A higher-order temporal logic for dynamical systems

David I. Spivak* and Patrick Schultz

Mathematics Department
Massachusetts Institute of Technology

NIST Applied Category Theory Workshop
March 16, 2018

An example system

The National Airspace System (NAS).

- Goals of NextGen:
  - Double the number of airplanes in the sky;
  - Remain extremely safe.

[1] Traffic Collision Avoidance System.

An example system

The National Airspace System (NAS).

- Goals of NextGen:
  - Double the number of airplanes in the sky;
  - Remain extremely safe.
- Safe separation problem:
  - Planes need to remain at a safe distance.
  - Can't generally communicate directly.
  - Use radars, pilots, ground control, radios, and TCAS.[1]

[1] Traffic Collision Avoidance System.

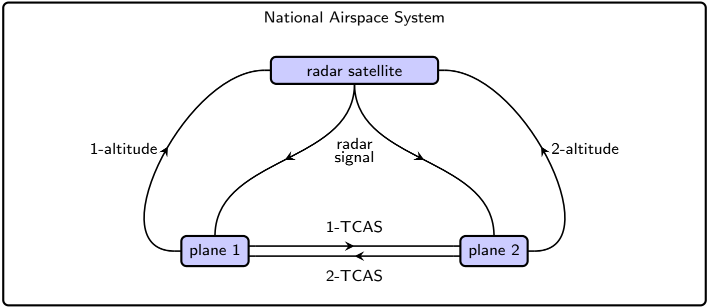
An example system

The National Airspace System (NAS).

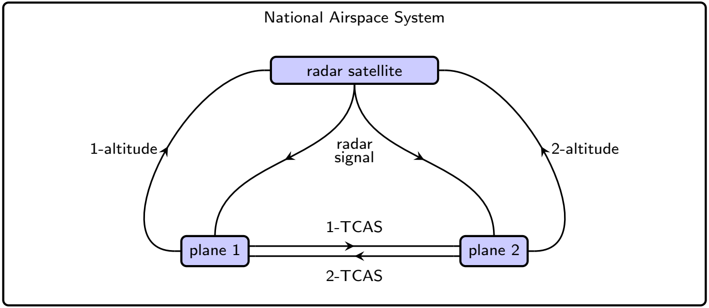
- Goals of NextGen:
  - Double the number of airplanes in the sky;
  - Remain extremely safe.
- Safe separation problem:
  - Planes need to remain at a safe distance.
  - Can't generally communicate directly.
  - Use radars, pilots, ground control, radios, and TCAS.[1]
- Systems of systems:
  - A great variety of interconnected systems.
  - Work in concert to enforce global property: safe separation.

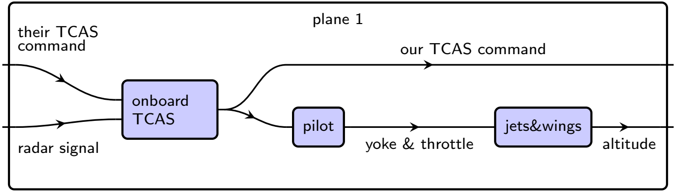
[1] Traffic Collision Avoidance System.

Systems of interacting systems in the NAS

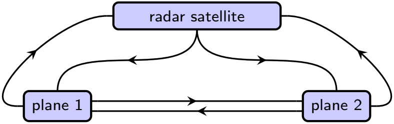


Systems of interacting systems in the NAS



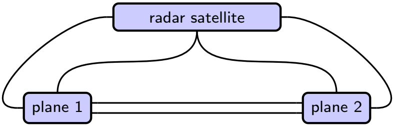


A hypergraph category



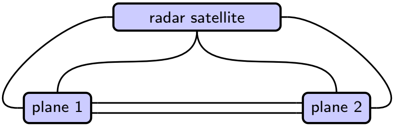
- What are these pictures?
    - Wires with arrows indicate "signal passing".

A hypergraph category



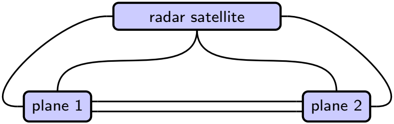
- What are these pictures?
  - Wires with arrows indicate "signal passing".
  - Drop the arrows for "variable sharing" perspective (Willems)
  - Either way, the planes and the radars are *constraints*.
  - "If I know you're close below me, I'll move up".

