Why Didn’t They Work?

• For compositions to work, machine must act same way regardless of what precedes low level input (high, low, nothing)

• *dog* does not meet this criterion
  – If first input is *stop_count*, *dog* emits 0
  – If high level input precedes *stop_count*, *dog* emits 0 or 1
State Machine Model

- 2-bit machine, levels *High*, *Low*, meeting 4 properties:
  1. For every input $i_k$, state $\sigma_j$, there is an element $c_m \in C^*$ such that $T^*(c_m, \sigma_j) = \sigma_n$, where $\sigma_n \neq \sigma_j$
     - $T^*$ is total function, inputs and commands always move system to a different state

Property 2

- There is an equivalence relation $\equiv$ such that:
  - If system in state $\sigma_j$ and high level sequence of inputs causes transition from $\sigma_i$ to $\sigma_j$, then $\sigma_i \equiv \sigma_j$
  - If $\sigma_i \equiv \sigma_j$ and low level sequence of inputs $i_1, \ldots, i_n$ causes system in state $\sigma_i$ to transition to $\sigma_j'$, then there is a state $\sigma_j'$ such that $\sigma_i' \equiv \sigma_j'$ and the inputs $i_1, \ldots, i_n$ cause system in state $\sigma_j$ to transition to $\sigma_j'$
- $\equiv$ holds if low level projections of both states are same
Property 3

• Let $\sigma_i \equiv \sigma_j$. If high level sequence of outputs $\sigma_1, \ldots, \sigma_n$ indicate system in state $\sigma_i$ transitioned to state $\sigma_i'$, then for some state $\sigma_j'$ with $\sigma_j' \equiv \sigma_i'$, high level sequence of outputs $\sigma_1', \ldots, \sigma_m'$ indicates system in $\sigma_j$ transitioned to $\sigma_j'$
  – High level outputs do not indicate changes in low level projection of states

Property 4

• Let $\sigma_i \equiv \sigma_j$, let $c$, $d$ be high level output sequences, $e$ a low level output. If $ced$ indicates system in state $\sigma_i$ transitions to $\sigma_i'$, then there are high level output sequences $c'$ and $d'$ and state $\sigma_j'$ such that $c'ed'$ indicates system in state $\sigma_j$ transitions to state $\sigma_j'$
  – Intermingled low level, high level outputs cause changes in low level state reflecting low level outputs only
Restrictiveness

- System is *restrictive* if it meets the preceding 4 properties

Composition

- Intuition: by 3 and 4, high level output followed by low level output has same effect as low level input, so composition of restrictive systems should be restrictive
Composite System

- System $M_1$’s outputs are $M_2$’s inputs
- $\mu_{1i}$, $\mu_{2i}$ states of $M_1$, $M_2$
- States of composite system pairs of $M_1$, $M_2$ states ($\mu_{1i}$, $\mu_{2i}$)
- $e$ event causing transition
- $e$ causes transition from state $(\mu_{1a}, \mu_{2a})$ to state $(\mu_{1b}, \mu_{2b})$ if any of 3 conditions hold

Conditions

1. $M_1$ in state $\mu_{1a}$ and $e$ occurs, $M_1$ transitions to $\mu_{1b}$; $e$ not an event for $M_2$; and $\mu_{2a} = \mu_{2b}$
2. $M_2$ in state $\mu_{2a}$ and $e$ occurs, $M_2$ transitions to $\mu_{2b}$; $e$ not an event for $M_1$; and $\mu_{1a} = \mu_{1b}$
3. $M_1$ in state $\mu_{1a}$ and $e$ occurs, $M_1$ transitions to $\mu_{1b}$; $M_2$ in state $\mu_{2a}$ and $e$ occurs, $M_2$ transitions to $\mu_{2b}$; $e$ is input to one machine, and output from other
Intuition

- Event causing transition in composite system causes transition in at least 1 of the components
- If transition occurs in exactly one component, event must not cause transition in other component when not connected to the composite system

