

# An Animation Module for TLA+

Will Schultz

What are TLA+ specifications useful for?

# Utility of TLA+ Specifications

1. **Communication:** codification and sharing
2. **Learning:** precise, abstract thinking
3. **Verification:** TLC, TLAPS

# Utility of TLA+ Specifications

1. **Communication**: codification and sharing
2. **Learning**: precise, abstract thinking
3. **Verification**: TLC, TLAPS

# Current State of TLA+

## **Communication Process**

Write a spec, have someone else read it.

Explore a model in the TLA+ Toolbox

## **Learning Process**

Think, Write, Verify, Iterate

# Visualization and TLA+

Visualization as a way to:

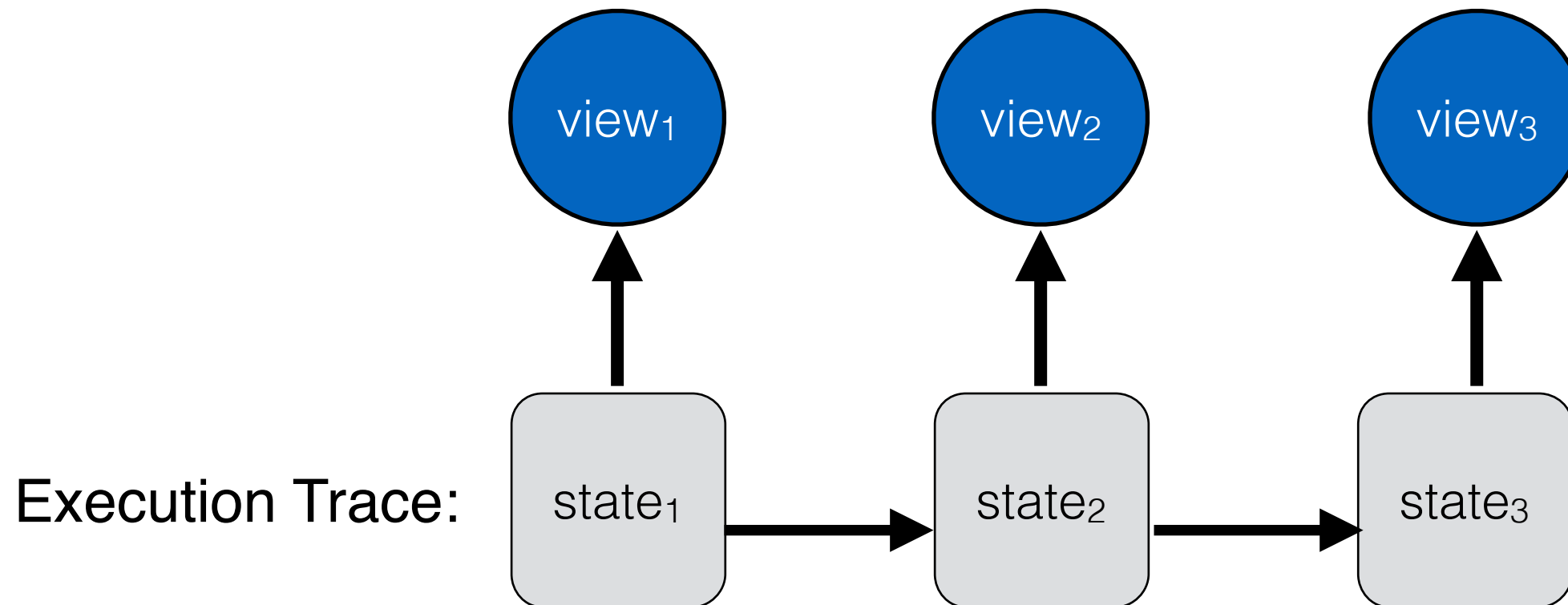
1. Improve the **communication** of TLA+ specs.
2. Enhance the TLA+ **learning** process.

# TLA+ Animation Module

- Visualize execution traces of a TLA+ spec & model using TLC

# TLA+ Animation Module

- Visualize execution traces of a TLA+ spec & model using TLC





# TLA+ Animation Module

- Module defines operators to describe and lay out graphical primitives (SVG) in TLA+

$$\begin{aligned} \text{Circle}(cx, cy, r, attrs) &\triangleq \\ &\text{LET } svgAttrs \triangleq [cx \mapsto \_str(cx), \\ &\quad \quad \quad cy \mapsto \_str(cy), \\ &\quad \quad \quad r \mapsto \_str(r)] \text{ IN} \\ &\text{SVGElem}(\text{"circle"}, \text{Merge}(svgAttrs, attrs), \langle \rangle) \end{aligned}$$

# TLA+ Animation Module

- Module defines operators to describe and lay out graphical primitives (SVG) in TLA+

$Group(children, attrs) \triangleq SVGElem("g", attrs, children)$

# TLA+ Animation Module

- **View expression:** TLA+ state expression that visualizes a single state by composing these graphical primitives



EXTENDS *Animation*

Represent the values of two numbers, 'x' and 'y', as a simple bar chart.

VARIABLE *x, y**vars*  $\triangleq$   $\langle x, y \rangle$ *Init*  $\triangleq$  $\wedge x = 0$  $\wedge y = 0$ *Next*  $\triangleq$  $\vee x' = x + 1 \wedge \text{UNCHANGED } y$  $\vee y' = y + 1 \wedge \text{UNCHANGED } x$ Basic  
Spec

Define an animation view.

*barHeight*  $\triangleq$  5*widthFactor*  $\triangleq$  10*View*  $\triangleq$ LET *xBar*  $\triangleq$  *Rect*(0, 10, *widthFactor* \* *x*, *barHeight*, ("fill" :> "blue"))*yBar*  $\triangleq$  *Rect*(0, 20, *widthFactor* \* *y*, *barHeight*, ("fill" :> "red"))IN*Group*( $\langle xBar, yBar \rangle$ ,  $\langle \rangle$ )

View

*AnimSpec*  $\triangleq$  $\wedge \text{AnimatedInit}(\text{Init}, \text{View})$  $\wedge \square[\text{AnimatedNext}(\text{Next}, \text{View}, \text{TRUE})]_{\langle vars, \text{AnimationVars} \rangle}$ Animated  
Spec

Define an animation view.

View

$barHeight \triangleq 5$

$widthFactor \triangleq 10$

$View \triangleq$

LET  $xBar \triangleq Rect(0, 10, widthFactor * x, barHeight, ("fill" :> "blue"))$   
 $yBar \triangleq Rect(0, 20, widthFactor * y, barHeight, ("fill" :> "red"))$  IN  
 $Group(\langle xBar, yBar \rangle, \langle \rangle)$

$AnimSpec \triangleq$

Animated Spec

$\wedge AnimatedInit(Init, View)$

$\wedge \square [AnimatedNext(Next, View, TRUE)] \langle vars, AnimationVars \rangle$

# TLA+ Animation Module

## Module internals:

- Defines a set of auxiliary variables to track animation related state.
- Records a sequence of frames, where each frame is a visualization of one state of a behavior
- When trace generation finishes, final state will contain an SVG string of the animation

Demos

# Concluding Thoughts

1. ProB Animator and Runway visualization tools similar, but not TLA+ native
2. Better performance, closer integration with TLC
3. Fully interactive trace exploration
4. Module & examples available at [https://github.com/will62794/tlaplus\\_animation](https://github.com/will62794/tlaplus_animation)