A higher-order temporal logic for dynamical systems

David I. Spivak

Department of Mathematics
Massachusetts Institute of Technology

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.

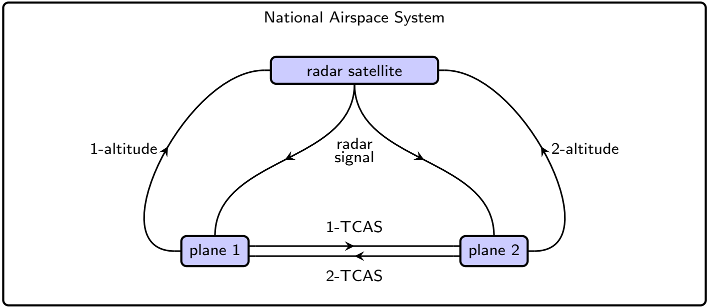
An example system

The National Airspace System (NAS).

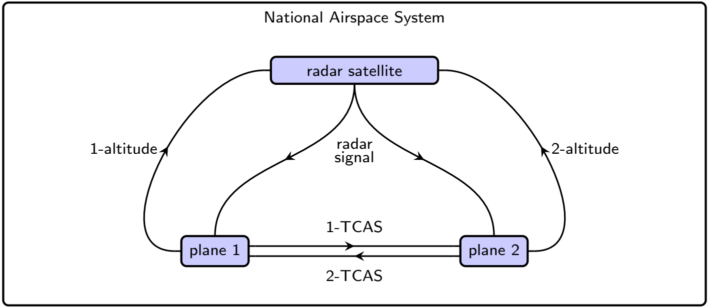
- Goals of NextGen:
  - Double the number of airplanes in the sky;
  - Remain extremely safe.
- Safe separation problem:
  - Planes need to remain at a safe distance.
  - Can't generally communicate directly.
  - Use radars, pilots, ground control, radios, and TCAS.[1]
- Systems of systems:
  - A great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

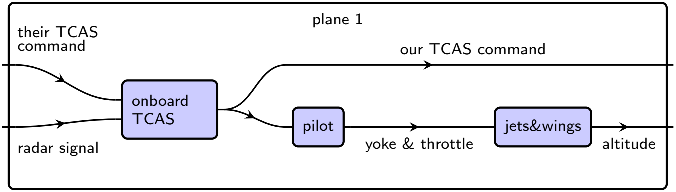
[1] Traffic Collision Avoidance System.

Systems of interacting systems in the NAS



Systems of interacting systems in the NAS





A poem

Anything that exists in time
has a certain set of possible behaviors
over any interval of time.

A poem

Anything that exists in time
has a certain set of possible behaviors
over any interval of time.

Its time-varying set
of behavioral possibilities
constitutes its type.

Behavior types

A *behavior type* T is a set of behavioral possibilities, changing in time.

- For each interval of time (a, b), a set $T(a, b)$.
- For each subinterval $(a', b') \subseteq (a, b)$, a function

$$\rho\colon T(a, b) \to T(a', b').$$

  "Restrict any T-possible movie clip to a sub-clip."
- And finally, $T(a, b) = \lim_{a<a'<b'<b} T(a', b')$.

Behavior types

A *behavior type* T is a set of behavioral possibilities, changing in time.

- For each interval of time (a, b), a set $T(a, b)$.
- For each subinterval $(a', b') \subseteq (a, b)$, a function

$$\rho\colon T(a, b) \to T(a', b').$$

"Restrict any T-possible movie clip to a sub-clip."

- And finally, $T(a, b) = \lim_{a<a'<b'<b} T(a', b')$.

This is (roughly) the definition of T being a *sheaf* on time-intervals.

Behavior types

A *behavior type* T is a set of behavioral possibilities, changing in time.

- For each interval of time (a, b), a set $T(a, b)$.
- For each subinterval $(a', b') \subseteq (a, b)$, a function

$$\rho \colon T(a, b) \to T(a', b').$$

  "Restrict any T-possible movie clip to a sub-clip."
- And finally, $T(a, b) = \lim_{a<a'<b'<b} T(a', b')$.

This is (roughly) the definition of T being a *sheaf* on time-intervals.

- Consider bicycle attitudes as a behavior type; what is $T(a, b)$?

Behavior types

A *behavior type* T is a set of behavioral possibilities, changing in time.

- For each interval of time (a, b), a set $T(a, b)$.
- For each subinterval $(a', b') \subseteq (a, b)$, a function

$$\rho\colon T(a, b) \to T(a', b').$$

  "Restrict any T-possible movie clip to a sub-clip."
- And finally, $T(a, b) = \lim_{a<a'<b'<b} T(a', b')$.

