Sheaf-theoretic analysis of networked dynamical systems

David I. Spivak

dspivak@math.mit.edu
Mathematics Department
Massachusetts Institute of Technology

Presented on 2017/06/08
at Emerging Topics in Network Dynamical Systems

Outline

A behavioral and compositional approach

Today I'll take a behavioral and compositional approach to systems.

- Behavioral: concentrate on how systems act, not how they're made.
  - I won't restrict myself to continuous, discrete, or hybrid systems.
  - Any type of behavior will be fine, as long as it occurs over time.
  - I'll make this precise by introducing *sheaves* (cf. alg. geometry).

A behavioral and compositional approach

Today I'll take a behavioral and compositional approach to systems.

- Behavioral: concentrate on how systems act, not how they're made.
  - I won't restrict myself to continuous, discrete, or hybrid systems.
  - Any type of behavior will be fine, as long as it occurs over time.
  - I'll make this precise by introducing *sheaves* (cf. alg. geometry).
- Compositional: we care about what happens when systems interact.
  - I'll discuss compositional information (e.g. bifurcation diagrams compose nicely),
  - And the compositional nature of "behavior contracts".

Main source of examples

Examples will be based on the US National Airspace System (NAS).

- It consists of airplanes, radars, pilots, air traffic control, etc.
- They are interacting together to compose a network.
  - And every node in it is composed of a sub-network.
  - E.g., the airplane is composed of motors, transponders, pilot, etc.
  - The NAS is a network of networks, a system of systems.

Main source of examples

Examples will be based on the US National Airspace System (NAS).

- It consists of airplanes, radars, pilots, air traffic control, etc.
- They are interacting together to compose a network.
  - And every node in it is composed of a sub-network.
  - E.g., the airplane is composed of motors, transponders, pilot, etc.
  - The NAS is a network of networks, a system of systems.
- Compositionality is all about chunking.
  - You can chunk any way you want (e.g. logically or physically).
  - The outcome is the same, however you chunk (associative law).
  - Choice of chunking should align with the problem to solve.

Behaviors in the national airspace system

Each node in the network has a certain type of possible behavior.

- When we chunk subnetworks, the behaviors are combined accordingly.
  - When parts interact, they influence and constrain each other.
  - All this interacting behavior creates the behavior of the whole.

Behaviors in the national airspace system

Each node in the network has a certain type of possible behavior.

- When we chunk subnetworks, the behaviors are combined accordingly.
  - When parts interact, they influence and constrain each other.
  - All this interacting behavior creates the behavior of the whole.
- The positional behavior of an airplane could be modeled as an ODE.
  - It is governed by control yoke (steering) and throttle (gas).
  - These are modeled as time-varying parameters, e.g. $a(t)$ in $\dot{x}(t) = f(x(t), a(t))$
  - A parameter, $\dot{x}(t) = f(x(t), a)$, is a special case with a constant.
- Other NAS elements might be modeled by labeled transition systems.
  - E.g. the TCAS: Traffic Collision Avoidance System.
  - It takes radar data and converts it into signals like *"climb!"*

We will model such types of behavior as sheaves.

Plan of the talk

- Discuss compositionality, including interfaces and networks.
- Describe their inhabitants: machines that have a type of behavior.
- Explain how behaviors compose; discuss compositional properties.
- Introduce higher-order temporal logic, composition of contracts.
- Conclude with a summary.

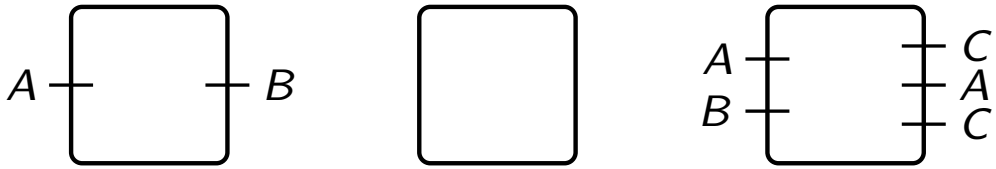
Outline

Interfaces

An *interface* is what connects an inner world to an outer world.

- We will draw interfaces as squares with ports.
  - Set-theoretically, an interface is just a finite set P, for "ports",
  - Where every $p \in P$ is labeled by a behavior type.
  - (We'll discuss behavior types in detail in the next section.)

Some interfaces:



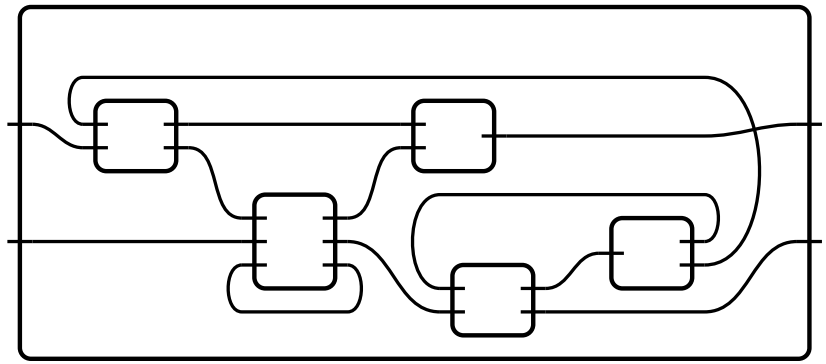
Sometimes we leave the labels off for typographical reasons.

- The next step will be to wire these together to form a network.

Wiring diagrams

Interfaces can be wired together inside another interface.

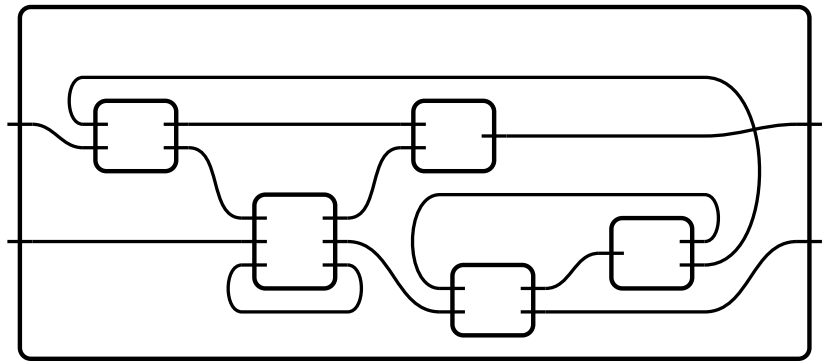
- When two interfaces share a wire, their behaviors are linked.
  - You might say they are "sharing a variable."
  - This makes sense even if one of them is the exterior interface.



