Formal Methods for Interactive Systems

Part 10 — Reverse Engineering with Matrix Algebra

Antonio Cerone

United Nations University

International Institute for Software Technology

Macau SAR China

email: `antonio@iist.unu.edu`

web: `www.iist.unu.edu`

Reverse Engineering

Define a model of an existing device

Reverse Engineering

Define a model of an existing device

Danger: model accuracy relies on accuracy of the reverse engineering

Reverse Engineering

Define a model of an existing device

Danger: model accuracy relies on accuracy of the reverse engineering

$\Longrightarrow$ manifacturers do not provide formal specifications

Reverse Engineering

Define a model of an existing device

Danger: model accuracy relies on accuracy of the reverse engineering

$\Longrightarrow$ manifacturers do not provide formal specifications

$\Longrightarrow$ formal specifications must be reconstructed

Reverse Engineering Process

Reconstruct formal specification from

- checking interaction results against the real device

Reverse Engineering Process

Reconstruct formal specification from

- checking interaction results against the real device
- observing user-machine interaction in the real operating environemt

Reverse Engineering Process

Reconstruct formal specification from

- checking interaction results against the real device
- observing user-machine interaction in the real operating environemt
- interviewing real users and manifacturer

Reverse Engineering Process

Reconstruct formal specification from

- checking interaction results against the real device
- observing user-machine interaction in the real operating environemt
- interviewing real users and manifacturer
- checking documentation (e.g., user manual)

RE Advantages

- model of a real system $\Longrightarrow$ no need to specify it in a paper)

RE Advantages

- model of a real system $\Longrightarrow$ no need to specify it in a paper)
- results can be verified against the device by the reader

RE Advantages

- model of a real system $\Longrightarrow$ no need to specify it in a paper)
- results can be verified against the device by the reader
- case studies are more credible $\Longleftarrow$ not tailored for the used modelling / verification approach

Matrix Algebra

to describe interaction human-machine

Matrix Algebra

to describe interaction human-machine

Matrices

- are standard mathematical objects with a long history

Matrix Algebra

to describe interaction human-machine

Matrices

- are standard mathematical objects
  with a long history
  $\Longrightarrow$ no need to introduce yet another notation
- are easy to calculate

Matrix Algebra

to describe interaction human-machine

Matrices

- are standard mathematical objects
  with a long history
  $\Longrightarrow$ no need to introduce yet another notation
- are easy to calculate
- semantics reflects human way of thinking

Matrix Algebra

to describe interaction human-machine

Matrices

- are standard mathematical objects
  with a long history
  $\Longrightarrow$ no need to introduce yet another notation
- are easy to calculate
- semantics reflects human way of thinking
- have

Matrix Algebra

to describe interaction human-machine

Matrices

- are standard mathematical objects
  with a long history
  $\Longrightarrow$ no need to introduce yet another notation
- are easy to calculate
- semantics reflects human way of thinking
- have
  - structure $\Longrightarrow$ easy modelling

Matrix Algebra

to describe interaction human-machine

Matrices

- are standard mathematical objects
  with a long history
  $\Longrightarrow$ no need to introduce yet another notation
- are easy to calculate
- semantics reflects human way of thinking
- have
  - structure $\Longrightarrow$ easy modelling
  - properties $\Longrightarrow$ verification tool

Matrix Algebra

to describe interaction human-machine

Matrices

- are standard mathematical objects
  with a long history
  $\Longrightarrow$ no need to introduce yet another notation
- are easy to calculate
- semantics reflects human way of thinking
- have
  - structure $\Longrightarrow$ easy modelling
  - properties $\Longrightarrow$ verification tool

[Thimbleby 04]

Example: Pushbutton Switch

A ligh bulb controlled by a pushbutton switch

Example: Pushbutton Switch

A ligh bulb controlled by a pushbutton switch

Pressing the switch alternatively turns the light on or off.

Example: Pushbutton Switch

A ligh bulb controlled by a pushbutton switch

Pressing the switch alternatively turns the light on or off.