Equivalence for Composite

- Equivalence relation for composite system
  \((\sigma_a, \sigma_b) \equiv_c (\sigma_c, \sigma_d) \iff \sigma_a \equiv \sigma_c \text{ and } \sigma_b \equiv \sigma_d\)
- Corresponds to equivalence relation in property 2 for component system
Composition Theorem

• System resulting from composition of two restrictive systems is itself restrictive

Information Flow

• How does information flow around a system?
Detour: Entropy

- Random variables
- Joint probability
- Conditional probability
- Entropy (or uncertainty in bits)
- Joint entropy
- Conditional entropy
- Applying it to secrecy of ciphers

Random Variable

- Variable that represents outcome of an event
  - $X$ represents value from roll of a fair die; probability for rolling $n$: $p(X = n) = 1/6$
  - If die is loaded so 2 appears twice as often as other numbers, $p(X = 2) = 2/7$ and, for $n \neq 2$, $p(X = n) = 1/7$
- Note: $p(X)$ means specific value for $X$ doesn’t matter
  - Example: all values of $X$ are equiprobable
Joint Probability

• Joint probability of $X$ and $Y$, $p(X, Y)$, is probability that $X$ and $Y$ simultaneously assume particular values
  – If $X$, $Y$ independent, $p(X, Y) = p(X)p(Y)$
• Roll die, toss coin
  – $p(X = 3, Y = \text{heads}) = p(X = 3)p(Y = \text{heads})$
    = $\frac{1}{6} \times \frac{1}{2} = \frac{1}{12}$

Two Dependent Events

• $X =$ roll of red die, $Y =$ sum of red, blue die rolls
  $p(Y=2) = \frac{1}{36}$    $p(Y=3) = \frac{2}{36}$    $p(Y=4) = \frac{3}{36}$    $p(Y=5) = \frac{4}{36}$
  $p(Y=6) = \frac{5}{36}$    $p(Y=7) = \frac{6}{36}$    $p(Y=8) = \frac{5}{36}$    $p(Y=9) = \frac{4}{36}$
  $p(Y=10) = \frac{3}{36}$   $p(Y=11) = \frac{2}{36}$   $p(Y=12) = \frac{1}{36}$
• Formula if events independent:
  $p(X=1, Y=11) = p(X=1)p(Y=11) = \left(\frac{1}{6}\right)\left(\frac{2}{36}\right) = \frac{1}{108}$
• But in reality, $Y = 11$ is possible only when $X = 5$ and blue die is 6, so:
  $p(X=1, Y=11) = 0$
Conditional Probability

• Conditional probability of $X$ given $Y$, $p(X|Y)$, is probability that $X$ takes on a particular value given $Y$ has a particular value.

• Continuing example ...
  - $p(Y=7|X=1) = 1/6$
  - $p(Y=7|X=3) = 1/6$

Relationship

• $p(X, Y) = p(X | Y) p(Y) = p(X) p(Y | X)$

• Example:
  - $p(X=3, Y=8) = p(X=3|Y=8) p(Y=8) = (1/5)(5/36) = 1/36$

• Note: if $X$, $Y$ independent:
  - $p(X|Y) = p(X)$
Entropy

- Uncertainty of a value, as measured in bits
- Example: $X$ value of fair coin toss; $X$ could be heads or tails, so 1 bit of uncertainty
  - Therefore entropy of $X$ is $H(X) = 1$
- Formal definition: random variable $X$, values $x_1, \ldots, x_n$; so $\sum_i p(X = x_i) = 1$
  $$H(X) = -\sum_i p(X = x_i) \log p(X = x_i)$$

Heads or Tails?

- $H(X) = -p(X=\text{heads}) \log p(X=\text{heads})$
  - $p(X=\text{tails}) \log p(X=\text{tails})$
  $$= - (1/2) \log (1/2) - (1/2) \log (1/2)$$
  $$= - (1/2) (-1) - (1/2) (-1) = 1$$
- Confirms previous intuitive result
### n-Sided Fair Die

\[ H(X) = - \sum_i p(X = x_i) \log p(X = x_i) \]

As \( p(X = x_i) = 1/n \), this becomes

\[ H(X) = - \sum_i (1/n) \log (1/n) = -n(1/n) (-\log n) \]

so

\[ H(X) = \log n \]

which is the number of bits in \( n \), as expected.

