CAS 745
Supervisory Control of Discrete-Event Systems

Slides 1: Algebraic Preliminaries

Dr. Ryan Leduc

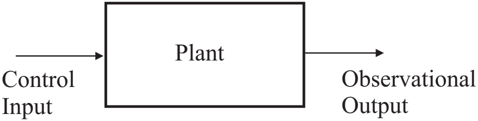
Department of Computing and Software
McMaster University

Material based on W. M. Wonham, Supervisory Control of Discrete-Event Systems, Department of Electrical and Computer Engineering, University of Toronto, July 2004 Lecture notes of Professor Wonham also used.

Control Systems

- The research area of *Supervisory Control* originated in systems control research.

- Area deals with dynamic systems: systems that have state space, state transition structure, and inputs/outputs.

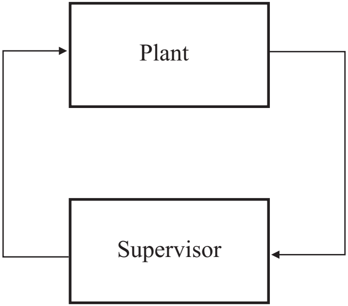


Control Systems - II

- Feedback control in the form of a controller is usually needed to ensure the desired behaviour.

- Typically, the controller has its own dynamics, and its the combination of the two (called the *closed-loop behaviour*) that provides the desired behaviour (*specification*).

- A suitable controller may not always exist for a given desired behaviour and plant.



Feedback Control

Continuous Control Systems Eg.

- Here our inputs/outputs are represented as continuous or discrete time functions, as well as the system's dynamics.
- Example shows a robot hand, grasping force, control system.

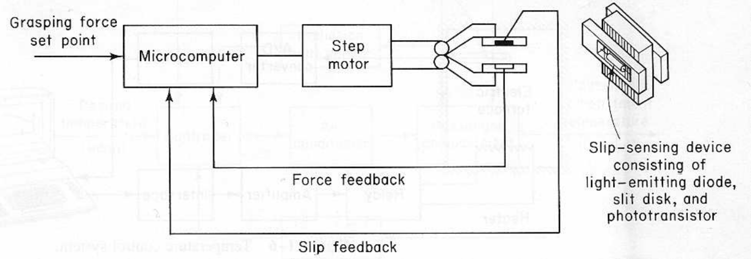


Diagram taken from Katsuhiko Ogata, *Modern Control Engineering, 2nd ed.*, Prentice Hall, 1990

Discrete-Event Systems Overview

- Discrete-event (dynamic) systems (DES or DEDS) are a relatively new area of research for control systems, but is now well represented at conferences and in the literature.
- Recently, continuous systems have been composed with DES to form a new research area called *Hybrid Systems.*
- The parent domains of DES are *Operations Research (OR)* and *Software Engineering.*

Operations Research

- OR deals with systems of *stores* and *servers* that are interconnected. Together, they process "items."

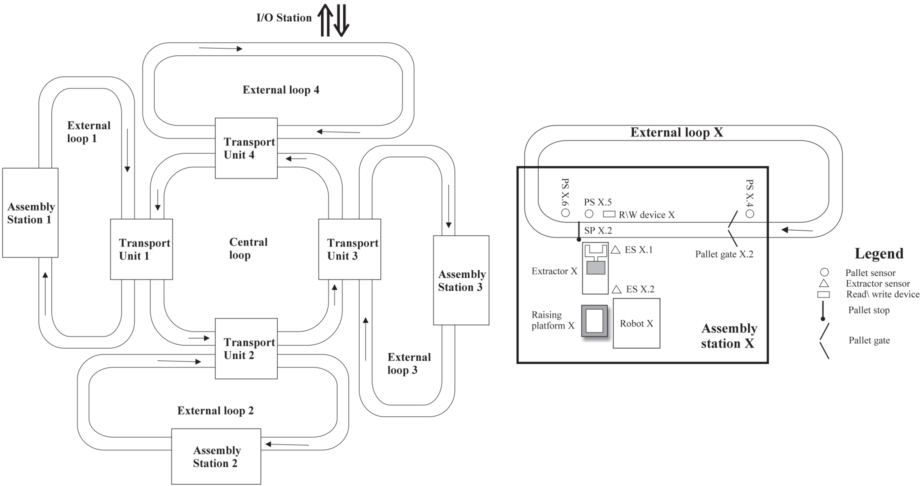
- **Focus of area:**
  - Measure quantitative performance and tradeoffs (throughput of operation versus cost etc.).

  - Optimize control parameters such as buffer size, maintenance schedules, uptime, etc.
- Some examples:

  - **Manufacturing Systems:** operates on *work pieces* which are stored in *queues or buffers.*

  - These work pieces are then operated on by *robots, machines, automated guided vehicles (AGVs)* etc.

Manufacturing Example



AIP example taken from R.J. Leduc, *Hierarchical Interface-based Supervisory Control.* Doctoral Thesis, Dept. of Elec. & Comp. Engrg., Univ. of Toronto, 2002.

Operations Research - II

- **Communication Systems:** operates on *messages* which are stored in queues and buffers.
- These are then served by *transmitters, channels, receivers, network nodes* etc.
- Other examples are: **Traffic systems, Database management systems,** etc.

Software Engineering and DES

- Some areas of software engineering that are relevant are operating systems (resource management etc.), concurrent computation, real-time (embedded and reactive) systems.

- Example applications are synchronization algorithms that control resource sharing and mutual exclusion enforcement for concurrent entities.