- 2 states: ON and OFF
- 1 operation: push

Example: Pushbutton Switch

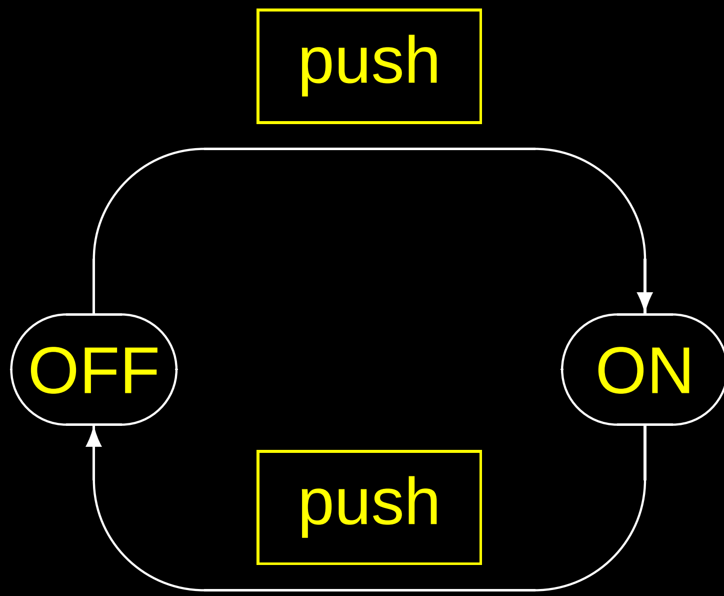
A ligh bulb controlled by a pushbutton switch

Pressing the switch alternatively turns the light on or off.

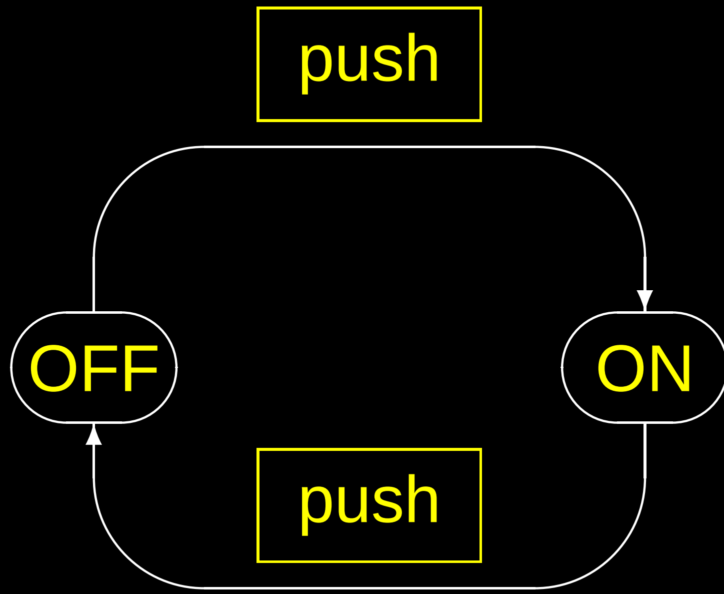
- 2 states: ON and OFF
- 1 operation: push

[Thimbleby 04]

Model of Pushbutton Switch

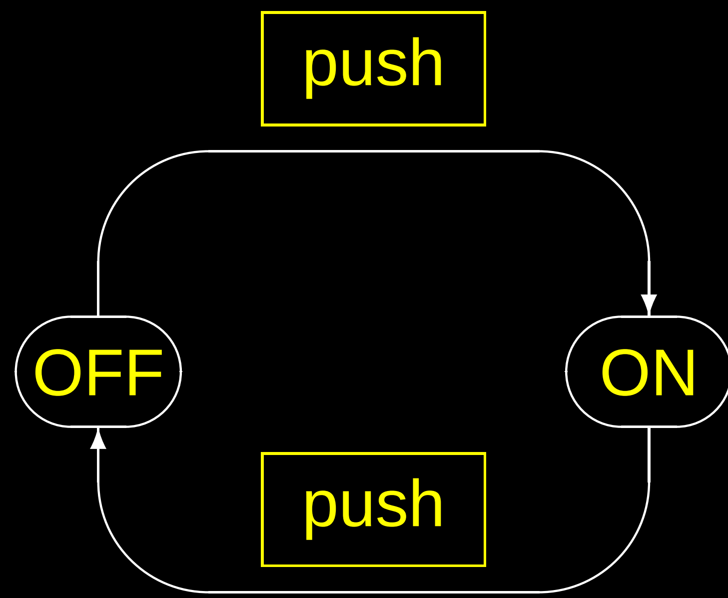


Model of Pushbutton Switch



$$\text{OFF} = \begin{bmatrix} 0 \, , \, 1 \end{bmatrix} \qquad \text{ON} = \begin{bmatrix} 1 \, , \, 0 \end{bmatrix}$$