A hypergraph category



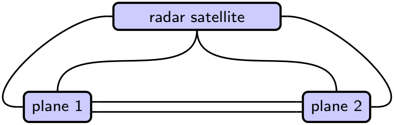
- What are these pictures?
  - Wires with arrows indicate "signal passing".
  - Drop the arrows for "variable sharing" perspective (Willems)
  - Either way, the planes and the radars are *constraints*.
  - "If I know you're close below me, I'll move up".
- What are these pictures formally?
  - Composition diagrams in a *hypergraph category*.
  - What we called "constraints" are formalized as relations.

Relations in a topos



Relations form a hypergraph category *in any topos* $\mathcal{E}$.

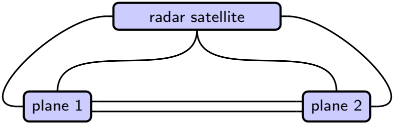
Relations in a topos



Relations form a hypergraph category *in any topos* $\mathcal{E}$.

- Example: relations in $\mathcal{E} = \mathbf{Set}$.
- Idea generalizes to arbitrary toposes.
- Every topos $\mathcal{E}$ has a subobject classifier Ω
- Relations on $A = A_1 \times \cdots \times A_n$ are morphisms $A \to \Omega$.

Relations in a topos



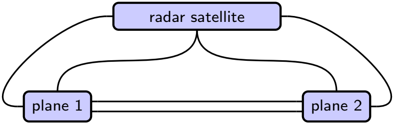
Relations form a hypergraph category *in any topos* $\mathcal{E}$.

- Example: relations in $\mathcal{E} = \mathbf{Set}$.
- Idea generalizes to arbitrary toposes.
- Every topos $\mathcal{E}$ has a subobject classifier Ω
- Relations on $A = A_1 \times \cdots \times A_n$ are morphisms $A \to \Omega$.

So... what's the topos for the National Airspace System?

- More generally, where do all these behaviors live?

Relations in a topos



Relations form a hypergraph category *in any topos* $\mathcal{E}$.

- Example: relations in $\mathcal{E} = \mathbf{Set}$.
- Idea generalizes to arbitrary toposes.
- Every topos $\mathcal{E}$ has a subobject classifier Ω
- Relations on $A = A_1 \times \cdots \times A_n$ are morphisms $A \to \Omega$.

So... what's the topos for the National Airspace System?

- More generally, where do all these behaviors live?
- They live in time.
- Goal: a good topos for studying behaviors (hence time).

NAS use-case as guide

What's the topos for the National Airspace System?

- This question was a major guide for our work.
- Need to combine many common frameworks into a "big tent".
  - Differential equations, continuous dynamical systems.
  - Labeled transition systems, discrete dynamical systems.
  - Delays, non-instantaneous rules.
  - Determinism, non-determinism.

NAS use-case as guide

What's the topos for the National Airspace System?

- This question was a major guide for our work.
- Need to combine many common frameworks into a "big tent".
  - Differential equations, continuous dynamical systems.
  - Labeled transition systems, discrete dynamical systems.
  - Delays, non-instantaneous rules.
  - Determinism, non-determinism.
- Need a logic in which to prove safety of the combined system.
  - Currently, combination process takes place in engineers' heads.
  - For NextGen, we may need to do better.

NAS use-case as guide

What's the topos for the National Airspace System?

- This question was a major guide for our work.
- Need to combine many common frameworks into a "big tent".
  - Differential equations, continuous dynamical systems.
  - Labeled transition systems, discrete dynamical systems.
  - Delays, non-instantaneous rules.
  - Determinism, non-determinism.
- Need a logic in which to prove safety of the combined system.
  - Currently, combination process takes place in engineers' heads.
  - For NextGen, we may need to do better.

Relationship to toposes:

- Toposes have an associated internal language and logic.
- Can use formal methods (proof assistants) to prove properties of NAS.

Plan of the talk

1. Define a topos $\mathcal{B}$ of behavior types.
2. Briefly discuss *temporal type theory*, which is sound in $\mathcal{B}$.
3. Return to our NAS use-case.

What is a topos and why?

Toposes—invented by Grothendieck—generalize topological spaces.

- Basic idea:
  - A topos tells you “what can live on a space”...
  - ...rather than telling you “what the space *is*”.
  - The space is just the habitat, or “site”, where stuff appears.

What is a topos and why?

Toposes—invented by Grothendieck—generalize topological spaces.