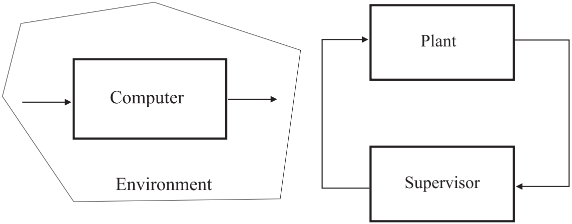
- **Focus of area:**

  - Guarantee safety ("nothing bad will happen"), mutual exclusion, deadlock prevention.

  - Guarantee liveness ("something good will eventually happen"), efficiency, absence of starvation or lockout.

Software Engineering and DES - II

- Diagram below shows different viewpoint of Software Engineering and discrete-event systems.



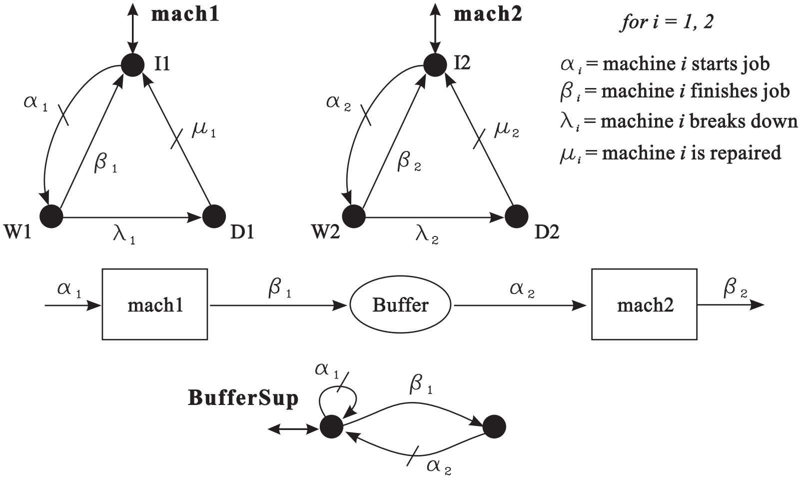
Discrete-Event Systems

- A Discrete-event system is a dynamic system that is:
  - Discrete in time and statespace.
  - Asynchronous or event driven. DES are driven by "events" (ie machine starts, machine stops etc.) or instantaneous "occurrences in time."
  - Occurrence of events is nondeterministic.
  - Means that the selection process that determines which event out of those currently possible to occur, is not modelled (could be chance, or some unspecified mechanism).

Focus of Area

- Logical correctness in presence of concurrency, timing constraints, etc.
- Quantitative performance.
- Feedback control synthesis and optimization.

Small Factory Example



Closed Loop: **sync** (mach1, mach2,BufferSup)

Industrial Chemical Process Eg.

- *Plant Components:* Reaction tank, valves, heater, mixer etc.

- *States:* Fluid level (low, medium, high, etc),
  temperature, pH (too low, OK, too high, etc).
  X
  valve states (open, closed, etc.)
  X
  heater and mixer states (off, on, etc.)

- *Transitions (events):* chemical mixture reaching desired composition, temperature entering a desired range, critical variables entering a dangerous range etc.

Discrete-Event Systems Challenges

Model size: DES state space increases combinatorially in the number of model components. Can quickly make computation intractable.

Structure: How to exploit regularities of a system's structure to improve scalability. Usually combined with appropriate architecture that can take advantage of structure.

Implementation: How to implement supervisors as hardware and software, taking care of necessary concurrency issues.

Binary Relations - §1.1

- Let X be an arbitrary set. The set of all ordered pairs of X is:

$$X \times X := \{(x_1, x_2) | \, x_1, x_2 \in X\}$$

- **Note:** (x_1, x_2) an ordered pair means that

$$(x_1, x_2) = (x_1', x_2') \Leftrightarrow x_1 = x_1' \;\&\; x_2 = x_2'$$

- **Defn:** A binary relation is any $R \subseteq X \times X$

- We write: $(x_1, x_2) \in R$ or $x_1 R x_2$ (infix notation).

Posets

- Let the symbol "$\leq$" denote a binary relation R with the properties:
  1. $(\forall x \in X)\ x \leq x$ *(reflexive)*
  2. $(\forall x, y, z \in X)\ x \leq y \ \&\ y \leq z \Rightarrow x \leq z$ *(transitive)*
  3. $(\forall x, y \in X)\ x \leq y \ \&\ y \leq x \Rightarrow x = y$ *(antisymmetric)*

Definition
A relation $\leq$ with the properties (1)-(3) is a partial order (p.o) on X.

- Poset stands for *partially ordered sets.*
- We refer to "the poset X" if the relation $\leq$ is understood. Otherwise, we refer to the poset $(X, \leq)$.

Total Ordering

- **Defn:** we say that $x, y \in X$ are comparable wrt $\leq$ if $x \leq y$ or $y \leq x$.
- If two elements $x, y \in X$ are not comparable, we write $x <> y$.
- A poset is a total ordering if every $x, y \in X$ is comparable.
- How do you know if the relation is a poset? Check that it satisfies the three properties.
- We can extend our notation as follows:
- If $x \leq y$, may write $y \geq x$
- If $x \leq y$ and $x \neq y$ then may write $x < y$ or $y > x$
- **NOTE:** $\neg(x \leq y) \Leftrightarrow (x <> y)$ or $(x > y)$

Poset Examples

1. Let $X = \mathbb{R}$ (set of real numbers), or $X = \mathbb{N} := \{0, 1, 2, \ldots\}$ (set of natural numbers) or $X = \mathbb{Z} := \{\ldots, -1, 0, +1, \ldots\}$ (set of integers) and let $\leq$ have normal meaning.

   Thus $2 \leq 3$ but not $4 \leq 3$.