This is (roughly) the definition of T being a *sheaf* on time-intervals.

- Consider bicycle attitudes as a behavior type; what is $T(a, b)$?
- How about for the earth-sun system or a web app?

Behavior types

A *behavior type* T is a set of behavioral possibilities, changing in time.

- For each interval of time (a, b), a set $T(a, b)$.
- For each subinterval $(a', b') \subseteq (a, b)$, a function

$$\rho\colon T(a, b) \longrightarrow T(a', b').$$

  "Restrict any T-possible movie clip to a sub-clip."
- And finally, $T(a, b) = \lim_{a<a'<b'<b} T(a', b')$.

This is (roughly) the definition of T being a *sheaf* on time-intervals.

- Consider bicycle attitudes as a behavior type; what is $T(a, b)$?
- How about for the earth-sun system or a web app?

More general than hybrid: continuously overlapping, not piecewise, models.

What are the consequences

We make a fairly simple commitment on behaving systems T:

- For each time-interval, T has a set of behavioral possibilities.
- For each sub-interval, you can "clip the movie"
- And a continuity requirement.

What are the consequences

We make a fairly simple commitment on behaving systems T:

- For each time-interval, T has a set of behavioral possibilities.
- For each sub-interval, you can "clip the movie"
- And a continuity requirement.
- Mathematically, T is a *sheaf*:
  - For each interval of time (a, b), a set $T(a, b)$.
  - For each $(a', b') \subseteq (a, b)$, function $\rho\colon T(a, b) \longrightarrow T(a', b')$.
  - Continuity, $T(a, b) = \lim_{a<a'<b'<b} T(a', b')$.

What are the consequences

We make a fairly simple commitment on behaving systems T:

- For each time-interval, T has a set of behavioral possibilities.
- For each sub-interval, you can "clip the movie"
- And a continuity requirement.
- Mathematically, T is a *sheaf*:
  - For each interval of time (a, b), a set $T(a, b)$.
  - For each $(a', b') \subseteq (a, b)$, function $\rho\colon T(a, b) \longrightarrow T(a', b')$.
  - Continuity, $T(a, b) = \lim_{a<a'<b'<b} T(a', b')$.

Sheaf / topos theory give a wealth of consequences for this commitment.

- Big one: a higher-order logic of time.
- A flexible system for building new behavior types from old.
- I'll discuss a new data structure for monitoring behavior.

Interconnection of behaviors

We can interconnect different systems using ‘sheaf morphisms’.

Interconnection of behaviors

We can interconnect different systems using 'sheaf morphisms'.

- Example: T is a bicycle, U is its pedals.
- A sheaf morphism $f\colon T \to U$ consists of:
  - ...for each interval (a, b), a function $f_{(a,b)}\colon T(a, b) \to U(a, b)$
  - ...such that all these 'commute' with restriction maps.

$$\begin{tikzcd} T(a,b) \arrow[r, "f_{(a,b)}"] \arrow[d, "\rho"'] & U(a,b) \arrow[d, "\rho"] \\ T(a',b') \arrow[r, "f_{(a',b')}"'] & U(a',b'). \end{tikzcd}$$

Interconnection of behaviors

We can interconnect different systems using 'sheaf morphisms'.

- Example: T is a bicycle, U is its pedals.
- A sheaf morphism $f\colon T \to U$ consists of:
  - ...for each interval (a, b), a function $f_{(a,b)}\colon T(a, b) \to U(a, b)$
  - ...such that all these 'commute' with restriction maps.

$$\begin{tikzcd} T(a,b) \arrow[r, "f_{(a,b)}"] \arrow[d, "\rho"'] & U(a,b) \arrow[d, "\rho"] \\ T(a',b') \arrow[r, "f_{(a',b')}"'] & U(a',b'). \end{tikzcd}$$

Systems interact by 'sharing variables': human $\to$ pedals $\leftarrow$ bicycle.

Interconnection of behaviors

We can interconnect different systems using 'sheaf morphisms'.

- Example: T is a bicycle, U is its pedals.
- A sheaf morphism $f \colon T \longrightarrow U$ consists of:
  - ...for each interval (a, b), a function $f_{(a,b)} \colon T(a, b) \longrightarrow U(a, b)$
  - ...such that all these 'commute' with restriction maps.

$$\begin{tikzcd} T(a,b) \arrow[r, "f_{(a,b)}"] \arrow[d, "\rho"'] & U(a,b) \arrow[d, "\rho"] \\ T(a',b') \arrow[r, "f_{(a',b')}"'] & U(a',b'). \end{tikzcd}$$

Systems interact by 'sharing variables': human $\longrightarrow$ pedals $\longleftarrow$ bicycle.