- Basic idea:
  - A topos tells you "what can live on a space"...
  - ...rather than telling you "what the space *is*".
  - The space is just the habitat, or "site", where stuff appears.
- Definition: a *topos* is the category of sheaves on a site.
- Two examples: topological spaces and databases.
  - Any topological space defines a site.
  - What lives there: vector fields, scalar fields; "bundles" of stuff.

What is a topos and why?

Toposes—invented by Grothendieck—generalize topological spaces.

- Basic idea:
  - A topos tells you "what can live on a space"...
  - ...rather than telling you "what the space *is*".
  - The space is just the habitat, or "site", where stuff appears.
- Definition: a *topos* is the category of sheaves on a site.
- Two examples: topological spaces and databases.
  - Any topological space defines a site.
  - What lives there: vector fields, scalar fields; "bundles" of stuff.
  - Any database schema S defines a site.
  - What lives there: all states of the database (S-instances).

What is a topos and why?

Toposes—invented by Grothendieck—generalize topological spaces.

- Basic idea:
  - A topos tells you "what can live on a space"...
  - ...rather than telling you "what the space *is*".
  - The space is just the habitat, or "site", where stuff appears.
- Definition: a *topos* is the category of sheaves on a site.
- Two examples: topological spaces and databases.
  - Any topological space defines a site.
  - What lives there: vector fields, scalar fields; "bundles" of stuff.
  - Any database schema S defines a site.
  - What lives there: all states of the database (S-instances).

Question: What's a good site on which *behaviors* can live?

What is a topos and why?

Toposes—invented by Grothendieck—generalize topological spaces.

- Basic idea:
  - A topos tells you "what can live on a space"...
  - ...rather than telling you "what the space *is*".
  - The space is just the habitat, or "site", where stuff appears.
- Definition: a *topos* is the category of sheaves on a site.
- Two examples: topological spaces and databases.
  - Any topological space defines a site.
  - What lives there: vector fields, scalar fields; "bundles" of stuff.
  - Any database schema S defines a site.
  - What lives there: all states of the database (S-instances).

Question: What's a good site on which *behaviors* can live?
Answer: roughly, Time. But what is that?

First guess: the space $\mathbb{R}$

A first guess: the space $\mathbb{R}$ as the site for behaviors.

First guess: the space $\mathbb{R}$

A first guess: the space $\mathbb{R}$ as the site for behaviors.

- What would a behavior type $B \in \mathsf{Shv}(\mathbb{R})$ be?
  - On objects:
    - For each open interval $(a, b) \subseteq \mathbb{R}$, a set $B(a, b)$
    - "The set of B-behaviors that can occur on (a, b)."

First guess: the space $\mathbb{R}$

A first guess: the space $\mathbb{R}$ as the site for behaviors.

- What would a behavior type $B \in \mathsf{Shv}(\mathbb{R})$ be?
  - On objects:
    - For each open interval $(a, b) \subseteq \mathbb{R}$, a set $B(a, b)$
    - "The set of B-behaviors that can occur on (a, b)."
  - On morphisms:
    - For each $a \leq a' < b' \leq b$, a function $B(a, b) \to B(a', b')$
    - "The B-way to restrict B-behaviors over subintervals."

First guess: the space $\mathbb{R}$

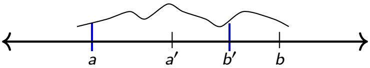
A first guess: the space $\mathbb{R}$ as the site for behaviors.

- What would a behavior type $B \in \mathrm{Shv}(\mathbb{R})$ be?
  - On objects:
    - For each open interval $(a, b) \subseteq \mathbb{R}$, a set $B(a, b)$
    - "The set of B-behaviors that can occur on (a, b)."
  - On morphisms:
    - For each $a \leq a' < b' \leq b$, a function $B(a, b) \to B(a', b')$
    - "The B-way to restrict B-behaviors over subintervals."
  - Gluing conditions:
    - "Continuity": $B(a, b) = \lim_{a<a'<b'<b} B(a', b')$.
    - "Composition": $B(a, b) = B(a, b') \times_{B(a',b')} B(a', b)$.

First guess: the space $\mathbb{R}$

A first guess: the space $\mathbb{R}$ as the site for behaviors.