2. Let $X = \underbrace{\mathbb{Z} \times \ldots \times \mathbb{Z}}_{\text{k-fold}}$ thus:

   $x \in X \Rightarrow x = (x_1, \ldots, x_k)$ with $x_i \in \mathbb{Z}$, $i = 1, \ldots, k$.

   Defn: $(\forall x, y \in X)\ x \leq y \Leftrightarrow x_i \leq y_i$ (normal meaning) for all $i = 1, \ldots, k$.

   eg. $(5, -2) \leq (7, -1)$ but $(5, -1) <> (7, -2)$ as $5 \leq 7$ but $-1 > -2$.

Poset Examples - II

3. Let A be an arbitrary set.

Let $X = \mathsf{Pwr}(A) = 2^A$.

The powerset of A (2^A) is the set of all subsets of A thus:
$x \in X \Rightarrow x \subseteq A$

eg. $x = \emptyset$, $x = \{a, b\}$ with $a, b \in A$, $x = A$

Defn: $(\forall x, y \in X)\ x \leq y \Leftrightarrow x \subseteq y$

eg. $A = \{\alpha, \beta, \gamma\}$ thus:

$$X = \{\emptyset, \{\alpha\}, \{\beta\}, \{\gamma\}, \{\alpha, \beta\}, \{\alpha, \gamma\}, \{\beta, \gamma\}, \{\alpha, \beta, \gamma\}\}$$

Have $\{\beta\} \leq \{\beta, \gamma\}$ but $\{\beta\} <> \{\alpha, \gamma\}$

Should be able to check for yourself that $(X, \leq)$ is a poset.

Meet

- Let $(X, \leq)$ be a poset, and let $x, y \in X$.
- **Defn:** Element $a \in X$ is a lower bound for x and y if $a \leq x$ and $a \leq y$.
- eg. for $X = \text{Pwr}(A)$, this means a is a subset of both x and y.
- **Defn:** Element $l \in X$ is a meet (greatest lower bound) for x and y iff

$$l \leq x \;\; \& \;\; l \leq y \;\; \& \;\; (\forall a \in X)\,(a \leq x \;\; \& \;\; a \leq y) \Rightarrow a \leq l$$

- In other words, l is a lower bound and beats any other lower bound.
- For a given x and y, the meet may not exist. If it does, it is unique and we denote it $x \wedge y$.
- **Defn:** We denote the bottom element by $\bot$. If it exists, it satisfies: $\bot \in X$ and $(\forall x \in X)\, \bot \leq x$.

Join

- **Defn:** Element $b \in X$ is an upper bound for x and y if $x \leq b$ and $y \leq b$.
- eg. for $X = \text{Pwr}(A)$, this means x and y are a subset of b.
- **Defn:** Element $u \in X$ is a join (least upper bound) of x and y iff

$$x \leq u \;\;\&\;\; y \leq u \;\;\&\;\; (\forall b \in X)\,(x \leq b \;\;\&\;\; y \leq b) \Rightarrow u \leq b$$

- In other words, u is a upper bound and is lower than any other upper bound.
- For a given x and y, the join may not exist. If it does, it is unique and we denote it $x \vee y$.
- **Defn:** We denote the top element by $\top$. If it exists, it satisfies: $\top \in X$ and $(\forall x \in X)\, x \leq \top$.

Meet and Join Examples

1. Let $X = \mathsf{Pwr}(A)$ and $x, y \in X$.

   We then have $x \wedge y = x \cap y$ (set intersection) and $x \vee y = x \cup y$ (set union).

   We also have $\bot = \emptyset$ and $\top = A$.

   For this X, the meet, join, bottom, and top always exist.

2. Let $X = \mathbb{Z} \times \mathbb{Z}$ and $x = (x_1, x_2), y = (y_1, y_2) \in X$. We then have:

$$\begin{aligned} x \wedge y &= (\min(x_1, y_1), \min(x_2, y_2)) \\ x \vee y &= (\max(x_1, y_1), \max(x_2, y_2)) \end{aligned}$$

   As $\mathbb{Z} := \{\ldots, -1, 0, +1, \ldots\}$, neither $\bot$ or $\top$ exist.

Lattices - §1.2

- **Defn:** A lattice L is a poset $(L, \leq)$ in which the meet and join is defined for all $x, y \in L$.

- The binary operators $\wedge$ and $\vee$ define the functions:

$$\wedge : L \times L \to L, \ \ \vee : L \times L \to L$$

- For $x, y, z \in L$ then the operators $\wedge$ and $\vee$ (denoted consistently throughout as $\star$) satisfy the following properties:
  1. $x \star x = x$ ($\star$ is *idempotent*)
  2. $x \star y = y \star x$ ($\star$ is *commutative*)
  3. $(x \star y) \star z = x \star (y \star z)$ ($\star$ is *associative*)

- eg. for $\vee$, point 2 gives: $x \vee y = y \vee x$

- We also have the following relationships:

$$x \wedge (x \vee y) = x \vee (x \wedge y) = x \qquad (\textit{absorption})$$

$$x \leq y \text{ iff } x \wedge y = x \text{ iff } x \vee y = y \qquad (\textit{consistency})$$

Absorption Example

Verify: $x \wedge (x \vee y) = x$

let $a = x \wedge (x \vee y)$

Proof: Must show 1) $x \leq a$ 2) $a \leq x$

Part 1) show $x \leq a$

Have $x \leq x$ by reflexivity of $\leq$

As $x \vee y$ is upper bound of x, y by definition of join, we have:
$x \leq x \vee y$

$\Rightarrow$ x is a lower bound of x and $x \vee y$

$\Rightarrow$ $x \leq x \wedge (x \vee y)$ by defn of meet.

