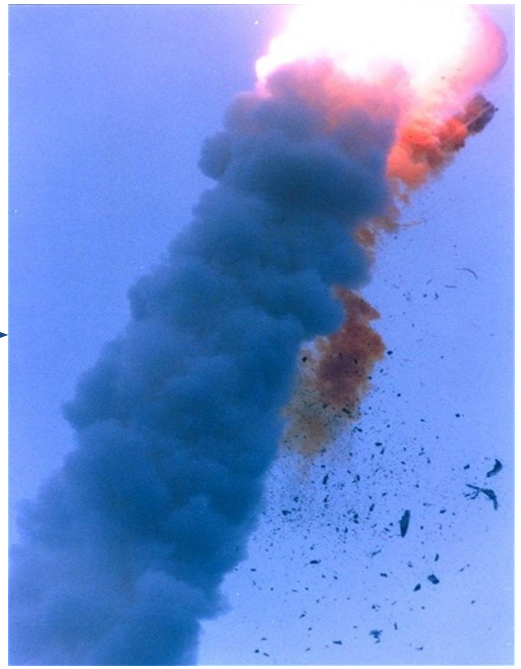
Deductive Verification vs. Model Checking for Physical Safety of a Feedback-Controlled Drone

Eric Yang and Can Afacan

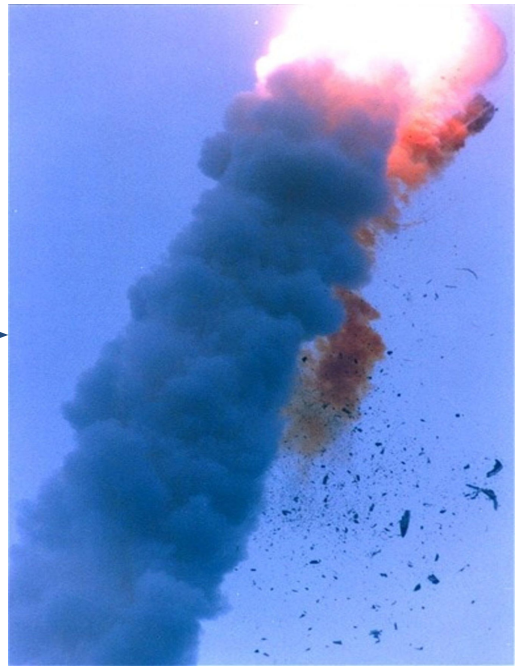
Northwestern

