TLA$^+$ for All: Model Checking in a Python Notebook

ETAPS 2025 TLA$^+$ Community Event

Konstantin Läufer *(presenter) and* George K. Thiruvathukal
Professors of Computer Science
Loyola University Chicago (USA)

Hello and welcome! Sali und willkumme zäme! ¡Hola y bienvenidos/as! Ciao e benvenuti/e!

- klaufer@luc.edu | laufer.cs.luc.edu | github.com/klaeufer
- At Loyola University Chicago since 1992
- Research and teaching in
  - (higher-order typed) programming languages
  - software frameworks and architecture
  - high-performance computing
  - formal methods, especially model checking
- Some industry consulting (Lucent/Bell Labs)

Context: education project goals

- Qualitatively *assess* the state of formal methods in computer science programs
- Construct level-appropriate *examples* that could be included midway into one's undergraduate studies
- Demonstrate how to address successive "failures" through progressively stringent *safety and liveness requirements*
- Establish an ongoing framework for assessing interest and relevance among students
- Develop *reusable curricular materials* linking discrete structures, formal methods, and adjacent courses
- ***Broaden adoption of formal methods in academia*** *(especially TLA$^+$)*

Motivation: formal methods and model checking

- FM around since the 60s starting with Hoare logic
- Model checking around since the early 80s for concurrent program verification, now well established
- Widely used in industry and government agencies, e.g.
  - NASA
  - Amazon Web Services (TLA$^+$)
- NSF CISE CCF/CNS Formal Methods in the Field (FMitF)

Adoption of FM and MC in education?

- Academia has been responding to the need for talent
- FM typically offered as more advanced courses
- MC offered in about 40% of courses
- Various tools used, including Leslie Lamport's TLA$^+$
- Formal Methods Education Database (FMEDB) fme-teaching.github.io/courses

Key challenges to teaching FM more broadly

- Various studies (referenced in our 2024 IEEE FIE and IEEE Computer papers)
- Insufficient mathematical background in students
- Lack of engaging case studies and tool documentation
- Misalignment with modern student learning styles: discovery-driven, solution-focused

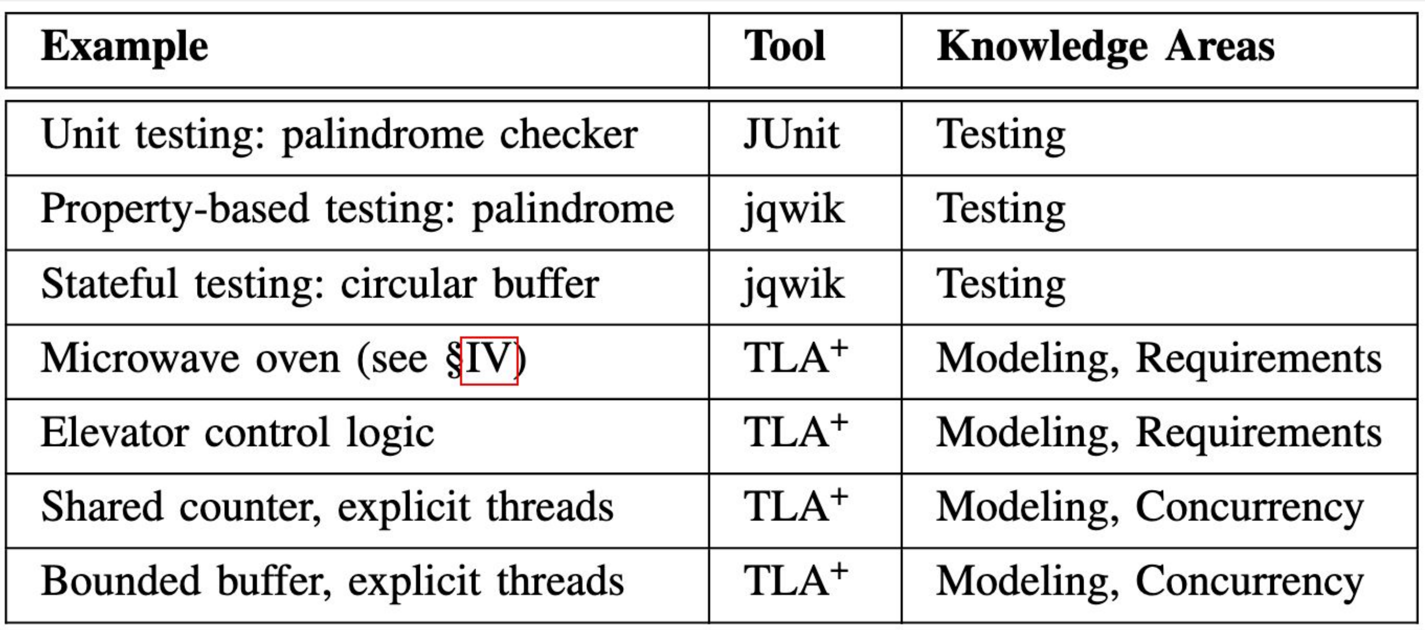
Institutional context: Loyola University Chicago