Wiring diagrams

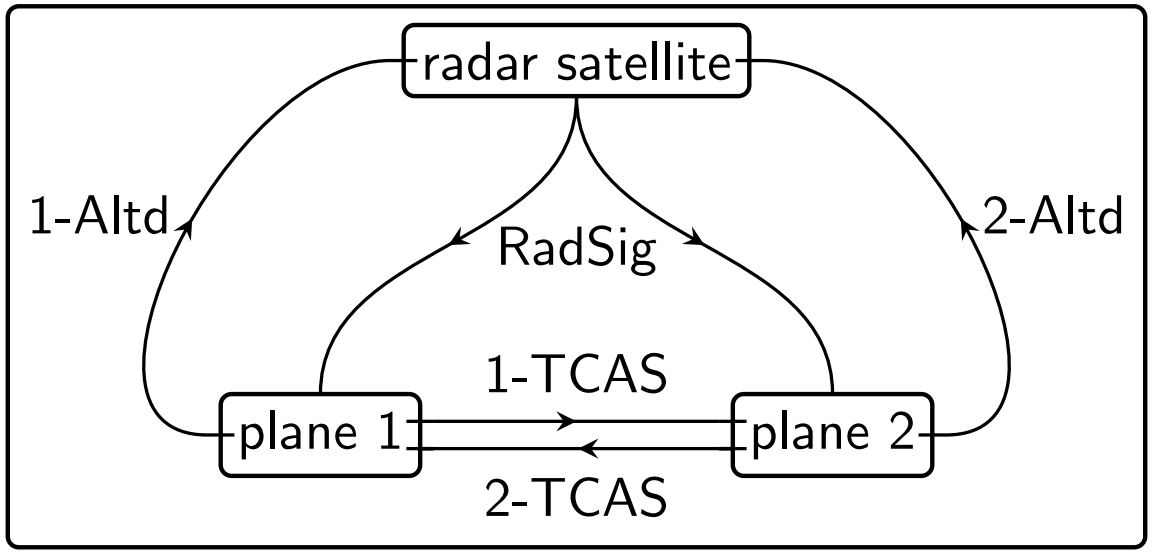
Interfaces can be wired together inside another interface.

- When two interfaces share a wire, their behaviors are linked.
  - You might say they are "sharing a variable."
  - This makes sense even if one of them is the exterior interface.

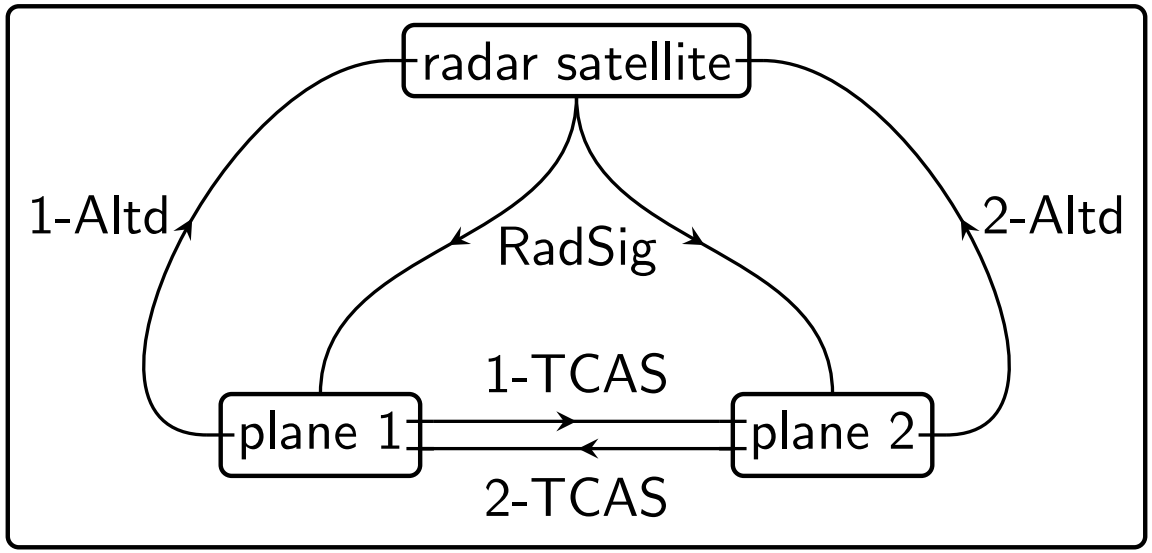


- This nesting of networks inside of interfaces is "fractal".
  - There are boxes within boxes, and chunking is associative.
  - The category-theoretic formalism for this is called an *operad*.
  - But we won't discuss operads today.

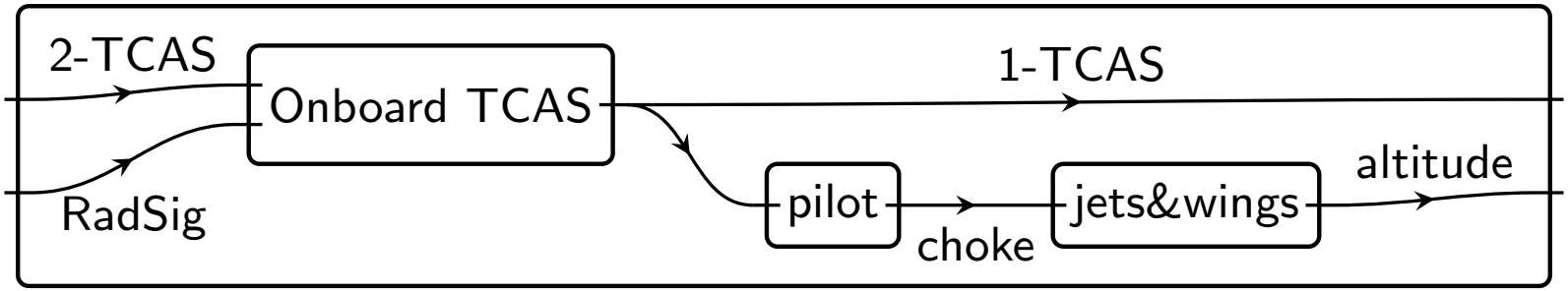
A network


radar satellite
1-Altd
RadSig
2-Altd
1-TCAS
plane 1
plane 2
2-TCAS

A network



Zooming into plane 1:



Syntax vs. semantics

These interfaces and wiring diagrams are a form of *syntax*.