Ariane 5 Flight 501



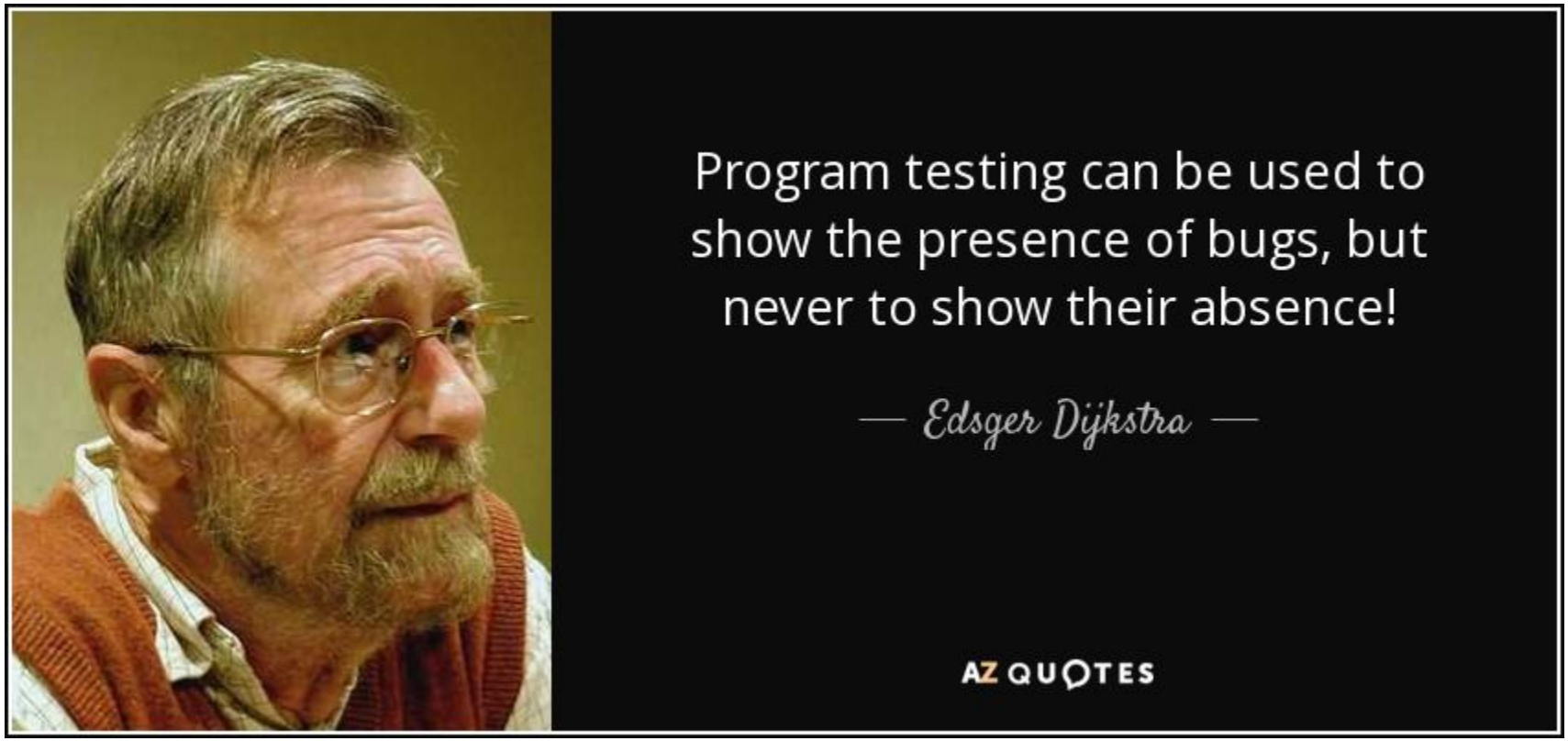
One missed test + physical system = catastrophic failure

Ariane 5 Flight 501

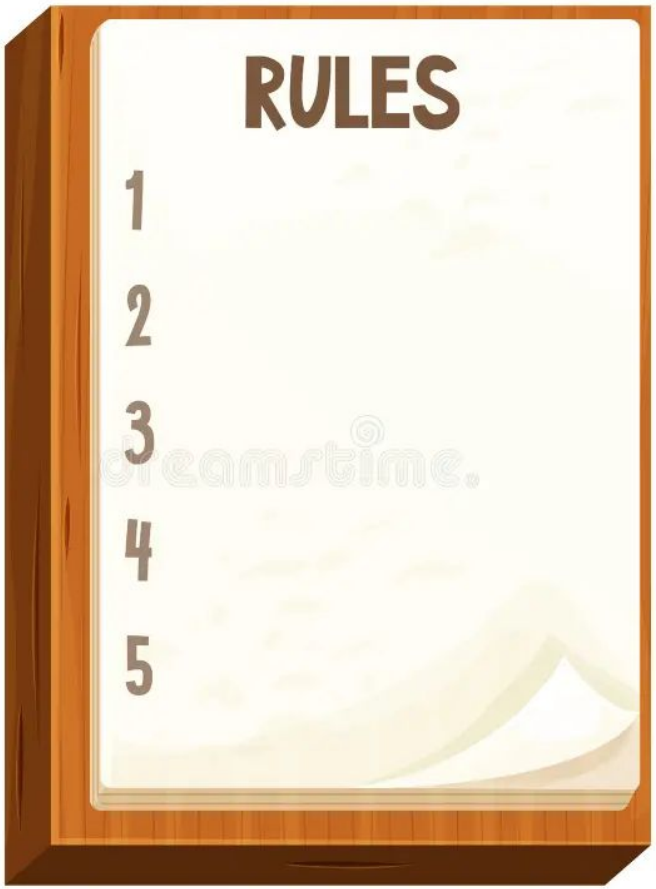


So, how do we prove safety before deployment?

