Compositional contracts for hybrid dynamical systems

David I. Spivak

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

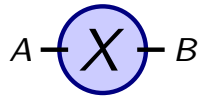
Presented on 2017/07/12
at SIAM Novel Approaches for Systems of Systems

Outline

Outline

Systems and composition

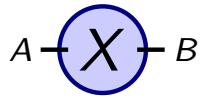
Imagine there is some sort of machine inside this circle:



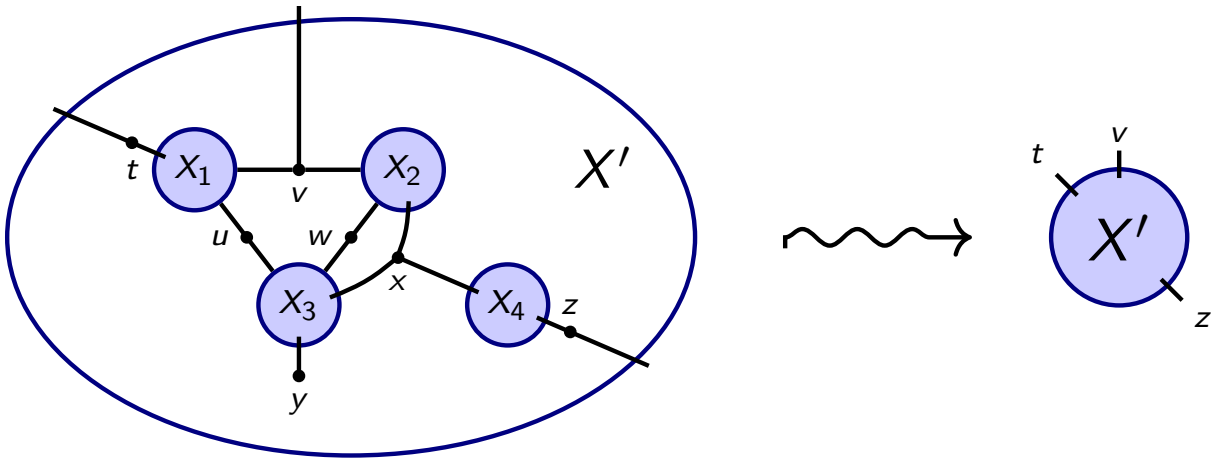
- Imagine the machine is called X and it has two *ports*, A and B.
  - You can think of A and B as input and output, or just as ports.
  - A and B are how machine X interfaces with the outside world.

Systems and composition

Imagine there is some sort of machine inside this circle:



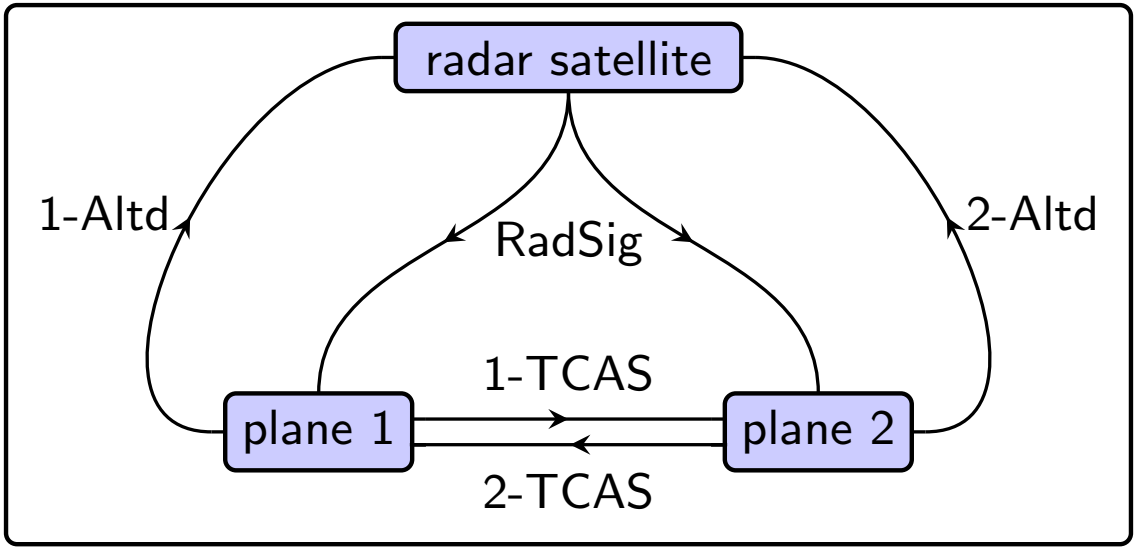
- Imagine the machine is called X and it has two *ports*, A and B.
  - You can think of A and B as input and output, or just as ports.
  - A and B are how machine X interfaces with the outside world.
- We interconnect machines along interfaces to form another machine.



- The behavior of X' is dictated by the X_i and their interconnection.

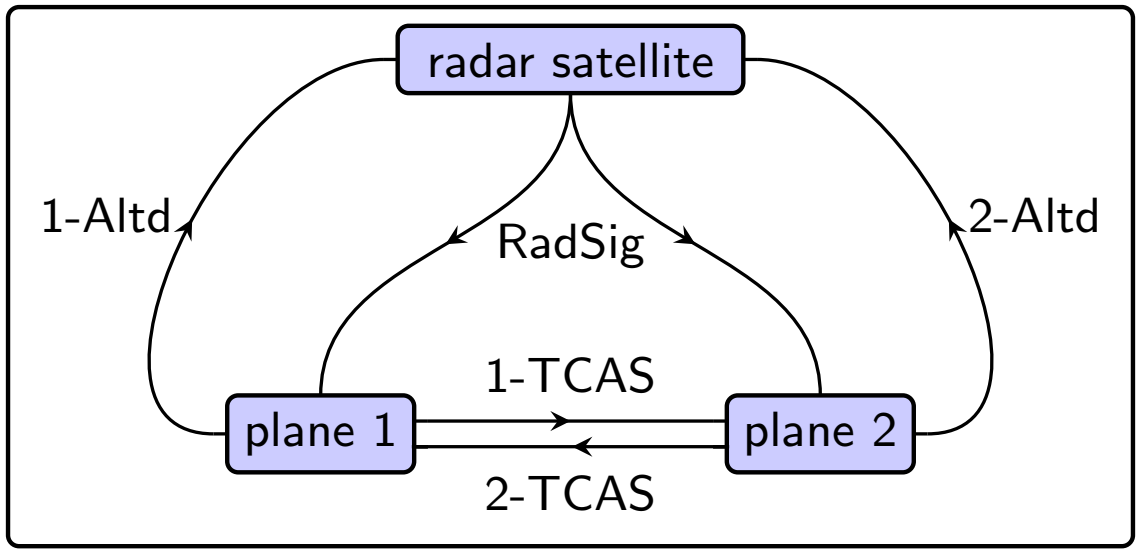
System of systems

Here's a more concrete example.