- Manipulating them, e.g. by nesting, is the main syntactic operation.
  - Compose interfaces in parallel or series.
  - Include feedback, splitting, etc. It's all compositional.
  - (The operadic language specifies all this syntax.)

Syntax vs. semantics

These interfaces and wiring diagrams are a form of *syntax*.

- Manipulating them, e.g. by nesting, is the main syntactic operation.
  - Compose interfaces in parallel or series.
  - Include feedback, splitting, etc. It's all compositional.
  - (The operadic language specifies all this syntax.)

Semantics is any model of this wiring diagram syntax.

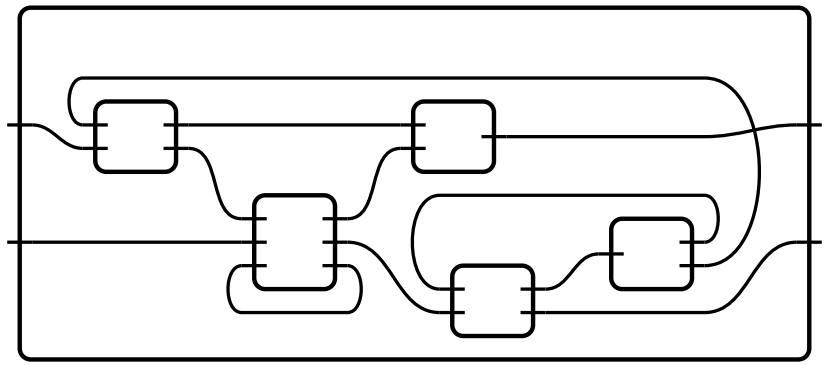
- There are many different semantic models for wiring diagrams.
  - E.g., WDs are a syntax for quantum processes (Coecke, Kissinger).
  - WDs are a syntax for set-theoretic relations.
  - WDs are a syntax for matrices.
  - WDs are a syntax for discrete dynamical systems.
  - We'll use WDs as syntax for combining general types of behavior.

We'll discuss semantics next.

Outline

Behaviors on wires and in machines

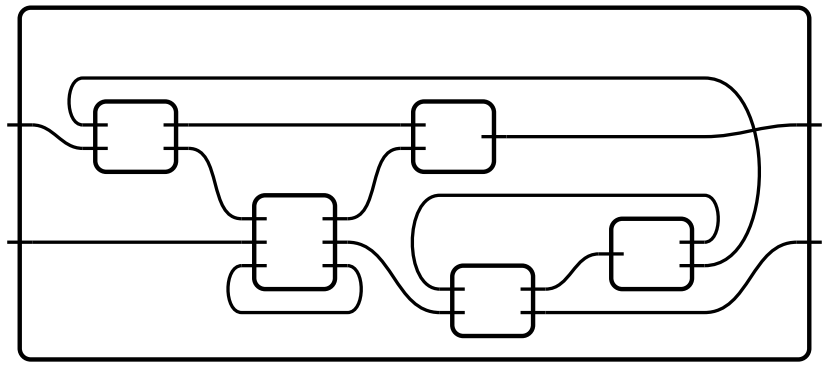
Where do we find behaviors taking place in a wiring diagram?



- Each interface houses a machine, which has some internal behavior.
- But that behavior is only exposed to others via wires.
  - A wire carries a type of signals; we're calling that a behavior too.
  - A machine's behavior constrains the signal behaviors on its ports.

Behaviors on wires and in machines

Where do we find behaviors taking place in a wiring diagram?



- Each interface houses a machine, which has some internal behavior.
- But that behavior is only exposed to others via wires.
  - A wire carries a type of signals; we're calling that a behavior too.
  - A machine's behavior constrains the signal behaviors on its ports.
- My behavior is the relationship between what I see and what I do.

My term behavior type includes that of machines and that of wire signals.

Informal introduction to behavior types as sheaves

What do we mean by a type of behavior B?

- It's a specification of "what can happen over intervals of time"...
  - What's possible over a 1-minute interval? 20 seconds?
  - Say $x \in B(60)$ or $y \in B(20)$.

Informal introduction to behavior types as sheaves

What do we mean by a type of behavior B?

- It's a specification of "what can happen over intervals of time"...
  - What's possible over a 1-minute interval? 20 seconds?
  - Say $x \in B(60)$ or $y \in B(20)$.
- ...and how these possibilities restrict to subintervals.
  - If $x \in B(60)$, what is its restriction $x\big|_{(30,50)}$?

Informal introduction to behavior types as sheaves

What do we mean by a type of behavior B?

- It's a specification of "what can happen over intervals of time"...
  - What's possible over a 1-minute interval? 20 seconds?
  - Say $x \in B(60)$ or $y \in B(20)$.
- ...and how these possibilities restrict to subintervals.
  - If $x \in B(60)$, what is its restriction $x\big|_{(30,50)}$?
- And—if you're feeling fancy—how do behaviors stitch together?
  - Suppose behaviors agree on some overlap: $x\big|_{(50,60)} = y\big|_{(0,10)}$.
  - We can imagine stitching them to form a behavior in $B(70)$.