Introducing Formal Verification

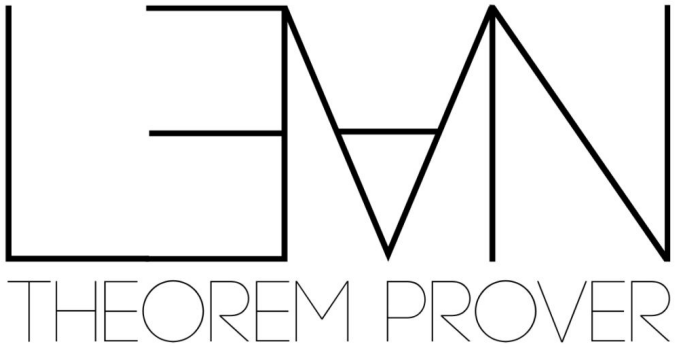


Specifications



Verification

Deductive Prover vs Model Checker

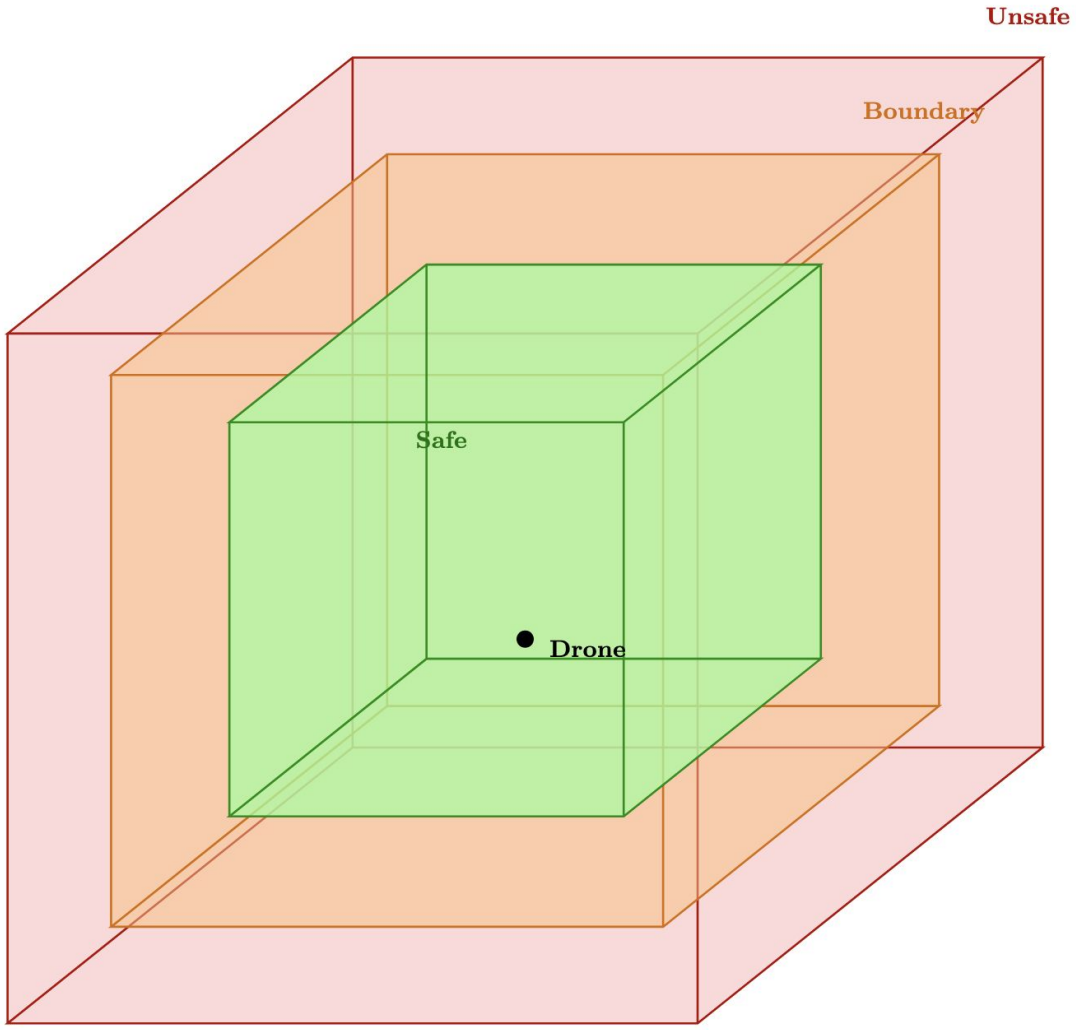


Deductive Prover vs Model Checker

LEAN
THEOREM PROVER

Experiment: Simulation in 3D Space

Simulation Setup



Specifications

P1. The drone always remains in the green+orange region:

$\square(q \in S)$

