Slide 0

Chap.5 Hierarchy of System Specifications

Time base

Slide 1

- time base is defined to be a structure $time = \langle T, < \rangle$
  where T is a set and $<$ is an ordering relation on elements of T
  $<$ is *transitive, irreflexive* and *antisymmetric*
- ordering
  - linear or total   if for every pair $(t, t^{'})$ either $t < t^{'}, t^{'} t$ or $t = t^{'}$
  - partial   otherwise
- past, future, and present
  - present: t is interpreted to be the present time
  - past: $T_{t)} = \{\tau | \tau \in T, \tau < t\}$
  - future: $T_{(t} = \{\tau | \tau \in T, t < \tau\}$
  - past including the present: $T_{t]} = \{\tau | \tau \in T, \tau \leq t\}$
  - future including the present: $T_{[t} = \{\tau | \tau \in T, t \leq \tau\}$
  - not critical whether we deal with closed or open intervals
    - $T_{t>}$ means either $T_{t)}$ or $T_{t]}$
    - $T_{<t}$ means either $T_{(t}$ or $T_{[t}$

Time base (Cont'd)

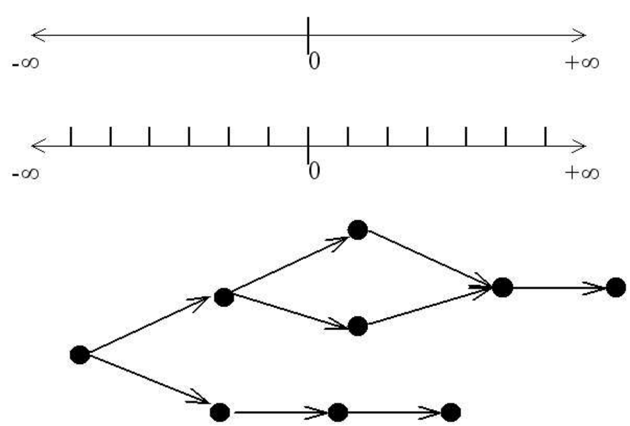
Slide 2

- time intervals
  * $T_{[t_1,t_2)} = \{\tau | \tau \in T, t_1 \leq \tau < t_2\}$
  * $T_{(t_1,t_2)} = \{\tau | \tau \in T, t_1 < \tau < t_2\}$
  * $T_{(t_1,t_2]} = \{\tau | \tau \in T, t_1 < \tau \leq t_2\}$
  * $T_{[t_1,t_2]} = \{\tau | \tau \in T, t_1 \leq \tau \leq t_2\}$
  * $T_{<t_1,t_2>}$ refers to one of the above intervals
    · $< t_1, t_1 >$ refers either to the singleton t_1 or to the empty interval

- examples of useful time bases
  - the real number $\Re$
  - the integers $\Im$
  - all sets $c * \Im, c \in \Re$ a constant

Time base (Cont'd)

Slide 3



Segments and trajectories

Slide 4

- trajectory
  - to describe how behavior occurs over time
  - let A denote a set, for example, an input, state, or output set
  - let T be a time base

    $f : T \to A$
    - f is called a time function, trajectory or signal
  - restricting f to a subset $T^{'}$ of T, $f|T^{'} : T^{'} \to A$

    where $f|T^{'}(t) := f(t)$ for all $t \in T^{'}$
  - with t denoting the present time
    - $f|T_{t>}$, also written $f_{t>}$, is called the past of f
    - $f|T_{<t}$, also written $f_{<t}$, is called the future of f

Segments and trajectories (Cont'd)

Slide 5

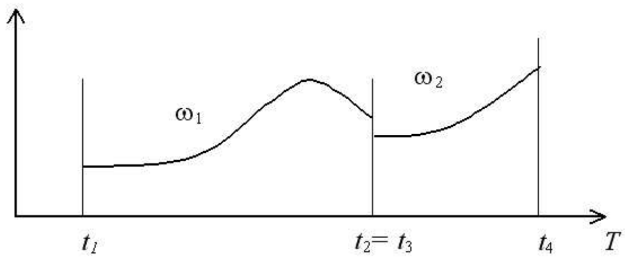
- segments
  - a restriction of f to a time interval $< t_1, t_2 >$ is called a segment
  - we usually write $\omega :< t_1, t_2 >\to A$ or $\omega_{<t_1,t_2>}$
  - a length operator $l : \Omega \to T_0^+$ by $l(\omega) = t_2 - t_1$

    where dom(ω) $= < t_1, t_2 >$ with dom(ω) denoting the domain of ω
  - *empty segment*
    - the segment with empty domain $< t_1, t_1 >$
    - denoted by special symbol Φ
    - empty segment has length 0
  - *contiguous segments*
    - a pair of segments $\omega_1 :< t_1, t_2 >\to A$ and $\omega_2 :< t_3, t_4 >\to A$
    - their domains are contiguous, i.e., $t_2 = t_3$

