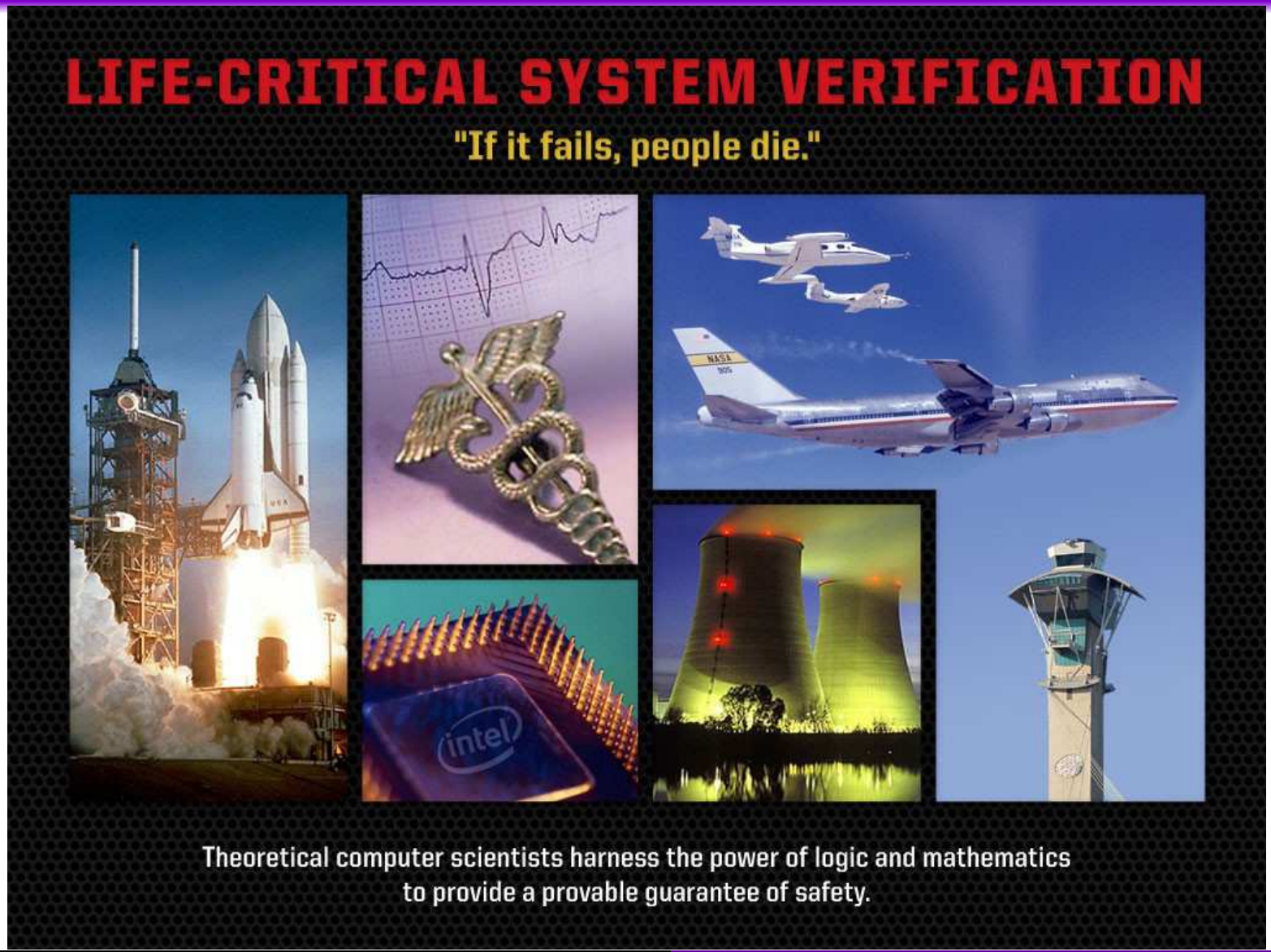
Logic for Learning: A Challenge Talk

Workshop on Learning and Logic

Kristin Yvonne Rozier
Iowa State University

February 10, 2025

Challenge: Some Systems Must Only Learn Safe Actions



– Safe Learning and Safe Acting –

What is learning?

- adding a behavior to an automated system in response to some observed pattern of operation
  - can be performed by a person or a machine
  - can take many forms (automated, semi-automated)
- "safe learning:" learned behavior is a safe action

– Safe Learning and Safe Acting –

What is learning?

- adding a behavior to an automated system in response to some observed pattern of operation
  - can be performed by a person or a machine
  - can take many forms (automated, semi-automated)
- "safe learning:" learned behavior is a safe action

What is safe acting?

- performing an action that:
  - does not harm humans
  - may prevent harm resulting from no action

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

- October 29, 2018

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

- October 29, 2018
- nose-up tendency
- Maneuvering Characteristics Augmentation System (MCAS) programmed to trim
- Left Primary Flight Display (PFD) fixed October 28

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

- October 29, 2018
- nose-up tendency
- Maneuvering Characteristics Augmentation System (MCAS) programmed to trim
- Left Primary Flight Display (PFD) fixed October 28
- AoA sensor passed pre-flight check

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

- October 29, 2018
- nose-up tendency
- Maneuvering Characteristics Augmentation System (MCAS) programmed to trim
- Left Primary Flight Display (PFD) fixed October 28
- AoA sensor passed pre-flight check
- AoA sensor detected approaching stall
- Left control column stick shaker alert

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

- October 29, 2018
- nose-up tendency
- Maneuvering Characteristics Augmentation System (MCAS) programmed to trim
- Left Primary Flight Display (PFD) fixed October 28
- AoA sensor passed pre-flight check
- AoA sensor detected approaching stall
- Left control column stick shaker alert
- PIC diagnosed *left PFD fault*
  - continuous 20° displacement of left AoA
  - left and right instruments indicating different altitudes
  - faulty AoA, altitude $\Rightarrow$ stall range
- automatic AND trim (*Aircraft Nose Down*)

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

- October 29, 2018
- nose-up tendency
- Maneuvering Characteristics Augmentation System (MCAS) programmed to trim
- Left Primary Flight Display (PFD) fixed October 28
- AoA sensor passed pre-flight check
- AoA sensor detected approaching stall
- Left control column stick shaker alert
- PIC diagnosed *left PFD fault*
  - continuous 20° displacement of left AoA
  - left and right instruments indicating different altitudes
  - faulty AoA, altitude ⇒ stall range
- automatic AND trim (*Aircraft Nose Down*)
- Crash into Java Sea 11 min after takeoff, killing all 189 on-board