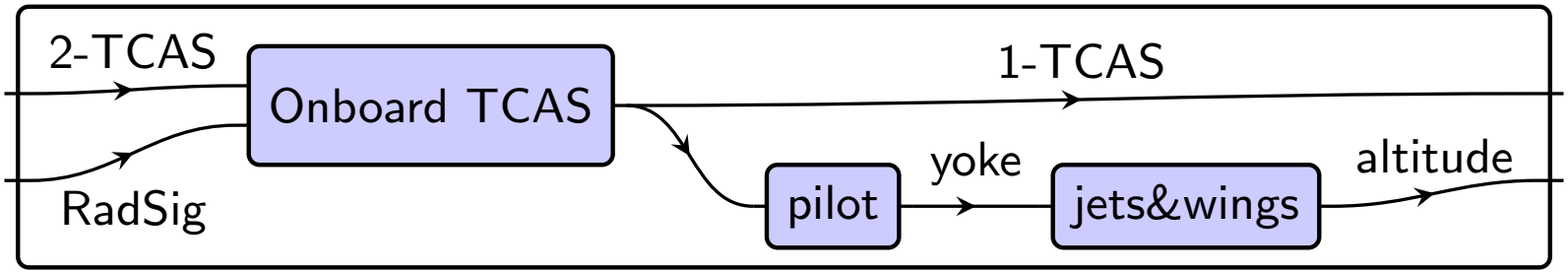


System of systems

Here's a more concrete example.



Often we can zoom into a component and see it as a system.



Behaviors in the national airspace system

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

- 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. $\dot{x}(t) = f\big(x(t), a(t), b(t)\big)$
  - Or, the behavior could be modeled by a diff. relation or inclusion.
- Other NAS elements should be modeled altogether differently.
  - Some systems in the NAS are nothing like ODEs.
  - The TCAS could be modeled as a labeled transition system.
  - The pilot could be modeled as a delay on TCAS commands.
- All of these are behavior types, which we will model as sheaves.
  - In this talk we give a new *temporal logic* for behavior types.
  - It is compositional: reason about disparate interacting machines.

Plan of talk

We will discuss the following.

- Behavior types as sheaves;
- Internal language of our sheaf topos:
  - The internal language and higher-order logic in any topos,
  - How this logic looks in our particular case;
- Compositionality of contracts;
- A simplified TCAS example.

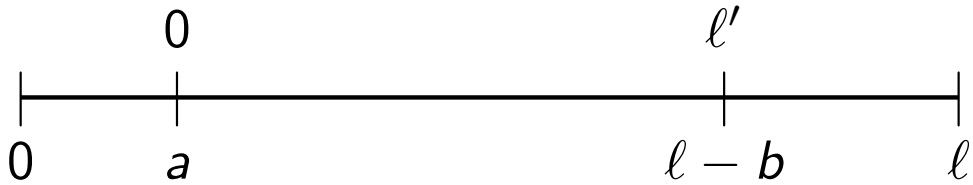
Then we will conclude with a summary.

Outline

Behavior types as sheaves

We model behavior types as sheaves on a certain site **Int**.

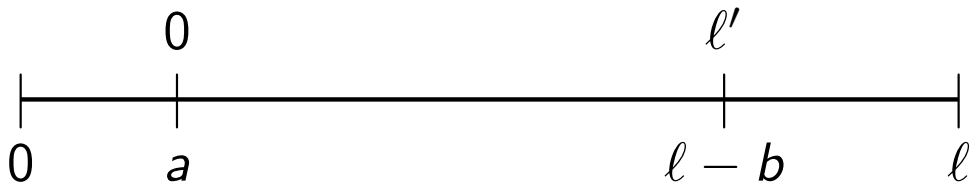
- As a category, the objects are lengths $\mathrm{Ob}(\mathbf{Int}) = \{\ell \in \mathbb{R} \mid \ell > 0\}$.
  - Morphisms are subinterval inclusions: $\mathbf{Int}(\ell', \ell) = \{i_{a,b} \mid a + \ell' + b = \ell\}$.
  - Composition is $i_{a,b} \circ i_{a',b'} = i_{a+a',b+b'}$.



Behavior types as sheaves

We model behavior types as sheaves on a certain site **Int**.

- As a category, the objects are lengths $\mathrm{Ob}(\mathbf{Int}) = \{\ell \in \mathbb{R} \mid \ell > 0\}$.
  - Morphisms are subinterval inclusions: $\mathbf{Int}(\ell', \ell) = \{i_{a,b} \mid a + \ell' + b = \ell\}$.
  - Composition is $i_{a,b} \circ i_{a',b'} = i_{a+a',b+b'}$.

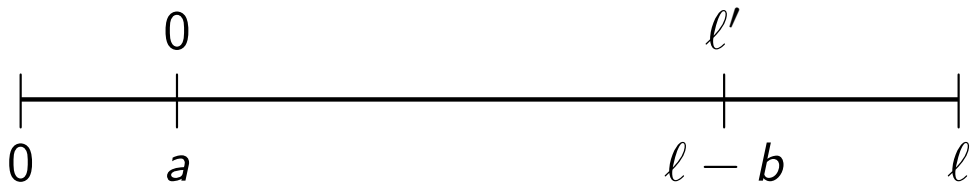


- Think of ℓ as representing the (translation-invariant) interval $(0, \ell)$.
- The coverage is: $\ell = \bigcup_{\{i_{a,b}\colon \ell' \to \ell \mid a,b>0\}} \ell'$

Behavior types as sheaves

We model behavior types as sheaves on a certain site **Int**.

- As a category, the objects are lengths $\mathrm{Ob}(\mathbf{Int}) = \{\ell \in \mathbb{R} \mid \ell > 0\}$.
  - Morphisms are subinterval inclusions: $\mathbf{Int}(\ell', \ell) = \{i_{a,b} \mid a + \ell' + b = \ell\}$.
  - Composition is $i_{a,b} \circ i_{a',b'} = i_{a+a',b+b'}$.