Segments and trajectories (Cont'd)

Slide 6

- *concatenation operation* •
  - $\omega_1 \bullet \omega_2 :< t_1, t_4 > \rightarrow A$
  - with
    - $\omega_1 \bullet \omega_2(t) = \omega_1(t)$ for $t \in < t_1, t_2 >$
    - $\omega_1 \bullet \omega_2(t) = \omega_2(t)$ for $t \in < t_3, t_4 >$

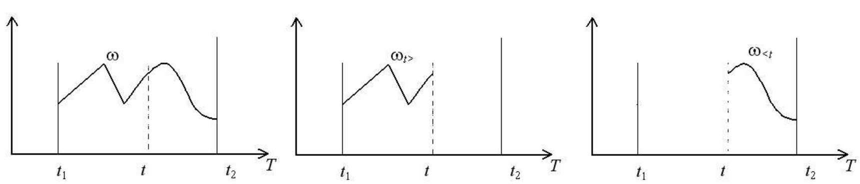


  - often denoted by $\omega_1 \omega_2$
  - *closed under concatenation*
    - if for every contiguous pair $\omega_1, \omega_2 \in \Omega$ also $\omega_1 \bullet \omega_2 \in \Omega$

Segments and trajectories (Cont'd)

Slide 7

- *left segment* and *right segment*
  - $\omega_{t>}$ or $\omega_{<t_1,t>}$ with $t \in < t_1, t_2 >$: left segment
  - $\omega_{<t}$ or $\omega_{<t,t_2>}$ with $t \in < t_1, t_2 >$: right segment
  - choose same type of original segment
    - original segment: $[t_1, t_2)$ →left segment: $\omega_{t)}$ and right segment: $\omega_{[t}$

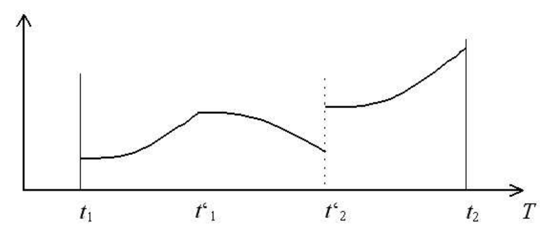


- split a segment into parts, it should be back again →$\omega_{t>} \bullet \omega_{<t} = \omega$
- *closed under left segmentation*
  - if every left segment is also in the set
  - if for every $\omega :< t_1, t_2 > \rightarrow A \in \Omega$ and $t \in < t_1, t_2 >$ we have $\omega_{t>} \in \Omega$
- *closed under right segmentation*
  - if for every $\omega :< t_1, t_2 > \rightarrow A \in \Omega$ and $t \in < t_1, t_2 >$ we have $\omega_{<t} \in \Omega$

Segments and trajectories (Cont'd)

Slide 8

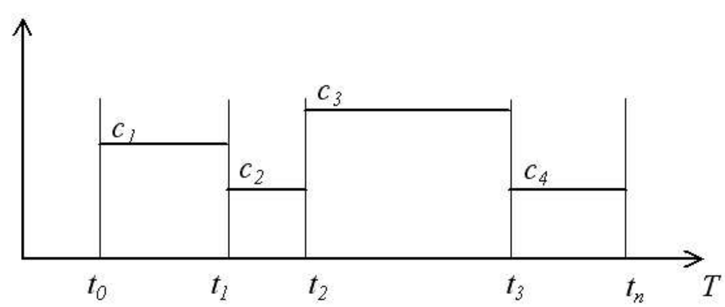
- piecewise continuous segments
  - *continuous segments*
    - a segment $\omega :< t_1, t_2 > \rightarrow \Re^n$ over a continuous time base if it is continuous at all points $t \in < t_1, t_2 >$
  - *piecewise continuous segments*
    - continuous at all points t except a finite number of points $t^{'} \in < t_1, t_2 >$
  - two segments often occur in differential equation modeling
  - *bounded piecewise continuous (bpc) segments* is a piecewise continuous segment with a finite upper bound



Segments and trajectories (Cont'd)