Model of Pushbutton Switch



$$\text{OFF} = \begin{bmatrix} 0 \;,\; 1 \end{bmatrix} \qquad \text{ON} = \begin{bmatrix} 1 \;,\; 0 \end{bmatrix}$$

$$\boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pushbutton Switch Transition

$$\mathrm{OFF} = \begin{bmatrix} 0 \; , \; 1 \end{bmatrix} \qquad \mathrm{ON} = \begin{bmatrix} 1 \; , \; 0 \end{bmatrix}$$

$$\boxed{\mathrm{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pushbutton Switch Transition

OFF

$$\text{OFF} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} \qquad \text{ON} = \begin{bmatrix} 1 \ , \ 0 \end{bmatrix}$$

$$\boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pushbutton Switch Transition

OFF $\boxed{\text{push}}$

$$\text{OFF} = \begin{bmatrix} 0 \;,\; 1 \end{bmatrix} \qquad \text{ON} = \begin{bmatrix} 1 \;,\; 0 \end{bmatrix}$$

$$\boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pushbutton Switch Transition

$$\text{OFF}\ \boxed{\text{push}} = \begin{bmatrix} 0\ ,\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\text{OFF} = \begin{bmatrix} 0\ ,\ 1 \end{bmatrix} \qquad \text{ON} = \begin{bmatrix} 1\ ,\ 0 \end{bmatrix}$$

$$\boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pushbutton Switch Transition

$$\text{OFF}\ \boxed{\text{push}} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 \ , \quad \end{bmatrix}$$

$$\text{OFF} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} \qquad \text{ON} = \begin{bmatrix} 1 \ , \ 0 \end{bmatrix}$$

$$\boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pushbutton Switch Transition

$$\text{OFF}\ \boxed{\text{push}} = \begin{bmatrix} 0 , 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 , 0 \end{bmatrix}$$

$$\text{OFF} = \begin{bmatrix} 0 , 1 \end{bmatrix} \qquad \text{ON} = \begin{bmatrix} 1 , 0 \end{bmatrix}$$

$$\boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Pushbutton Switch Transition

$$\text{OFF}\ \boxed{\text{push}} = \begin{bmatrix} 0 \,,\, 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 \,,\, 0 \end{bmatrix} = \text{ON}$$

$$\text{OFF} = \begin{bmatrix} 0 \,,\, 1 \end{bmatrix} \qquad \text{ON} = \begin{bmatrix} 1 \,,\, 0 \end{bmatrix}$$

$$\boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Switch Properties

$$\text{OFF}\ \boxed{\text{push}} = \text{ON}$$

Switch Properties

OFF $\boxed{\text{push}}$ = ON

ON $\boxed{\text{push}}$ = OFF

Switch Properties

OFF $\boxed{\text{push}}$ = ON

ON $\boxed{\text{push}}$ = OFF

$\boxed{\text{push}}$ $\boxed{\text{push}}$

Switch Properties

$$\text{OFF}\ \boxed{\text{push}} = \text{ON} \qquad\qquad \text{ON}\ \boxed{\text{push}} = \text{OFF}$$

$$\boxed{\text{push}}\ \boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Switch Properties

$$\text{OFF}\ \boxed{\text{push}} = \text{ON} \qquad \text{ON}\ \boxed{\text{push}} = \text{OFF}$$

$$\boxed{\text{push}}\ \boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Switch Properties

$$\text{OFF}\ \boxed{\text{push}} = \text{ON} \qquad \text{ON}\ \boxed{\text{push}} = \text{OFF}$$

$$\boxed{\text{push}}\ \boxed{\text{push}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Switch Safety

$$\text{OFF} \; \boxed{\quad \text{x} \quad} = \begin{bmatrix} 0 \;,\; 1 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \;,\; 1 \end{bmatrix} = \text{OFF}$$