- Think of ℓ as representing the (translation-invariant) interval $(0, \ell)$.
- The coverage is: $\ell = \bigcup_{\{i_{a,b}\colon \ell' \to \ell \mid a,b>0\}} \ell'$
- A sheaf B on **Int** is the formal definition of behavior type.
  - To every length ℓ, assign a set $B(\ell)$; elements are called *behaviors*.
  - To every inclusion $i_{a,b}\colon \ell' \to \ell$, a restriction map $B(\ell) \to B(\ell')$.

Example behavior types

Here are some examples of behavior types:

- The sheaf C_n of continuous curves in $\mathbb{R}^n$: $C_n(\ell) = \{f : (0, \ell) \to \mathbb{R}^n\}$.
- For any set, there is an associated constant sheaf; e.g. $\mathbb{Q}(\ell) = \mathbb{Q}$.
- A sheaf $\mathrm{Time} \subseteq C_1$ given by $\mathrm{Time}(\ell) = \{f \in C_1 \mid f \text{ isometry}\}$.

Example behavior types

Here are some examples of behavior types:

- The sheaf C_n of continuous curves in $\mathbb{R}^n$: $C_n(\ell) = \{f : (0, \ell) \to \mathbb{R}^n\}$.
- For any set, there is an associated constant sheaf; e.g. $\mathbb{Q}(\ell) = \mathbb{Q}$.
- A sheaf $\mathrm{Time} \subseteq C_1$ given by $\mathrm{Time}(\ell) = \{f \in C_1 \mid f \text{ isometry}\}$.
- Custom sheaves:
  - Solutions to ODE, $\dot{x}(t) = f(x(t), a(t))$, with parameter $a(t)$:

$$B(\ell) = \{(x, a) \in C_n \times C_m \mid \dot{x} = f(x, a)\}$$

  - Walks through a graph G, where edges are instantaneous and one remains on a vertex for an arbitrary positive amount of time.
  - Take any movie: $M(\ell) = \{\text{snippets of length } \ell\}$.

Example behavior types

Here are some examples of behavior types:

- The sheaf C_n of continuous curves in $\mathbb{R}^n$: $C_n(\ell) = \{f : (0, \ell) \to \mathbb{R}^n\}$.
- For any set, there is an associated constant sheaf; e.g. $\mathbb{Q}(\ell) = \mathbb{Q}$.
- A sheaf $\mathrm{Time} \subseteq C_1$ given by $\mathrm{Time}(\ell) = \{f \in C_1 \mid f \text{ isometry}\}$.
- Custom sheaves:
  - Solutions to ODE, $\dot{x}(t) = f(x(t), a(t))$, with parameter $a(t)$:

$$B(\ell) = \{(x, a) \in C_n \times C_m \mid \dot{x} = f(x, a)\}$$

  - Walks through a graph G, where edges are instantaneous and one remains on a vertex for an arbitrary positive amount of time.
  - Take any movie: $M(\ell) = \{\text{snippets of length } \ell\}$.

In fact, behavior types are the objects of a topos, Shv(**Int**).

- Thus we can add, multiply, exponentiate, etc.: $B + C$, $B \times C$, B^C, ...
- There is a sheaf $\Omega = \mathrm{Prop}$ of propositions, the subobject classifier.

Prop: propositions as behaviors

Next we'll discuss the internal language of our topos $\mathcal{E} := \mathrm{Shv}(\mathbf{Int})$.

- The subobject classifier Prop will play a special role.
  - It's the sheaf of truth values, like "true/false", but not boolean.
  - Prop is like an auditor's record book: "compliant here...".

Prop: propositions as behaviors

Next we'll discuss the internal language of our topos $\mathcal{E} := \mathrm{Shv}(\mathbf{Int})$.

- The subobject classifier Prop will play a special role.
  - It's the sheaf of truth values, like "true/false", but not boolean.
  - Prop is like an auditor's record book: "compliant here...".
- Predicates on a sheaf X are sheaf morphisms $\phi\colon X \to \mathrm{Prop}$.
  - Predicates cut out subsheaves, $\{x : X \mid \phi(x)\} \subseteq X$.
  - E.g. the sheaf of curves in $\mathbb{R}^2$ that stay within the unit disk.

Prop: propositions as behaviors

Next we'll discuss the internal language of our topos $\mathcal{E} := \mathrm{Shv}(\mathbf{Int})$.

- The subobject classifier Prop will play a special role.
  - It's the sheaf of truth values, like "true/false", but not boolean.
  - Prop is like an auditor's record book: "compliant here...".
- Predicates on a sheaf X are sheaf morphisms $\phi\colon X \to \mathrm{Prop}$.
  - Predicates cut out subsheaves, $\{x : X \mid \phi(x)\} \subseteq X$.
  - E.g. the sheaf of curves in $\mathbb{R}^2$ that stay within the unit disk.
- As a sheaf, $\mathrm{Prop}(\ell) =$"open subsets of $(0, \ell)$ in the interval domain".
  - The *interval domain* is a non-Hausdorff topological space.
  - Points are closed intervals $[d, u]$, where $d \leq u$.
  - There is a base of open sets $(a, b) = \{[d, u] \mid a < d \leq u < b\}$.
- There is a way to view these open sets as "continuous Dyck paths".

$\mathrm{Prop}(\ell)$ as Dyck paths

Prop will classify sub-types of behaviors, specifying where they're compliant.

- How can we visualize an element $P \in \mathrm{Prop}(\ell)$?
- P is an open subset in a strange topological space. Two rules:
  - Down-closed: If $[d, u] \in P$, then $[d', u'] \in P$ for all $d \leq d' \leq u' \leq u$.
  - Rounded: If $[d, u] \in P$ then $[d', u'] \in P$ for some $d' < d \leq u < u'$.