- Private, urban, mid-size, R1 as of 3/2025
- Students: 600+ undergrad majors, 120 MS, 9 PhD
    - Many from underrepresented groups
    - Quite a few first-generation college-goers
- Faculty: 13 TT, 4 NTT, around 10 PT
- Research-active, support from NSF, NSA, NIH, industry

Curricular context

- Five-course, three-semester foundation sequence
  - Intro to Programming (CS1)
  - Data Structures I (CS2) (linear data structures)
  - Intro to Computer Systems (CS3)
  - ***Discrete Structures***
  - Intro to Linux Command Line

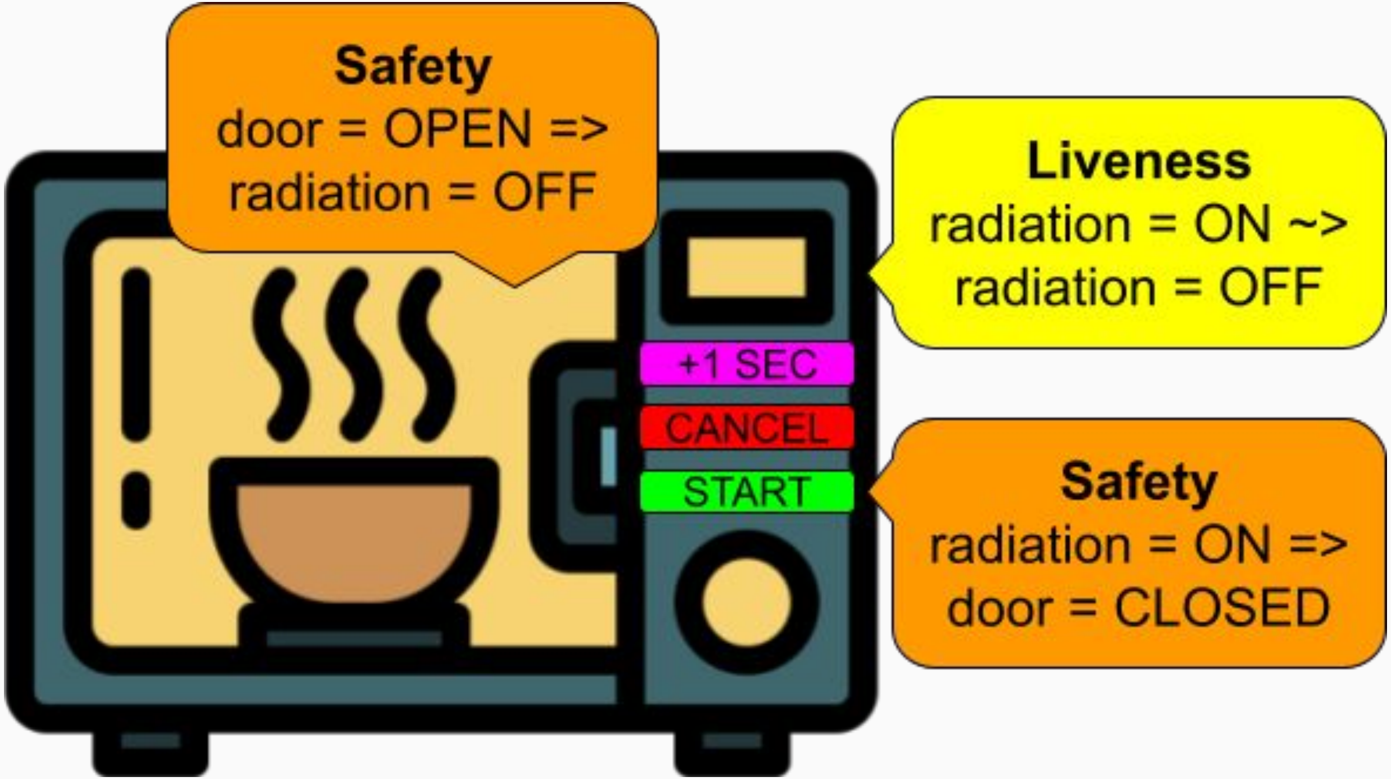
Progression of course examples (with corresponding ACM/IEEE knowledge areas)