- Variable sharing between X and Y is captured with sheaf morphisms

$$X \xrightarrow{f} V \xleftarrow{g} Y.$$

- Coupled system's behavior type is the sheaf $\{(x, y) \mid f(x) = g(y)\}$.

Mathematics in a topos

In the last line above we said "the sheaf $\{(x, y) \mid f(x) = g(y)\}$."

...Isn't that notation for a set?

Mathematics in a topos

In the last line above we said "the sheaf $\{(x, y) \mid f(x) = g(y)\}$."

...Isn't that notation for a set?

- Sheaves act to an astounding degree like sets.
  - You can take unions, products, subsets, etc. of sheaves.
  - Abstractions: just like matrices and reals form a *ring*...
  - ... both sets and sheaves form a *topos*.

Mathematics in a topos

In the last line above we said "the sheaf $\{(x, y) \mid f(x) = g(y)\}$."

...Isn't that notation for a set?

- Sheaves act to an astounding degree like sets.
  - You can take unions, products, subsets, etc. of sheaves.
  - Abstractions: just like matrices and reals form a *ring*...
  - ... both sets and sheaves form a *topos*.
- In particular, writing in set-notation unambiguously defines a sheaf.
  - The above refers to the sheaf sending any (a, b) to the set

$$\{(x, y) \in X(a, b) \times Y(a, b) \mid f_{(a,b)}(x) = g_{(a,b)}(y)\}.$$

Mathematics in a topos

In the last line above we said "the sheaf $\{(x, y) \mid f(x) = g(y)\}$."

...Isn't that notation for a set?

- Sheaves act to an astounding degree like sets.
  - You can take unions, products, subsets, etc. of sheaves.
  - Abstractions: just like matrices and reals form a *ring*...
  - ... both sets and sheaves form a *topos*.
- In particular, writing in set-notation unambiguously defines a sheaf.
  - The above refers to the sheaf sending any (a, b) to the set

$$\{(x, y) \in X(a, b) \times Y(a, b) \mid f_{(a,b)}(x) = g_{(a,b)}(y)\}.$$

All math constructions make sense in the topos of behavior types.

- Write usual def. of groups, rings, vector spaces, manifolds, etc.
- As sheaves, resulting mathematical objects *change through time*.
- I.e. groups, rings, etc. that change through time.

Logic for sets vs. sheaves

Like in sets, one can apply logic to sheaves.

- We have statements like $\forall(y : Y).\, \exists(x : X).\, f(x) = y$.
- It means that X simulates every Y-behavior, via $f : X \to Y$.

Logic for sets vs. sheaves

Like in sets, one can apply logic to sheaves.

- We have statements like $\forall(y : Y).\, \exists(x : X).\, f(x) = y$.
- It means that X simulates every Y-behavior, via $f : X \to Y$.

Sets are like behaviors if space-time consisted of just one point.

- A proposition like "13 is prime" is timeless.
- But consider "I'll visit you at least once every year."
- How do you track the truth of this behavioral assertion over time?

Logic for sets vs. sheaves

Like in sets, one can apply logic to sheaves.

- We have statements like $\forall(y : Y).\, \exists(x : X).\, f(x) = y$.
- It means that X simulates every Y-behavior, via $f : X \to Y$.

Sets are like behaviors if space-time consisted of just one point.

- A proposition like "13 is prime" is timeless.
- But consider "I'll visit you at least once every year."
- How do you track the truth of this behavioral assertion over time?

Truth-values for sets are different than truth values for behaviors.

- The truth values for sets are: `true`, `false`.
- These make sense for timeless things, like properties of integers.
- Time-varying behaviors need time-varying truth values.

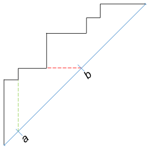
Truth values via the 'intervallic monitor'

A time-varying behavior satisfies a proposition sometimes and not others.

Truth values via the ‘intervallic monitor’

A time-varying behavior satisfies a proposition sometimes and not others.

- We can monitor such a proposition through time.
- Namely, plot at each moment “duration of perfect compliance.”