Part 1 done.

Absorption Example - II

Part 2) show $a \leq x$

By defn of meet, $a \leq x$ and $a \leq x \vee y$

Part 2 done.

By parts 1) - 2), we have $x \leq a$ and $a \leq x$.

$\Rightarrow x = a$ by antisymmetry of Posets.

QED

Infinum and Supremum

- **Defn:** Let L be a lattice and let S be a nonempty, possibly infinite subset of L.

- The infimum of S, denoted *inf*(S), is an element $l \in L$ with the properties:

$$(\forall y \in S)\, l \leq y \;\;\&\;\; (\forall z \in L)\, ((\forall y \in S)\, z \leq y) \Rightarrow z \leq l$$

- Note: $l \in L$ but is possible $l \notin S$

- **Defn:** Let L be a lattice and let S be a nonempty, possibly infinite subset of L.

- The supremum of S, denoted *sup*(S), is an element $u \in L$ with the properties:

$$(\forall y \in S)\, y \leq u \;\;\&\;\; (\forall z \in L)\, ((\forall y \in S)\, y \leq z) \Rightarrow u \leq z$$

- Note: $u \in L$ but is possible $u \notin S$

Complete Lattices

- If lattice L is finite, the inf(S) and sup(S) reduce to the meet and join of a finite number of elements of L.
- In this case, they always exist as L is a lattice.
- If S is an infinite subset, they need not exist. If they do, they are unique.
- **Defn:** A lattice L is complete if for any nonempty subset S of L, the inf(S) and sup(S) always exists.

Lattice Examples

1. Let A be an arbitrary finite set. Let $L = \text{Pwr}(A)$. Let $S \subseteq L$ and $S \neq \emptyset$.

$$\inf(S) = \bigcap_{y \in S} y \quad \sup(S) = \bigcup_{y \in S} y$$

   Clearly, $(L, \cap, \cup)$ is complete.

2. Let $L = \mathbb{Z} \times \mathbb{Z}$ and $x = (x_1, x_2), y = (y_1, y_2) \in L$. We then have:

$$\begin{aligned} x \wedge y &= (\min(x_1, y_1), \min(x_2, y_2)) \\ x \vee y &= (\max(x_1, y_1), \max(x_2, y_2)) \end{aligned}$$

   Let $S := \{(x, y) \in \mathbb{Z} \times \mathbb{Z} |\, y \geq 1\}$.

   Thus S is an infinite subset, with y component able to be arbitrarily large. Clearly no max value exists for S, thus $\sup(S)$ not defined.

   Thus, L is not complete.

When S is Empty

- Regardless of whether L is complete, if $\sup(L) = \top$ exists, then we can include $S = \emptyset \subseteq L$ in our definition of $\inf(S)$ by defining:

$$\inf(\emptyset) = \sup(L)$$

- Similarly if $\inf(L) = \bot$ exists, we can define:

$$\sup(\emptyset) = \inf(L)$$

- This is a result of "empty set logic."

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

- If $p = F$ then the statement $p \Rightarrow q$ is always true for any q.

When S is Empty - II

- The statement $y \in \emptyset$ is false for any y in the universe, thus statement $y \in \emptyset \Rightarrow q$ is true for any q.

- To show that inf$(\emptyset) = \top$, we need to show:
  1. $(\forall y \in \emptyset)\, \top \leq y$

     Equivalent to showing: $(\forall y)\, y \in \emptyset \Rightarrow \top \leq y$

     This statement would only be false if:

     $$(\exists y)\, y \in \emptyset \;\;\&\;\; \neg(\top \leq y)$$

     NOTE: Similarly $(\forall z \in L)(\forall y \in \emptyset)\, z \leq y$ is true.

  2. $(\forall z \in L)\, z \leq \top$ true by defn of $\top$

Equivalence Relations - §1.3

- Let X be a nonempty set, and let $E \subseteq X \times X$ be a binary relation on X.
- **Defn:** The relation E is an equivalence relation on X if:
  1. $(\forall x \in X)\, xEx$ (E is *reflexive*)
  2. $(\forall x, x' \in X)\, xEx' \Rightarrow x'Ex$ (E is *symmetric*)
  3. $(\forall x, x', x'' \in X)\, xEx' \;\&\; x'Ex'' \Rightarrow xEx''$ (E is *transitive*)
- For xEx' we may write $x \equiv x' \,(\text{mod} E)$

Equivalence Relations Examples

1. Let X be the set of all motorized vehicles.

   We could define an equivalence relation, E, as follows:

   - All two wheeled vehicles (mopeds, motorbikes etc) are equivalent, AND
   - All four wheeled vehicles (cars, trucks, buses) are equivalent, but no two wheel is equivalent to a four wheel vehicle, AND
   - All vehicles that do not fall into the above types are equivalent.

2. Any arbitrary partition (ie doesn't need any physical meaning) of X could be used to create an equivalence relation.

Equivalence Classes (Cosets)

- For $x \in X$, let $[x]$ represent the subset of elements that are equivalent to x:

$$[x] := \{x' \in X \,|\, x' E x\} \subseteq X$$

- **Defn:** We refer to the subset $[x]$ as the coset (or equivalence class) of x wrt to E.

- By reflexivity, we have $x \in [x]$ thus all cosets are non empty.

- **Proposition 1.3.1**

$$(\forall x, y \in X) \text{ either } \; [x] = [y] \text{ or } \; [x] \cap [y] = \emptyset$$

Proof: Assume one statement is false, and show this implies the other is true.

Partitions

- The equivalence relation divides X into nonempty, non-overlapping subsets, inducing a partition of X.