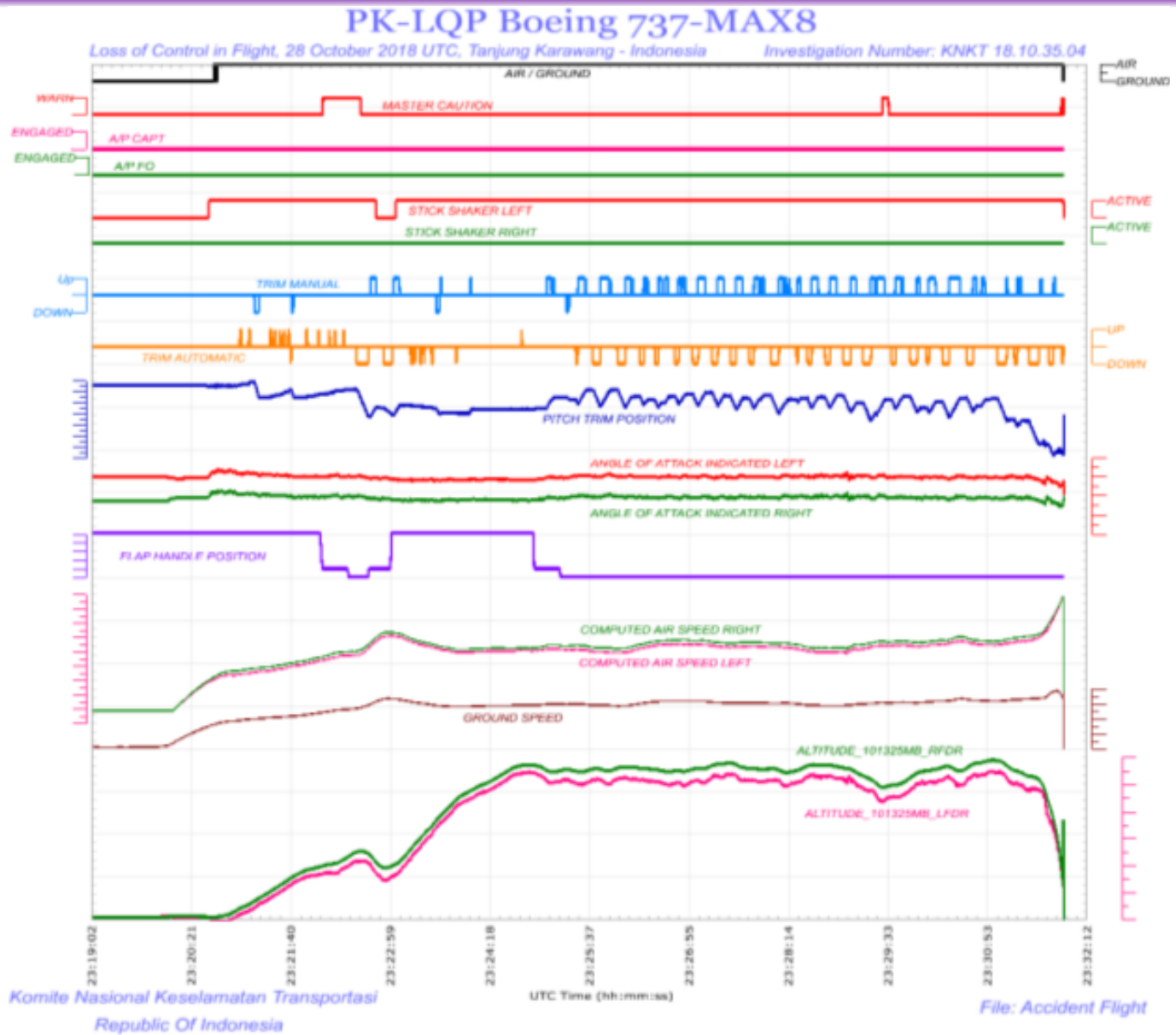


Figure 5: The significant parameters from the accident flight

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

This is Unsafe Learning!

- The 737-8 MAX tends to pitch up
- High AoA + low airspeed + low altitude = possible stall, crash
- Left and right sensors tend to agree; only need to check one
- Fixed sensors for left PFD passed pre-flight check; tend to function correctly

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

This is Unsafe Learning!

- The 737-8 MAX tends to pitch up
- High AoA + low airspeed + low altitude = possible stall, crash
- Left and right sensors tend to agree; only need to check one
- Fixed sensors for left PFD passed pre-flight check; tend to function correctly

Learning was done by humans!
Is there hope for autonomous systems?

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

How can pilots compensate for a malfunction if they are not aware of how it works?[1]

What we want:

"I'm doing this because the **left AoA sensor** indicates AoA is **above the threshold** of X; at **low altitude threshold** Y from **left altitude sensor** this indicates a **likely stall**."

[1] Les Abend. "Lion Air crash: Is it safe to get on a Boeing 737 MAX plane?" CNN Opinion, Updated 11:17 PM ET, November 28, 2018.

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

"Sanity Check" Specifications Relevant to this Mission:

- The AoA cannot be 20° different between two sides of the aircraft
- The altitude cannot be multiple values simultaneously.
- Altitude, airspeed verified by ATC to pilots, not autonomous system
- The pilot should not be fighting the stick in manual flight mode.
- The left and right PFD should agree; PIC and SIC should not have different control column modes like stick shake

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

"Sanity Check" Specifications Relevant to this Mission:

- The AoA cannot be 20° different between two sides of the aircraft
- The altitude cannot be multiple values simultaneously.
- Altitude, airspeed verified by ATC to pilots, not autonomous system
- The pilot should not be fighting the stick in manual flight mode.
- The left and right PFD should agree; PIC and SIC should not have different control column modes like stick shake

These *sanity checks* might have prevented the crash

A Recent Motivation...
Crash of Lion Air's Flight 610 Boeing 737-8 MAX

"Sanity Check" Specifications Relevant to this Mission:

- The AoA cannot be 20° different between two sides of the aircraft
- The altitude cannot be multiple values simultaneously.
- Altitude, airspeed verified by ATC to pilots, not autonomous system
- The pilot should not be fighting the stick in manual flight mode.
- The left and right PFD should agree; PIC and SIC should not have different control column modes like stick shake

These *sanity checks* might have prevented the crash

Safety specifications enable safe learning/acting, explainability

How do we know an action is safe?

How do we know an action is safe?

- Need a **proof**!

How do we know an action is safe?

- Need a **proof**!
  - Proof that the action is within a safety region?

How do we know an action is safe?

- Need a **proof**!
  - Proof that the action is within a safety region?
  - Proof that harmful actions aren't within the behavior space?

How do we know an action is safe?

- Need a **proof**!
  - Proof that the action is within a safety region?
  - Proof that harmful actions aren't within the behavior space?
  - ...