---

### Ann, Pam, and Paul

Ann, Pam twice as likely to win as Paul

\( W \) represents the winner. What is its entropy?

- \( w_1 = Ann, w_2 = Pam, w_3 = Paul \)
- \( p(W = w_1) = p(W = w_2) = 2/5, p(W = w_3) = 1/5 \)

- So \( H(W) = - \sum_i p(W = w_i) \log p(W = w_i) \)
  
  \[ = -(2/5) \log (2/5) - (2/5) \log (2/5) - (1/5) \log (1/5) \]
  
  \[ = \log 5 - (4/5) \log 2 = \log 5 - (4/5) \approx 1.52 \]

- If all equally likely to win, \( H(W) = \log 3 = 1.58 \)
Joint Entropy

• $X$ takes values from $\{ x_1, \ldots, x_n \}$
  $- \sum_i p(X=x_i) = 1$

• $Y$ takes values from $\{ y_1, \ldots, y_m \}$
  $- \sum_i p(Y=y_i) = 1$

• Joint entropy of $X$, $Y$ is:
  $- H(X, Y) = -\sum_j \sum_i p(X=x_i, Y=y_j) \lg p(X=x_i, Y=y_j)$

Example

$X$: roll of fair die, $Y$: flip of coin

• $p(X=1, Y=\text{heads}) = p(X=1)p(Y=\text{heads}) = 1/12$
  $- \text{As } X \text{ and } Y \text{ are independent}$

• $H(X, Y) = -\sum_j \sum_i p(X=x_i, Y=y_j) \lg p(X=x_i, Y=y_j)$
  $= -2 \left[ 6 \left[ (1/12) \lg (1/12) \right] \right] = \lg 12$
Conditional Entropy

- $X$ takes values from $\{x_1, \ldots, x_n\}$
  - $\Sigma_i p(X=x_i) = 1$
- $Y$ takes values from $\{y_1, \ldots, y_m\}$
  - $\Sigma_i p(Y=y_i) = 1$
- Conditional entropy of $X$ given $Y=y_j$ is:
  - $H(X \mid Y=y_j) = -\Sigma_i p(X=x_i \mid Y=y_j) \log p(X=x_i \mid Y=y_j)$
- Conditional entropy of $X$ given $Y$ is:
  - $H(X \mid Y) = -\Sigma_j p(Y=y_j) \Sigma_i p(X=x_i \mid Y=y_j) \log p(X=x_i \mid Y=y_j)$

Example

- Roll of red die, $Y$ sum of red, blue roll
- Note $p(X=1 \mid Y=2) = 1$, $p(X=i \mid Y=2) = 0$ for $i \neq 1$
  - If the sum of the rolls is 2, both dice were 1
- $H(X \mid Y=2) = -\Sigma_i p(X=x_i \mid Y=2) \log p(X=x_i \mid Y=2) = 0$
- Note $p(X=i, Y=7) = 1/6$
  - If the sum of the rolls is 7, the red die can be any of 1, ..., 6 and the blue die must be 7—roll of red die
- $H(X \mid Y=7) = -\Sigma_i p(X=x_i \mid Y=7) \log p(X=x_i \mid Y=7)$
  - $= -6 \left( \frac{1}{6} \right) \log \left( \frac{1}{6} \right) = \log 6$
Perfect Secrecy

- Cryptography: knowing the ciphertext does not decrease the uncertainty of the plaintext
- $M = \{m_1, \ldots, m_n\}$ set of messages
- $C = \{c_1, \ldots, c_n\}$ set of ciphers
- Cipher $c_i = E(m_i)$ achieves perfect secrecy if $H(M \mid C) = H(M)$