$\mathrm{Prop}(\ell)$ as Dyck paths

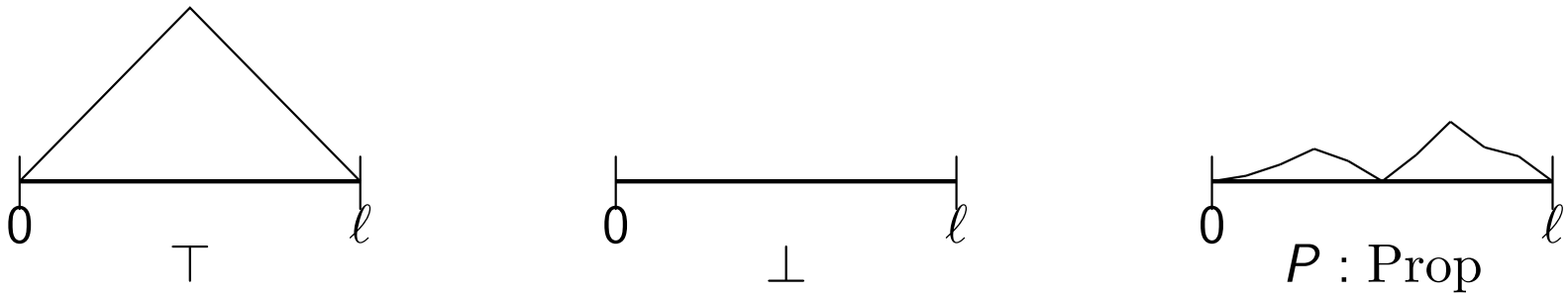
Prop will classify sub-types of behaviors, specifying where they're compliant.

- How can we visualize an element $P \in \mathrm{Prop}(\ell)$?
- P is an open subset in a strange topological space. Two rules:
  - Down-closed: If $[d, u] \in P$, then $[d', u'] \in P$ for all $d \leq d' \leq u' \leq u$.
  - Rounded: If $[d, u] \in P$ then $[d', u'] \in P$ for some $d' < d \leq u < u'$.
- In fact we can draw these using what I call "continuous Dyck paths".
  - Using discrete time instead, Prop would consist of ordinary Dyck paths.
  - Dyck paths are well-known in combinatorics (Catalan numbers).

$\mathrm{Prop}(\ell)$ as Dyck paths

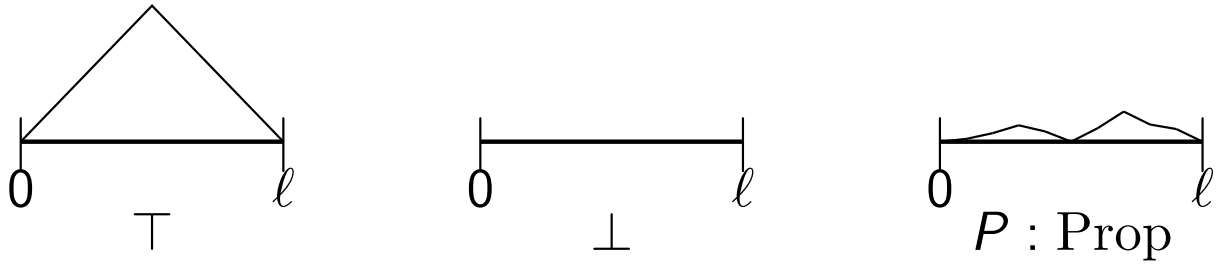
Prop will classify sub-types of behaviors, specifying where they're compliant.

- How can we visualize an element $P \in \mathrm{Prop}(\ell)$?
- P is an open subset in a strange topological space. Two rules:
  - Down-closed: If $[d, u] \in P$, then $[d', u'] \in P$ for all $d \leq d' \leq u' \leq u$.
  - Rounded: If $[d, u] \in P$ then $[d', u'] \in P$ for some $d' < d \leq u < u'$.
- In fact we can draw these using what I call "continuous Dyck paths".
  - Using discrete time instead, Prop would consist of ordinary Dyck paths.
  - Dyck paths are well-known in combinatorics (Catalan numbers).
- A continuous Dyck path over an interval of length ℓ is:
  - A 1-Lipschitz function $p\colon [0, \ell] \to \mathbb{R}_{\geq 0}$ such that $p(0) = 0 = p(\ell)$.
  - $P \in \mathrm{Prop}(\ell)$ corresponds to $p(x) = \sup\{y \mid [x - y, x + y] \in P\}$.



Prop^X as a Heyting algebra

Prop has the structure of a Heyting algebra.

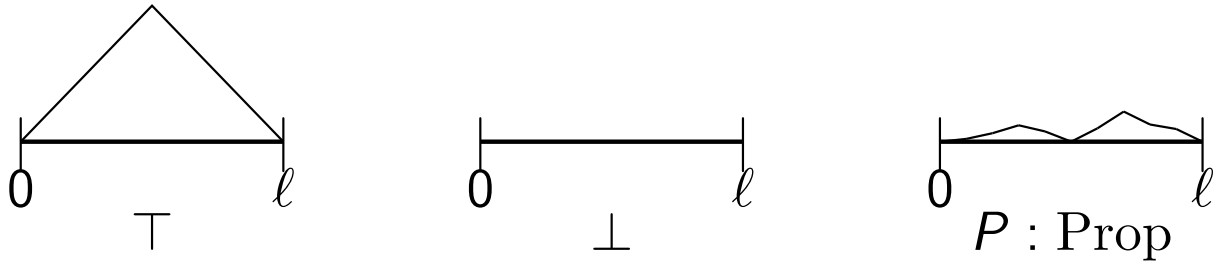


- Given two Dyck paths P, Q, one may be under the other, $P \leq Q$.
- There is a maximal Dyck path, $\top$, and a minimal one, $\bot$.
- The pointwise max of two Dyck paths is a Dyck path, $P \vee Q$.
- The pointwise min of two Dyck paths is a Dyck path, $P \wedge Q$.
- There is an operation $P \Rightarrow Q$, a Dyck path such that

$$R \leq (P \Rightarrow Q) \quad \text{iff} \quad (R \wedge P) \leq Q.$$

Prop^X as a Heyting algebra

Prop has the structure of a Heyting algebra.



- Given two Dyck paths P, Q, one may be under the other, $P \leq Q$.
- There is a maximal Dyck path, $\top$, and a minimal one, $\bot$.
- The pointwise max of two Dyck paths is a Dyck path, $P \vee Q$.
- The pointwise min of two Dyck paths is a Dyck path, $P \wedge Q$.
- There is an operation $P \Rightarrow Q$, a Dyck path such that

$$R \leq (P \Rightarrow Q) \quad \text{iff} \quad (R \wedge P) \leq Q.$$

So you can do constructive logic in Prop.

- For any sheaf X, the sheaf Prop^X inherits a Heyting algebra structure.
- This will give us the internal logic for the topos Shv(**Int**).