We need a specification of what is safe!

How do we know an action is safe?

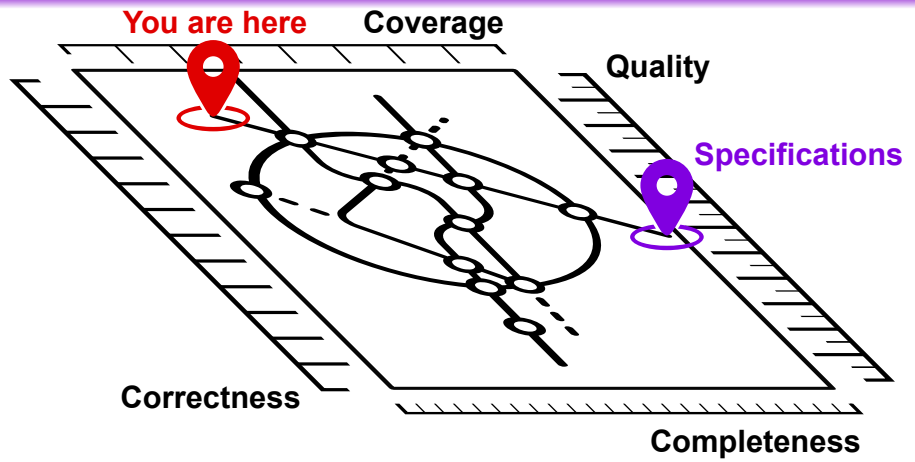
- Need a **proof**!
  - Proof that the action is within a safety region?
  - Proof that harmful actions aren't within the behavior space?
  - ...
- Need a **specification**!
  - What are the safety requirements?
  - What are the assumed safety bounds?
  - How do we identify a violation?

How do we know an action is safe?

- Need a **proof**!
  - Proof that the action is within a safety region?
  - Proof that harmful actions aren't within the behavior space?
  - ...
- Need a **specification**!
  - What are the safety requirements?
  - What are the assumed safety bounds?
  - How do we identify a violation?
- Need a way of checking the **implementation** follows the **proof**, generated from the **specification**

What Is Wrong With This Picture?

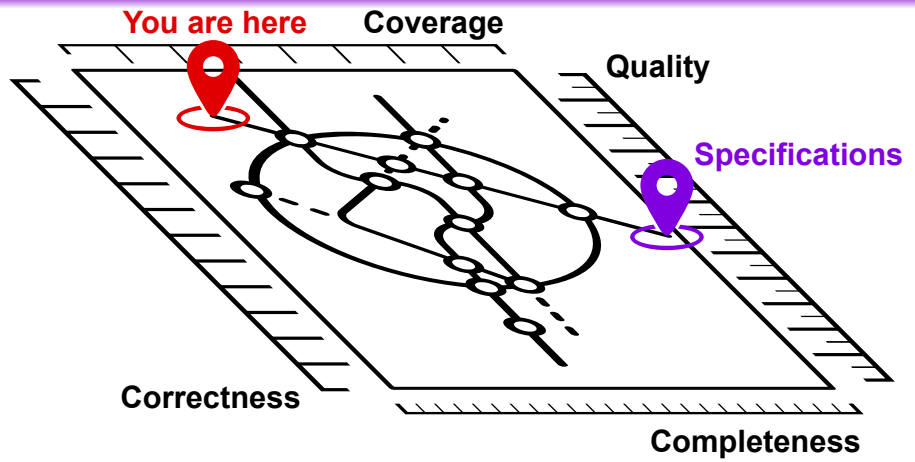
Specification: The Biggest Bottleneck in Formal Methods and Autonomy[2]



- Where are we now?
  - Continuously re-assess . . .
- Where will we get specifications from?
- How should we measure specification quality?
- How do we best use specifications?
- How should we organize specifications?

[2] For expansions on these ideas, see: K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

Specification: The Biggest Bottleneck in Formal Methods and Autonomy[2]



- Where are we now?
  - Continuously re-assess . . .
- Where will we get specifications from?
- How should we measure specification quality?
- How do we best use specifications?
- How should we organize specifications?

. . . in the context of learning, autonomously acting systems?

[2] For expansions on these ideas, see: K.Y.Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE, 2016.

Down a Level: What is Safe Learning?

What are the inputs and outputs?

Safe Learning In Six Definitions

1. Learning within **safety bounds**

Safe Learning In Six Definitions

1. Learning within **safety bounds**
2. Learning safe behaviors

Safe Learning In Six Definitions

1. Learning within **safety bounds**
2. Learning safe behaviors $\rightarrow$ learning safety requirements

Safe Learning In Six Definitions

1. Learning within **safety bounds**
2. Learning safe behaviors → learning safety requirements → **safe behavior genesis**

Safe Learning In Six Definitions

1. Learning within **safety bounds**
2. Learning safe behaviors → learning safety requirements → **safe behavior genesis**
3. **Refining behaviors** to be more safe/conservative

Safe Learning In Six Definitions

1. Learning within **safety bounds**
2. Learning safe behaviors → learning safety requirements → **safe behavior genesis**
3. **Refining behaviors** to be more safe/conservative
4. Learning that **generates verification artifacts**

Safe Learning In Six Definitions

1. Learning within **safety bounds**
2. Learning safe behaviors → learning safety requirements → **safe behavior genesis**
3. **Refining behaviors** to be more safe/conservative
4. Learning that **generates verification artifacts**
   - Learning that **passes verification tests**

Safe Learning In Six Definitions

1. Learning within **safety bounds**
2. Learning safe behaviors → learning safety requirements → **safe behavior genesis**
3. **Refining behaviors** to be more safe/conservative
4. Learning that **generates verification artifacts**
   - Learning that **passes verification tests**
5. Learning that **obeys temporal contracts**, enforced my MC or RV[3]