Basics

- Bell-LaPadula Model embodies information flow policy
  - Given compartments $A$, $B$, info can flow from $A$ to $B$ iff $B \text{ dom } A$
- Variables $x$, $y$ assigned compartments $x$, $y$ as well as values
  - If $x = A$ and $y = B$, and $A \text{ dom } B$, then $y := x$ allowed but not $x := y$
Entropy and Information Flow

• Idea: info flows from x to y as a result of a sequence of commands c if you can deduce information about x before c from the value in y after c

• Formally:
  – s time before execution of c, t time after
  – \( H(x_s | y_t) < H(x_s | y_s) \)
  – If no y at time s, then \( H(x_s | y_t) < H(x_s) \)

Example 1

• Command is \( x := y + z \); where:
  – \( 0 \leq y \leq 7 \), equal probability
  – \( z = 1 \) with prob. 1/2, \( z = 2 \) or \( z = 3 \) with prob. 1/4 each

• s state before command executed; t, after; so
  – \( H(y_s) = H(y_t) = -8(1/8) \log_2 (1/8) = 3 \)
  – \( H(z_s) = H(z_t) = -(1/2) \log_2 (1/2) -2(1/4) \log_2 (1/4) = 1.5 \)

• If you know \( x_t \), \( y_s \) can have at most 3 values, so \( H(y_s | x_t) = -3(1/3) \log_2 (1/3) = \log_3 3 \)
Example 2

- Command is
  - if \( x = 1 \) then \( y := 0 \) else \( y := 1 \);

where:
  - \( x, y \) equally likely to be either 0 or 1
- \( H(x_s) = 1 \) as \( x \) can be either 0 or 1 with equal probability
- \( H(x_s \mid y_t) = 0 \) as if \( y_t = 1 \) then \( x_s = 0 \) and vice versa
  - Thus, \( H(x_s \mid y_t) = 0 < 1 = H(x_s) \)
- So information flowed from \( x \) to \( y \)

Implicit Flow of Information

- Information flows from \( x \) to \( y \) without an explicit assignment of the form \( y := f(x) \)
  - \( f(x) \) an arithmetic expression with variable \( x \)
- Example from previous slide:
  - if \( x = 1 \) then \( y := 0 \)
    else \( y := 1 \);
- So must look for implicit flows of information to analyze program
Notation

• \( x \) means class of \( x \)
  – In Bell-LaPadula based system, same as “label of security compartment to which \( x \) belongs”
• \( x \leq y \) means “information can flow from an element in class of \( x \) to an element in class of \( y \)
  – Or, “information with a label placing it in class \( x \) can flow into class \( y \)”

Information Flow Policies

Information flow policies are usually:

• reflexive
  – So information can flow freely among members of a single class

• transitive
  – So if information can flow from class 1 to class 2, and from class 2 to class 3, then information can flow from class 1 to class 3
Non-Transitive Policies

- Betty is a confidant of Anne
- Cathy is a confidant of Betty
  - With transitivity, information flows from Anne to Betty to Cathy
- Anne confides to Betty she is having an affair with Cathy’s spouse
  - Transitivity undesirable in this case, probably

Non-Lattice Transitive Policies

- 2 faculty members co-PIs on a grant
  - Equal authority; neither can overrule the other
- Grad students report to faculty members
- Undergrads report to grad students
- Information flow relation is:
  - Reflexive and transitive
- But some elements (people) have no “least upper bound” element
  - What is it for the faculty members?
Confidentiality Policy Model

- Lattice model fails in previous 2 cases
- Generalize: policy \( I = (SC_I, \leq_I, \text{join}_I) \):
  - \( SC_I \) set of security classes
  - \( \leq_I \) ordering relation on elements of \( SC_I \)
  - \( \text{join}_I \) function to combine two elements of \( SC_I \)
- Example: Bell-LaPadula Model
  - \( SC_I \) set of security compartments
  - \( \leq_I \) ordering relation \( \text{dom} \)
  - \( \text{join}_I \) function \( \text{lub} \)