Switch Safety

$$\text{OFF} \ \boxed{\quad x \quad} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} = \text{OFF}$$

$$\text{ON} \ \boxed{\quad x \quad} = \begin{bmatrix} 1 \ , \ 0 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} = \text{OFF}$$

Switch Safety

$$\text{OFF}\ \boxed{\text{x}} = \begin{bmatrix} 0 , 1 \end{bmatrix} \begin{bmatrix} \text{x}_{1,1} & \text{x}_{1,2} \\ \text{x}_{2,1} & \text{x}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 , 1 \end{bmatrix} = \text{OFF}$$

$$\text{ON}\ \boxed{\text{x}} = \begin{bmatrix} 1 , 0 \end{bmatrix} \begin{bmatrix} \text{x}_{1,1} & \text{x}_{1,2} \\ \text{x}_{2,1} & \text{x}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 , 1 \end{bmatrix} = \text{OFF}$$

$\text{x}_{2,1} = 0 \quad \text{x}_{2,2} = 1$

Switch Safety

$$\text{OFF} \; \boxed{\quad \mathsf{x} \quad} = \begin{bmatrix} 0 \; , \; 1 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \; , \; 1 \end{bmatrix} = \text{OFF}$$

$$\text{ON} \; \boxed{\quad \mathsf{x} \quad} = \begin{bmatrix} 1 \; , \; 0 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \; , \; 1 \end{bmatrix} = \text{OFF}$$

$$x_{1,1} = 0 \quad x_{1,2} = 1$$
$$x_{2,1} = 0 \quad x_{2,2} = 1$$

Switch Safety

$$\text{OFF} \; \boxed{\quad \mathsf{x} \quad} = \begin{bmatrix} 0 \;, & 1 \end{bmatrix} \begin{bmatrix} \mathsf{x}_{1,1} & \mathsf{x}_{1,2} \\ \mathsf{x}_{2,1} & \mathsf{x}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \;, & 1 \end{bmatrix} = \text{OFF}$$

$$\text{ON} \; \boxed{\quad \mathsf{x} \quad} = \begin{bmatrix} 1 \;, & 0 \end{bmatrix} \begin{bmatrix} \mathsf{x}_{1,1} & \mathsf{x}_{1,2} \\ \mathsf{x}_{2,1} & \mathsf{x}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \;, & 1 \end{bmatrix} = \text{OFF}$$

$$\begin{matrix} \mathsf{x}_{1,1} = 0 & \mathsf{x}_{1,2} = 1 \\ \mathsf{x}_{2,1} = 0 & \mathsf{x}_{2,2} = 1 \end{matrix} \quad \Longrightarrow \quad \boxed{\quad \mathsf{x} \quad} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

Switch Safety

$$\text{OFF} \ \boxed{\mathsf{x}} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} \begin{bmatrix} \mathsf{x}_{1,1} & \mathsf{x}_{1,2} \\ \mathsf{x}_{2,1} & \mathsf{x}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} = \text{OFF}$$

$$\text{ON} \ \boxed{\mathsf{x}} = \begin{bmatrix} 1 \ , \ 0 \end{bmatrix} \begin{bmatrix} \mathsf{x}_{1,1} & \mathsf{x}_{1,2} \\ \mathsf{x}_{2,1} & \mathsf{x}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} = \text{OFF}$$

$$\begin{matrix} \mathsf{x}_{1,1} = 0 & \mathsf{x}_{1,2} = 1 \\ \mathsf{x}_{2,1} = 0 & \mathsf{x}_{2,2} = 1 \end{matrix} \quad \Longrightarrow \quad \boxed{\mathsf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$$\exists\, n \quad \text{such that} \quad \boxed{\text{push}}^{\,n} = \boxed{\mathsf{x}} \ ?$$

Switch Safety

$$\text{OFF}\ \boxed{\text{x}} = \begin{bmatrix} 0 , 1 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 0 , 1 \end{bmatrix} = \text{OFF}$$

$$\text{ON}\ \boxed{\text{x}} = \begin{bmatrix} 1 , 0 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 0 , 1 \end{bmatrix} = \text{OFF}$$

$$\begin{matrix} x_{1,1} = 0 & x_{1,2} = 1 \\ x_{2,1} = 0 & x_{2,2} = 1 \end{matrix} \implies \boxed{\text{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

$\exists\, n$ such that $\boxed{\text{push}}^{n} = \boxed{\text{x}}$?