Example	Tool	Knowledge Areas
Unit testing: palindrome checker	JUnit	Testing
Property-based testing: palindrome	jqwik	Testing
Stateful testing: circular buffer	jqwik	Testing
Microwave oven (see §IV)	TLA^+	Modeling, Requirements
Elevator control logic	TLA^+	Modeling, Requirements
Shared counter, explicit threads	TLA^+	Modeling, Concurrency
Bounded buffer, explicit threads	TLA^+	Modeling, Concurrency

← recently inserted content module on Alloy (relational logic)

Example: microwave oven — overview



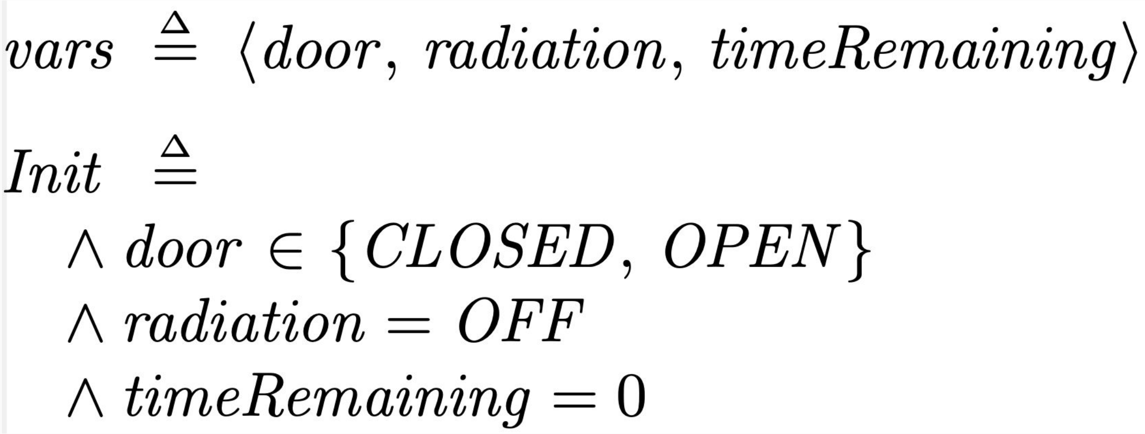
Safety
door = OPEN =>
radiation = OFF

Liveness
radiation = ON ~>
radiation = OFF

Safety
radiation = ON =>
door = CLOSED

Example: microwave oven in TLA$^+$