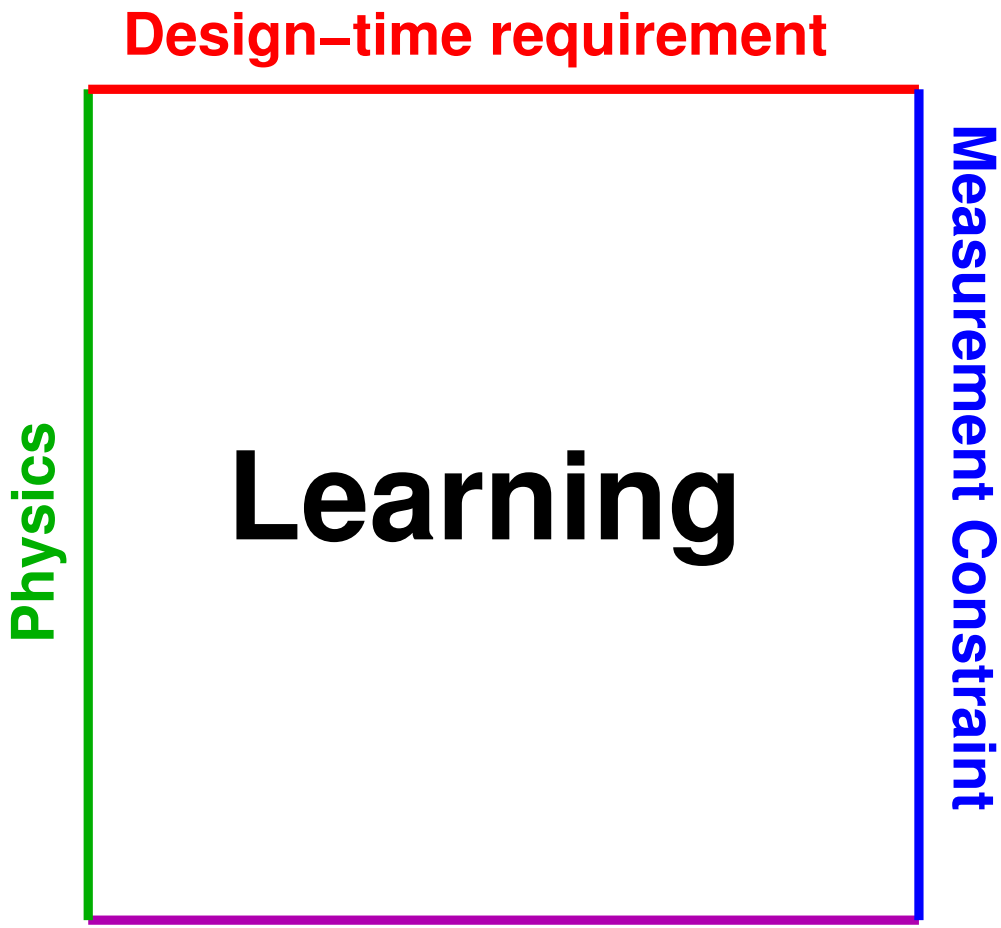
[3] Blohm, Pauline, Julius Adelt, and Paula Herber. "Safe Integration of Learning in SystemC using Timed Contracts and Model Checking." In Proceedings of the 21st ACM-IEEE International Conference on Formal Methods and Models for System Design, pp. 12-22. 2023.

Safe Learning In Six Definitions

1. Learning within **safety bounds**
2. Learning safe behaviors → learning safety requirements → **safe behavior genesis**
3. **Refining behaviors** to be more safe/conservative
4. Learning that **generates verification artifacts**
   - Learning that **passes verification tests**
5. Learning that **obeys temporal contracts**, enforced my MC or RV[3]
6. Learning of **proofs**

[3] Blohm, Pauline, Julius Adelt, and Paula Herber. "Safe Integration of Learning in SystemC using Timed Contracts and Model Checking." In Proceedings of the 21st ACM-IEEE International Conference on Formal Methods and Models for System Design, pp. 12-22. 2023.

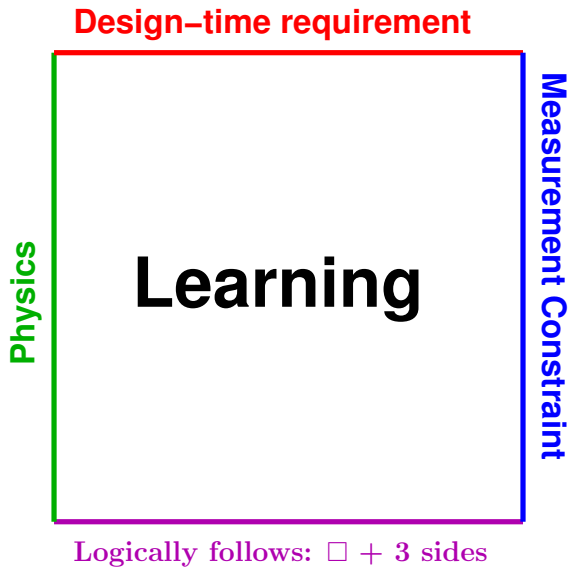
Learning in a Safety Region



Safety Bounds

- Can use logical deduction (e.g., bound by SAT/SMT)
- Can use a priori known bounds (e.g., bounded learning)

- Can we use design-time requirements? Or temporal contracts?
- Can we use technological limits?
  - what we can measure
  - computational complexity
  - what we can verify



Bottleneck: Where do we get these bounds from?

Safety Bound Extraction from Learning

Post Learning: What Safety Bounds Were Learned?

- Rule extraction for Deep Neural Networks[4]
- ML feature selection
- ML feature extraction[5]

[4] T. Hailesilassie. "Rule Extraction Algorithm for Deep Neural Networks: A Review." IJCSIS, Vol. 14, No. 7, 2016

[5] S. Khalid, T. Khalil, S. Nasreen. "A Survey Of Feature Selection And Feature Extraction Techniques In Machine Learning." Science and Information Conference, 2014

An Observation...

These bounds look a lot like sanity checks ...

Dynamic Sanity Checking: Some Challenges

Dynamic Sanity Checks:

- change with different mission modes
- accommodate re-planning
- respond to unexpected environmental conditions
- allow human interaction
  - how to explain the purpose behind findings to humans
  - how to create and monitor additional sanity checks per human request
  - how to allow humans to refine definition of safety

Challenge: What Do The Bounds Look Like?

To be useful, bounds must obey patterns...

What are the patterns?

- Measurable
- Precise
- Domain-specific (in the system domain, level of abstraction, units of the action being bounded)
- Translatable: English $\Longleftrightarrow$ System-level
- (Semi-) Automatable
- What else?