Two topological spaces

Consider the (non-Hausdorff) topological space $\mathbb{IR}$ defined as follows:

- A point in $\mathbb{IR}$ is any closed interval $[a, b] \subseteq \mathbb{R}$.
- For any $d < u$, the set of closed intervals $[a, b] \subseteq (d, u)$ is open.
- These (d, u)'s form a basis for the topology on $\mathbb{IR}$.
  - So notice that $(0, 5) \neq (0, 3) \cup (2, 5)$ because it misses $[1, 4]$.
  - However, $(0, 5) = \bigcup_{n \in \mathbb{N}} (\frac{1}{n+1}, 5)$.

Two topological spaces

Consider the (non-Hausdorff) topological space $\mathbb{IR}$ defined as follows:

- A point in $\mathbb{IR}$ is any closed interval $[a, b] \subseteq \mathbb{R}$.
- For any $d < u$, the set of closed intervals $[a, b] \subseteq (d, u)$ is open.
- These (d, u)'s form a basis for the topology on $\mathbb{IR}$.
  - So notice that $(0, 5) \neq (0, 3) \cup (2, 5)$ because it misses $[1, 4]$.
  - However, $(0, 5) = \bigcup_{n \in \mathbb{N}} (\frac{1}{n+1}, 5)$.
- This space is a *domain*, (the "interval domain"), if that rings a bell.
  - Some points contain others; call the smaller ones *specializations*.
  - One often writes $[a, d] \sqsubseteq [b, c]$ if $a \leq b \leq c \leq d$.
  - In a domain, the poset of points determines the lattice of opens.
  - And the lattice of open sets determines the poset of points.
- The subspace of maximal pts in $\mathbb{IR}$ is homeomorphic to the usual $\mathbb{R}$.
  - Because the maximal points are $[a, a]$.
  - $\mathbb{R}$ has the same basis, but there we have $(0, 5) = (0, 3) \cup (2, 5)$.

What is a sheaf on a topological space X?

A sheaf is a data structure associated to a topological space.

- Let X be a topological space and Ω_X its poset of open sets.
- A *sheaf B on X* is an assignment of lots of sets and functions:

What is a sheaf on a topological space X?

A sheaf is a data structure associated to a topological space.

- Let X be a topological space and Ω_X its poset of open sets.
- A *sheaf B on X* is an assignment of lots of sets and functions:
  - To each open set $U \in \Omega_X$, we assign a set $B(U)$.
  - To each subset relation $V \subseteq U$, we assign a function $\rho_{V,U} \colon B(U) \to B(V)$.
  - We will call each element $b \in B(U)$ a *behavior*.
  - Given $b \in B(U)$, we will call $\rho_{V,U}(b) \in B(V)$ its *restriction* to V.

What is a sheaf on a topological space X?

A sheaf is a data structure associated to a topological space.

- Let X be a topological space and Ω_X its poset of open sets.
- A *sheaf B on X* is an assignment of lots of sets and functions:
  - To each open set $U \in \Omega_X$, we assign a set $B(U)$.
  - To each subset relation $V \subseteq U$, we assign a function $\rho_{V,U} \colon B(U) \to B(V)$.
  - We will call each element $b \in B(U)$ a *behavior*.
  - Given $b \in B(U)$, we will call $\rho_{V,U}(b) \in B(V)$ its *restriction* to V.
- To be a sheaf, the $B(U)$'s and ρ's need to satisfy three conditions:
  - Of course $\rho_{U,U} = \mathrm{id}_{B(U)}$.
  - Whenever $W \subseteq V \subseteq U$, we have $\rho_{W,U} = \rho_{W,V} \circ \rho_{V,U}$.
  - Whenever $U = \bigcup_{i \in I} V_i$, the set of behaviors on U is determined by those on the V_i:

$$B(U) \cong \{(b_i \in V_i)_{i \in I} \mid \rho_{V_i, V_i \cap V_j}(b_i) = \rho_{V_j, V_i \cap V_j}(b_j)\}.$$

Behaviors types as sheaves

- What's a sheaf on $\mathbb{IR}$?
  - It assigns a set $B(a, b)$ to each basis open (a, b).
  - This is the set of possible behaviors that can occur on (a, b).
  - It assigns a function $B(a, b) \to B(a', b')$ when $a \leq a' < b' \leq b$.
  - A behavior on (a, b) is restricted to a behavior on (a', b').
  - Behaviors on $\bigcup_{i \in I} V_i$ are determined by behaviors on the V_i.

Behaviors types as sheaves

- What's a sheaf on $\mathbb{IR}$?
  - It assigns a set $B(a, b)$ to each basis open (a, b).
  - This is the set of possible behaviors that can occur on (a, b).
  - It assigns a function $B(a, b) \to B(a', b')$ when $a \leq a' < b' \leq b$.
  - A behavior on (a, b) is restricted to a behavior on (a', b').
  - Behaviors on $\bigcup_{i \in I} V_i$ are determined by behaviors on the V_i.
- What's a sheaf on $\mathbb{R}$?
  - Same description! What's the difference?
  - Fewer open sets in $\mathbb{R}$ than in $\mathbb{IR}$: $(0, 3) \cup (2, 5) =^{?} (0, 5)$.

Translation invariance

I'll summarize sheaves on the next slide, but one more thing.

- We often want to consider translation-invariant behaviors B:
  - Translation-invariant means independent of a global clock.
  - That is, there are compatible isomorphisms $B(a, b) \cong B(a + r, b + r)$ for any r and $a < b$.
  - For any length $\ell > 0$, we have a set $B(\ell)$.
  - Translation-invariant implies: $B(0, 10) \cong B(10) \cong B(5, 15)$.
- What I call a behavior type is a translation-invariant sheaf.
- Sheaves actually make sense on generalizations of spaces, called *sites*.

Translation invariance

I'll summarize sheaves on the next slide, but one more thing.

- We often want to consider translation-invariant behaviors B:
  - Translation-invariant means independent of a global clock.
  - That is, there are compatible isomorphisms $B(a, b) \cong B(a + r, b + r)$ for any r and $a < b$.
  - For any length $\ell > 0$, we have a set $B(\ell)$.
  - Translation-invariant implies: $B(0, 10) \cong B(10) \cong B(5, 15)$.
- What I call a behavior type is a translation-invariant sheaf.
- Sheaves actually make sense on generalizations of spaces, called *sites*.
  - There is a site **Int** whose sheaves are translation-invariant sheaves on $\mathbb{IR}$.
  - There's also a site for translation-invariant sheaves on $\mathbb{R}$.
- There are many settings (sites) on which we can discuss sheaves.