State: (mathematical) variables and their initial values



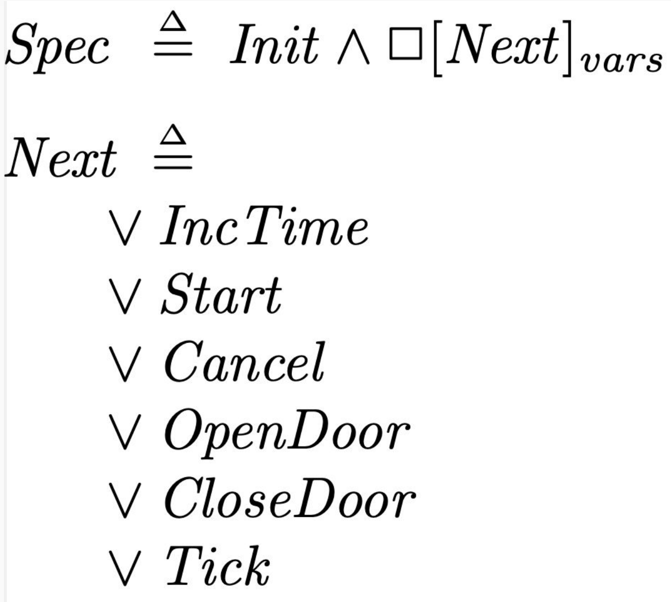
$$vars \triangleq \langle door,\ radiation,\ timeRemaining \rangle$$

$$\begin{aligned}
Init \triangleq\ & \\
& \land\ door \in \{CLOSED,\ OPEN\} \\
& \land\ radiation = OFF \\
& \land\ timeRemaining = 0
\end{aligned}$$

Example: microwave oven in TLA$^+$ (abridged)

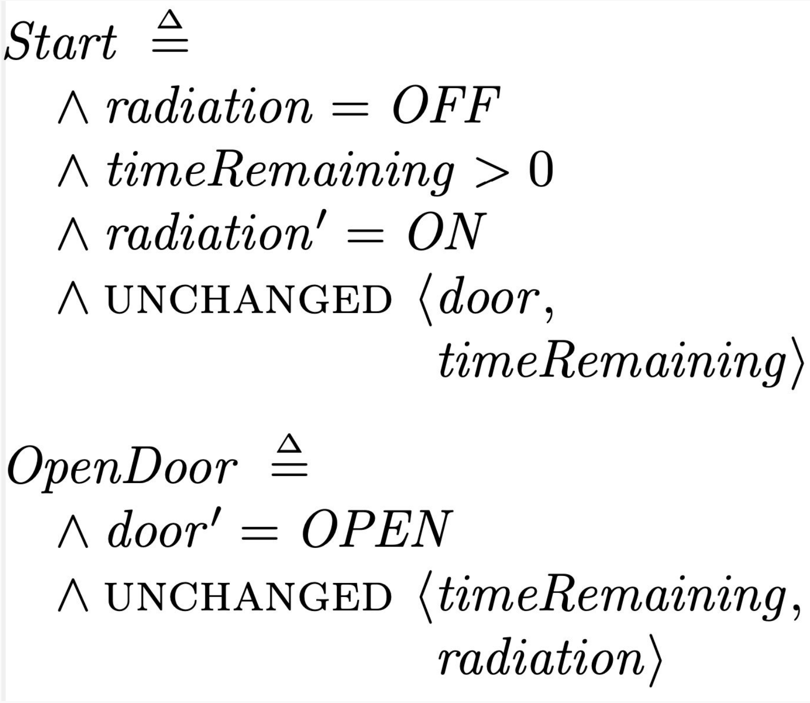
Behavior: Next action models events as state transitions

▢ (box) temporal operator means “always”



$$Spec \triangleq Init \land \Box[Next]_{vars}$$

$$\begin{aligned} Next \triangleq\ & \lor\ IncTime \\ & \lor\ Start \\ & \lor\ Cancel \\ & \lor\ OpenDoor \\ & \lor\ CloseDoor \\ & \lor\ Tick \end{aligned}$$