Outline

The internal language of a topos

Toposes connect geometry, algebra, and logic.

- Toposes are just nice categories.
  - You have finite limits, exponentials, and a subobject classifier.
  - The category of sheaves on any site is a topos.

The internal language of a topos

Toposes connect geometry, algebra, and logic.

- Toposes are just nice categories.
  - You have finite limits, exponentials, and a subobject classifier.
  - The category of sheaves on any site is a topos.
- A topos $\mathcal{E}$ has an internal language in which to do higher-order logic.
  - That is, there is a corresponding formal language (type theory).
  - Objects / morphisms in $\mathcal{E}$ become types / terms in the language.
  - Terms of type Prop are logical formulas in this language.
  - One can express facts about the topos as axioms in this language.
- The topos is a model of these axioms, using higher-order logic.
  - Anything provable from the axioms is true in the topos!
  - Such proofs can be verified in the natural language of Coq/Lean.

Examples of statements in the internal language

What kind of things can you express in a topos $\mathcal{E}$?

- You can express products, coproducts, exponentials, etc.
- You can express subsheaves of a given sheaf X using predicates.
  - Given a predicate $\phi : X \to \mathrm{Prop}$, we have $\{x : X \mid \phi(x)\}$.

Examples of statements in the internal language

What kind of things can you express in a topos $\mathcal{E}$?

- You can express products, coproducts, exponentials, etc.
- You can express subsheaves of a given sheaf X using predicates.
  - Given a predicate $\phi : X \to \mathrm{Prop}$, we have $\{x : X \mid \phi(x)\}$.
- Example: given types $X, Y \in \mathcal{E}$, we can write

$$\mathbf{Epi}(X, Y) \coloneqq \{f : X \to Y \mid \forall(y : Y).\, \exists(x : X).\, f(x) = y\}$$

  - This formula is written in a kind of set-theoretic syntax.
  - It seems to describe the epimorphisms from X to Y.
  - Whether that's accurate or not, it does describe an object in $\mathcal{E}$.
  - But in fact it is accurate: the elements of $\mathbf{Epi}(X, Y)$ are exactly the epimorphisms $X \to Y$.

This story is thanks to Bill Lawvere, André Joyal, and others.

The internal language of behavior types

Our topos Shv(**Int**) has an internal language.

- We call it a "higher-order temporal logic".
  - It may be related to Halperin-Shoham *interval temporal logic*.
  - The syntax and proof rules are standard higher-order logic.
  - What makes it temporal? It has a type Time and related axioms.

The internal language of behavior types

Our topos Shv(**Int**) has an internal language.

- We call it a "higher-order temporal logic".
  - It may be related to Halperin-Shoham *interval temporal logic*.
  - The syntax and proof rules are standard higher-order logic.
  - What makes it temporal? It has a type Time and related axioms.
- This language is pretty general and expressive.
  - For example, we can express derivatives with respect to $t : \mathrm{Time}$.
  - Using that, we can express ODEs completely within the logic.
  - We can also express labeled transition systems within the logic.
- Next we will discuss some modalities in this logic.

Modalities in general

Modalities on $\mathcal{E}$ allow us to simulate subtoposes inside of $\mathcal{E}$.

- Synonyms: *modality* = *local operator* = *Lawvere-Tierney topology*.
- A modality is just a map $j : \mathrm{Prop} \to \mathrm{Prop}$ satisfying three properties:
  - Closure: For all $P : \mathrm{Prop}$, we have $P \Rightarrow jP$ and $jjP \Rightarrow jP$.
  - Functoriality: For all P, Q, we have $(P \Rightarrow Q) \Rightarrow (jP \Rightarrow jQ)$. [1]
- Given modalities j_1, j_2, we write $j_1 \Rightarrow j_2$ iff $\forall(P : \mathrm{Prop}).\, j_1 P \Rightarrow j_2 P$.
- Modalities on $\mathcal{E}$ correspond 1-1 with subtoposes $\mathcal{E}' \subseteq \mathcal{E}$.
  - id : $\mathrm{Prop} \to \mathrm{Prop}$ gives back $\mathcal{E}$ and $\top$ gives the empty topos.
  - The modality $P \mapsto \neg\neg P$ gives the smallest dense subtopos.
  - If $j_1 \Rightarrow j_2$, then j_2 corresponds to a subtopos of j_1.

We will next discuss modalities in Shv(**Int**).

[1] Given closure, functoriality is equivalent to the more familiar, $j(P \wedge Q) \Leftrightarrow (jP \wedge jQ)$.

Modalities in Shv(Int)

In our setting, modalities correspond to subspaces of the interval domain.

- For any $t : \mathrm{Time}$, there is a modality $\xi_0^t : \mathrm{Prop} \to \mathrm{Prop}$.
  - $\xi_0^t(P)$ is read "See $t = 0$, P".
  - It corresponds to the closure of $[0, 0]$ in the interval domain.
  - Logically, $\xi_0^t(P) := t \# 0 \vee P$, i.e. $\xi_0^t(P) = (t < 0) \vee (t > 0) \vee P$.

Modalities in Shv(**Int**)

In our setting, modalities correspond to subspaces of the interval domain.

- For any $t : \mathrm{Time}$, there is a modality $\xi_0^t : \mathrm{Prop} \to \mathrm{Prop}$.
  - $\xi_0^t(P)$ is read "See $t = 0$, P".
  - It corresponds to the closure of $[0, 0]$ in the interval domain.
  - Logically, $\xi_0^t(P) := t\#0 \vee P$, i.e. $\xi_0^t(P) = (t < 0) \vee (t > 0) \vee P$.
- For any $t : \mathrm{Time}$, there is a modality $@_0^t : \mathrm{Prop} \to \mathrm{Prop}$.
  - $@_0^t(P)$ is read "At $t = 0$, P".
  - It corresponds to the point $[0, 0]$ in the interval domain.
  - Logically, $@_0^t(P) := (P \Rightarrow t\#0) \Rightarrow t\#0$.

Modalities in Shv(Int)

In our setting, modalities correspond to subspaces of the interval domain.