Recap

From now on, when we discuss a behavior type B we mean

- Formally, it's a translation-invariant sheaf B on $\mathbb{IR}$.
  - An assignment of sets and functions:
  - To each open (a, b), we have a set $B(a, b)$ of behaviors
  - Subject to translation-invariance: $B(a, b) \cong B(a + r, b + r)$.
  - Whenever $a \leq a' < b' \leq b$ we have $\rho\colon B(a, b) \to B(a', b')$.
- Informally a sheaf specifies "a space of possibilities".

Recap

From now on, when we discuss a behavior type B we mean

- Formally, it's a translation-invariant sheaf B on $\mathbb{IR}$.
  - An assignment of sets and functions:
  - To each open (a, b), we have a set $B(a, b)$ of behaviors
  - Subject to translation-invariance: $B(a, b) \cong B(a + r, b + r)$.
  - Whenever $a \leq a' < b' \leq b$ we have $\rho\colon B(a, b) \to B(a', b')$.
- Informally a sheaf specifies "a space of possibilities".
- Examples: over an interval of length ℓ
  - ODE: B follows any length-ℓ trajectory of a given ODE.
  - Graph: B traverses edges in a given graph for ℓ seconds.
  - Stopwatch: B runs a stopwatch with any start time for ℓ seconds.
  - Delay: B is running two behaviors b_1, b_2 of a given type B' for ℓ seconds, and b_1 is d-seconds behind b_2 for a given d.

The topos of sheaves on $\mathbb{IR}$

The category of behavior types is called a *topos*. It includes

- Behavior types X, Y, Z (these are sheaves.)
- An ability to multiply behavior types $X \times Y$ (behaviors are pairs)
- Maps of behavior types $X \to Y$, and compositions thereof.
  - In fact, there is a type Y^X recording all maps $X \to Y$.

The topos of sheaves on $\mathbb{IR}$

The category of behavior types is called a *topos*. It includes

- Behavior types X, Y, Z (these are sheaves.)
- An ability to multiply behavior types $X \times Y$ (behaviors are pairs)
- Maps of behavior types $X \to Y$, and compositions thereof.
  - In fact, there is a type Y^X recording all maps $X \to Y$.
- An interesting type of behavior, called Prop.
  - Technically, Prop is called the subobject classifier of the topos.
  - Logically, it's the type of propositions (akin to true false).
  - We'll discuss Prop as a sheaf next.

Toposes are very rich structures, connecting geometry and logic.

Prop and predicates

Prop is like an auditor's record book: "compliant here...".

- $\mathrm{Prop}(a, b)$ is the set of open subsets $U \subseteq (a, b)$. "Compliant on U"
- Given $a \leq a' < b' \leq b'$ the map $\rho\colon \mathrm{Prop}(a, b) \to \mathrm{Prop}(a', b')$ is given by $U \mapsto U \cap (a', b')$.

Prop and predicates

Prop is like an auditor's record book: "compliant here...".

- $\mathrm{Prop}(a, b)$ is the set of open subsets $U \subseteq (a, b)$. "Compliant on U"
- Given $a \leq a' < b' \leq b'$ the map $\rho\colon \mathrm{Prop}(a, b) \to \mathrm{Prop}(a', b')$ is given by $U \mapsto U \cap (a', b')$.
- A mapping $f : B \to \mathrm{Prop}$ is called a *predicate* on X.
  - Given any behavior $x \in B(a, b)$, we get an open set $f(x) \subseteq (a, b)$
  - "x was compliant with f on $f(x)$."
  - The set of fully compliant behaviors are $\{x : X \mid f(x) = (a, b)\}$.

Prop and predicates

Prop is like an auditor's record book: "compliant here...".

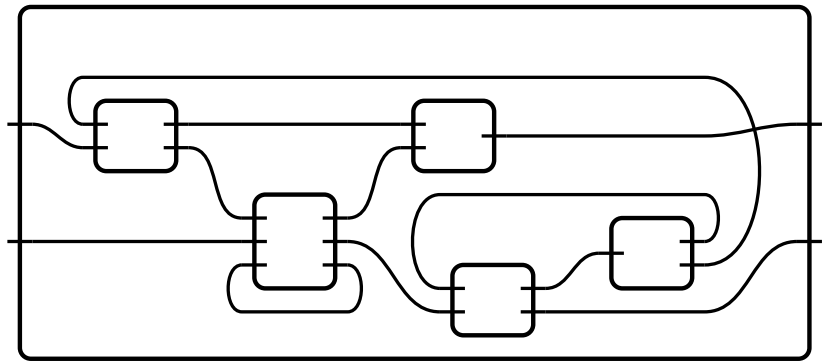
- $\mathrm{Prop}(a, b)$ is the set of open subsets $U \subseteq (a, b)$. "Compliant on U"
- Given $a \leq a' < b' \leq b'$ the map $\rho\colon \mathrm{Prop}(a, b) \to \mathrm{Prop}(a', b')$ is given by $U \mapsto U \cap (a', b')$.
- A mapping $f : B \to \mathrm{Prop}$ is called a *predicate* on X.
  - Given any behavior $x \in B(a, b)$, we get an open set $f(x) \subseteq (a, b)$
  - "x was compliant with f on $f(x)$."
  - The set of fully compliant behaviors are $\{x : X \mid f(x) = (a, b)\}$.
- Prop is called the *subobject classifier* because...
  - The fully compliant behaviors form a new sheaf $\{X \mid f\} \subseteq X$.
  - For every subsheaf $W \subseteq X$ there is some $f : X \to \mathrm{Prop}$ such that $W \cong \{X \mid f\}$.
- Prop plays a big role in the upcoming logic.

Outline

Behaviors on wires and in machines

Here's what we said before:

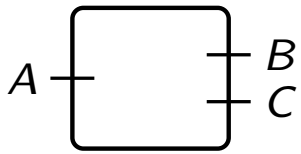
- Where do we find behaviors taking place in a wiring diagram?