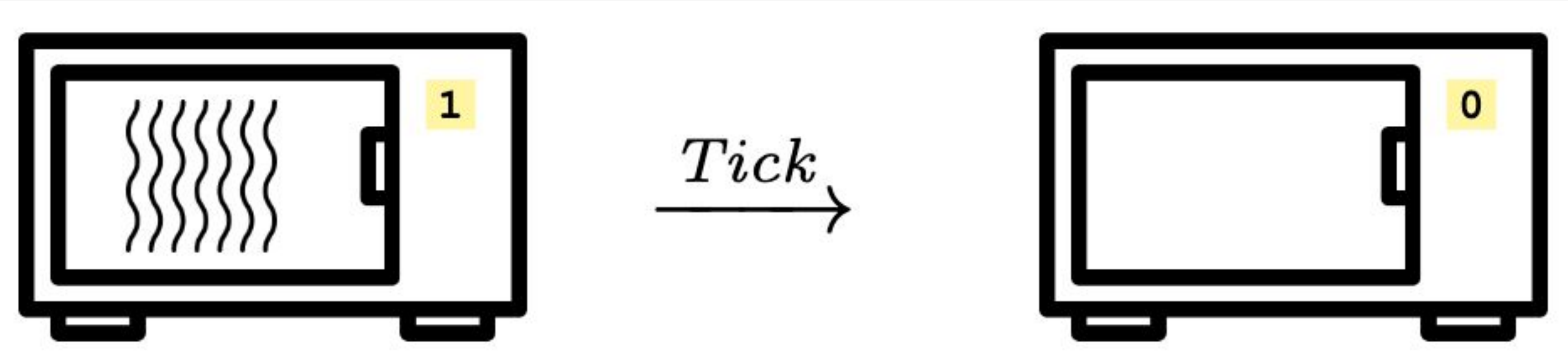
$$\begin{aligned} Start \triangleq\ & \land\ radiation = OFF \\ & \land\ timeRemaining > 0 \\ & \land\ radiation' = ON \\ & \land\ \text{UNCHANGED}\ \langle door, \\ & \qquad\qquad timeRemaining \rangle \end{aligned}$$

$$\begin{aligned} OpenDoor \triangleq\ & \land\ door' = OPEN \\ & \land\ \text{UNCHANGED}\ \langle timeRemaining, \\ & \qquad\qquad radiation \rangle \end{aligned}$$

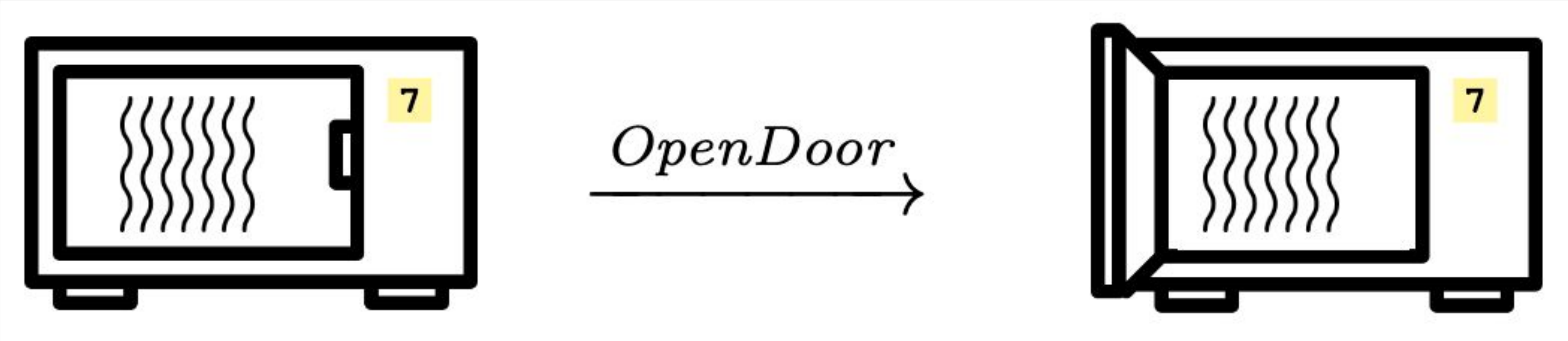
So what does the model checker (TLC) do?

