'$:

- $f(q_0) = q_0'$
- $(q_1, x, y, q_2) \in h \Rightarrow (f(q_1), x, y, f(q_2)) \in h'$

Then $L(M) = L(M')$.

Step 1.
$M = (Q, q_0, I, O, h), M' = (Q', q'_0, I, O, h') \in Sig.$
Suppose $\exists f : Q \to Q'$:

homomorphism

- $f(q_0) = q'_0$
- $(q_1, x, y, q_2) \in h \Rightarrow (f(q_1), x, y, f(q_2)) \in h'$

Then $L(M) = L(M')$.

Proof of step 1.

Suppose $f : Q \to Q', f(q_0) = q'_0$, is a homomorphism.
Then $\forall x_1 \dots x_k/y_1 \dots y_k \in L(M)$, $\exists q_i \in Q$, $i = 1, \dots, k$:

$q_0 \stackrel{x_1/y_1}{\to} q_1 \stackrel{x_2/y_2}{\to} \dots \stackrel{x_k/y_k}{\to} q_k$ and

$q'_0 = f(q_0) \stackrel{x_1/y_1}{\to} f(q_1) \stackrel{x_2/y_2}{\to} \dots \stackrel{x_k/y_k}{\to} f(q_k)$

Then $x_1 \dots x_k/y_1 \dots y_k \in L(M')$ and $L(M) \subseteq L(M')$.
Since M, M' are input complete and deterministic,
we have $L(M') = L(M)$.

Step 2.

$M' = (Q', q'_0, I, O, h') \in \mathcal{D}_m$.

Suppose M' pass $V \oplus \bigcup_{i=0}^{m-n} \Sigma^i \oplus W$

Then $V \oplus \bigcup_{i=0}^{m-n} \Sigma^i$ is a state cover of M' and

$V \oplus \bigcup_{i=0}^{m-n+1} \Sigma^i$ is a transition cover of M'.

Proof of step 2.

1. $M' \underline{\textbf{pass}} (V. \bigcup_{i=0}^{m-n+1} \Sigma^i.W) \cap L(M)$
   $\Rightarrow (V. \bigcup_{i=0}^{m-n+1} \Sigma^i.W) \cap L(M) = (V. \bigcup_{i=0}^{m-n+1} \Sigma^i.W) \cap L(M')$
   $\Rightarrow V \oplus \bigcup_{i=0}^{m-n+1} \Sigma^i \oplus W = V \oplus' \bigcup_{i=0}^{m-n+1} \Sigma^i \oplus' W$

2. $|\{q_0'\text{-after-}\pi) \mid \pi \in V\}| \geq n$

3. $\{q_0'\text{-after-}\pi \mid \pi \in V \oplus \bigcup_{i=0}^{m-n} \Sigma^i\} = Q'$

Proof of $|\{\{\mathbf{q_0'}\text{-after-}\pi \mid \pi \in \mathbf{V}\}| \geq \mathbf{n}$

Let

- $\pi_0 = \varepsilon$
- $\{\pi_i \mid i = 0, \ldots, n-1\} \subseteq V$ state cover of M
- $q_0 \xrightarrow{\pi_i} q_i$
- $q_0' \xrightarrow{\pi_i} q_i'$

Then $i \neq j \Rightarrow q'_i \neq q'_j$:

1. $q_i \neq q_j \Rightarrow \exists \tau_i \neq \tau_j \in W : \tau_{i_I} = \tau_{j_I}, \tau_i \in L(q_i), \tau_j \in L(q_j)$

   (W is a characterisation set of M)

2. $\tau_i \in L'(q'_i), \tau_j \in L'(q'_j) \wedge \tau_{i_I} = \tau_{j_I}$

3. M' is deterministic $\Rightarrow L'(q'_i) \neq L'(q'_j) \Rightarrow q'_i \neq q'_j$

Hence $n = |\{q'_0, \ldots, q'_{n-1}\}| \leq |\{q'_0\text{-after-}\pi \mid \pi \in V\}|$

Step 3.

$M' = (Q', q'_0, I, O, h') \in \mathcal{D}_m$.

Suppose M' pass $V \oplus \bigcup_{i=0}^{m-n+1} \Sigma^i \oplus W$.

Then $\exists f : M' \to M$ homomorphism.