- Each interface houses a machine with a type of internal behavior.
- But that behavior is only exposed in terms of the ports.

Machines in terms of behavior types

An interface is a bunch of ports, each labeled with a behavior type.

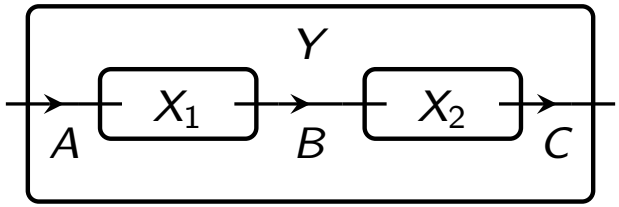


A machine X, housed within this interface, consists of:

- A behavior type X, together with a map $p\colon X \to A \times B \times C$.
- X is the internal behavior and p is its expressions in the outside world.
- The image $p(X) \subseteq A \times B \times C$ is the set of possible expressions.
- But for a given (a, b, c), there may be many (or no) internal behaviors so expressed.

E.g. perhaps X is delayed addition: $A(t+1) = B(t) + C(t)$ for all time t.

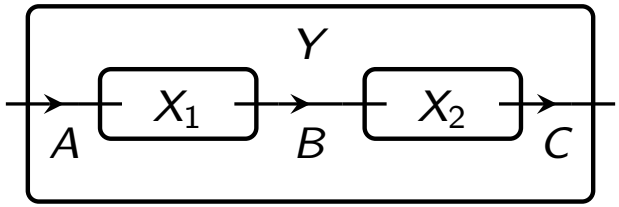
Composing behaviors in a wiring diagram



In this picture, we have machines X_1, X_2 wired together inside Y.

- Each interior machine has a map from internal behavior to ports
- Namely $X_1 \to A \times B$ and $X_2 \to B \times C$.
- The behavior type of the whole is often denoted $Y := X_1 \times_B X_2$.
  - A behavior on Y is a pair of B-matching behaviors on X_1, X_2.
  - Clearly there is a map to the exposed variables, $Y \to A \times C$.

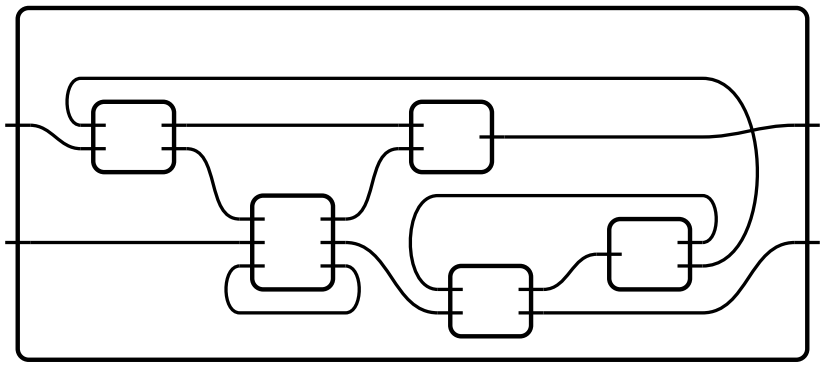
Composing behaviors in a wiring diagram



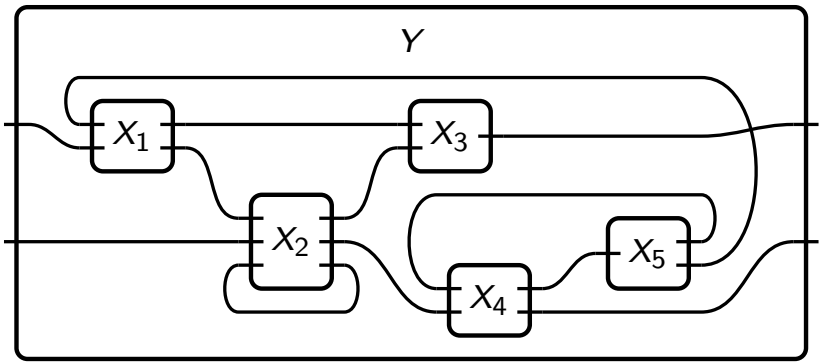
In this picture, we have machines X_1, X_2 wired together inside Y.

- Each interior machine has a map from internal behavior to ports
- Namely $X_1 \to A \times B$ and $X_2 \to B \times C$.
- The behavior type of the whole is often denoted $Y := X_1 \times_B X_2$.
  - A behavior on Y is a pair of B-matching behaviors on X_1, X_2.
  - Clearly there is a map to the exposed variables, $Y \to A \times C$.

The same idea makes sense for any wiring diagram.



Other semantics of wiring diagrams

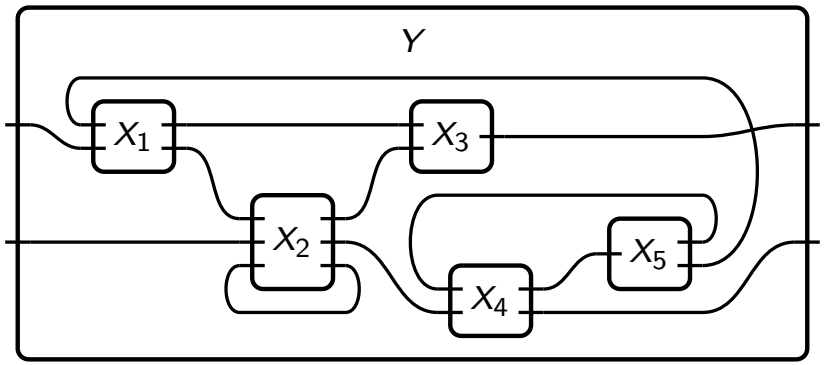


Matrix mult. formula for this wiring diagram:
$Y(a_0, a_1, a_{10}, a_{11}) = \sum_{a_2,\ldots,a_9} \prod_{i=1}^{5} X_i(\bar{a})$

Wiring diagrams are syntax that models many different semantics.