- For any $t : \mathrm{Time}$, there is a modality $\xi_0^t : \mathrm{Prop} \to \mathrm{Prop}$.
  - $\xi_0^t(P)$ is read "See $t = 0$, P".
  - It corresponds to the closure of $[0, 0]$ in the interval domain.
  - Logically, $\xi_0^t(P) := t\#0 \vee P$, i.e. $\xi_0^t(P) = (t < 0) \vee (t > 0) \vee P$.
- For any $t : \mathrm{Time}$, there is a modality $@_0^t : \mathrm{Prop} \to \mathrm{Prop}$.
  - $@_0^t(P)$ is read "At $t = 0$, P".
  - It corresponds to the point $[0, 0]$ in the interval domain.
  - Logically, $@_0^t(P) := (P \Rightarrow t\#0) \Rightarrow t\#0$.
- There's a modality $\pi : \mathrm{Prop} \to \mathrm{Prop}$.
  - πP is read "Pointwise, P".
  - It corresponds to the subtopos of maximal points.
  - As a subspace of the interval domain, it's the usual real line $\mathbb{R}$.
  - Logically, $\pi P := \forall(t : \mathrm{Time}).\, @_0^t(P)$.

Variable real numbers and derivatives

- In any topos $\mathcal{E}$ there is a real numbers object $\mathbb{R}_{\mathcal{E}}$.
  - Real numbers are defined internally, as Dedekind cuts $\mathbb{Q} \to \mathrm{Prop}$.
  - In Shv(**Int**), it is the *constant reals*; sections are constant.
  - In the π-subtopos (add π to all formulas), it is C_1, *variable reals*.
  - In the $@_0^t$-subtopos, it is what we call *momentary reals*.

Variable real numbers and derivatives

- In any topos $\mathcal{E}$ there is a real numbers object $\mathbb{R}_{\mathcal{E}}$.
  - Real numbers are defined internally, as Dedekind cuts $\mathbb{Q} \to \mathrm{Prop}$.
  - In Shv(**Int**), it is the *constant reals*; sections are constant.
  - In the π-subtopos (add π to all formulas), it is C_1, *variable reals*.
  - In the $@_0^t$-subtopos, it is what we call *momentary reals*.
- One can define derivatives of variable reals.
  - Externally, if $x, \dot{x} : C_1$, we can decide if $\dot{x}$ really is the derivative.
  - Internally, (theorem:) the following formula captures that notion.

$$q \leq \dot{x} \Leftrightarrow \forall(r_1, r_2 : \mathbb{R}).\, (r_1 < r_2) \Rightarrow q < \frac{x^{@}(r_2) - x^{@}(r_1)}{r_2 - r_1}$$

Variable real numbers and derivatives

- In any topos $\mathcal{E}$ there is a real numbers object $\mathbb{R}_{\mathcal{E}}$.
  - Real numbers are defined internally, as Dedekind cuts $\mathbb{Q} \to \mathrm{Prop}$.
  - In Shv(**Int**), it is the *constant reals*; sections are constant.
  - In the π-subtopos (add π to all formulas), it is C_1, *variable reals*.
  - In the $@_0^t$-subtopos, it is what we call *momentary reals*.
- One can define derivatives of variable reals.
  - Externally, if $x, \dot{x} : C_1$, we can decide if $\dot{x}$ really is the derivative.
  - Internally, (theorem:) the following formula captures that notion.

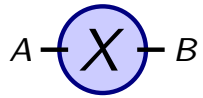
$$q \leq \dot{x} \Leftrightarrow \forall (r_1, r_2 : \mathbb{R}).\, (r_1 < r_2) \Rightarrow q < \frac{x^{@}(r_2) - x^{@}(r_1)}{r_2 - r_1}$$

- From here, ODEs are easily definable, e.g. $\dot{x} = f(x, a, b)$.
  - They cut out subsheaves: all behaviors satisfying the equation.
  - Obviously, one could use differential relations, inclusions, etc.

Outline

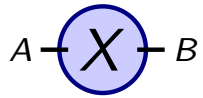
Behavior contracts

Imagine there is some sort of machine inside this circle:



Behavior contracts

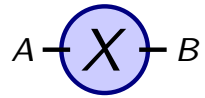
Imagine there is some sort of machine inside this circle:



- This actually denotes a sheaf morphism $f : X \to A \times B$.
  - X represents the total behavior; A, B are the interface behavior.
  - f tells us what we can see of X's behavior from outside.

Behavior contracts

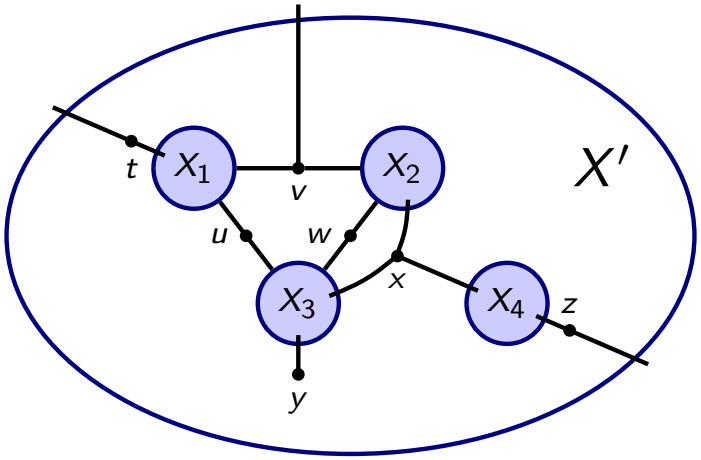
Imagine there is some sort of machine inside this circle:



- This actually denotes a sheaf morphism $f : X \to A \times B$.
  - X represents the total behavior; A, B are the interface behavior.
  - f tells us what we can see of X's behavior from outside.
- There may be a logical property ϕ that X is guaranteed to satisfy.
  - Such is called a *behavior contract on* X. Write $X \models \phi$.
  - Mathematically, ϕ corresponds to a subsheaf of $A \times B$.
  - Then $X \models \phi$ means $\mathrm{im}(f) \subseteq \{(a, b) : A \times B \mid \phi(a, b)\}$.

Combining independent behavior contracts

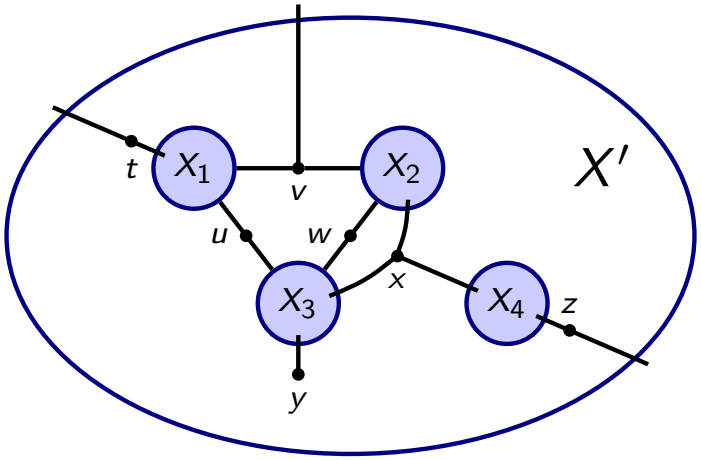
In the picture below, imagine machine X_2 has no idea about X_1 or X_3, etc.