- Valid states are those reachable by the spec:
  - Start in initial state (Init)
  - Perform zero or more steps (Next) according to □
- The model checker computes a finite subset of the valid state space
- It reports violations of invariants and temporal properties
- Handles very large state spaces
- *Unbiased, finds violations we might not think about*

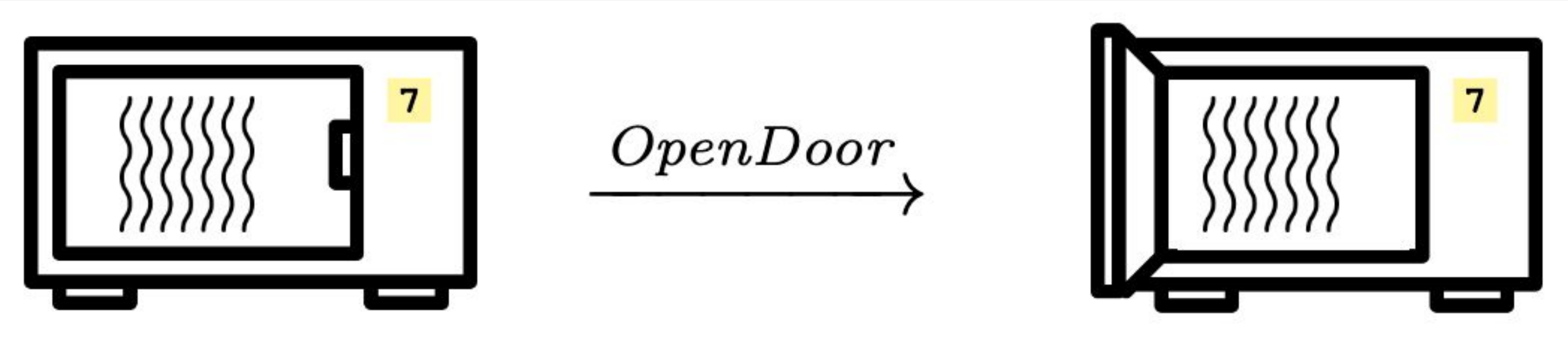
Example: one second in the life of a microwave oven
Scenario 1: Normal operation



Example: microwave oven
Scenario 2



Example: microwave oven
Scenario 2



Invariant DoorSafety is violated.