Proof of step 3.
Let

- $Q' =: \{q'_0, \ldots, q'_{m-1}\}$
- $\{\varepsilon\} \cup \{\pi_i \mid i = 1, \ldots m-1\} \subseteq V \oplus \bigcup_{i=0}^{m-n} \Sigma^i$, a state cover of M'.
- $f : Q' \to Q$, $f(q'_0) = q_0$ and $f(q'_i) = q_0$-after-π_i.

Proof of step 3.
Let

- $Q' =: \{q'_0, \ldots, q'_{m-1}\}$
- $\{\varepsilon\} \cup \{\pi_i \mid i = 1, \ldots m-1\} \subseteq V \oplus \bigcup_{i=0}^{m-n} \Sigma^i$, a state cover of M'.
- $f : Q' \to Q$, $f(q'_0) = q_0$ and $f(q'_i) = q_0$-after-π_i.

Then $f(q)$ is the unique state of M such that
$\pi \in L(q') \Leftrightarrow \pi \in L(f(q)), \forall \pi \in W$

Proof of step 3.
Let

- $Q' =: \{q'_0, \ldots, q'_{m-1}\}$
- $\{\varepsilon\} \cup \{\pi_i \mid i = 1, \ldots m-1\} \subseteq V \oplus \bigcup_{i=0}^{m-n} \Sigma^i$, a state cover of M'.
- $f : Q' \to Q$, $f(q'_0) = q_0$ and $f(q'_i) = q_0$-after-π_i.

Then $f(q)$ is the unique state of M such that
$\pi \in L(q') \Leftrightarrow \pi \in L(f(q)), \forall \pi \in W$

Then

$$x/y \in L(q'_i) \Leftrightarrow x/y \in L(q_i), \qquad \forall x/y \in \Sigma$$

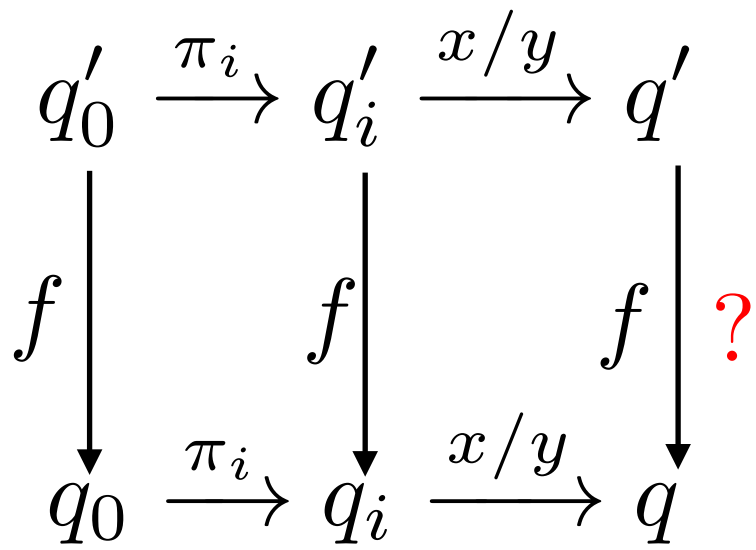
$$[q_0, q'_0\ \underline{\mathrm{pass}}(V \oplus \bigcup_{i=0}^{m-n} \Sigma^i) \oplus \Sigma]$$

$$\tau \in L(q'_i) \Leftrightarrow \tau \in L(q_i), \qquad \forall \tau \in W$$

$$[q_0, q'_0\ \underline{\mathrm{pass}}(V \oplus \bigcup_{i=0}^{m-n} \Sigma^i) \oplus W]$$

$$(x/y).\tau \in L(q'_i) \Leftrightarrow (x/y).\tau \in L(q_i), \qquad \forall x/y \in \Sigma, \tau \in W$$

$$[q_0, q'_0\ \underline{\mathrm{pass}}(V \oplus \bigcup_{i=0}^{m-n} \Sigma^i) \oplus \Sigma \oplus W]$$



$$\pi \in L(q') \Leftrightarrow \pi \in L(q), \forall \pi \in W$$
$$\Rightarrow f(q') = q$$