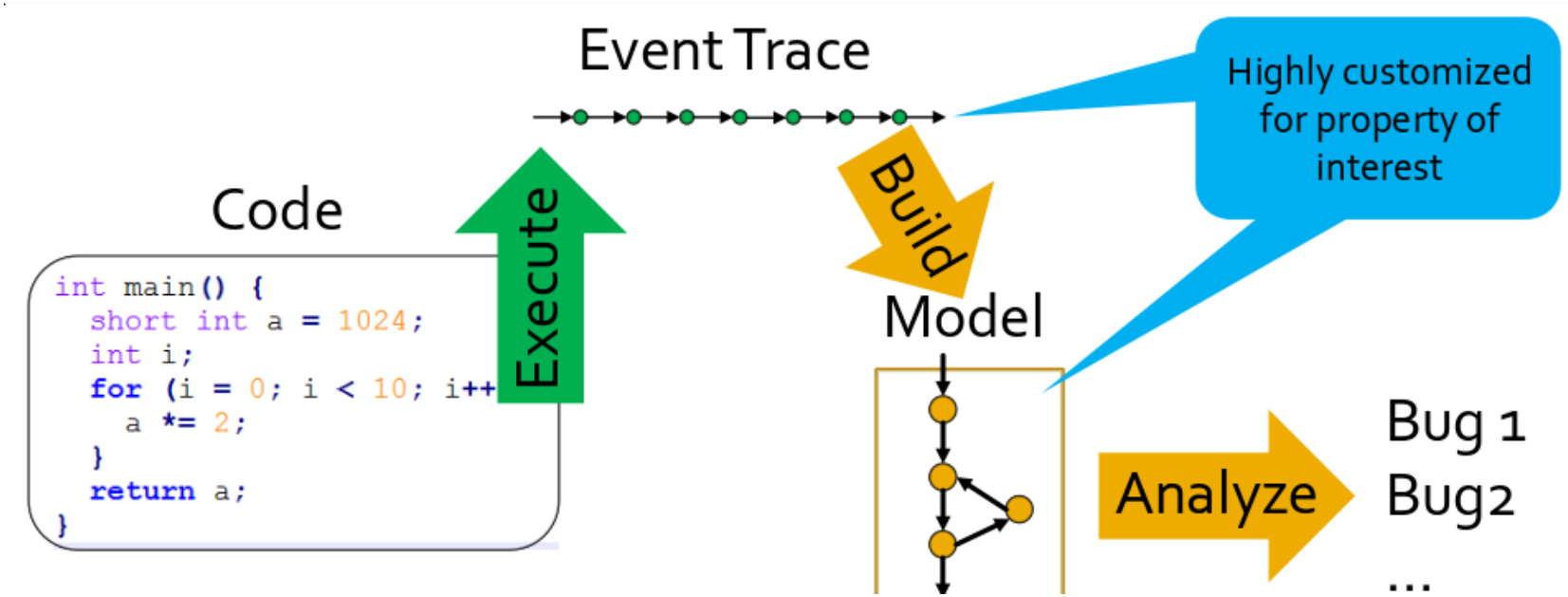
Challenge: What Do The Bounds Look Like?

To be useful, bounds must obey patterns...

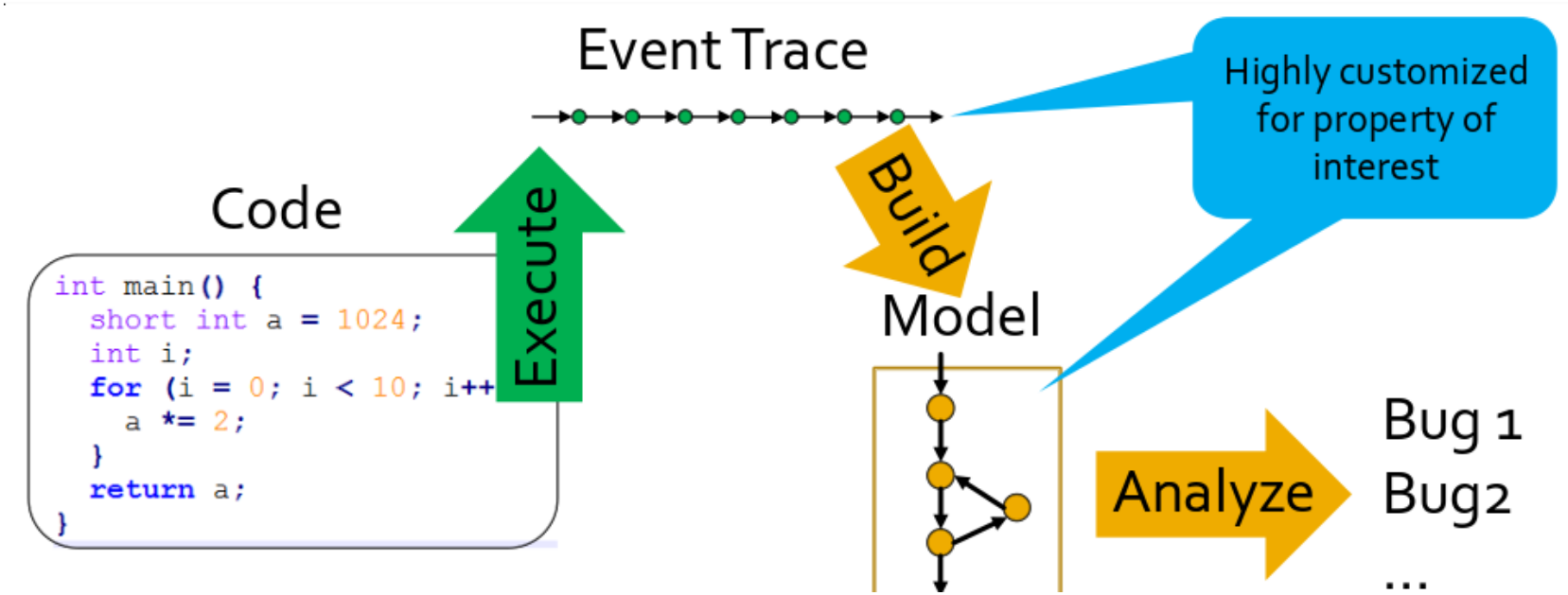
What are the patterns?

- Measurable
- Precise
- Domain-specific (in the system domain, level of abstraction, units of the action being bounded)
- Translatable: English $\Longleftrightarrow$ System-level
- (Semi-) Automatable
- What else?

Learning from Simulation or Runtime Verification?



Learning from Simulation or Runtime Verification?



6 7

[6] Grigore Rosu and Klaus Havelund, 2001, https://www.runtimeverification.com/presentations/

[7] Kristin Yvonne Rozier. "From Simulation to Runtime Verification and Back: Connecting Single-Run Verification Techniques." In Spring Simulation Conference (SpringSim'19) 2019.

Purpose

The purpose of simulation is insight [8] whereas the purpose of RV is fault detection [9].

[8] Leemis, L. M., and S. K. Park. 2006. Discrete-event simulation: A first course. Pearson Prentice Hall Upper Saddle River, NJ.

[9] Leucker, M., and C. Schallhart. 2009. "A brief account of runtime verification". The Journal of Logic and Algebraic Programming vol. 78 (5), pp. 293–303.

Purpose

The purpose of simulation **and learning?** is insight [10] whereas the purpose of RV **and learning?** is fault detection [11].

[10] Leemis, L. M., and S. K. Park. 2006. Discrete-event simulation: A first course. Pearson Prentice Hall Upper Saddle River, NJ.