No because $\boxed{\text{push}}^{2} = I$

Safer System

$$\text{OFF}\ \boxed{\quad \text{x} \quad} = \begin{bmatrix} 0 & , & 1 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & , & 0 \end{bmatrix} = \text{ON}$$

$$\text{ON}\ \boxed{\quad \text{y} \quad} = \begin{bmatrix} 1 & , & 0 \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{1,2} \\ y_{2,1} & y_{2,2} \end{bmatrix} = \begin{bmatrix} 0 & , & 1 \end{bmatrix} = \text{OFF}$$

Safer System

$$\text{OFF} \ \boxed{\quad x \quad} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 1 \ , \ 0 \end{bmatrix} = \ \text{ON}$$

$$\text{ON} \ \boxed{\quad y \quad} = \begin{bmatrix} 1 \ , \ 0 \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{1,2} \\ y_{2,1} & y_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} = \ \text{OFF}$$

$$x_{2,1} = 1 \quad x_{2,2} = 0 \qquad \Longrightarrow \qquad \boxed{\quad x \quad} = \begin{bmatrix} - & - \\ 1 & 0 \end{bmatrix}$$

Safer System

$$\text{OFF} \; \boxed{\quad \mathsf{x} \quad} = \begin{bmatrix} 0 , 1 \end{bmatrix} \begin{bmatrix} \mathsf{x}_{1,1} & \mathsf{x}_{1,2} \\ \mathsf{x}_{2,1} & \mathsf{x}_{2,2} \end{bmatrix} = \begin{bmatrix} 1 , 0 \end{bmatrix} = \text{ON}$$

$$\text{ON} \; \boxed{\quad \mathsf{y} \quad} = \begin{bmatrix} 1 , 0 \end{bmatrix} \begin{bmatrix} \mathsf{y}_{1,1} & \mathsf{y}_{1,2} \\ \mathsf{y}_{2,1} & \mathsf{y}_{2,2} \end{bmatrix} = \begin{bmatrix} 0 , 1 \end{bmatrix} = \text{OFF}$$

$$\mathsf{x}_{2,1} = 1 \quad \mathsf{x}_{2,2} = 0 \quad \Longrightarrow \quad \boxed{\quad \mathsf{x} \quad} = \begin{bmatrix} - & - \\ 1 & 0 \end{bmatrix}$$

$$\mathsf{y}_{1,1} = 0 \quad \mathsf{y}_{1,2} = 1 \quad \Longrightarrow \quad \boxed{\quad \mathsf{y} \quad} = \begin{bmatrix} 0 & 1 \\ - & - \end{bmatrix}$$

Safer System

$$\text{OFF}\ \boxed{\quad \text{x} \quad} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} \\ x_{2,1} & x_{2,2} \end{bmatrix} = \begin{bmatrix} 1 \ , \ 0 \end{bmatrix} = \text{ON}$$

$$\text{ON}\ \boxed{\quad \text{y} \quad} = \begin{bmatrix} 1 \ , \ 0 \end{bmatrix} \begin{bmatrix} y_{1,1} & y_{1,2} \\ y_{2,1} & y_{2,2} \end{bmatrix} = \begin{bmatrix} 0 \ , \ 1 \end{bmatrix} = \text{OFF}$$

$$x_{2,1} = 1 \quad x_{2,2} = 0 \quad \Longrightarrow \quad \boxed{\quad \text{x} \quad} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} = \boxed{\quad \text{on} \quad}$$

$$y_{1,1} = 0 \quad y_{1,2} = 1 \quad \Longrightarrow \quad \boxed{\quad \text{y} \quad} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \boxed{\quad \text{off} \quad}$$

Two-position Switch

A ligh bulb controlled by a two-position switch

Two-position Switch

A ligh bulb controlled by a two-position switch

One switch position (on) turns the light on; the other (off) turns the light off.