Slide 9

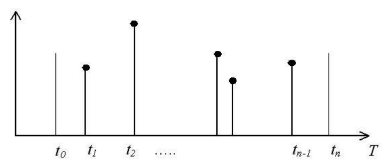
- piecewise constant segments
  - a subclass of piecewise continuous segments
  - a finite set of time points $t_1, t_2, t_3, \ldots, t_{n-1} \in < t_0, t_n >$ and values $c_1, c_2, \ldots, c_n \in A$ such that $\omega = \omega_{<t_0,t_1>} \bullet \omega_{<t_1,t_2>} \bullet \ldots \bullet \omega_{<t_{n-1},t_n>}$ and $\omega_{<t_{i-1},t_i>} = c_i$, for all $t \in < t_{i-1}, t_i >$
  - this segment occur in discrete event models as state trajectories



Segments and trajectories (Cont'd)

Slide 10

- event segments
  - represent a sequence of events ordered in time
  - $\omega :< t_0, t_n > \rightarrow A \cup \{\varnothing\}$ over a continuous time base
    where an arbitrary set $A \cup \{\varnothing\}$
    $\{\varnothing\}$ denotes the *nonevent* and is an element not in A
  - ω is an event segment
    * if there exists a finite set of time points $t_1, t_2, t_3, \ldots, t_{n-1} \in < t_0, t_n >$ such that $\omega(t_i) = a_i \in A$ for $i = 1, \ldots, n-1$, and $\omega(t) = \varnothing$ for all other $t \in < t_0, t_n >$
    * an event segment $\omega :< t_1, t_2 > \rightarrow A \cup \{\varnothing\}$ does not contain any events $\varnothing_{<} t_1, t_2 >$
  - this segment occur in discrete event models as input and output trajectories



Segments and trajectories (Cont'd)

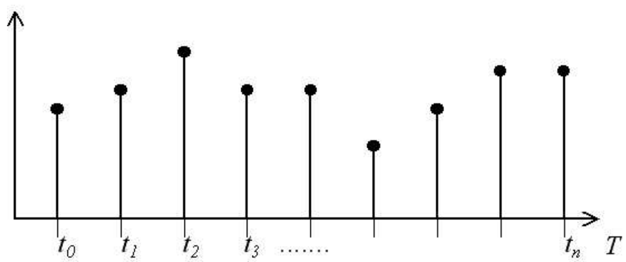
Slide 11

- correspondence between piecewise constant and event segments
  - event segment →piecewise constant segment
    * let $\omega :< t_0, t_n > \rightarrow A \cup \{\varnothing\}$ be an event segment with event times $t_1, t_2, t_3, \ldots, t_{n-1} \in < t_0, t_n >$
    * corresponding piecewise constant segment $w^{'} :< t_0, t_n > \rightarrow A$ by $\omega^{'}(t) = \omega(t_x)$ with t_x being the largest time in $\{t_1, t_2, t_3, \ldots, t_{n-1}\}$ with $t_x \leq t$
      if such a number t_x does not exist, then $\omega^{'}(t) = \varnothing$
  - piecewise constant segment →event segment
    * let $w^{'} :< t_0, t_n > \rightarrow A$ be a piecewise constant segment with event times $t_1, t_2, t_3, \ldots, t_{n-1} \in < t_0, t_n >$ and constant values $c_1, c_2, \ldots, c_n \in A$
    * corresponding event trajectory $\omega :< t_0, t_n > \rightarrow A \cup \{\varnothing\}$ by
      $\omega(t) = \omega^{'}(t) = c_{i+1}$ if $t = t_i \in \{t_0, t_1, t_2, t_3, \ldots, t_{n-1}\}$ and $\omega(t) = \varnothing$ otherwise

Segments and trajectories (Cont'd)

Slide 12

- sequences
  - all segments over a discrete time base are called *finite sequences* or for short just *sequences*
  - sequences are employed in discrete time models as input, state, and output trajectories



I/O observation frame

Slide 13

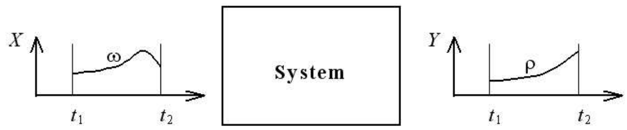
- IO $= (T, X, Y)$

  where
  - T is a time base
  - X is a set–the input values set
  - Y is a set–the output values set
- very simply
  - we have set X of values that can appear at the input of the system and an abstract set Y of values that can appear at its output
- it is a black box specification