Specifications

P1. The drone always remains in the green+orange region:

$\square(q \in S)$

P2. The drone never exceeds the prescribed speed limit:

$\square(\|v\| \leq v\_max)$

Specifications

P1. The drone always remains in the green+orange region:

$\square(q \in S)$

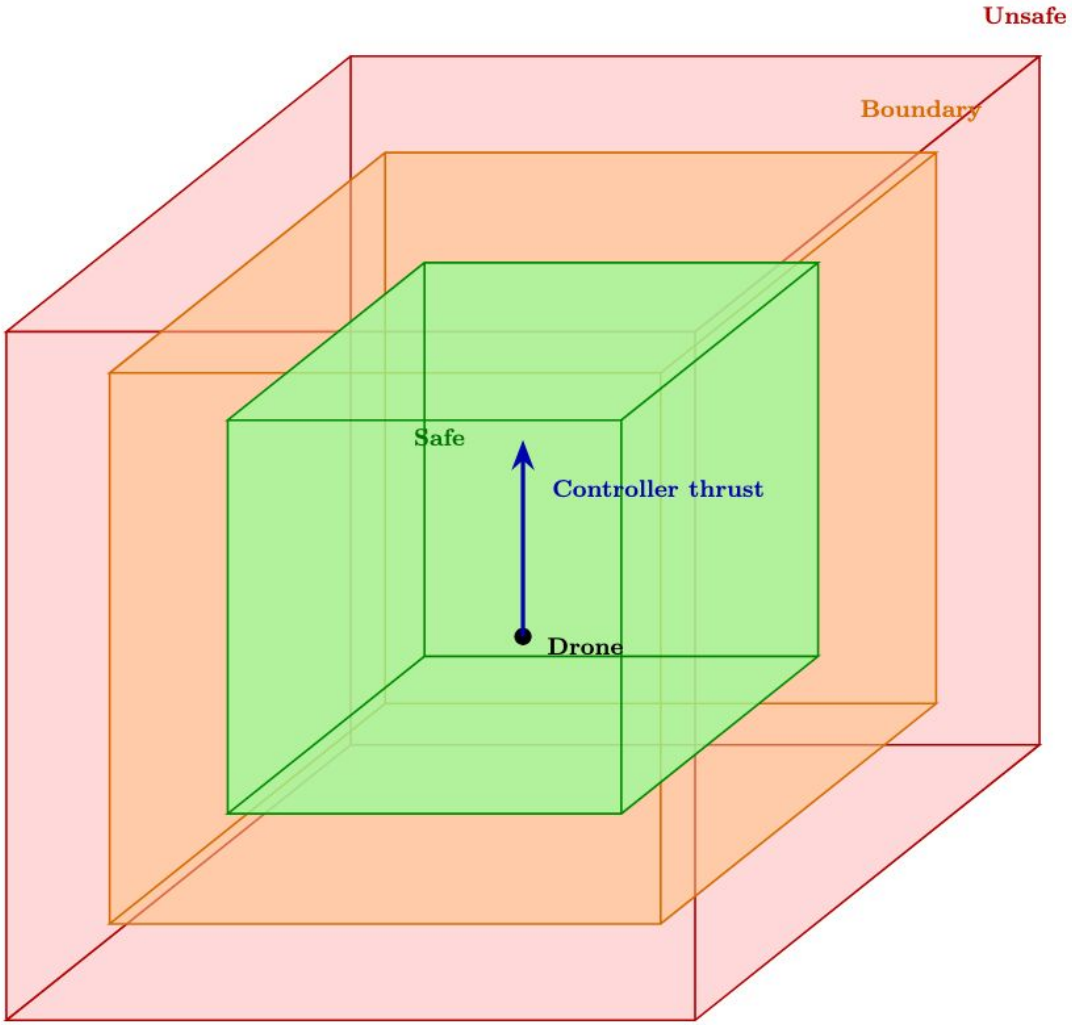
P2. The drone never exceeds the prescribed speed limit:

$\square(\|v\| \leq v\_max)$

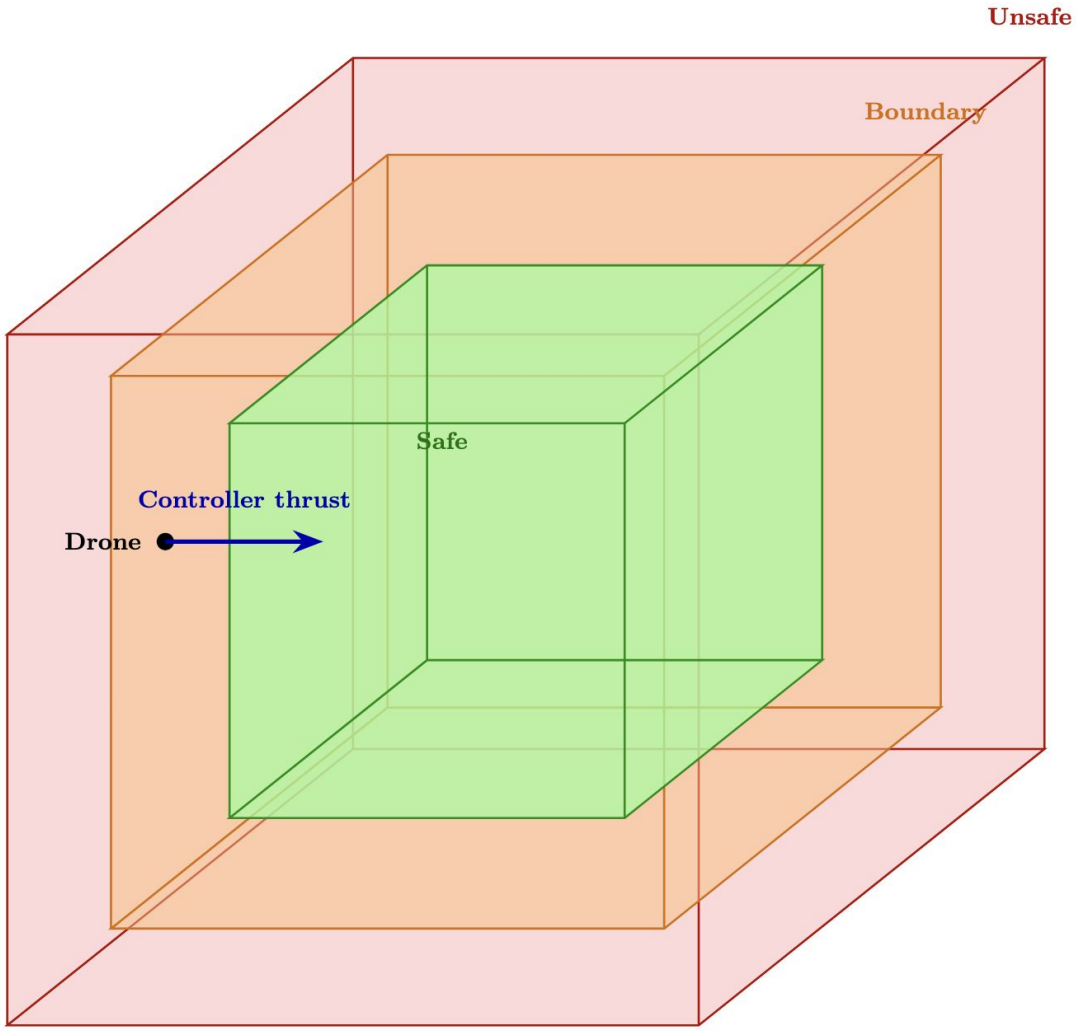
P3. Whenever the drone enters the orange region, it eventually returns to the green region:

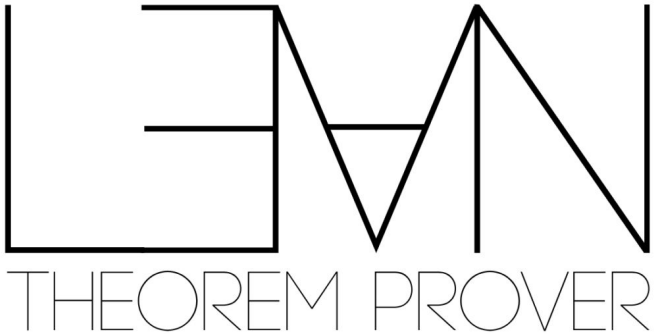
$\square(q \in B \rightarrow \Diamond(q \in S\_in))$

Controller Design



Controller Design





Running Formal Verification

What are the tradeoffs for each method of formal verification in this experiment?

Quantitative Comparison

Method	Lines of Code	Verification Evidence	Runtime
TLA+/ TLC	144	6.89 million reachable states	75 s
nuSMV	166	19.74 billion symbolic states	14.5 s
Lean 4	497	3 specification theorems proved	Instant

Tradeoffs for Physical Safety

TLA & nuSMV:

- Fast and compact
- Less expressive (Finite or Discrete State Systems)
- Less effort

Tradeoffs for Physical Safety

TLA & nuSMV:

- Fast and compact
- Less expressive (Finite or Discrete State Systems)
- Less effort

Lean:

- More involved and time consuming
- More scalable
- More expressive

Tradeoffs for Physical Safety

TLA & nuSMV:

- Fast and compact
- Less expressive (Finite or Discrete State Systems)
- Less effort

Lean:

- More involved and time consuming
- More scalable
- More expressive

```
DroneLean/
├── Controller.lean
├── Dynamics.lean
├── Invariants.lean
├── Properties.lean
├── SafetyProofs.lean
├── Types.lean
└── VelocityBound.lean
```

Conclusions

- We modeled a controller-operated drone with safety specifications.
- Use different verification paradigm to verify the same safety goals.
- TLA+/TLC and nuSMV exhaustive checked the possible states of the model.
- Lean 4 proved the safety result mathematically.

Conclusions

- We modeled a controller-operated drone with safety specifications.
- Use different verification paradigm to verify the same safety goals.
- TLA+/TLC and nuSMV exhaustive checked the possible states of the model.
- Lean 4 proved the safety result mathematically.
- Main takeaway: **model checkers are faster for exploring behavior; Lean scales better, gives stronger mathematical assurance but costs more proof effort.**

Future Directions

- Add richer physics (e.g. Wind, Disturbances)

Future Directions

- Add richer physics (e.g. Wind, Disturbances)
- Add more complex controller logic

Future Directions

- Add richer physics (e.g. Wind, Disturbances)
- Add more complex controller logic
- Multi-agent system

Thank you!