Two-position Switch

A ligh bulb controlled by a two-position switch

One switch position (on) turns the light on; the other (off) turns the light off.

Properties

- The system is closed

Two-position Switch

A ligh bulb controlled by a two-position switch

One switch position (on) turns the light on; the other (off) turns the light off.

Properties

- The system is closed
- any sequence of actions is equivalent to the last actions the user does

Remarks

- matrix algebra can be used to find interesting properties

Remarks

- matrix algebra can be used to find interesting properties through
- direct manipulation of matrices (operations) rather than vectors (states)

Remarks

- matrix algebra can be used to find interesting properties through
- direct manipulation of matrices (operations) rather than vectors (states)
  $\Longleftarrow$ number of operations much smaller than number of states

Remarks

- matrix algebra can be used to find interesting properties through
- direct manipulation of matrices (operations) rather than vectors (states)
  $\Longleftarrow$ number of operations much smaller than number of states
- the methodology is
  - scalable

Remarks

- matrix algebra can be used to find interesting properties through
- direct manipulation of matrices (operations) rather than vectors (states)
  $\Longleftarrow$ number of operations much smaller than number of states
- the methodology is
  - scalable
  - mechanisable [Gow and Thimbleby 04]

Calculator

- Model: CASIO HL-820LC

Calculator

- Model: CASIO HL-820LC
- State

Calculator

- Model: CASIO HL-820LC
- State
  - d $=$ display contents

Calculator

- Model: CASIO HL-820LC
- State
  - d $=$ display contents
  - m $=$ memory contents

Calculator

- Model: CASIO HL-820LC
- State
  - d $=$ display contents
  - m $=$ memory contents
  - state representation: vector s $=$ [d,m]

Calculator

- Model: CASIO HL-820LC
- State
  - d $=$ display contents
  - m $=$ memory contents
  - state representation: vector s $=$ [d,m]
- Operations: binary 2×2 matrices

Calculator Functions

- $\boxed{\text{M}+}$ add display to memory

Calculator Functions

- $\boxed{\text{M}+}$ add display to memory
- $\boxed{\text{M}-}$ subtract display from memory

Calculator Functions

- $\boxed{\text{M}+}$ add display to memory
- $\boxed{\text{M}-}$ subtract display from memory
- $\boxed{\text{AC}}$ clear display

Calculator Functions

- $\boxed{\text{M}+}$ add display to memory
- $\boxed{\text{M}-}$ subtract display from memory
- $\boxed{\text{AC}}$ clear display
- $\boxed{\text{MRC}}$ recall memory

Calculator Functions

- $\boxed{\text{M}+}$ add display to memory
- $\boxed{\text{M}-}$ subtract display from memory
- $\boxed{\text{AC}}$ clear display
- $\boxed{\text{MRC}}$ recall memory
- $\boxed{\text{MRC}}$ $\boxed{\text{MRC}}$ recall and clear memory

MRC

What is the precise semantics of MRC?

- pressed once

MRC

What is the precise semantics of MRC?

- pressed once
  - if $m = 0$ then no effect

MRC

What is the precise semantics of MRC?

- pressed once
  - if $m = 0$ then no effect
  - if $m \neq 0$ then $d := m$

MRC

What is the precise semantics of MRC?

- pressed once
  - if $m = 0$ then no effect
  - if $m \neq 0$ then $d := m$
- pressed twice

MRC

What is the precise semantics of MRC?

- pressed once
  - if $m = 0$ then no effect
  - if $m \neq 0$ then $d := m$
- pressed twice
  - $m := 0$

Remarks on MRC

- $\boxed{\text{MRC}}$ $\boxed{\text{MRC}}$ is not the same as the product $\boxed{\text{MRC}} \cdot \boxed{\text{MRC}}$

Remarks on MRC

- $\boxed{\text{MRC}}\ \boxed{\text{MRC}}$ is not the same as the product $\boxed{\text{MRC}} \cdot \boxed{\text{MRC}}$
- Semantics of pressed once depends on previous content of memory

Remarks on MRC

- $\boxed{\text{MRC}}\ \boxed{\text{MRC}}$ is not the same as the product $\boxed{\text{MRC}} \cdot \boxed{\text{MRC}}$
- Semantics of pressed once depends on previous content of memory
- Is $\boxed{\text{MRC}}(\ \boxed{\text{MRC}}\ \boxed{\text{MRC}}\ ) = (\ \boxed{\text{MRC}}\ \boxed{\text{MRC}}\ )\ \boxed{\text{MRC}}$?