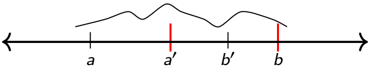
- What would a behavior type $B \in \mathrm{Shv}(\mathbb{R})$ be?
  - On objects:
    - For each open interval $(a, b) \subseteq \mathbb{R}$, a set $B(a, b)$
    - "The set of B-behaviors that can occur on (a, b)."
  - On morphisms:
    - For each $a \leq a' < b' \leq b$, a function $B(a, b) \to B(a', b')$
    - "The B-way to restrict B-behaviors over subintervals."
  - Gluing conditions:
    - "Continuity": $B(a, b) = \lim_{a<a'<b'<b} B(a', b')$.
    - "Composition": $B(a, b) = B(a, b') \times_{B(a',b')} B(a', b)$.



First guess: the space $\mathbb{R}$

A first guess: the space $\mathbb{R}$ as the site for behaviors.

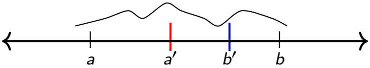
- What would a behavior type $B \in \mathrm{Shv}(\mathbb{R})$ be?
  - On objects:
    - For each open interval $(a, b) \subseteq \mathbb{R}$, a set $B(a, b)$
    - "The set of B-behaviors that can occur on (a, b)."
  - On morphisms:
    - For each $a \leq a' < b' \leq b$, a function $B(a, b) \longrightarrow B(a', b')$
    - "The B-way to restrict B-behaviors over subintervals."
  - Gluing conditions:
    - "Continuity": $B(a, b) = \lim_{a<a'<b'<b} B(a', b')$.
    - "Composition": $B(a, b) = B(a, b') \times_{B(a',b')} B(a', b)$.



First guess: the space $\mathbb{R}$

A first guess: the space $\mathbb{R}$ as the site for behaviors.

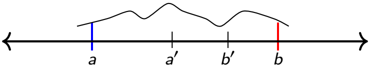
- What would a behavior type $B \in \mathrm{Shv}(\mathbb{R})$ be?
  - On objects:
    - For each open interval $(a, b) \subseteq \mathbb{R}$, a set $B(a, b)$
    - "The set of B-behaviors that can occur on (a, b)."
  - On morphisms:
    - For each $a \leq a' < b' \leq b$, a function $B(a, b) \longrightarrow B(a', b')$
    - "The B-way to restrict B-behaviors over subintervals."
  - Gluing conditions:
    - "Continuity": $B(a, b) = \lim_{a<a'<b'<b} B(a', b')$.
    - "Composition": $B(a, b) = B(a, b') \times_{B(a',b')} B(a', b)$.



First guess: the space $\mathbb{R}$

A first guess: the space $\mathbb{R}$ as the site for behaviors.

- What would a behavior type $B \in \mathrm{Shv}(\mathbb{R})$ be?
  - On objects:
    - For each open interval $(a, b) \subseteq \mathbb{R}$, a set $B(a, b)$
    - "The set of B-behaviors that can occur on (a, b)."
  - On morphisms:
    - For each $a \leq a' < b' \leq b$, a function $B(a, b) \to B(a', b')$
    - "The B-way to restrict B-behaviors over subintervals."
  - Gluing conditions:
    - "Continuity": $B(a, b) = \lim_{a<a'<b'<b} B(a', b')$.
    - "Composition": $B(a, b) = B(a, b') \times_{B(a',b')} B(a', b)$.



Why $\mathbb{R}$ is not preferable as the site

Two reasons *not to use* $\mathrm{Shv}(\mathbb{R})$ as our topos.

- 1. Often want to consider **non-composable** behaviors!
  - "Roughly monotonic": $\forall(t_1, t_2).\, t_1 + 5 \leq t_2 \Rightarrow f(t_1) \leq f(t_2)$.
  - "Don't move much": $\forall(t_1, t_2).\, -5 < f(t_1) - f(t_2) < 5$.
  - Neither of these have the "composition gluing".

Why $\mathbb{R}$ is not preferable as the site

Two reasons *not to use* $\mathrm{Shv}(\mathbb{R})$ as our topos.

- 1. Often want to consider **non-composable** behaviors!
  - "Roughly monotonic": $\forall (t_1, t_2).\, t_1 + 5 \leq t_2 \Rightarrow f(t_1) \leq f(t_2)$.
  - "Don't move much": $\forall (t_1, t_2).\, -5 < f(t_1) - f(t_2) < 5$.
  - Neither of these have the "composition gluing".
- 2. Want to compare behavior across different time windows.
  - Example: a delay is "the same behavior at different times."
  - $\mathrm{Shv}(\mathbb{R})$ sees no relationship between $B(0, 3)$ and $B(2, 5)$.

Why $\mathbb{R}$ is not preferable as the site

Two reasons *not to use* $\mathrm{Shv}(\mathbb{R})$ as our topos.