- Let $\mathcal{P}$ be a family of subsets of X, with index set A:

$$\mathcal{P} = \{C_\alpha \subseteq X \,|\, \alpha \in A\}$$

Defn: $\mathcal{P}$ is a partition of X if:

1. $(\forall \alpha \in A)\, C_\alpha \neq \emptyset$
2. $(\forall x \in X)(\exists \alpha \in A)\, x \in C_\alpha$
3. $(\forall \alpha, \beta \in A)\, \alpha \neq \beta \Rightarrow C_\alpha \cap C_\beta = \emptyset$

- The subsets C_α are called the *cells* of $\mathcal{P}$.

- Clearly, the collection of distinct cosets of an equivalence relationship is a partition.

Inducing Equivalence Relations

- Given a partition $\mathcal{P}$ of X, we can define a corresponding equivalence relation E as follows:

$$(\forall x, y \in X)\, xEy \text{ iff } (\exists \alpha \in A)\, x \in C_\alpha \;\&\; y \in C_\alpha$$

- Will speak of equivalence relations and partitions interchangeably.

Lattice of Partitions

- Let $\mathcal{E}(X)$ (or simply $\mathcal{E}$) be the set of all equivalence relations on (partitions of) X. We will now define a partial order on $\mathcal{E}$ and then show that it is a complete lattice:

$$(\forall E_1, E_2 \in \mathcal{E})\, E_1 \leq E_2 \text{ iff } (\forall x, y \in X)\, xE_1y \Rightarrow xE_2y$$

- Implies every coset of E_1 is a subset of exactly one coset of E_2.

- **Defn:** If $E_1 \leq E_2$, we say that E_1 refines E_2, or E_1 is finer than E_2, or that E_2 is coarser than E_1.

Existence of Meet

- **Proposition 1.3.2:** In the poset $(\mathcal{E}, \leq)$ the meet $E = E_1 \wedge E_2$ always exists and is given by:

$$(\forall x, x' \in X)\, x \equiv x' (\text{mod}\, E)$$
$$\text{iff} x \equiv x' (\text{mod}\, E_1) \;\&\; x \equiv x' (\text{mod}\, E_2)$$

- The relation $E_1 \wedge E_2$ is the coarsest partition that is still finer than E_1 and E_2.

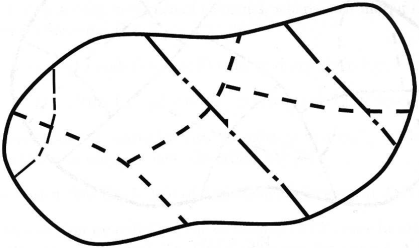


Fig. 1.3.1
Meet of Two Partitions

—— - —— - —— boundaries of E-cells
.... boundaries of F-cells
Any boundary line is a boundary of an $(E \wedge F)$-cell

Existence of Join

- **Proposition 1.3.3:** In the poset $(\mathcal{E}, \leq)$ the join $E = E_1 \vee E_2$ always exists and is given by:

$$(\forall x, x' \in X)\, x \equiv x' (\text{mod}\, E)$$
$$\text{iff}\, (\exists\, \text{integer}\, k \geq 1)(\exists x_0, x_1, \ldots, x_k \in X) x_0 = x \;\&\; x_k = x'$$
$$\&(\forall i) 1 \leq i \leq k \Rightarrow [x_i \equiv x_{i-1} (\text{mod}\, E_1)\, \text{or}\, x_i \equiv x_{i-1} (\text{mod}\, E_2)]$$

- The relation $E_1 \vee E_2$ is the finest partition that is still coarser than E_1 and E_2.

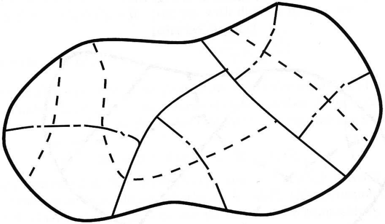
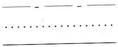


Fig. 1.3.2
Join of Two Partitions



boundaries of E-cells
boundaries of F-cells
common boundaries of E-cells and F-cells, forming boundaries of $(E \vee F)$-cells

Completeness of $\mathcal{E}$

- **Proposition 1.3.4:** Let $\mathcal{F} \subseteq \mathcal{E}$ be a nonempty collection of equivalence relations on X. Then $E = \inf(\mathcal{F})$ exists; in fact:

$$(\forall x, x' \in X)\, xEx' \text{ iff } (\forall F \in \mathcal{F})\, xFx'$$

- Also, $E = \sup(\mathcal{F})$ exists; in fact:
$(\forall x, x' \in X)\, xEx'$
  iff $(\exists \text{ integer } k \geq 1)(\exists F_1, \ldots, F_k \in \mathcal{F})(\exists x_0, x_1, \ldots, x_k \in X)$
    $x_0 = x \;\&\; x_k = x' \;\&\; (\forall i) 1 \leq i \leq k \Rightarrow x_i \equiv x_{i-1} (\text{mod}\, F_i)$

- We can also state $\bot = \inf(\mathcal{E})$ and $\top = \sup(\mathcal{E})$ as follows:

$$x \equiv x' (\text{mod}\, \bot) \text{ iff } x = x' \qquad x \equiv x' (\text{mod}\, \top) \text{ iff } \textit{true}$$