Remarks on MRC

- $\boxed{\text{MRC}}\ \boxed{\text{MRC}}$ is not the same as the product $\boxed{\text{MRC}} \cdot \boxed{\text{MRC}}$
- Semantics of pressed once depends on previous content of memory
- Is $\boxed{\text{MRC}}(\ \boxed{\text{MRC}}\ \boxed{\text{MRC}}\ ) = (\ \boxed{\text{MRC}}\ \boxed{\text{MRC}}\ )\ \boxed{\text{MRC}}$?
- Calculator:
  $\boxed{\text{MRC}}\ (\ \boxed{\text{MRC}}\ \boxed{\text{MRC}}\ ) = \boxed{\text{MRC}}\ \boxed{\text{MRC}}$

Bad Design?

- Do we really need to give such a special function to $\boxed{\text{MRC}}$ $\boxed{\text{MRC}}$?

Bad Design?

- Do we really need to give such a special function to $\boxed{\text{MRC}}\ \boxed{\text{MRC}}$?
- No!
  $\boxed{\text{MRC}}\ \boxed{\text{MRC}} = \boxed{\text{MRC}}\ \boxed{\text{M}-}$

Bad Design?

- Do we really need to give such a special function to $\boxed{\text{MRC}}\ \boxed{\text{MRC}}$?
- No!
  $\boxed{\text{MRC}}\ \boxed{\text{MRC}} = \boxed{\text{MRC}}\ \boxed{\text{M}-}$
- $\boxed{\text{MRC}}\ \boxed{\text{MRC}}$ is a bad design choice!

Bad Design?

- Do we really need to give such a special function to $\boxed{\text{MRC}}$ $\boxed{\text{MRC}}$?
- No!
  $\boxed{\text{MRC}}$ $\boxed{\text{MRC}}$ $=$ $\boxed{\text{MRC}}$ $\boxed{\text{M}-}$
- $\boxed{\text{MRC}}$ $\boxed{\text{MRC}}$ is a bad design choice!

[Thimbleby 00]

Safety-critical

- Is a calculator safety-critical?

Safety-critical

- Is a calculator safety-critical?
- May a calculator-like interface be safety-critical?

Safety-critical

- Is a calculator safety-critical?
- May a calculator-like interface be safety-critical?
- Yes!
  Syringe pump have calculator-like interfaces

Examinations

Seminar 6 — Usable Calculator

Topic: Calculator Interfaces

Towards a truly usable calculator

- Harold Thimbleby
  Computer Algebra for User Interface Design, 2004
- Harold Thimbleby
  A Novel Pen-based Calculator and its Evaluation, 2004
- Will Thimbleby and Harold Thimbleby
  A Novel Gesture-based Calculator and its Design Principles, 2005
- Websites on
  - Computer Algebra: http://www.cs.swan.ac.uk/~csharold/CA/
  - Calculator: http://www.cs.swan.ac.uk/calculators/

References

Reverse Engineering

- [Thimbleby 04]: Reverse Engineering and Matrix Algebra
- [Gow and Thimbleby 2004]: Matrix Algebra and Tools
- [Thimbleby 00]: Reverse Engineering

[Thimbleby 04]

Harold Thimbleby.
User interface design with matrix algebra.
ACM Transactions on Computer-Human Interaction, Vol. 11, No. 2, 2004, pages 181–236.

About:

- Reverse Engineering
- Matrix Algebra

[Gow and Thimbleby 04]

Jeremy Gow and Harold Thimbleby.
MAUI: An Interface Design Tool Based on Matrix Algebra.
In *Proceedings of CADUI 2004*, Springer, 2004, pages 81–94.

About:

- Tools

[Thimbleby 00]

Harold Thimbleby.
Calculators are needlessly bad.
International Journal of Computer-Human Studies, Vol. 52, No. 6, 2000, pages 1031–1069.

About:

- Reverse Engineering of Calculatos