- 1. Often want to consider **non-composable** behaviors!
  - "Roughly monotonic": $\forall(t_1, t_2).\, t_1 + 5 \le t_2 \Rightarrow f(t_1) \le f(t_2)$.
  - "Don't move much": $\forall(t_1, t_2).\, -5 < f(t_1) - f(t_2) < 5$.
  - Neither of these have the "composition gluing".
- 2. Want to compare behavior across different time windows.
  - Example: a delay is "the same behavior at different times."
  - $\mathrm{Shv}(\mathbb{R})$ sees no relationship between $B(0, 3)$ and $B(2, 5)$.
  - To fix this, replace interval (a, b) by duration $b - a$.
  - "Translation invariance."

Why $\mathbb{R}$ is not preferable as the site

Two reasons *not to use* $\mathrm{Shv}(\mathbb{R})$ as our topos.

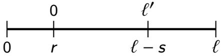
- 1. Often want to consider **non-composable** behaviors!
  - "Roughly monotonic": $\forall(t_1, t_2).\, t_1 + 5 \leq t_2 \Rightarrow f(t_1) \leq f(t_2)$.
  - "Don't move much": $\forall(t_1, t_2).\, -5 < f(t_1) - f(t_2) < 5$.
  - Neither of these have the "composition gluing".
- 2. Want to compare behavior across different time windows.
  - Example: a delay is "the same behavior at different times."
  - $\mathrm{Shv}(\mathbb{R})$ sees no relationship between $B(0, 3)$ and $B(2, 5)$.
  - To fix this, replace interval (a, b) by duration $b - a$.
  - "Translation invariance."

Discard composition gluing, add translation invariance.

Our choice of topos $\mathcal{B}$

Use $\mathbb{IR}_{/\rhd}$ the following site:

- Objects $= \{\ell \in \mathbb{R}_{\geq 0}\}$.
- $\mathrm{Hom}(\ell', \ell) = \{\langle r, s\rangle \mid r + \ell' + s = \ell\}$



- Coverage $\{\langle r, s\rangle \colon \ell' \to \ell \mid r > 0, s > 0\}$.
- When $r, s > 0$, write $\ell' \rightsquigarrow \ell$.[2]

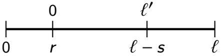
The topos of behavior types: $\mathcal{B} = \mathrm{Shv}(\mathbb{IR}_{/\rhd})$.

[2] Johnstone-Joyal's notation in "Continuous categories and exponentiable toposes".

Our choice of topos $\mathcal{B}$

Use $\mathbb{IR}_{/\triangleright}$ the following site:

- Objects $= \{\ell \in \mathbb{R}_{\geq 0}\}$.
- $\mathrm{Hom}(\ell', \ell) = \{\langle r, s\rangle \mid r + \ell' + s = \ell\}$



- Coverage $\{\langle r, s\rangle \colon \ell' \to \ell \mid r > 0, s > 0\}$.
- When $r, s > 0$, write $\ell' \rightsquigarrow \ell$.[2]

The topos of behavior types: $\mathcal{B} = \mathrm{Shv}(\mathbb{IR}_{/\triangleright})$.

- A sheaf X assigns a set of possible behaviors to each ℓ,
- And a restriction map to each included subinterval $\langle r, s\rangle \colon \ell' \to \ell$,
- Such that $X(\ell) \cong \lim_{\ell' \rightsquigarrow \ell} X(\ell')$.

[2] Johnstone-Joyal's notation in "Continuous categories and exponentiable toposes".

Type theory and toposes

Type theory is useful, e.g. in computer science.

- It's basically a bunch of language rules.

Type theory and toposes

Type theory is useful, e.g. in computer science.

- It's basically a bunch of language rules.
- E.g. simply-typed lambda calculus with sum types and quotient types.
  - Start with atomic types and atomic terms.
  - Build new types, terms, and propositions using constructors.
    - Types: $\mathbb{N}$, $\mathtt{Prop}$, products, arrows, sums, quotients.
    - Terms: tupling, projection, lambda abstraction, evaluation, etc.
    - Propositions: $\exists, \forall, \wedge, \vee, \neg, \Rightarrow, \Leftrightarrow, \top, \bot$.
  - Add axioms, which are logical statements.

Type theory and toposes

Type theory is useful, e.g. in computer science.

