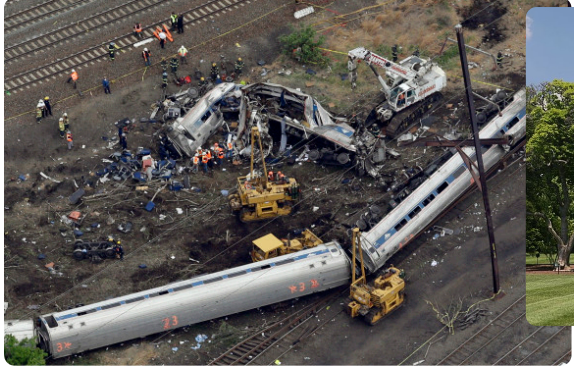
Towards Foundational Verification of Cyber-physical Systems

Gregory Malecha Daniel Ricketts
Mario M. Alvarez Sorin Lerner

SoSCYPS 2016

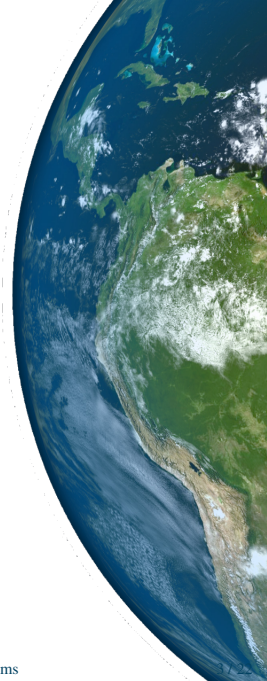



JOHNSON, STEVEN
117
77
87.9
42.2
22.3
86
58
108
88
97
95

System
(Hw+C)

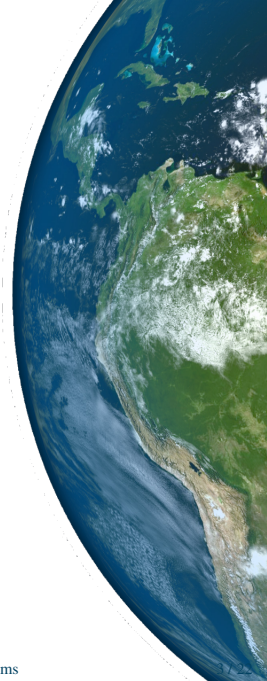
System
(Hw+C)

Specification
(Safety)



System
(Hw+C)

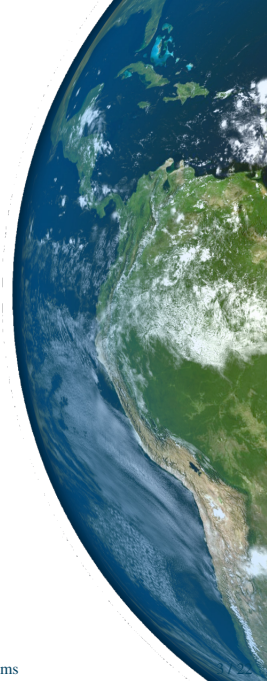
Verifier
(KeYmaera)



System
(Hw+C)

Model
(∂DL)

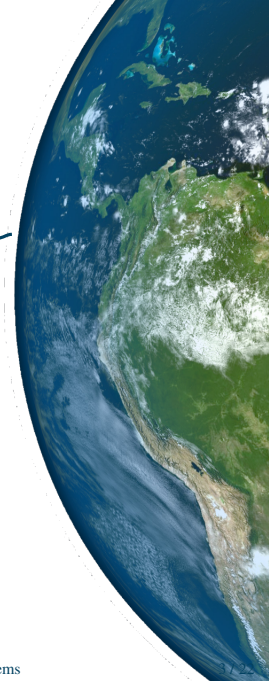
Verifier
(KeYmaera)

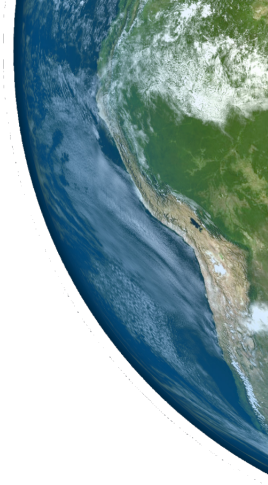


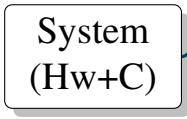
System (Hw+C)

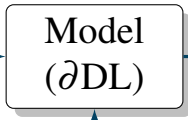
Model (∂DL)

Verifier (KeYmaera)