[11] Leucker, M., and C. Schallhart. 2009. "A brief account of runtime verification". The Journal of Logic and Algebraic Programming vol. 78 (5), pp. 293–303.

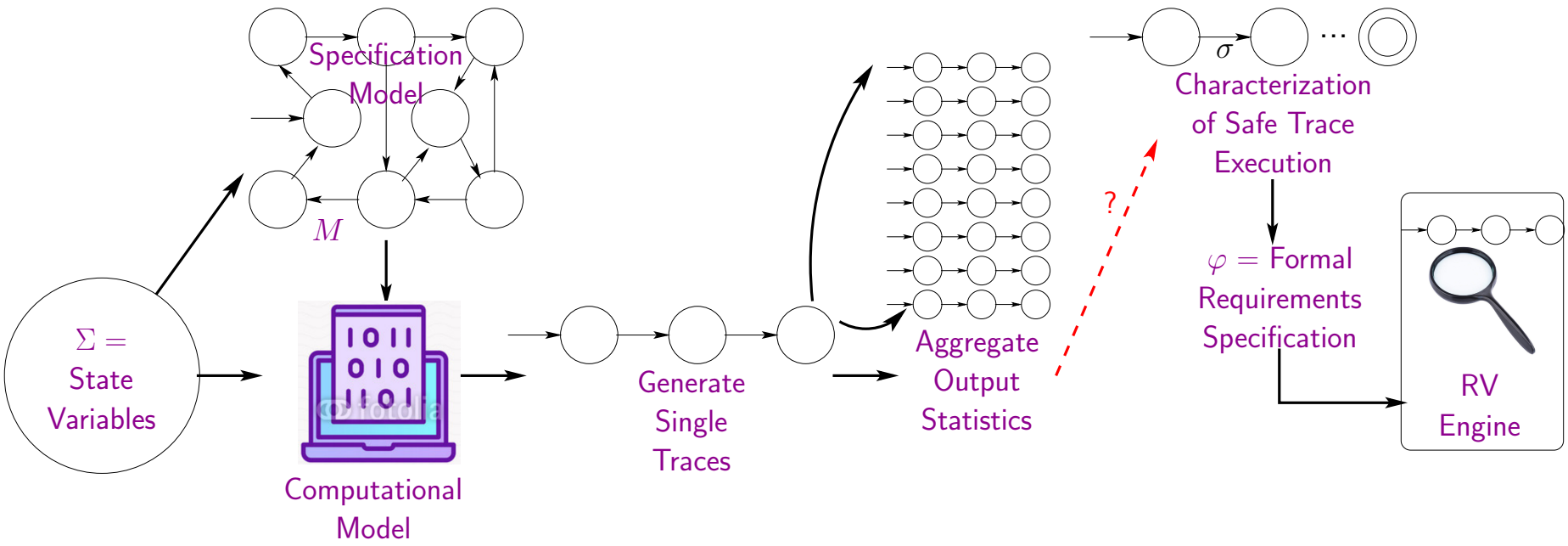
The Specification Bottleneck

Specification is the biggest bottleneck to RV.[12]

Can learning provide RV requirements?

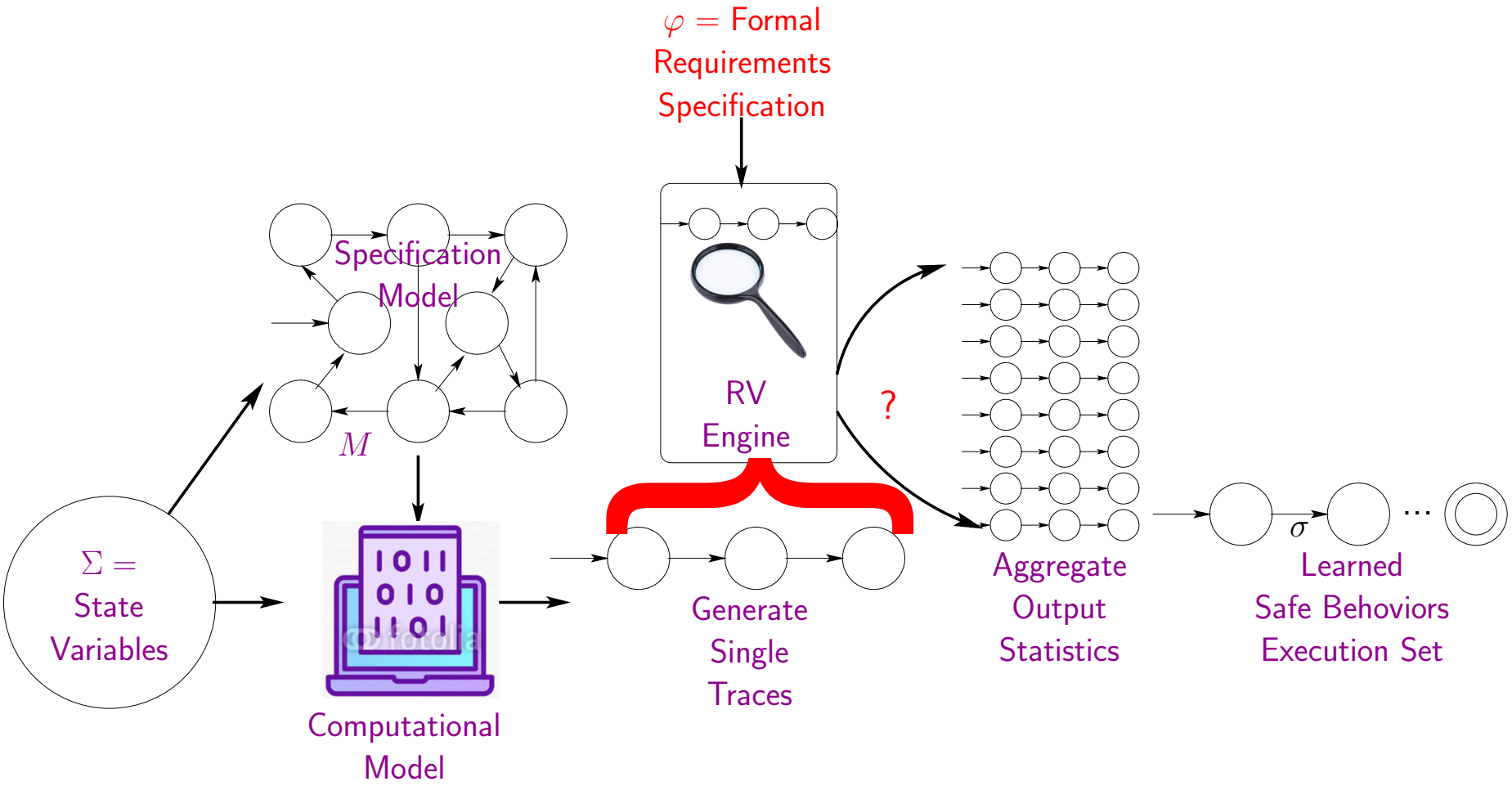
[12] Rozier, K. Y. 2016, July. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy". In Proceedings of 8th Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE 2016), Volume 9971 of LNCS, pp. 1–19. Toronto, ON, Canada, Springer-Verlag.