- It's basically a bunch of language rules.
- E.g. simply-typed lambda calculus with sum types and quotient types.
  - Start with atomic types and atomic terms.
  - Build new types, terms, and propositions using constructors.
    - Types: $\mathbb{N}$, $\mathtt{Prop}$, products, arrows, sums, quotients.
    - Terms: tupling, projection, lambda abstraction, evaluation, etc.
    - Propositions: $\exists, \forall, \wedge, \vee, \neg, \Rightarrow, \Leftrightarrow, \top, \bot$.
  - Add axioms, which are logical statements.

I thought this was dreadfully boring. Until I witnessed...

The Kripke-Joyal semantics

The Kripke-Joyal semantics is pretty neat.

- Start with atomic types, terms, and axioms from your topos.
- Kripke-Joyal is a machine that turns logic into topos-proofs.

The Kripke-Joyal semantics

The Kripke-Joyal semantics is pretty neat.

- Start with atomic types, terms, and axioms from your topos.
- Kripke-Joyal is a machine that turns logic into topos-proofs.
- Suppose you have any expression in the type theory.
  - It automatically has semantics in your topos.
  - That is, it means something about sheaves X.
  - $\forall(x : X)$ – "for all restriction maps and sections x..."
  - $\exists(x : X)$ – "there is a covering family and a section x in each..."
  - Each connective $\wedge, \vee, \Rightarrow$, means something sheafy.
- Statements and proofs are recursive, tree-like structures.
  - Kripke-Joyal recurses over that structure.
  - At each step, it unwinds the logic into restrictions, covers, sections.
  - It manages all the topos stuff and lets you believe you're in **Set**.

The Kripke-Joyal semantics: doing the heavy lifting.

Types in the topos $\mathcal{B}$

In this topos, you can study any sort of mathematical object

- You can study groups, topological spaces, databases, etc.
- There's only one caveat: everything occurs in time.

Types in the topos $\mathcal{B}$

In this topos, you can study any sort of mathematical object

- You can study groups, topological spaces, databases, etc.
- There's only one caveat: everything occurs in time.
    - A group object in this topos is a group that can change in time.
    - A database schema is one that can change in time.

Types in the topos $\mathcal{B}$

In this topos, you can study any sort of mathematical object

- You can study groups, topological spaces, databases, etc.
- There's only one caveat: everything occurs in time.
  - A group object in this topos is a group that can change in time.
  - A database schema is one that can change in time.
- There is a type $\mathbb{R}_{\mathsf{var}}$ of real numbers that change continuously in time.
  - It is a topological ring object just like real numbers always are.
  - Define temporal derivatives, rate of change through time, within the logic.
  - We prove logically that it satisfies the usual rules (linear, Leibniz)
  - And we check semantically that it actually is the derivative.

Differential equations

As a logical expression, derivatives work like anything else.

- Consider a differential equation, like

$$f(\dot{x}, \ddot{x}, a, b) = 0.$$

Differential equations

As a logical expression, derivatives work like anything else.

- Consider a differential equation, like

$$f(\dot{x}, \ddot{x}, a, b) = 0.$$

- Maybe $a, b : \mathbb{R}_{\mathrm{var}}$ are continuous functions of time.
- Regardless, $f(\dot{x}, \ddot{x}, a, b) = 0$ is just an equation in the logic.
  - Use it with $\top, \bot, \neg, \vee, \wedge, \Rightarrow, \exists, \forall$.
  - Can be combined with any other property.

The problem: safe altitude

Simplifying the safe separation problem.

- Real problem: safe separation for pairs of planes.
  - Components: Radars, pilots, thrusters/actuators.
  - Behavior types: Discrete signals, (continuous) diff-eqs, delays.

The problem: safe altitude

Simplifying the safe separation problem.

- Real problem: safe separation for pairs of planes.
    - Components: Radars, pilots, thrusters/actuators.
    - Behavior types: Discrete signals, (continuous) diff-eqs, delays.
- Simplification: safe altitude for one plane.
    - One radar, one pilot, one thruster.
    - Same behavior types: discrete, continuous, delay.

The problem: safe altitude

Simplifying the safe separation problem.

- Real problem: safe separation for pairs of planes.
    - Components: Radars, pilots, thrusters/actuators.
    - Behavior types: Discrete signals, (continuous) diff-eqs, delays.
- Simplification: safe altitude for one plane.
    - One radar, one pilot, one thruster.
    - Same behavior types: discrete, continuous, delay.

Goal: combine disparate guarantees to prove useful result.