- E.g., relations, matrices, quantum processes, and now behaviors.
- There are high-level maps, called *functors*, that connect these.
- For example, there's a steady-state functor from behaviors to matrices.
  - To each wire, associate the set of all constant (steady) behaviors.
  - Each machine gives a matrix (tensor) of constant behaviors.
  - These can be plotted as bifurcation diagrams.
  - The bifurcation diagram for Y will be the product of the bifurcation diagrams for the X_i.

Next up: the semantics of contracts

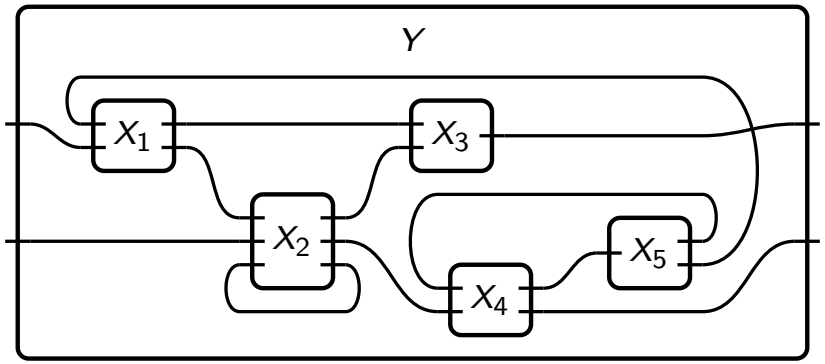


In the last part of the talk, I'll discuss compositional contracts.

- Like bifurcation diagrams, these give a compositional way of analyzing behaviors.
  - A contract is a guarantee for how a machine will behave.
  - Given a contract on each interior box, $X_1, \ldots, X_n$
  - We can produce a contract on the outer box Y.
- Useful when designing a system with parts from disparate suppliers.
  - Don't have to know implementations; just work with contracts.
  - Composed machines satisfy composed contracts.

Outline

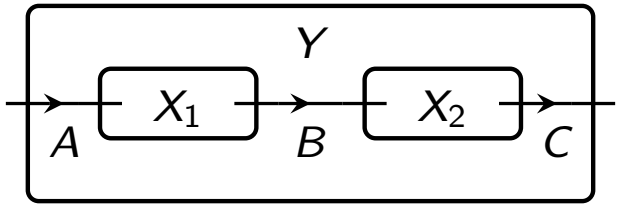
Compositional contracts



We want to compose contracts like we compose machines.

- Suppose each interior machine X_i is made by a different supplier.
  - They won't tell us the implementation of their machine.
  - But they will make guarantees about its behavior, i.e. contracts.
- These guarantees should have a syntax and a semantics.
  - The semantics will be in terms of sheaves.
  - The syntax will be in terms of a corresponding logic.
- The composed machine (in Y) should satisfy a composed contract.

Example

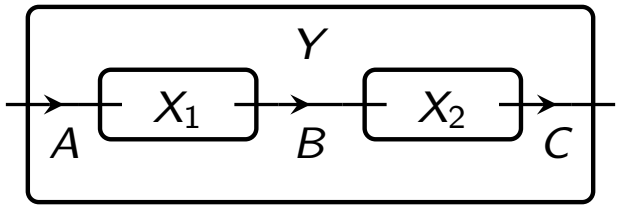


Suppose X_1 and X_2 guarantee the following contracts.

- X_1: If I receive two reds within a second, I'll output a green within five seconds of the first.
- X_2: If I ever receive a green, I'll output a blue within three seconds.

What will be the composed contract on Y?

Example



Suppose X_1 and X_2 guarantee the following contracts.

- X_1: If I receive two reds within a second, I'll output a green within five seconds of the first.
- X_2: If I ever receive a green, I'll output a blue within three seconds.

What will be the composed contract on Y?

- Y: If I receive two reds within a second, I'll output a blue within eight seconds.

Contracts are predicates on interfaces

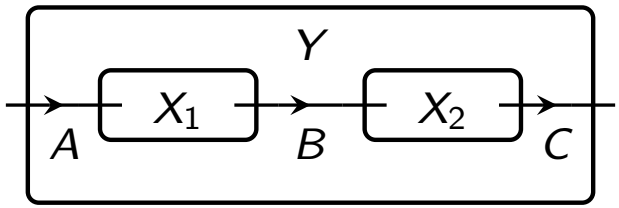
We said that sheaves on any space form a topos.

- We discussed that maps $\phi\colon X \to \mathrm{Prop}$ are predicates on X.
  - They correspond one-to-one with subobjects $X' \subseteq X$.
  - The subobject could be written $X' = \{X \mid \phi\} = \{x : X \mid \phi(x)\}$.
  - It's the subobject of behaviors that validate the contract.

Contracts are predicates on interfaces

We said that sheaves on any space form a topos.

- We discussed that maps $\phi\colon X \to \mathrm{Prop}$ are predicates on X.
  - They correspond one-to-one with subobjects $X' \subseteq X$.
  - The subobject could be written $X' = \{X \mid \phi\} = \{x : X \mid \phi(x)\}$.
  - It's the subobject of behaviors that validate the contract.



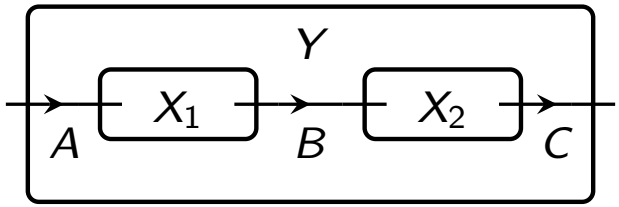
- By a contract on a machine, we just mean a predicate on its interface.
  - A contract for X_1 is a predicate $\phi_1\colon A \times B \to \mathrm{Prop}$.
  - A contract for X_2 is a predicate $\phi_2\colon B \times C \to \mathrm{Prop}$.
  - These cut out subobjects $X_1' \subseteq A \times B$ and $X_2' \subseteq B \times C$.

Combining contracts syntactically

Here's how you compose contracts in a wiring diagram:

- Each interior box has a contract, i.e. a predicate on its interface.
- Multiply (AND) them, and existentially quantify over unexposed wires.