I/O relation observation

Slide 14

- apply to the input a segment from (X, T) and observe at the output a segment from (Y, T)
- an input segment ($\omega \in (X, T)$) and an output segment ($\rho \in (Y, T)$) →input-output pair



- collection of I/O pair →I/O relation
- IORO $= (T, X, \Omega, Y, R)$

  where $T, X,$ and Y are the same as for IO frame and
  - Ω is a set–the set of allowable input segments
  - R is a relation–the I/O relation with the constraints that $(a)\Omega \subseteq (X, T)$ and $(b)R \subseteq \Omega \times (Y, T)$ where $(\omega, \rho) \in R \Rightarrow$ dom(ω) = dom(ρ)

I/O relation observation (Cont'd)

Slide 15

- an example
  - a differential equation

$$\frac{d^3y}{dt^3} + \frac{2d^2y}{dt^2} + \frac{8dy}{dt} + 7y = x$$

  - we have $T = X = Y = \Re$
  - Ω is not specified by the equation, but one choice is the set of bounded piecewise continuous segments (bpc)

    $R = \{(\omega, \rho) | \omega \in bpc,$ dom(ω) = dom$(\rho)\}$ and

    $\frac{d^3\rho(t)}{dt^3} + \frac{2d^2\rho(t)}{dt^2} + \frac{8d\rho(t)}{dt} + 7\rho(t) = \omega(t)$ for all $t \in$ dom(ω)

I/O function observation

Slide 16

- I/O relation observation
  - a set of output segments with allowable input segments
  - depending on its inner (unknown) state, the system may respond with different output segments to a particular input segment
- I/O function observation
  - we can partition the relation R into a set of functions $F = \{f_1, f_2, \ldots, f_i, \ldots\}$ such that for each function f_i produces one unique output segment $\rho = f_i(\omega)$
- IOFO $= (T, X, \Omega, Y, F)$

  where T, X, Y, Ω are the same as for IORO and F is a set of I/O functions with the constraint that $f \in F \Rightarrow f \subseteq \Omega \times (Y, T)$ is a function, and if $\rho = f(\omega)$, then dom(ρ) = dom(ω)
- association
  - given an IOFO (T, X, Ω, Y, F), we associate with it an IORO (T, X, Ω, Y, R) where $R = \cup_{f \in F} f$

I/O function observation (Cont'd)

Slide 17

- an example
  - a differential equation

$$\frac{d^3y}{dt^3} + \frac{2d^2y}{dt^2} + \frac{8dy}{dt} + 7y = x$$

    $F = \{f_{a,b,c} | a, b, c \in R\}$, where $f_{a,b,c} : \Omega \to (Y, T)$ is defined by $f_{a,b,c}(\omega) = \rho$ where $\frac{d^3\rho(t)}{dt^3} + \frac{2d^2\rho(t)}{dt^2} + \frac{8d\rho(t)}{dt} + 7\rho(t) = \omega(t)$ for all $t \in$ dom$(\omega) =< t_1, t_2 >$ and $\frac{d^2\rho(t_1)}{dt^2} = a, \frac{d\rho(t_1)}{dt} = b, \rho(t_1) = c$
  - in general, given a differential equation operator L of order n, we require n parameters to specify an I/O function, corresponding to the specification of the initial values of the derivatives $\frac{d^{n-1}y}{dt^{n-1}}, \cdots, \frac{dy}{dt}, y$

I/O system

Slide 18

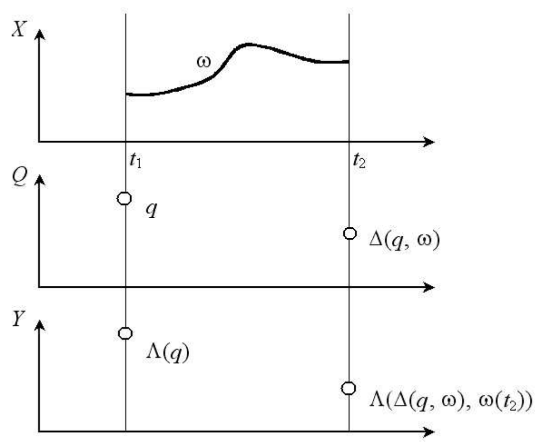
- I/O function observation
  - description of the initial state in that if it is in a state represented by f, then the input segment ω will yield a unique response $f(\omega)$
  - but not its immediate and final states
- I/O system
  - employ a state set and state transition function for modeling the interior of a system
  - the future is uniquely determined by the current state and the future input →*semigroup* or *composition property*
- S $= (T, X, \Omega, Y, Q, \Delta, \Lambda)$

  where $T, X, Y,$ and Ω are the same as before and
  - Q is the set of states
  - $\Delta : Q \times \Omega \rightarrow Q$ is the global state transition function
  - $\Lambda : Q \times X \rightarrow Y$(or $\Lambda : Q \rightarrow Y$) is the output function
  - subject to the following constraints