Setup

Variables to be used, and their types:

$$t : \mathtt{Time}. \quad T, P : \mathrm{Cmnd}. \quad a : \mathbb{R}_{\pi}. \quad \mathtt{safe}, \mathtt{margin}, \mathtt{del}, \mathtt{rate} : \mathbb{Q}.$$

What these mean:

$t : \mathtt{Time}$.	time-line	(a clock).
$a : \mathbb{R}_{\mathrm{var}}$.	altitude	(continuously changing).
$T : \mathrm{Cmnd}$.	TCAS command	(occurs at discrete instants).
$P : \mathrm{Cmnd}$.	pilot's command	(occurs at discrete instants).
$\mathtt{safe} : \mathbb{Q}$.	safe altitude	(constant).
$\mathtt{margin} : \mathbb{Q}$.	margin-of-error	(constant).
$\mathtt{del} : \mathbb{Q}$.	pilot delay	(constant).
$\mathtt{rate} : \mathbb{Q}$.	maximal ascent rate	(constant).

Behavior contracts

$t : \mathtt{Time}$.	time-line	(a clock).
$a : \mathbb{R}_{\mathsf{var}}$.	altitude	(continuously changing).
T : Cmnd.	TCAS command	(occurs at discrete instants).
P : Cmnd.	pilot's command	(occurs at discrete instants).
$\mathtt{safe} : \mathbb{Q}$.	safe altitude	(constant).
$\mathtt{margin} : \mathbb{Q}$.	margin-of-error	(constant).
$\mathtt{del} : \mathbb{Q}$.	pilot delay	(constant).
$\mathtt{rate} : \mathbb{Q}$.	maximal ascent rate	(constant).

- $\theta_1 \coloneqq (\mathtt{margin} > 0) \wedge (a \geq 0)$.

Behavior contracts

t : $\mathtt{Time}$.	time-line	(a clock).
a : $\mathbb{R}_{\mathrm{var}}$.	altitude	(continuously changing).
T : Cmnd.	TCAS command	(occurs at discrete instants).
P : Cmnd.	pilot's command	(occurs at discrete instants).
$\mathtt{safe}$: $\mathbb{Q}$.	safe altitude	(constant).
$\mathtt{margin}$: $\mathbb{Q}$.	margin-of-error	(constant).
$\mathtt{del}$: $\mathbb{Q}$.	pilot delay	(constant).
$\mathtt{rate}$: $\mathbb{Q}$.	maximal ascent rate	(constant).

- $\theta_1 \coloneqq (\mathtt{margin} > 0) \wedge (a \geq 0)$.
- $\theta_2 \coloneqq (a > \mathtt{safe} + \mathtt{margin} \Rightarrow T = \mathtt{level})$.
- $\theta_2' \coloneqq (a < \mathtt{safe} + \mathtt{margin} \Rightarrow T = \mathtt{climb})$.

Behavior contracts

t : $\mathtt{Time}$.	time-line	(a clock).
a : $\mathbb{R}_{\text{var}}$.	altitude	(continuously changing).
T : Cmnd.	TCAS command	(occurs at discrete instants).
P : Cmnd.	pilot's command	(occurs at discrete instants).
$\mathtt{safe} : \mathbb{Q}$.	safe altitude	(constant).
$\mathtt{margin} : \mathbb{Q}$.	margin-of-error	(constant).
$\mathtt{del} : \mathbb{Q}$.	pilot delay	(constant).
$\mathtt{rate} : \mathbb{Q}$.	maximal ascent rate	(constant).

- $\theta_1 \coloneqq (\mathtt{margin} > 0) \wedge (a \geq 0)$.
- $\theta_2 \coloneqq (a > \mathtt{safe} + \mathtt{margin} \Rightarrow T = \mathtt{level})$.
- $\theta_2' \coloneqq (a < \mathtt{safe} + \mathtt{margin} \Rightarrow T = \mathtt{climb})$.
- $\theta_3 \coloneqq (P = \mathtt{level} \Rightarrow \dot{a} = 0) \wedge (P = \mathtt{climb} \Rightarrow \dot{a} = \mathtt{rate})$.

Behavior contracts

t : $\mathtt{Time}$.	time-line	(a clock).
a : $\mathbb{R}_{\mathrm{var}}$.	altitude	(continuously changing).
T : Cmnd.	TCAS command	(occurs at discrete instants).
P : Cmnd.	pilot's command	(occurs at discrete instants).
$\mathtt{safe} : \mathbb{Q}$.	safe altitude	(constant).
$\mathtt{margin} : \mathbb{Q}$.	margin-of-error	(constant).
$\mathtt{del} : \mathbb{Q}$.	pilot delay	(constant).
$\mathtt{rate} : \mathbb{Q}$.	maximal ascent rate	(constant).