Canonical Projection - §1.4

- Let X and Y be sets. Will use π, ρ etc. for elements of $\mathcal{E}(X)$.
- For $\pi \in \mathcal{E}(X)$, let $\overline{X} = X/\pi$ be the set of distinct cells of the equivalence relation π.
- **Defn:** The canonical projection associated with π, denoted P_π, is defined to be the surjective function:

$$P_\pi : X \to X/\pi : x \mapsto [x]$$

Equivalence Kernel

- Let $f : X \to Y$ be a function with the given domain and codomain. For $A \subseteq X$, we define:

$$f(A) := \{f(x) | \, x \in A\}$$

- **Defn:** We also associate with f the inverse image function $f^{-1} : \mathsf{Pwr}(Y) \to \mathsf{Pwr}(X)$ as follows:

$$f^{-1}(B) := \{x \in X | \, f(x) \in B\}, \quad B \in \mathsf{Pwr}(Y)$$

- **Defn:** The equivalence kernel of $f : X \to Y$, denoted $\ker f$, is the equivalence relation in $\mathcal{E}(X)$, defined as follows:

$$(\forall x, x' \in X) \, x \equiv x' (\text{mod } \ker f) \text{ iff } f(x) = f(x')$$

- For $\pi \in \mathcal{E}(X)$ and the canonical projection $P_\pi : X \to X/\pi$, we have $\ker P_\pi = \pi$

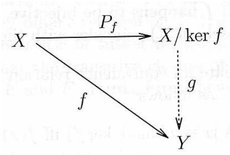
Equivalence Kernel Example

- Let the set X be a set of people in a room and let each person's age be in the integer range $\{21, 22, \ldots, 25\}$.
- Let $Y = \mathbb{N} := \{0, 1, 2, \ldots\}$, and $f : X \to Y$ map a person to their corresponding age.
- The equivalence relation ker$f$ groups the people based on their age.
- Let $\overline{X}$ be the cells of ker$f$. Treating $\overline{X}$ as a set of labels for the cosets, we could take $\overline{X} = \{21, 22, \ldots, 25\}$, representing the five possible ages of the people.
- Let $P_f : X \to \overline{X}$.
- See diagram in class.

Canonical Factorization

- Let $f : X \to Y$ and let $P_f : X \to X/\text{ker}f$.
- **Defn:** We refer to $f = g \circ P_f$ as the *canonical factorization* of f.
- If we only care about evaluating f, we can convert X to $\overline{X}$, and then just use g to evaluate f using the typically much smaller set $\overline{X}$.
- *Note:* $\circ$ denotes function composition. In this case, equivalent to saying:

$$(\forall x \in X)\, f(x) = g(P_f(x))$$



Canonical Factorization Proof

Claim: There exists a unique function $g : X/\text{ker}f \to Y$ such that $f = g \circ P_f$.

Proof: Let $\overline{x} \in X/\text{ker}f = \overline{X}$.

$\Rightarrow (\exists a \in X)\,[a] = \overline{x}$ by definition of $\overline{X}$

Define $g(\overline{x}) = f(a)$ **(1)**

If there exists other $a' \in X$ such that $[a'] = \overline{x}$ then:
$a \equiv a'(\text{mod ker}f)$

$\Rightarrow f(a) = f(a')$ (by definition of ker$f$), thus function g is well defined. **(2)**

Now need to verify that: $f = g \circ P_f$

Sufficient to show: $(\forall x \in X)\, f(x) = g(P_f(x))$

Canonical Factorization Proof - II

Let $x \in X$, then $P_f(x) = [x] = \overline{x}$

$\Rightarrow g(\overline{x}) = f(x)$ by definition.

thus $f(x) = g(P_f(x))$ as required. **(3)**

Finally, we need to show that g is unique.

Let $\hat{g} : X/\text{ker}f \rightarrow Y$ and assume that $f = \hat{g} \circ P_f$. **(4)**

Will show this implies $g = \hat{g}$.

Must show: $(\forall \overline{x} \in \overline{X})\, g(\overline{x}) = \hat{g}(\overline{x})$

Let $\overline{x} \in \overline{X}$, then $\overline{x} = [a]$ for some $a \in X$.

$\Rightarrow \overline{x} = P_f(a)$

Canonical Factorization Proof - III

$\Rightarrow \hat{g}(\overline{x}) = \hat{g}(P_f(a)) = f(a)$ by **(4)**

As $g(\overline{x}) = f(a)$ by definition, we have $\hat{g}(\overline{x}) = g(\overline{x})$

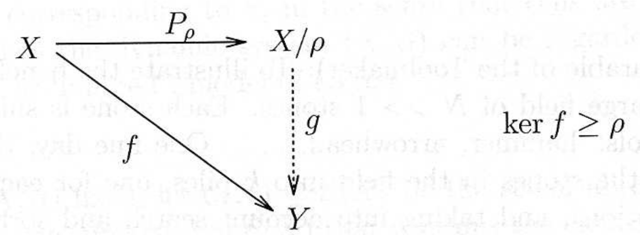
By **(1)**, **(2)**, and **(3)** we have defined a suitable function g, and by **(4)** we have shown it is unique.

QED

Other Factorizations

- **Proposition 1.4.1:** Suppose $f : X \to Y$ and let $\rho \in \mathcal{E}(X)$ with $\ker f \geq \rho$.

- There exists a unique map $g : X/\rho \to Y$ such that $f = g \circ P_\rho$

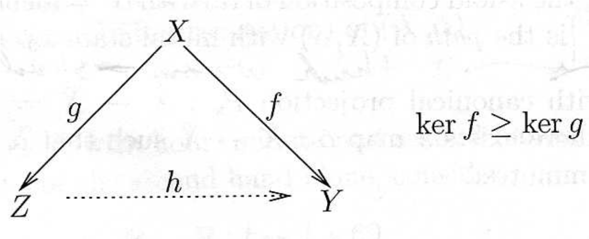


Other Factorizations - II

- **Proposition 1.4.2:** Suppose $f : X \to Y$ and let $g : X \to Z$ and let ker$f \geq$ ker$g$.