Learning from Simulation → RV



Kristin Yvonne Rozier. "From Simulation to Runtime Verification and Back: Connecting Single-Run Verification Techniques." In Spring Simulation Conference (SpringSim'19) 2019.

Learning from RV → Simulation



A Pathological Example of Good Bounds
. . . For Human Learning

U.S. Department
of Transportation
Federal Aviation Administration

October 12, 2017

Aeronautical

Information

Manual Official Guide to
Basic Flight Information and ATC Procedures

A Pathological Example of Good Bounds
. . . For Human Learning

U.S. Department
of Transportation
Federal Aviation Administration

October 12, 2017

766 pages

Aeronautical

Information

Manual Official Guide to
Basic Flight Information and ATC Procedures

A Pathological Example of Good Bounds
. . . For Human Learning

U.S. Department
of Transportation
Federal Aviation Administration

October 12, 2017

766 pages

Aeronautical

Information

Manual Official Guide to
Basic Flight Information and ATC Procedures

"fundamentals required in order to fly in the United States NAS [including] factors affecting flight safety"

A Pathological Example of Good Bounds
. . . For Human Learning

U.S. Department
of Transportation
Federal Aviation Administration

October 12, 2017

766 pages

Aeronautical

Information

Manual Official Guide to
Basic Flight Information and ATC Procedures

**"fundamentals required
in order to fly
in the United States NAS [including]
factors affecting flight safety"**

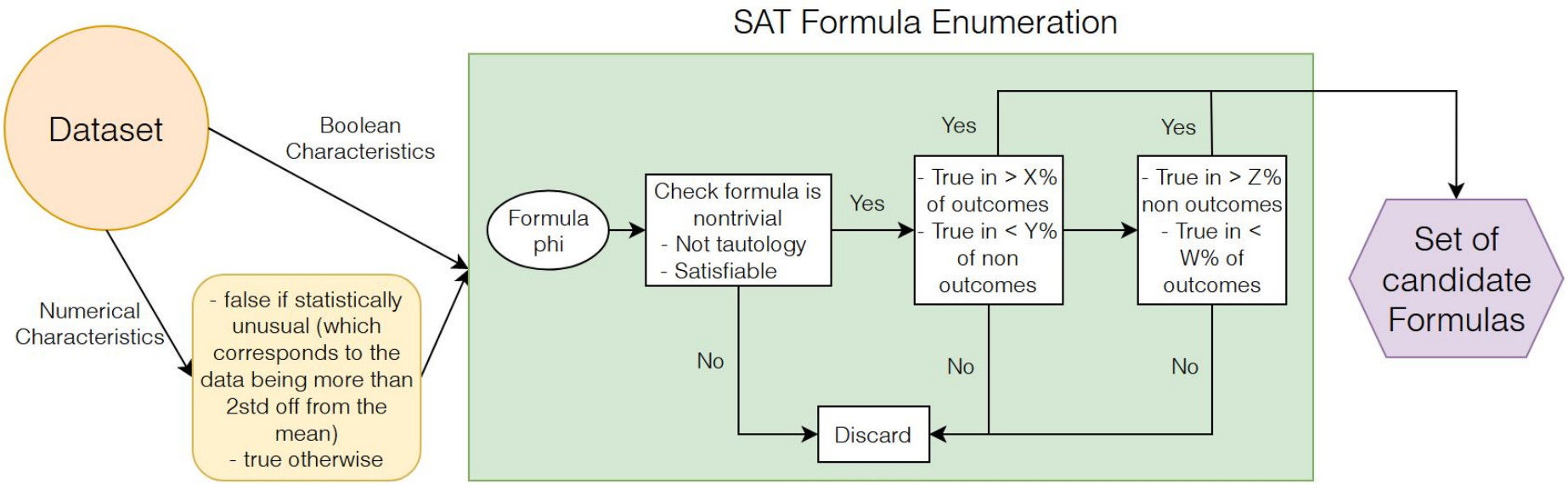
**Declarative statements
and reference definitions**

Safety-Bounding Human Learning: AIM, page 5–4–51

Temperature Limits. For aircraft using barometric vertical navigation (without temperature compensation) to conduct the approach, low and high-temperature limits are identified on the procedure. Cold temperatures reduce the glidepath angle while high temperatures increase the glidepath angle. Aircraft using baro VNAV with temperature compensation or aircraft using an alternate means for vertical guidance (e.g., SBAS) may disregard the temperature restrictions. The charted temperature limits are evaluated for the final approach segment only. Regardless of charted temperature limits or temperature compensation by the FMS, the pilot may need to manually compensate for cold temperature on minimum altitudes and the decision altitude.

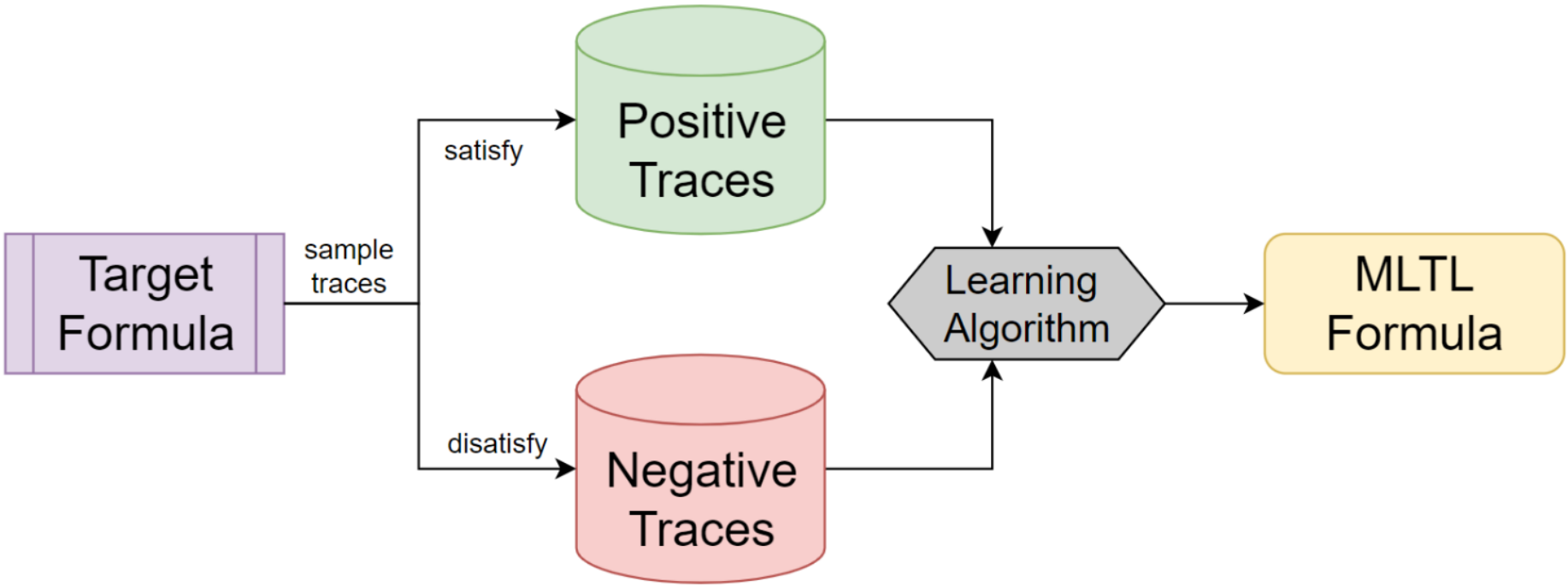
What about when data is purposely absent/corrupted?