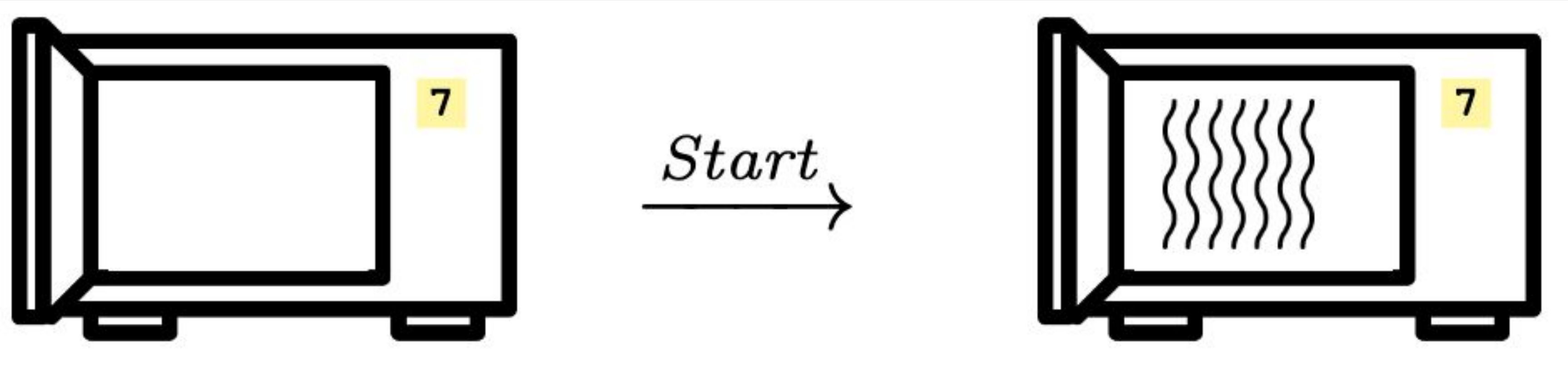
…

/\ door = OPEN

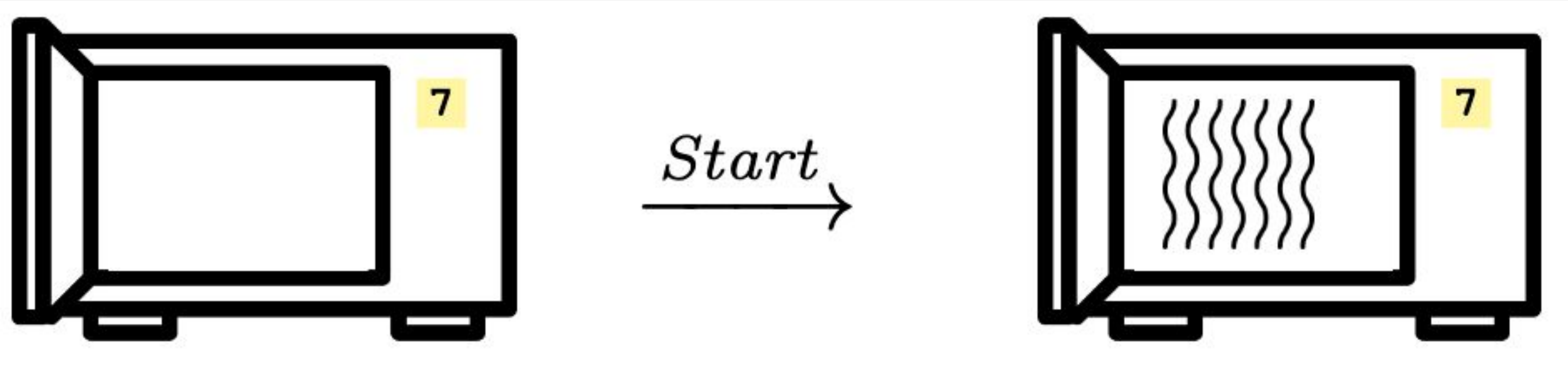
/\ timeRemaining = 1

/\ radiation = ON

Example: microwave oven
Scenario 3



Example: microwave oven
Scenario 3



Invariant DoorSafety is violated.

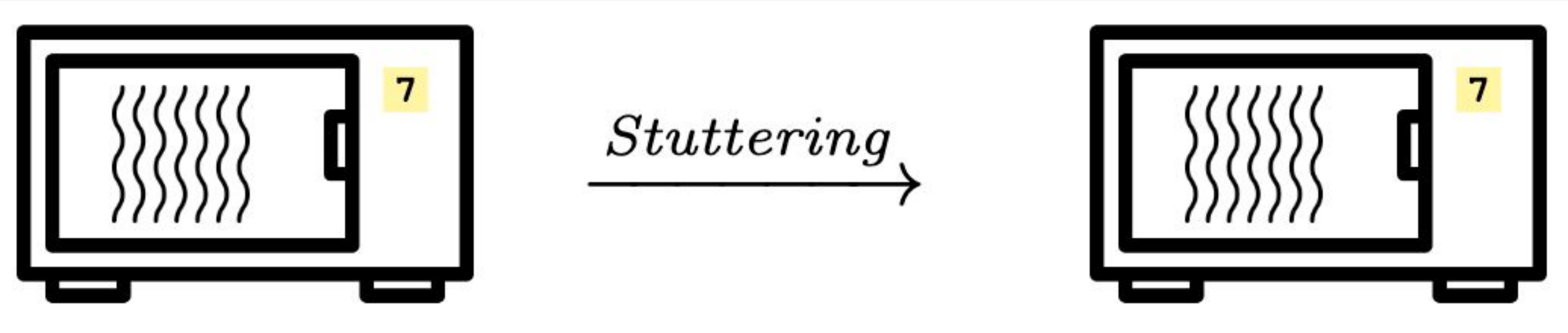
...