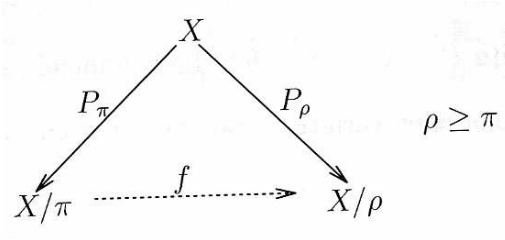
- There exists a map $h : Z \to Y$ such that $f = h \circ g$.

- Furthermore, h is uniquely defined on the image $g(X)$ of X in Z; that is, the restriction $h_{|g(X)}$ is unique.



Other Factorizations - III

- **Proposition 1.4.3:** If $\pi, \rho \in \mathcal{E}(X)$ and $\pi \leq \rho$, there is a unique function $f : X/\pi \to X/\rho$ such that $P_\rho = f \circ P_\pi$.



Dynamic Systems

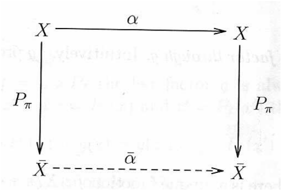
- **Defn:** A dynamic system on a set X is a map $\alpha : X \to X$ with interpretation X are the system's *states*, and α is the *state transition function* .
- We can then select an initial state $x_0 \in X$ and the system will evolve through states $x_1 = \alpha(x_0), x_2 = \alpha(x_1), \dots$
- We write α^k for the *k-fold composition* of α, taking $\alpha^0 =$ identity. ie. for $k = 2$ get $\alpha^2 = \alpha \circ \alpha$.
- The *sequence* $\langle \alpha^0(x_0), \alpha^1(x_0), \alpha^2(x_0), \dots \rangle$ is the path of (X, α) with initial state x_0.

Congruences of Dynamic Systems

- **Defn:** Let $\pi \in \mathcal{E}(X)$ with canonical projection $P_\pi : X \to \overline{X} := X/\pi$.

  Relation π is a congruence for α if there exists a map $\overline{\alpha} : \overline{X} \to \overline{X}$ such that $\overline{\alpha} \circ P_\pi = P_\pi \circ \alpha$. ie., the diagram below commutes.

- We say that $\overline{\alpha}$ is the map induced by α on $\overline{X}$.



Congruences of Dynamic Systems - II

- Fixing (X, α), let $\mathcal{C}(X) \subseteq \mathcal{E}(X)$ be the set of all congruences for α.
- It can be shown that $\mathcal{C}(X)$ is a complete sublattice of $\mathcal{E}(X)$ that contains the elements $\perp$ and $\top$ of $\mathcal{E}(X)$.
- **Defn:** For Lattice $L = (X, \leq)$ and $Y \subseteq X$, we say that $M = (Y, \leq)$ is a sublattice of L if Y is closed under the meet and join operations of L.
- By closed, we mean that the meet and join of any two elements of Y are also a member of Y.

Congruence Condition

- How do we know when π is a congruence?

- **Proposition 1.4.4:** π is a congruence for α iff

$$\ker P_\pi \leq \ker(P_\pi \circ \alpha)$$

  namely $(\forall x, x' \in X)\,(x, x') \in \pi \Rightarrow (\alpha(x), \alpha(x')) \in \pi$

  Proof: Immediate from **Proposition 1.4.1**, with identifications $(Y, f, \rho, g) = (\overline{X}, P_\pi \circ \alpha, \pi, \overline{\alpha})$.

  QED

- Condition says that α maps all elements in the same cell of π to the same new cell of π.

- Can take the dynamic system $(\overline{X}, \overline{\alpha})$, with initial state $\overline{x_0} = [x_0]$, as a consistent aggregated (high level) model of (X, α).

Observers

- Let $\alpha : X \to X$ be an arbitrary function. Let $\omega \in \mathcal{E}(X)$.
- **Defn:** Let $\omega \cdot \alpha \in \mathcal{E}(X)$ be defined as:

$$(\forall x, x' \in X)\, x \equiv x' (\mathrm{mod}\, \omega \cdot \alpha) \text{ iff } \alpha(x) \equiv \alpha(x') (\mathrm{mod}\, \omega)$$

- ie. $\omega \cdot \alpha = \ker(P_\omega \circ \alpha)$.
- Thus, ω is a congruence for α iff $\omega \leq \omega \cdot \alpha$.
- **Defn:** Let $\gamma : X \to Y$ be an arbitrary function. The observer for the triple (X, α, γ) is defined to be the equivalence relation:

$$\omega_o := \sup\{\omega \in \mathcal{E}(X) |\, \omega \leq (\ker \gamma) \wedge (\omega \cdot \alpha)\}$$

- Will show the observer is the coarsest congruence for α that is finer than $\ker \gamma$.

Observers Proof

Let $\mathcal{F} = \{\omega \in \mathcal{E}(X) | \, \omega \leq (\ker \gamma) \wedge (\omega \cdot \alpha)\}$.

Thus, $\mathcal{F}$ is the set of all congruences finer than the $\ker \gamma$.

Claim: $\omega_o \in \mathcal{F}$

To prove this, we must show that ω_o is a congruence for α and that $\omega_o \leq \ker \gamma$.

Subclaim 1: ω_o is a congruence for α.

Proof: Clearly, $\mathcal{F} \subseteq \mathcal{C}(X)$, the set of all congruences for α, as every $\omega \in \mathcal{F}$ has property $\omega \leq \omega \cdot \alpha$ and is thus a congruence.

As $\mathcal{C}(X)$ (the set of all congruences for α) is a complete sublattice of $\mathcal{E}(X)$, we have $\omega_o = \sup(\mathcal{F}) \in \mathcal{C}(X)$, as required.