Example:



- Say the interior contracts are $\phi_1(a, b)$ and $\phi_2(b, c)$.
- Their composition is then $\exists(b : B).\, \phi(a, b) \wedge \phi_2(b, c)$.
- This is a predicate $A \times C \to \mathrm{Prop}$.

Safety properties

All possible contracts in our system are safety properties.

- What is a safety property P?
  - For any behavior B and interval of time $[a, b]$,
  - Write $B \models_{[a,b]} P$ if B complies with P over $[a, b]$.
  - Then P is a safety property if, for all B and $a \leq a' \leq b' \leq b$,
  - If $B \models_{[a,b]} P$ then $B \models_{[a',b']} P$.
  - "Compliance on an interval means compliance throughout."

Safety properties

All possible contracts in our system are safety properties.

- What is a safety property P?
  - For any behavior B and interval of time $[a, b]$,
  - Write $B \models_{[a,b]} P$ if B complies with P over $[a, b]$.
  - Then P is a safety property if, for all B and $a \leq a' \leq b' \leq b$,
  - If $B \models_{[a,b]} P$ then $B \models_{[a',b']} P$.
  - "Compliance on an interval means compliance throughout."
- The propositions in our language are all safety properties.
  - This is part of the "sheaf" semantics.
  - The rest: if $B \models_{[a',b']} P$ for all $a < a' \leq b' < b$ then $B \models_{[a,b]} P$.
- In other words, $\wedge, \vee, \Rightarrow, \neg, \forall, \exists$ preserve this.
  - It's not something to prove about a property; it's built in.

Back to toposes again

Toposes $\mathcal{E}$ have internal type theories with proofs in higher-order logic.

- The types are objects $X \in \mathcal{E}$,
- Terms in context are morphisms $X_1 \times \cdots \times X_n \to X'$,
- Formulas are terms of type Prop,
- Connectives, $\wedge, \vee, \Rightarrow$, etc., are morphisms $\mathrm{Prop} \times \mathrm{Prop} \to \mathrm{Prop}$.
- Given $\phi\colon X \to \mathrm{Prop}$, we can also form $\exists(x\colon X).\,\phi$ and $\forall(x\colon X).\,\phi$
- The logic is constructive: to prove $\exists(x\colon X).\,\phi$, exhibit such an x.

More details of the language

- We will not discuss the internal logic of the **Int**-sheaf topos in detail.
  - It probably wouldn't be appreciated by the audience.
  - Ask me afterwards if you're interested.
- Briefly, there are types such as Time, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{R}_{\mathsf{var}}$, and Prop.
- There are morphisms, e.g. $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and "< 0" : $\mathbb{R}_{\mathsf{var}} \to \mathrm{Prop}$.

More details of the language

- We will not discuss the internal logic of the **Int**-sheaf topos in detail.
  - It probably wouldn't be appreciated by the audience.
  - Ask me afterwards if you're interested.
- Briefly, there are types such as Time, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{R}_{\mathsf{var}}$, and Prop.
- There are morphisms, e.g. $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and "< 0" : $\mathbb{R}_{\mathsf{var}} \to \mathrm{Prop}$.
- There are useful *modalities*, i.e. maps $j : \mathrm{Prop} \to \mathrm{Prop}$,
  - satisfying three axioms: $jj\phi = j\phi$, $\phi \Rightarrow j\phi$, $j\phi \wedge j\psi \Rightarrow j(\phi \wedge \psi)$.
  - Example: $@_{t=0}(\phi)$ means that ϕ holds in a neighborhood of 0.
  - Example: $\xi_{t=0}(\phi)$ means that ϕ holds in every nbhd of 0.
  - Example: $\mathrm{ptw}(\phi) = \forall(t : \mathrm{Time}).\, @_{t=0}(\phi)$.

TCAS contracts and derivatives

The TCAS contracts involve several constructions.

- We won't go into them here; a paper should be out in a few months.
- We write contracts and combine them using the internal logic.
- We also prove that (a *vast* simplification of) the NAS is safe.

One of the most interesting and important is that of *derivatives*.

- It is a time-derivative, phrased in terms of variable real numbers.
- Given $f, g : \mathbb{R}_{\text{var}}$, we write $f = \dot{g}$ if the following proposition holds:

$$\forall(q : \mathbb{Q}).\, q < f \Leftrightarrow \forall(t : \text{Time})(h : \mathbb{R}).\, h > 0 \Rightarrow q < \frac{@_{t=h}(g) - @_{t=0}(g)}{h}$$

- Thus we can write differential equations, e.g. $\ddot{x} - 3\dot{x} = \sin(x)$.
- They are contracts, e.g. on $x : \mathbb{R}_{\text{var}}$.

Our logic vs. other temporal logics

- Most temporal logics have special connectives and proof rules.
  - Ours has neither. It is only temporal in that its subject is behaviors over time.
  - We use standard higher-order logic.
  - We can thus model it directly in a proof assistant like Coq.
- Most temporal logics have a "next" time, or are otherwise discrete.
  - This one is more general and expressive, e.g. derivative.
  - It is likely that other logics are more convenient for specific behavior types.
  - Ours is designed as a big tent: combine all sorts of different behavior types.

Outline

Summary

Today I talked about compositionality and behavior.

- I discussed composition in terms of wiring diagrams.
  - Interfaces are boxes with ports, each port is labeled.
  - The ports are labeled by behavior types.
  - The interfaces are inhabited by machines, which also have behavior types.
  - The interfaces are wired together inside a larger interface.
  - The interior behaviors combined form the exterior behavior.

Summary

Today I talked about compositionality and behavior.

- I discussed composition in terms of wiring diagrams.
  - Interfaces are boxes with ports, each port is labeled.
  - The ports are labeled by behavior types.
  - The interfaces are inhabited by machines, which also have behavior types.
  - The interfaces are wired together inside a larger interface.
  - The interior behaviors combined form the exterior behavior.
- I discussed behavior types in terms of sheaves.
  - A set of behaviors over any interval of time, and restriction maps.
  - There is a topos of behavior types, and thus an internal logic.
  - Contracts for machines are predicates on their interfaces.
  - Combine contracts for disparate systems in this big tent.

Thanks for inviting me to speak!