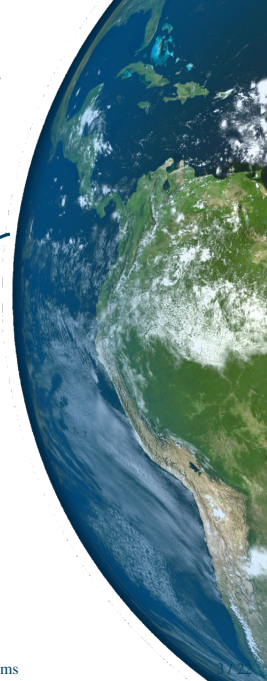


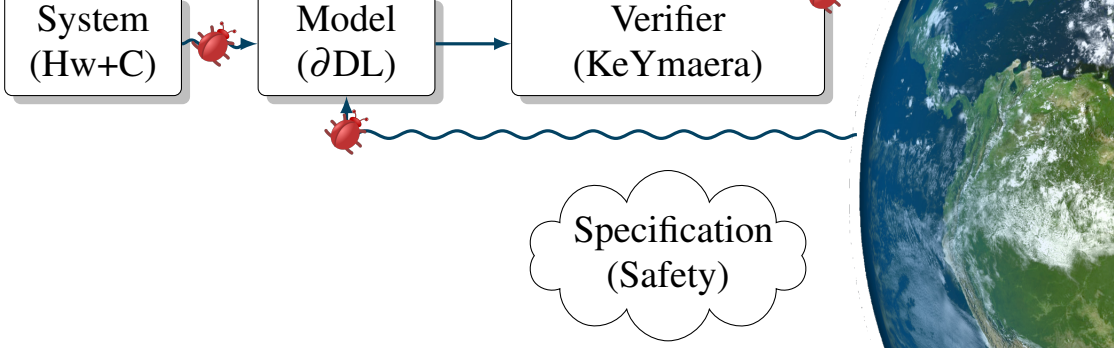


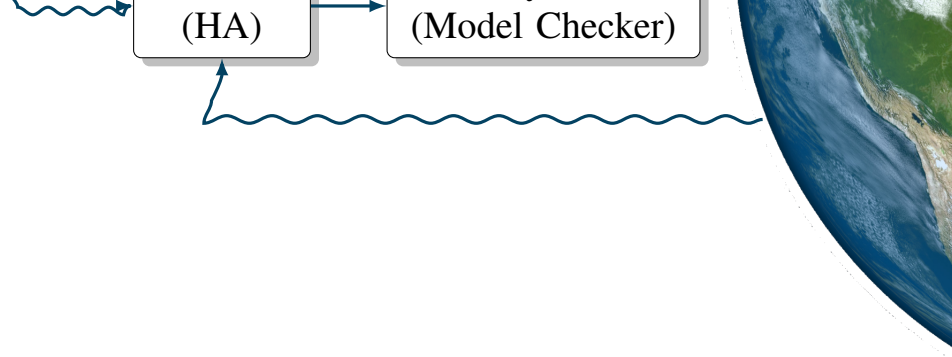

System
(Hw+C)

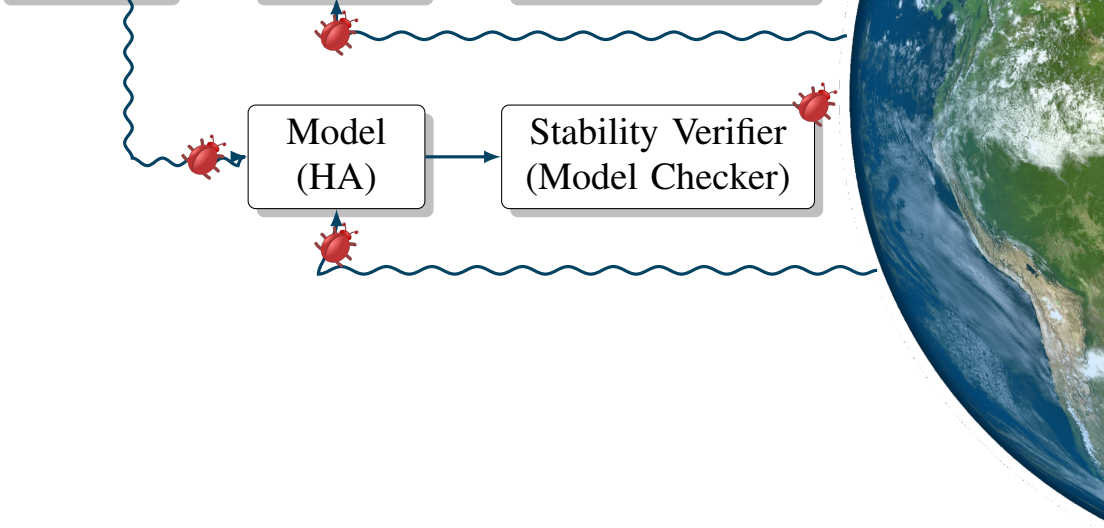

Model
(∂DL)