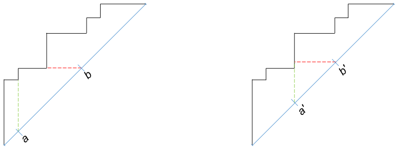


The target is in the field-of-view of at least one sensor for the duration.

Truth values via the 'intervallic monitor'

A time-varying behavior satisfies a proposition sometimes and not others.

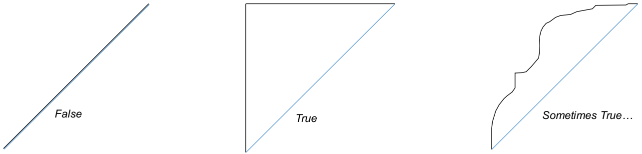
- We can monitor such a proposition through time.
- Namely, plot at each moment "duration of perfect compliance."



The target is in the field-of-view of at least one sensor for the duration.

What are the truth values?

- Rather than `true` or `false` as truth values, ...
- ...the 'intervalic monitors' are the truth values of the topos.



Logic of behavior types

Logic involves operations and quantification.

- Logic requires operations: AND, OR, IMPLIES, NOT ($\wedge, \vee, \Rightarrow, \neg$)...
- ...and quantification: FORALL and EXISTS ($\forall, \exists$).

Logic of behavior types

Logic involves operations and quantification.

- Logic requires operations: AND, OR, IMPLIES, NOT ($\wedge, \vee, \Rightarrow, \neg$)...
- ...and quantification: FORALL and EXISTS ($\forall, \exists$).

These all make sense for intervallic monitors P, Q.

- Take $P \wedge Q$ to be $\min(P, Q)$.
- Take $P \vee Q$ to be $\max(P, Q)$.

Logic of behavior types

Logic involves operations and quantification.

- Logic requires operations: AND, OR, IMPLIES, NOT ($\wedge, \vee, \Rightarrow, \neg$)...
- ...and quantification: FORALL and EXISTS ($\forall, \exists$).

These all make sense for intervallic monitors P, Q.

- Take $P \wedge Q$ to be $\min(P, Q)$.
- Take $P \vee Q$ to be $\max(P, Q)$.
- Similarly we can define the other connectives and quantification.
- All rules of standard higher-order logic hold for intervallic monitors.

HOTL: ordinary higher order logic + $\texttt{Time}$ + axioms

Higher-order logic is more expressive than first-order logic

- With HOL, you can reason over functions or subsets.
- Reasoning is almost the same, except constructive.
- This means you can use a proof assistant, like Coq or Lean.

$(\texttt{Threat} = 1) \Rightarrow \exists (p_1, p_2 : \mathsf{Pilot}).\, p_1 \neq p_2 \wedge \texttt{InFlight}(p_1) \wedge \texttt{InFlight}(p_2)$

HOTL: ordinary higher order logic + $\mathtt{Time}$ + axioms

Higher-order logic is more expressive than first-order logic

- With HOL, you can reason over functions or subsets.
- Reasoning is almost the same, except constructive.
- This means you can use a proof assistant, like Coq or Lean.

$(\mathtt{Threat} = 1) \Rightarrow \exists (p_1, p_2 : \mathsf{Pilot}).\, p_1 \neq p_2 \wedge \mathtt{InFlight}(p_1) \wedge \mathtt{InFlight}(p_2)$

HOTL, our temporal Logic, adds a type $\mathtt{Time}$ and about 10 axioms.

- Behavior of $\mathtt{Time}$ is that of a perfect clock (with any $t = 0$ point).
- Example axiom: $\mathtt{Time}$ is an $\mathbb{R}$-torsor:

$$\forall (t : \mathtt{Time})(r : \mathbb{R}).\, \exists ! (t' : \mathtt{Time}).\, t + r = t'.$$

$$\forall (t, t' : \mathtt{Time}).\, \exists ! (r : \mathbb{R}).\, t + r = t'.$$

HOTL is quite expressive

Completely within the logic, one can define derivatives.

- This lets you define differential equations, inclusions, etc.
- Pure logic: $\forall(t : \mathtt{Time}).\, \exists(a : \mathbb{R}).\, 0 < \ddot{x} + \dot{x} - \sin(x^2) + a < 5t.$

HOTL is quite expressive

Completely within the logic, one can define derivatives.

- This lets you define differential equations, inclusions, etc.
- Pure logic: $\forall (t : \mathtt{Time}).\, \exists (a : \mathbb{R}).\, 0 < \ddot{x} + \dot{x} - \sin(x^2) + a < 5t$.

One can also define hybrid systems (discrete jumps) in the logic.