I/O system (Cont'd)

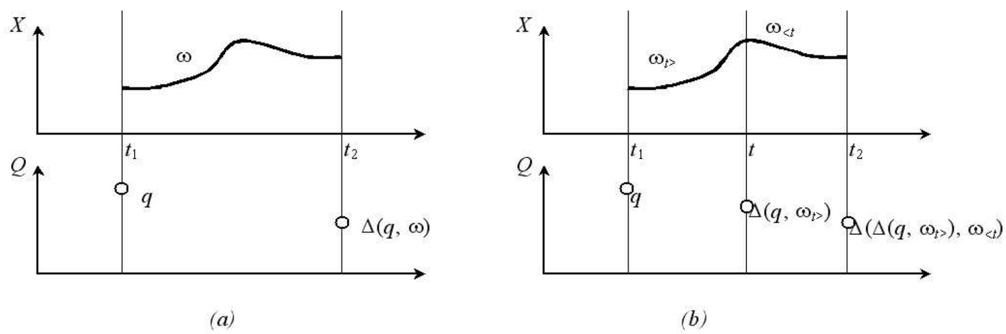
Slide 19

1. *closure property*: Ω is closed under concatenation as well as under left segmentation
2. *composition (or semigroup) property*: for every pair of contiguous input segments $\omega, \omega' \in \Omega, \Delta(q, \omega \bullet \omega') = \Delta((\Delta(q, \omega), \omega')$



I/O system (Cont'd)

Slide 20



I/O system (Cont'd)

Slide 21

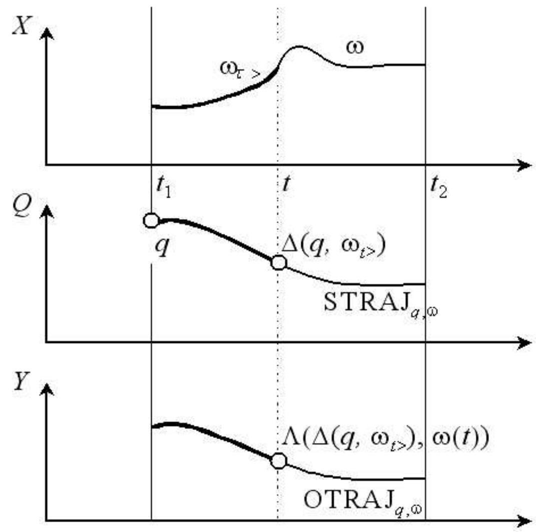
- going from system structure to behavior
  - internal structure →generate its behavior
  - I/O system structure S →I/O function observation and an I/O relation observation
  - state trajectory and output trajectory
    - global state transition function only determines the final state
    - but admissible input segments are closed under left segmentation
    - we can define a unique state and output trajectory given an input segment and an initial state

  → for every $q \in Q$ and for every $\omega \in \Omega, \omega :< t_1, t_2 > \rightarrow X$, $\text{STRAJ}_{q,w} :< t_1, t_2 > \rightarrow Q$ with $\text{STRAJ}_{q,w}(t) = \Delta(q, \omega_{t>})$ for all $t \in < t_1, t_2 >$

  → and $\text{OTRAJ}_{q,w} :< t_1, t_2 > \rightarrow Y$ with $\text{OTRAJ}_{q,w}(t) \ = \Lambda(\text{STRAJ}_{q,w}(t), \omega(t))$ for all $t \in < t_1, t_2 >$

I/O system (Cont'd)

Slide 22



I/O system (Cont'd)