/\ door = OPEN

/\ timeRemaining = 1

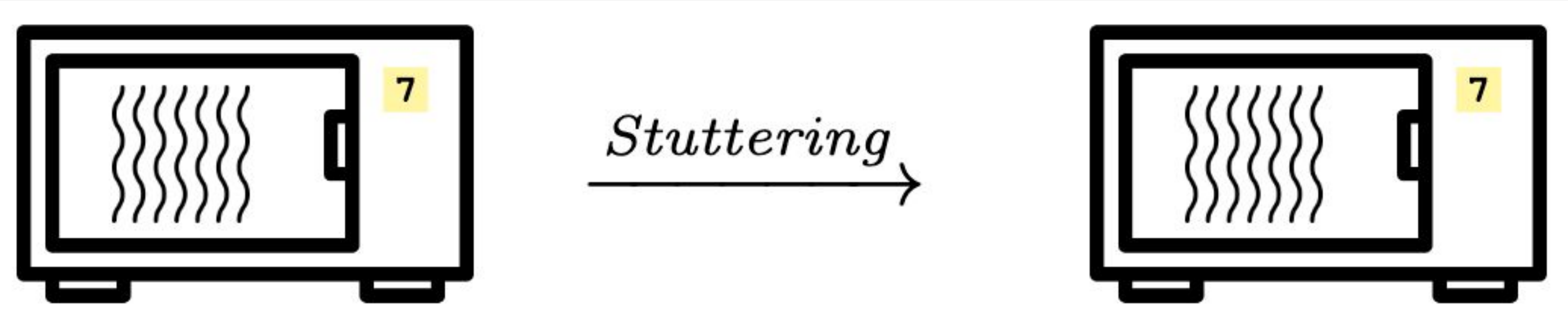
/\ radiation = ON

Example: microwave oven
Scenario 4



Does this violate anything?

Example: microwave oven
Scenario 4



HeatLiveness ==

□ (radiation = ON ~> radiation = OFF)

Other fun examples

- Puzzles
  - Die Hard water jug challenge
  - River crossing: farmer, wolf, goat, cabbage
  - River crossing with flashlight
- Simulations
  - Vending machine
  - Elevator
- Classic concurrency examples
  - Concurrent increment of a variable
- *Lots of examples* at homes.cs.aau.dk/~kgl/esv04/exercises

How to run the examples in practice?

- Specific IDE and CLI: lamport.azurewebsites.net/tla/toolbox.html
- Standard IDE: Visual Studio Code with TLA$^+$ extension
- Browser-based standard IDE (quasi-zero install): gitpod.io/#https://github.com/lucformalmethodscourse/microwave-tla
- ***Notebook (zero install)*** – see next slide

How to run the examples in practice? Notebook

- Motivation
    - Experiment with TLA+ models instantly
- Benefits
    - Student engagement
    - Easy to share and reproduce
    - Ease of maintenance

How to run the examples in practice? Notebook

- Still needed
  - pretty-printing using tlatex
  - state graph viz using Graphviz
  - Broader tool integration, e.g. Alloy
  - …

Preliminary evaluation results

- Fall 2022: 15 students, standardized course evaluation
- Spring 2024: 22 students, standardized + course-specific 15-question survey given as pretest-posttest
  - Marked improvement in student receptiveness to FM
  - Enhanced ability to conceptualize and apply
  - Deeper understanding of software correctness
  - Composite scores increased from 3.0 to 4.1 (n = 19)
  - 84% reported a positive experience
- Spring 2025: still ongoing – including effectiveness of notebook

Conclusion

- Successful integration of model checking using TLA$^+$ into our intermediate undergraduate curriculum
- Five courses/three semesters of prerequisites
- Grounded in ACM/IEEE Computing Curricula
- Open source/access/participation curricular material (including code) at lucformalmethodscourse.github.io

Future plans

- Continue offering the Formal Methods course every spring semester, continually updating and evaluating it
    - *Added module on Alloy (relational logic)*
- *Integrate more closely with introductory Discrete Structures course*
    - ***Add introductory modules on using automated proof checker (LEAN)***
- Disseminate *reusable curricular materials* linking discrete structures, formal methods, and *adjacent courses*
- Continue related AI4SE/AI4FM research on LLM-enabled automated test and model synthesis

Future plans

- Obtain feedback at the TLA$^+$ event — ***klaufer@luc.edu***
  - Compare notes with industry
- Network with other universities, starting regionally
- Consider offering workforce development workshop
- Continue contributing to Formal Methods Education Database (FMEDB) — fme-teaching.github.io

Thank you! Questions?

klaufer@luc.edu

doi.org/10.6084/m9.figshare.27122916.v1