Inferring Boolean Formulas for Forced Labor



Schumm, J., Montabon, F., Rozier, K.Y. , Wang, Z., and Burken., W. Formal Methods: A novel methodology to address complex sustainability issues with real world data constraints. Under submission.

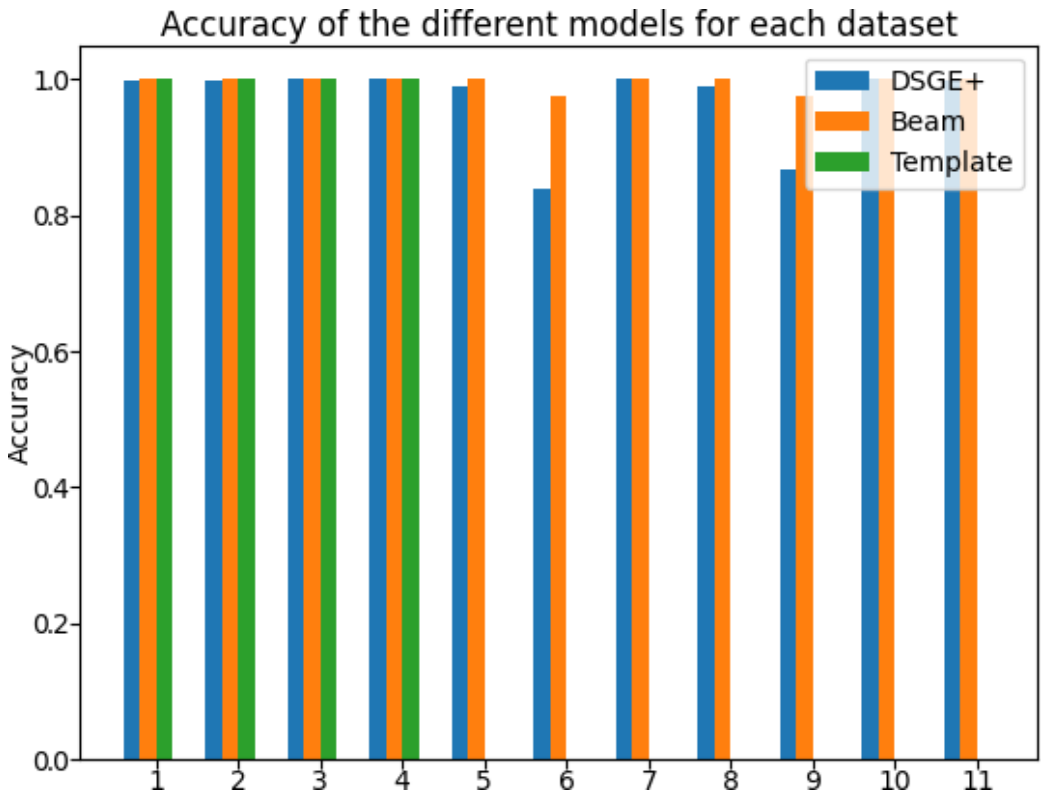
Learning MLTL From Synthetic Data



Wang, Z., Marzen, L., Swaminathan, J., Tran, Nhan. Rozier, K.Y. Learning Mission-time Linear Temporal Logic (MLTL) from Data, In preparation.

Approaches to Learning MLTL from Synthetic Data

- **Beam Search:** Limited BFS using heuristic "Best First Search"
- **Genetic Algorithm:** Grammatical Evolution (GE) search over the space of a formal MLTL grammar
- **Template-driven Search:** Search over templates of "common" formula shapes, iterative counterexample-guided refinement of formulas



Wang, Z., Marzen, L., Swaminathan, J., Tran, Nhan. Rozier, K.Y. Learning Mission-time Linear Temporal Logic (MLTL) from Data, In preparation.

Safe Learning Challenges: Verification Artifacts and Proofs

How can learning algorithms **generate verification inputs**?

Safe Learning Challenges: Verification Artifacts and Proofs

How can learning algorithms **generate verification inputs**?

Can any learning algorithms **generate verification artifacts**?

Safe Learning Challenges: Verification Artifacts and Proofs

How can learning algorithms **generate verification inputs**?

Can any learning algorithms **generate verification artifacts**?

Can they generate **explainability artifacts**?

Safe Learning Challenges: Verification Artifacts and Proofs

How can learning algorithms **generate verification inputs**?

Can any learning algorithms **generate verification artifacts**?

Can they generate **explainability artifacts**?

Can we even start to **generate proofs**?

Logic Enabling Learning

Learning driven by a **formal specification** that is **checkable**, with the **provable** result of minimizing harm to humans (through action or inaction):

1. Learning within **safety bounds**
2. Learning safe behaviors → learning safety requirements → **safe behavior genesis**
3. **Refining behaviors** to be more safe/conservative
4. Learning that **generates verification artifacts**
   - Learning that **passes verification tests**
5. Learning that **obeys temporal contracts**
6. Learning of **proofs**

BACKUP SLIDES

NASA Autonomy Operating System: Landing Order

NASA Autonomy Operating System: Landing Order

- Use SMT as an approximation for common sense
- Need to explain back to ATC:
  "I believe I'm n^{th} in line for landing" from SMT query

Research problem: how to translate NLP to SMT and back?