Observers Proof - II

Subclaim 2: $\omega_o \leq \ker\gamma$

Proof: Let $x, x' \in X$. Assume $x(\sup(\mathcal{F}))x'$ **(1)**

Will now show implies $x(\ker\gamma)x'$

From definition of sup, **(1)** implies: **(2)**
$(\exists\,\text{integer}\,k \geq 1)(\exists F_1, \ldots, F_k \in \mathcal{F})(\exists x_0, x_1, \ldots, x_k \in X)$
$$x_0 = x \;\&\; x_k = x' \;\&\; (\forall i)1 \leq i \leq k \Rightarrow x_i \equiv x_{i-1}(\text{mod}\, F_i)$$

From definition of $\mathcal{F}$, $(\forall F \in \mathcal{F})\, F \leq \ker\gamma$

$\Rightarrow (\forall i)1 \leq i \leq k \Rightarrow x_i \equiv x_{i-1}(\text{mod}\ker\gamma)$ **(3)**

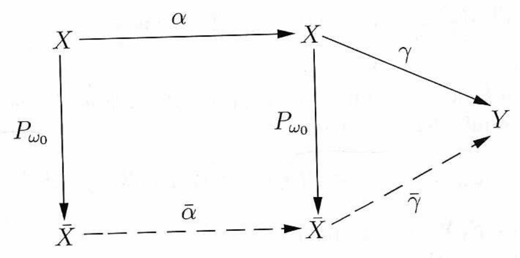
$\Rightarrow x(\ker\gamma)x'$ by the transitive property of equivalence relations.

By **subclaims 1** and **2**, we have that ω_o is a congruence for α and that $\omega_o \leq \ker\gamma$, as required.

QED

Observer Diagram

- If ω_o is an observer for (X, α, γ), then it will induce the maps $\overline{\alpha}$ and $\overline{\gamma}$, such that the diagram below will commute.
- Think of γ as the output map for the system.
- We can thus take the dynamic system $(\overline{X}, \overline{\alpha}, \overline{\gamma})$ as a consistent aggregated (high level) model of (X, α, γ).

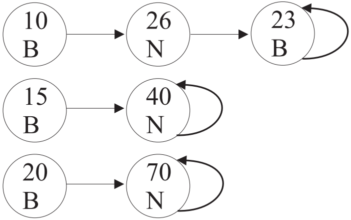


Observer Example

- Assume we have the dynamic system (X, α, γ) shown in the figure and table below.

- Define $\gamma : X \to Y := \{B, N, D\}$ as follows:

$$\begin{aligned}\gamma(x) &= B \text{ if } x < 25, \quad \gamma(x) = N \text{ if } 25 \leq x \leq 100, \\ \gamma(x) &= D \text{ if } x > 100\end{aligned}$$



x	10	15	20	23	26	40	70
$\alpha(x)$	26	40	70	23	23	40	70
$\gamma(x)$	B	B	B	B	N	N	N

Observer Example - II

- We will use the function $\Psi : \mathcal{E}(X) \rightarrow \mathcal{E}(X)$ to calculate our observer, ω_o:

$$\Psi(\omega) := (\ker \gamma) \wedge (\omega \cdot \alpha), \quad \omega \in \mathcal{E}(X)$$

- It can be shown that, if we start with $\omega = \ker \gamma$, Ψ is monotone and that ω_o is the greatest fixed point of Ψ.

- Our refinement algorithm will be:
  1. Set $\omega' = \ker \gamma$
  2. $\omega = \Psi(\omega') := (\ker \gamma) \wedge (\omega' \cdot \alpha)$
  3. if $\omega \neq \omega'$ then $\omega' = \omega$; goto step 2
  4. $\omega_o = \omega$; stop

Observer Example - III

- From table: $\ker\gamma = \{\{10, 15, 20, 23\}, \{26, 40, 70\}\}$

i	$\omega' = \ker\gamma$ calculate: $\omega' \cdot \alpha = \{\{10, 15, 20, 40, 70\}, \{23, 26\}\}$ calculate: $\omega = (\ker\gamma) \wedge (\omega' \cdot \alpha) = \{\{10, 15, 20\}, \{23\}, \{26\}, \{40, 70\}\}$ $\omega \neq \omega'$ so $\omega' = \omega$
ii	calculate: $\omega' \cdot \alpha = \{\{10\}, \{15, 20, 40, 70\}, \{23, 26\}\}$ calculate: $\omega = \{\{10\}, \{23\}, \{26\}, \{15, 20\}, \{40, 70\}\}$ $\omega \neq \omega'$ so $\omega' = \omega$
iii	calculate: $\omega' \cdot \alpha = \{\{10\}, \{15, 20, 40, 70\}, \{23, 26\}\}$ calculate: $\omega = \{\{10\}, \{23\}, \{26\}, \{15, 20\}, \{40, 70\}\}$ $\omega = \omega'$ so $\omega_o = \omega$; stop

Observer Example - IV

- Our observer is thus:

$$\omega_o = \{\{10\}, \{23\}, \{26\}, \{15, 20\}, \{40, 70\}\}$$

- Define: $\overline{X} := \{a, c, b, d, e\}$ to label cells:

$$\{10\}, \{23\}, \{26\}, \{15, 20\}, \{40, 70\}$$

- We can now define our aggregated system $(\overline{X}, \overline{\alpha}, \overline{\gamma})$ as follows:

$\overline{x}$	a	b	c	d	e
$\overline{\alpha}(\overline{x})$	b	c	c	e	e
$\overline{\gamma}(\overline{x})$	B	N	B	B	N