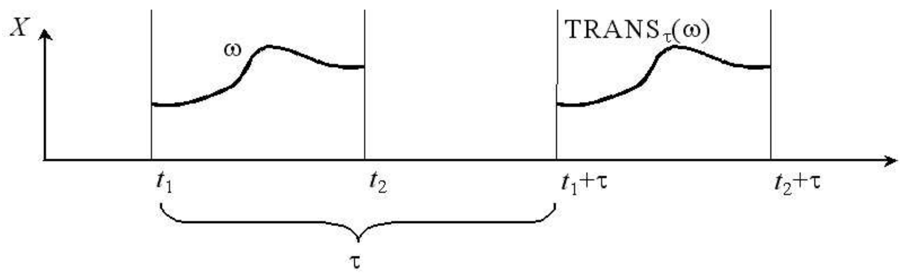
Slide 23

- I/O function of state q
  * $\tilde{\beta}_q : \Omega \rightarrow (Y, T)$
  * $\tilde{\beta}_q(w) =$ OTRAJ$_{q,w}$ where for each $\omega \in \Omega$
  * the set of functions $\tilde{\beta}_S = \{\tilde{\beta}_q | q \in Q\}$ is called the *I/O behavior of S* (IOFO)
  * the union of all functions $\tilde{\beta}_q \in \tilde{\beta}_S$ (IORO) where $R_S = \{(\omega, \rho) | \ \omega \in \Omega, \rho =$ OTRAJ$_{q,w}, q \in Q\}$
- the final output function
  * with each state $q \in Q$, a function $\beta : \Omega \rightarrow Y$
  * $\beta_q(\omega) = \Lambda(\Delta(q, w))$
  * the relationship between $\tilde{\beta}_q$ and β_q is given by $\beta_q(\omega_{t>}) = \tilde{\beta}_q(w)(t)$ for $t \in$ dom(ω)

I/O system (Cont'd)

- time-invariant systems
  - translation operator
    - define a unary operator on segments TRANS$_\tau$: $(X, T) \rightarrow (X, T)$ where if $\omega :< t_1, t_2 >\rightarrow X$ and TRANS$_\tau(\omega) = \omega^{'}$, then $\omega^{'} :< t_1 + \tau, t_2 + \tau >\rightarrow X$ and $\omega^{'}(t + \tau) = \omega(t)$ for all $t \in< t_1, t_2 >$

Slide 24



    - $\Omega \subseteq (X, T)$ is closed under translation if $\omega \in \Omega$ for every $\tau \in T$, TRANS$_\tau(\omega) \in \Omega$

I/O system (Cont'd)

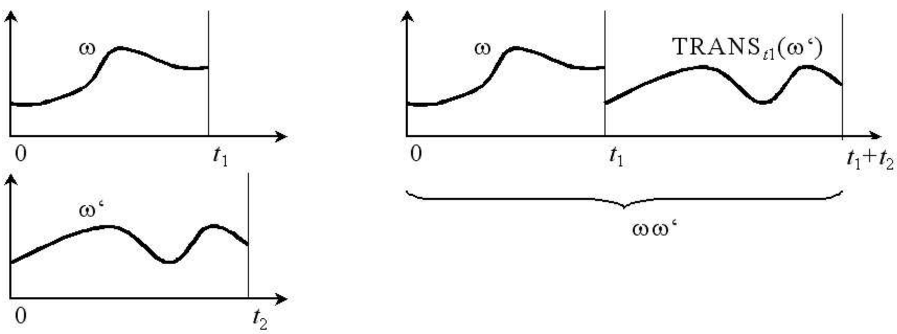
  - time-invariant systems
    - a system $S = (T, X, \Omega, Y, Q, \Delta, \Lambda)$ is time invariant if
      a) Ω is closed under translation
      b) Δ is time independent: for every $\tau \in T, \omega \in \Omega$, and $q \in Q$
      $\Delta(q, \omega) = \Delta(q,$ TRANS$_\tau(\omega))$
    - anywhere the same shaped segment is applied to the same initial state, the same result will be achieved

Slide 25

  - composition of segment
    - two segments beginning at zero are concatenated
    - concatenation operation $\bullet_0 : \Omega_0 \times \Omega_0 \rightarrow \Omega_0$ by $\omega \bullet_0 \omega^{'} = \omega \bullet_0$ TRANS$_{l(\omega)}(\omega^{'})$ for all $\omega, \omega^{'} \in \Omega_0$ and STR$(\omega) = TRANS_{-t_1}(\omega)$ for $\omega :< t_1, t_2 >\rightarrow X$
    - $S_0 = (T, X, \Omega_0, Y, Q, \Delta_0, \Lambda)$ where T, X, Y, Q, and Λ are the same as before but Ω_0 is the set of segments beginning at time 0 and $\Delta_0 : Q \times \Omega_0 \rightarrow Q$ satisfies the composition property: for all $q \in Q, \omega, \omega^{'} \in \Omega_0, \Delta_0(q, \omega \bullet \omega^{'}) = \Delta_0(\Delta_0(q, \omega), \omega^{'})$
    - expanded form of S_0 is the structure, $S = (T, X, \Omega, Y, Q, \Delta, \Lambda)$, where Ω is the translation closure of Ω_0 and $\Delta : Q \times \Omega \rightarrow Q$ is defined by $\Delta(q, \omega) = \Delta_0(q,$ STR$(\omega))$ for all $q \in Q, \omega \in \Omega$