- Advantage of defining things "in the logic":
- It's a self-contained system.
- The logic is not man-made, but drops out of topos theory.

HOTL is quite expressive

Completely within the logic, one can define derivatives.

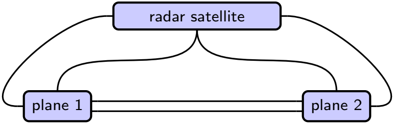
- This lets you define differential equations, inclusions, etc.
- Pure logic: $\forall(t : \mathtt{Time}).\, \exists(a : \mathbb{R}).\, 0 < \ddot{x} + \dot{x} - \sin(x^2) + a < 5t$.

One can also define hybrid systems (discrete jumps) in the logic.

- Advantage of defining things “in the logic”:
- It’s a self-contained system.
- The logic is not man-made, but drops out of topos theory.

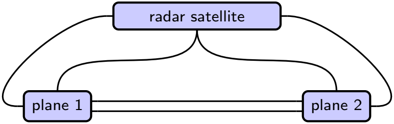
A “big tent” formalism for any sort of behavior.

Summary



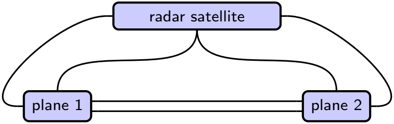
- Idea: sheaves for integrating very general behavior systems.

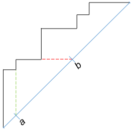
Summary



- Idea: sheaves for integrating very general behavior systems.
- 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.

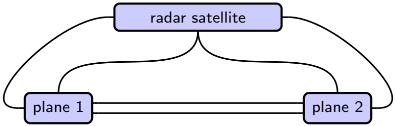
Summary

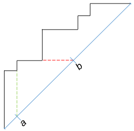




- Idea: sheaves for integrating very general behavior systems.
- Many different formalisms for behavior, but they all occur in time.
  - We say that time occurs in intervals, which can be restricted.
  - Sheaves are behavior types: what can occur over intervals.
- The sheaf topos has a native ‘internal logic’.
  - Looks like usual set theory, $\forall, \exists, \wedge, \vee, \Rightarrow, \neg$; use formal methods
  - Truth values are intervallic monitors. A 2-D concept of time.
  - Essential for capturing time-patterns: dwell-time, synchrony, etc.

Summary





- Idea: sheaves for integrating very general behavior systems.
- 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 sheaf topos has a native ‘internal logic’.
  - Looks like usual set theory, $\forall, \exists, \wedge, \vee, \Rightarrow, \neg$; use formal methods
  - Truth values are intervallic monitors. A 2-D concept of time.
  - Essential for capturing time-patterns: dwell-time, synchrony, etc.
- “Big tent”: import proofs from simpler systems and combine them.

If you're interested in reading more

- Book (Springer-Birkäuser).
  - *Temporal Type Theory*.
  - Freely available: `https://arxiv.org/abs/1710.10258`
  - Technical.

If you're interested in reading more

- Book (Springer-Birkäuser).
  - *Temporal Type Theory*.
  - Freely available: `https://arxiv.org/abs/1710.10258`
  - Technical.
- Book (Cambridge University Press).
  - *Seven Sketches in Compositionality*.
  - Freely available: `https://arxiv.org/abs/1803.05316`
  - Chapter 7 is about this material.
  - Friendly!

If you're interested in reading more

- Book (Springer-Birkäuser).
  - *Temporal Type Theory*.
  - Freely available: `https://arxiv.org/abs/1710.10258`
  - Technical.
- Book (Cambridge University Press).
  - *Seven Sketches in Compositionality*.
  - Freely available: `https://arxiv.org/abs/1803.05316`
  - Chapter 7 is about this material.
  - Friendly!

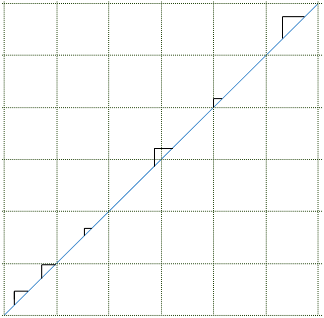
Questions and comments are welcome. Thanks!

Extra slide on intervallic monitors

Sample assertion: "I will visit you at least once per month."

$$\forall(t : \texttt{Time}).\, {\downarrow}^{t}_{[0,1]} \exists(r : \mathbb{R}).\, (0 < r < 1) \wedge @^{t}_{[r,r]}\texttt{Visit}$$

I'm visiting



I'm compliant with my assertion