- $\theta_1 \coloneqq (\mathtt{margin} > 0) \wedge (a \geq 0)$.
- $\theta_2 \coloneqq (a > \mathtt{safe} + \mathtt{margin} \Rightarrow T = \mathtt{level})$.
- $\theta_2' \coloneqq (a < \mathtt{safe} + \mathtt{margin} \Rightarrow T = \mathtt{climb})$.
- $\theta_3 \coloneqq (P = \mathtt{level} \Rightarrow \dot{a} = 0) \wedge (P = \mathtt{climb} \Rightarrow \dot{a} = \mathtt{rate})$.
- $\theta_4 \coloneqq \mathrm{is\_delayed}(\mathtt{del}, T, P)$.

θ_4 is an abbreviation for a longer logical condition.

Behavior contracts

$t : \mathtt{Time}$.	time-line	(a clock).
$a : \mathbb{R}_{\mathrm{var}}$.	altitude	(continuously changing).
T : Cmnd.	TCAS command	(occurs at discrete instants).
P : Cmnd.	pilot's command	(occurs at discrete instants).
$\mathtt{safe} : \mathbb{Q}$.	safe altitude	(constant).
$\mathtt{margin} : \mathbb{Q}$.	margin-of-error	(constant).
$\mathtt{del} : \mathbb{Q}$.	pilot delay	(constant).
$\mathtt{rate} : \mathbb{Q}$.	maximal ascent rate	(constant).

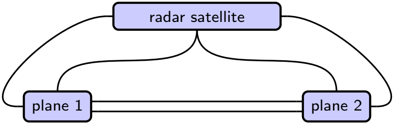
- $\theta_1 := (\mathtt{margin} > 0) \wedge (a \geq 0)$.
- $\theta_2 := (a > \mathtt{safe} + \mathtt{margin} \Rightarrow T = \mathtt{level})$.
- $\theta_2' := (a < \mathtt{safe} + \mathtt{margin} \Rightarrow T = \mathtt{climb})$.
- $\theta_3 := (P = \mathtt{level} \Rightarrow \dot{a} = 0) \wedge (P = \mathtt{climb} \Rightarrow \dot{a} = \mathtt{rate})$.
- $\theta_4 := \mathrm{is\_delayed}(\mathtt{del}, T, P)$.

θ_4 is an abbreviation for a longer logical condition.

- Can prove safe separation

$$\forall (t : \mathtt{Time}). \downarrow_0^t \left(t > \mathtt{del} + \frac{\mathtt{safe}}{\mathtt{rate}} \;\Rightarrow\; a \geq \mathtt{safe}\right).$$

Summary



- Idea: topos theory for integrating systems in a big tent.
- Many different formalisms for behavior, but they all occur in time.
  - We say that time occurs in intervals, which can be restricted.
  - Sheaves are behavior types: what can occur over intervals.
- The topos has a native "internal" logic.
  - Looks like usual set theory, $\forall, \exists, \wedge, \vee, \Rightarrow, \neg$; use formal methods
  - It compiles via Kripke-Joyal into complex facts about sheaves.

This temporal type theory is quite general, and fully compositional.

If you're interested in reading more

- Book (to be published by Springer).
  - *Temporal Type Theory*.
  - Freely available: `https://arxiv.org/abs/1710.10258`
  - Very technical.

If you're interested in reading more

- Book (to be published by Springer).
  - *Temporal Type Theory*.
  - Freely available: `https://arxiv.org/abs/1710.10258`
  - Very technical.
- Book (hopefully to be published by MIT Press).
  - *Seven Sketches in Compositionality*.
  - Freely available: `https://arxiv.org/abs/1803.05316`
  - Friendly! Chapter 7 is about this material.

If you're interested in reading more

- Book (to be published by Springer).
  - *Temporal Type Theory*.
  - Freely available: `https://arxiv.org/abs/1710.10258`
  - Very technical.
- Book (hopefully to be published by MIT Press).
  - *Seven Sketches in Compositionality*.
  - Freely available: `https://arxiv.org/abs/1803.05316`
  - Friendly! Chapter 7 is about this material.

Questions and comments are welcome. Thanks!