I/O system (Cont'd)

Slide 26



Iterative specification of systems

Slide 27

- I/O system
  - describe system behavior with a global transition function that determines the final state given the initial state and the applied input segment
  - since the input segments are left segmentable and closed under composition, the state and output values along the entire input interval can be generated

  → such an approach is not very practical
  - we need a way to generate state and output trajectories in an iterative way going from one state along the trajectory to the next
- iterative specification
  - generator segments
    - consider (Z,T) the set of all segments $\{\omega :< 0, t_1 >\rightarrow Z | t_1 \in T\}$, which is a semigroup under concatenation
    - for a subset Γ of (Z,T), we designate by Γ^+ the concatenation closure of Γ, then Γ^+ can be constructed as follows
    - let $\Gamma^1 = \Gamma$ and $\Gamma^{i+1} = \Gamma^i \bullet \Gamma = \{\omega\omega' | \omega \in \Gamma^i$ and $\omega' \in \Gamma\}$
    - then $\Gamma^+ = \cup_{i \in \Im} \Gamma^i$

Iterative specification of systems (Cont'd)

Slide 28

- Γ generates Ω
  - given a set of segments Ω, we are interested in finding a set Γ with the property that $\Gamma^+ = \Omega$
  - we say that $\omega_1, \omega_2, \ldots, \omega_n$ is a decomposition of ω by Γ if $\omega \in \Gamma$ for each $i = 1, \ldots, n$ and $\omega = \omega_1 \bullet \omega_2 \bullet \cdots \bullet \omega_n$
  - we cannot expect a unique decomposition by Γ
  - we are interested in selecting a single representative, or canonical decomposition $\rightarrow$maximal length segmentation
- maximal length segmentation
  - first find ω_1, the longest generator in Γ that is also left segment of ω
  - this process is repeated with what remains of ω after ω_1 is removed, generating ω_2, and so on

  $\rightarrow$ maximal length segment (mls) decomposition
- admissible
  - Γ is an admissible set of generators for Ω if Γ generates Ω and for each $\omega \in \Omega$, a unique mls decomposition of ω by Γ exists
  - Γ admissibly generates Ω

Iterative specification of systems (Cont'd)

Slide 29

  - sufficient conditions for admissibility
    a) existence of longest segments: $\omega \in \Gamma^+ \Rightarrow$ max$\{t|\omega_{t>} \in \Gamma\}$ exists
    b) closure under right segmentation: $\omega \in \Gamma \Rightarrow \omega_{<t} \in \Gamma$ for all $\tau \in$ dom(ω)

- generator state transition systems
  - let Ω_G be an admissible generating set for Ω and a function $\delta : Q \times \Omega_G \rightarrow Q$ is a single segment transition function
  - let $\omega_1, \omega_2, \ldots, \omega_n$ be the mls decomposition of ω and δ defined for each segment ω_i
  - if q_0 is the initial state, $q_1 = \delta(q_0, \omega_1)$ is the state after injecting ω_1, $q_2 = \delta(q_1, \omega_2)$ is the state after injecting ω_2, and so on
  - generator state transition
    - a function $\delta^+ : Q \times \Omega_G^+ \rightarrow Q$, called the extension of δ to Ω_G^+ by

$$\delta^+(q,\omega) = \begin{cases} \delta(q,\omega) & \text{if } \omega \in \Omega_G \\ \delta^+(\delta(q,\omega_1), \omega_2 \bullet \cdots \bullet \omega_n) & \text{if } \omega \in \Omega_G^+ \end{cases}$$

    where $\omega_1 \bullet \omega_2 \bullet \cdots \bullet \omega_n$ is the mls decomposition for ω

Iterative specification of systems (Cont'd)