They were designed independently, by different suppliers $i = 1, 2, 3, 4$.

Combining independent behavior contracts

In the picture below, imagine machine X_2 has no idea about X_1 or X_3, etc.

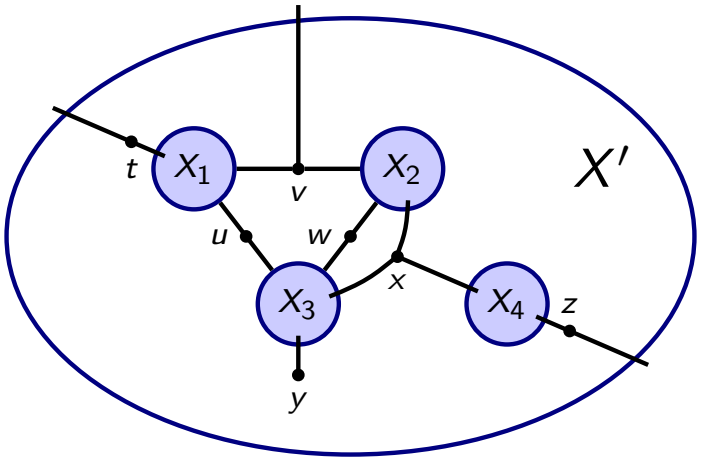


They were designed independently, by different suppliers $i = 1, 2, 3, 4$.

- Luckily, each supplier i gave us a behavior contract for its product.
  - Let $\Gamma_1 = t, u, v$, etc. (let $\Gamma_2 = v, w, x$, let $\Gamma_3 = u, w, x, y$, let $\Gamma_4 = x, z$, and let $\Gamma' = t, v, z$).
  - Γ_i is the context in which formula ϕ_i makes sense: $\Gamma_i \vdash \phi_i$.
  - Each supplier i provides a guarantee: $X_i \models \phi_i$.

We want to produce a global contract (ϕ' on X') from those on the X_i.

Composing behavior contracts



Suppose given behavior contracts $\Gamma_i \vdash \phi_i$ and guarantees $X_i \models \phi_i$.

- The wiring diagram tells us how variables from the Γ_i align.
- Given a X'-contract $\Gamma' \vdash \phi'$, we try to prove it's guaranteed, $X' \models \phi'$.
- The required proof in the above case is to show:

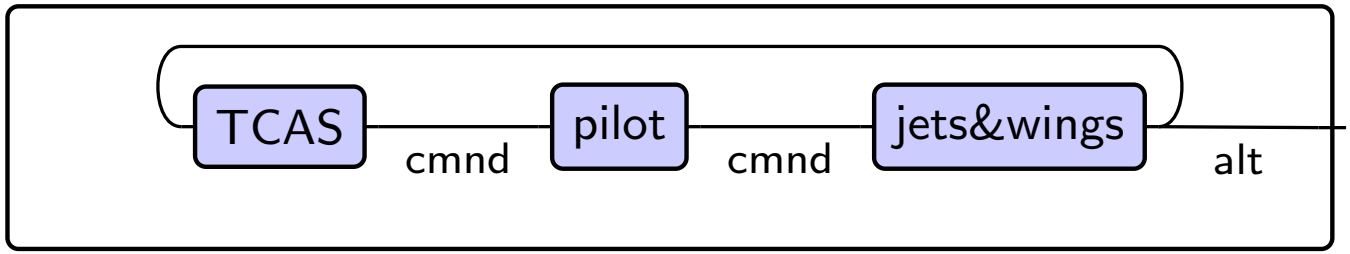
$$t, v, z \vdash \forall(u, w, x, y).\, (\phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4) \Rightarrow \phi'$$

- There is a universal contract ϕ' such that $X' \models \phi'$, namely

$$\exists(u, w, x, y).\, \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4$$

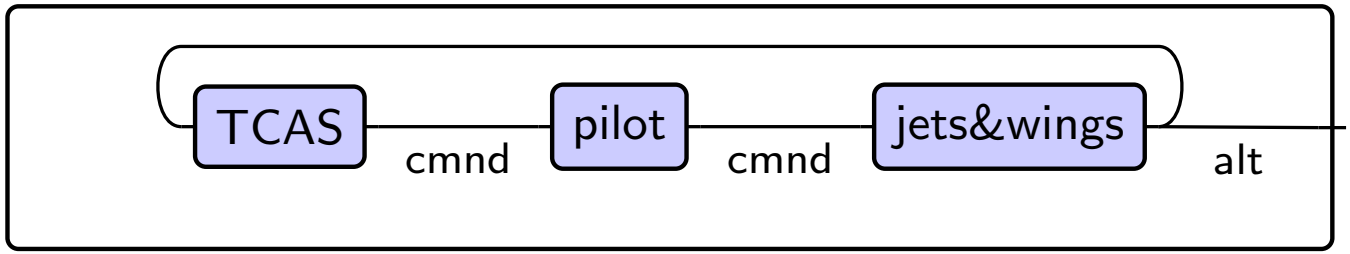
A simplified TCAS example

Simplify the TCAS system: a single plane trying to reach a safe altitude.



A simplified TCAS example

Simplify the TCAS system: a single plane trying to reach a safe altitude.



Our model combines different behavior types:

- TCAS acts as a labeled transition system (labeled by altitude ranges).
- The pilot acts as a delay (transferring commands from TCAS).
- The jets&wings act as an ODE, taking commands as input.

A simplified TCAS example

I won't give the details, but here is the idea.

- Let $a(t)$ denote the altitude of the plane at time t.

A simplified TCAS example

I won't give the details, but here is the idea.