Wp-Method

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

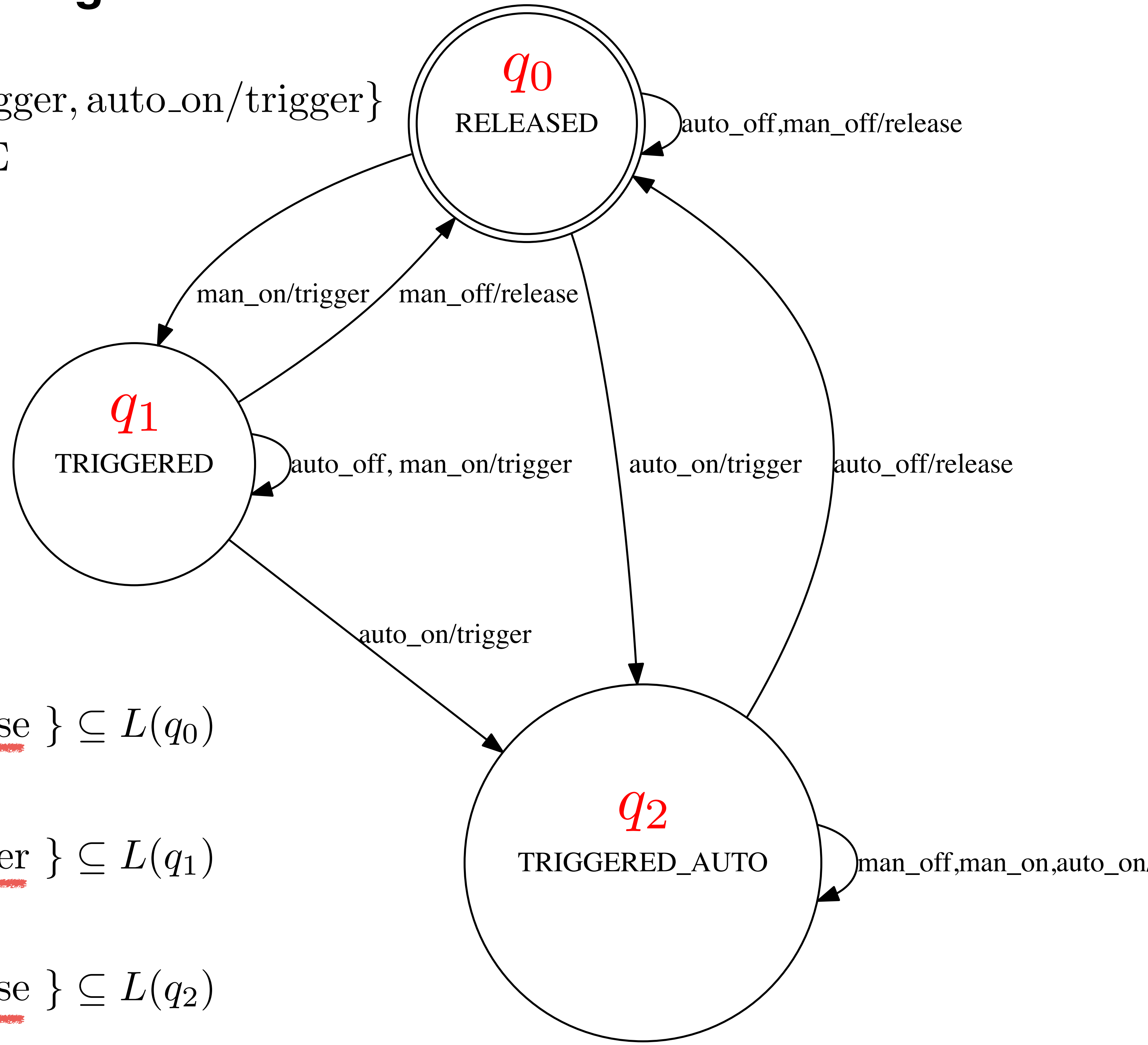
$\mathcal{D}_m = \{M' = (Q', q'_0, I, O, h') \subseteq Sig \mid |Q'| \leq m\}$

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 $P = V \cup V \oplus \Sigma$

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



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

W0

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

W1

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

W2

- 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

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

$U \oplus \{W_0, \ldots, W_{n-1}\} =$
$\bigcup_{\pi \in U \wedge q_i = q_0\text{-after-}\pi} \{\pi\}.W_i$

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.