Slide 30

- iterative specification
  - $G = < T, X, \Omega_G, Y, Q, \delta, \lambda >$
    where T, X, Y, Q have the same interpretation as for I/O system,
    - Ω_G is the set of input segment generators
    - $\delta : Q \times \Omega_G \rightarrow Q$ is the single segment state transition function
    - $\lambda : Q \times X \rightarrow Y$ is the output function
  - sufficient conditions for iterative specification
    (a) existence of longest segments: $\omega \in \Omega_G^+ \Rightarrow$ max$\{t | \omega_{t>} \in \Omega_G\}$ exists
    (b) closure under right segmentation: $\omega \in \Omega_G \Rightarrow \omega_{<t} \in \Omega_G$ for $t \in$ dom(ω)
    (c) closure under left segmentation: $\omega \in \Omega_G \Rightarrow \omega_{t>} \in \Omega_G$ for $t \in$ dom(ω)
    (d) consistency of composition: $\omega_1, \omega_2, \ldots, \omega_n \in \Omega_G$ and $\omega_1 \bullet \omega_2 \bullet \cdots \bullet \omega_n \in \Omega_G^+ \Rightarrow \delta^+(q, \omega_1 \bullet \omega_2 \bullet \cdots \bullet \omega_n) = \delta(\delta(\cdots \delta(\delta(q, \omega_1), \omega_2), \cdots), \omega_n)$

Network of system specifications (coupled systems)

Slide 31

- a higher level of specification
  - component systems are coupled through their output and input interfaces
  - $\rightarrow$ coupled system specification
  - $\rightarrow$ network of system specification
- coupled system specification
  - $N = < T, X_N, Y_N, D, \{M_d | d \in D\}, \{I_d | d \in D \cup \{N\}\}, \{Z_d | d \in D \cup \{N\}\} >$
    where
    - X_N is the set of inputs of the network–the external inputs
    - Y_N is the set of outputs of the network–the external outputs
    - D is the set of component references

Network of system specifications (coupled systems) (Cont'd)

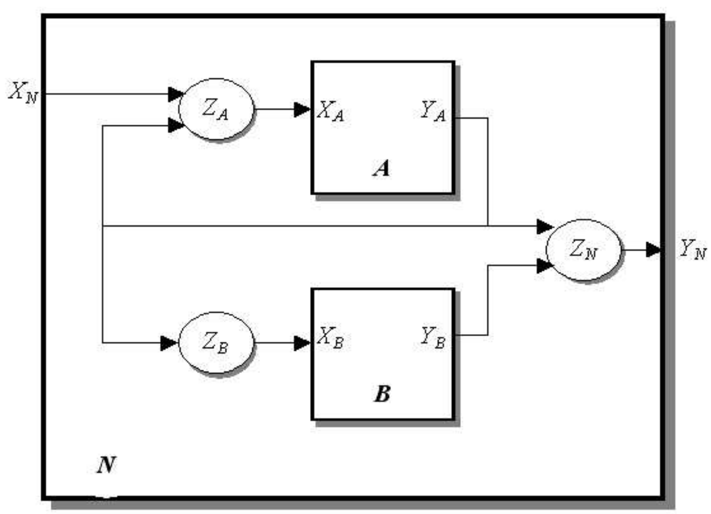
Slide 32

* for all $d \in D$, $M_d =< T, X_d, Y_d, \Omega, Q, \Delta, \Lambda >$ is an I/O system
* for all $d \in D \cup \{N\}$,
  · $I_d \subseteq D \cup \{N\}$ is the set influencers of d
  · $Z_d : \times_{i \in I_d} YX_i \rightarrow XY_d$ is the interface map for d
  with

$$\begin{aligned} YX_i &= X_i && \text{if } i = N \\ &= Y_i && \text{if } i \neq N \\ XY_d &= Y_d && \text{if } d = N \\ &= X_d && \text{if } i \neq N \end{aligned}$$

Network of system specifications (coupled systems) (Cont'd)

Slide 33



$I_A = \{N, A\}$, $I_B = \{A\}$, $I_N = \{A, B\}$

Network of system specifications (coupled systems) (Cont'd)

Slide 34

- coupled system specification at the structured system level
  - coupling is done by directly connecting output and input ports
  - almost exclusively used in modeling practice
  - coupling is specified by three parts
    - the *external input coupling (EIC)*: coupling of network input ports to input ports of some components
    - the *external output coupling (EOC)*: coupling of component output ports to output ports of the network
    - the *internal coupling (IC)*: coupling of component output ports to output ports of components
  - $N =< T, X_N, Y_N, D, \{M_d | d \in D\}, EIC, EOC, IC >$

Network of system specifications (coupled systems) (Cont'd)

Slide 35