- Let $a(t)$ denote the altitude of the plane at time t.
- There are constants (positive rationals) $\texttt{safe}$, $\texttt{margin}$, $\texttt{delay}$, $\texttt{rate}$.
- TCAS says "climb" if a below $\texttt{safe} + \texttt{margin}$; "level" otherwise.

$$(a < \texttt{safe} + \texttt{margin} \Rightarrow T = \texttt{climb}) \wedge (a > \texttt{safe} + \texttt{margin} \Rightarrow T = \texttt{level})$$

A simplified TCAS example

I won't give the details, but here is the idea.

- Let $a(t)$ denote the altitude of the plane at time t.
- There are constants (positive rationals) $\texttt{safe}$, $\texttt{margin}$, $\texttt{delay}$, $\texttt{rate}$.
- TCAS says "climb" if a below $\texttt{safe} + \texttt{margin}$; "level" otherwise.

$$(a < \texttt{safe} + \texttt{margin} \Rightarrow T = \texttt{climb}) \wedge (a > \texttt{safe} + \texttt{margin} \Rightarrow T = \texttt{level})$$

- The pilot does whatever TCAS says, but after a delay.

$$\forall(t{:}\mathrm{Time})(r{:}\mathbb{R})(c{:}\mathrm{Cmnd}).\xi_{[0,\texttt{delay}+r]}\Big(0<t<r\Rightarrow(T=c)\Leftrightarrow\texttt{delay}<t<\texttt{delay}+r\Rightarrow(P=c)\Big)$$

A simplified TCAS example

I won't give the details, but here is the idea.

- Let $a(t)$ denote the altitude of the plane at time t.
- There are constants (positive rationals) $\texttt{safe}$, $\texttt{margin}$, $\texttt{delay}$, $\texttt{rate}$.
- TCAS says "climb" if a below $\texttt{safe} + \texttt{margin}$; "level" otherwise.

$$(a < \texttt{safe} + \texttt{margin} \Rightarrow T = \texttt{climb}) \wedge (a > \texttt{safe} + \texttt{margin} \Rightarrow T = \texttt{level})$$

- The pilot does whatever TCAS says, but after a delay.

$$\forall(t{:}\mathrm{Time})(r{:}\mathbb{R})(c{:}\mathrm{Cmnd}).\xi_{[0,\texttt{delay}+r]}\Big(0{<}t{<}r{\Rightarrow}(T{=}c){\Leftrightarrow}\texttt{delay}{<}t{<}\texttt{delay}{+}r{\Rightarrow}(P{=}c)\Big)$$

- The airplane ascends at $\texttt{rate}$, or stays level, as instructed by the pilot.

$$(P = \texttt{level} \Rightarrow \dot{a} = 0) \wedge (P = \texttt{climb} \Rightarrow \dot{a} = \texttt{rate})$$

From this we can prove a global contract. Let $M := \frac{\texttt{safe}}{\texttt{rate}} + \texttt{delay}$.

- Given a perfect clock, if you see time 0, the plane will be safe when $t > M$.

$$\forall(t : \mathrm{Time}).\, \xi_0(t > M \Rightarrow a > \texttt{safe})$$

Justifying and moving beyond over-simplification

The above example was simplified. What about that?

- Here's what we were trying to show:
  - Very different sorts of contracts encoded in a common language.
  - Nothing was ad-hoc about combining them.

Justifying and moving beyond over-simplification

The above example was simplified. What about that?

- Here's what we were trying to show:
  - Very different sorts of contracts encoded in a common language.
  - Nothing was ad-hoc about combining them.
- To make it more complex: first add another airplane.
  - I don't think this will be much more difficult.
  - First: replace altitude a with difference in altitude, $|a_1 - a_2|$.
  - Second: deal with the conditional given by sign of $a_1 - a_2$.
  - Ongoing work: find and deal with additional complexities.

Justifying and moving beyond over-simplification

The above example was simplified. What about that?

- Here's what we were trying to show:
  - Very different sorts of contracts encoded in a common language.
  - Nothing was ad-hoc about combining them.
- To make it more complex: first add another airplane.
  - I don't think this will be much more difficult.
  - First: replace altitude a with difference in altitude, $|a_1 - a_2|$.
  - Second: deal with the conditional given by sign of $a_1 - a_2$.
  - Ongoing work: find and deal with additional complexities.
- Another thing to consider: the wiring diagram changes over time.
  - Planes go in and out of communication with each other in time.
  - This is mathematically straightforward.
  - Rather than equating behaviors, $s_1 = s_2$, use $P \Rightarrow (s_1 = s_2)$.
  - "If P happens, we'll communicate." E.g. $a < t < b \Rightarrow (s_1 = s_2)$.

Outline

Summary

We discussed

- Behavior types as sheaves, forming a topos Shv(**Int**).

Summary

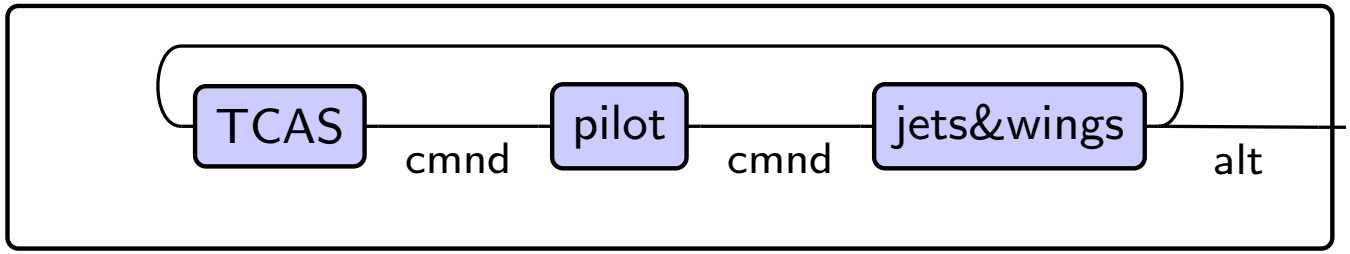
We discussed

- Behavior types as sheaves, forming a topos Shv(**Int**).
- Internal language of our sheaf topos:
  - The internal language and higher-order logic in any topos,
  - How this logic looks in our case ($\mathrm{Time}, \mathrm{Prop}$, modalities).

Summary

We discussed

- Behavior types as sheaves, forming a topos Shv(**Int**).
- Internal language of our sheaf topos:
  - The internal language and higher-order logic in any topos,
  - How this logic looks in our case ($\mathrm{Time}, \mathrm{Prop}$, modalities).
- Compositionality of behavior contracts.
- A simplified TCAS example.



Thanks for your time!