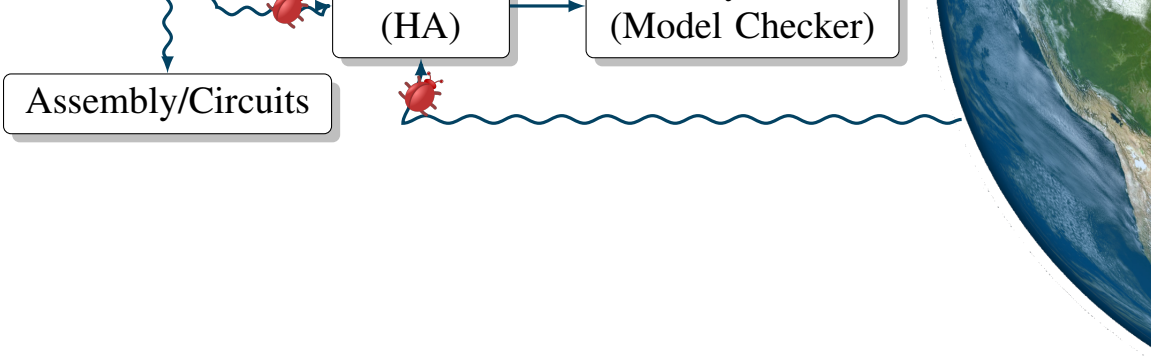

Verifier
(KeYmaera)

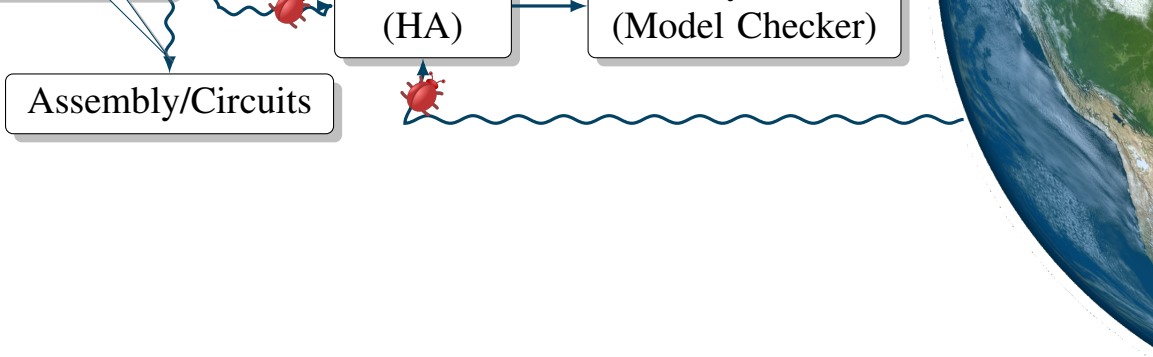


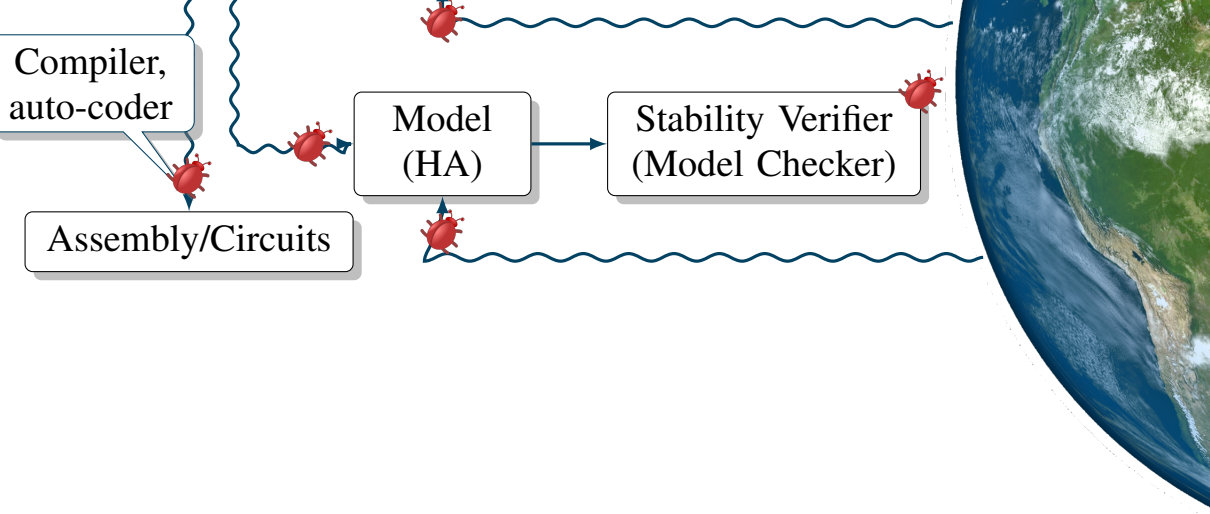

System
(Hw+C)
Model
(∂DL)
Verifier
(KeYmaera)
Specification
(Safety)

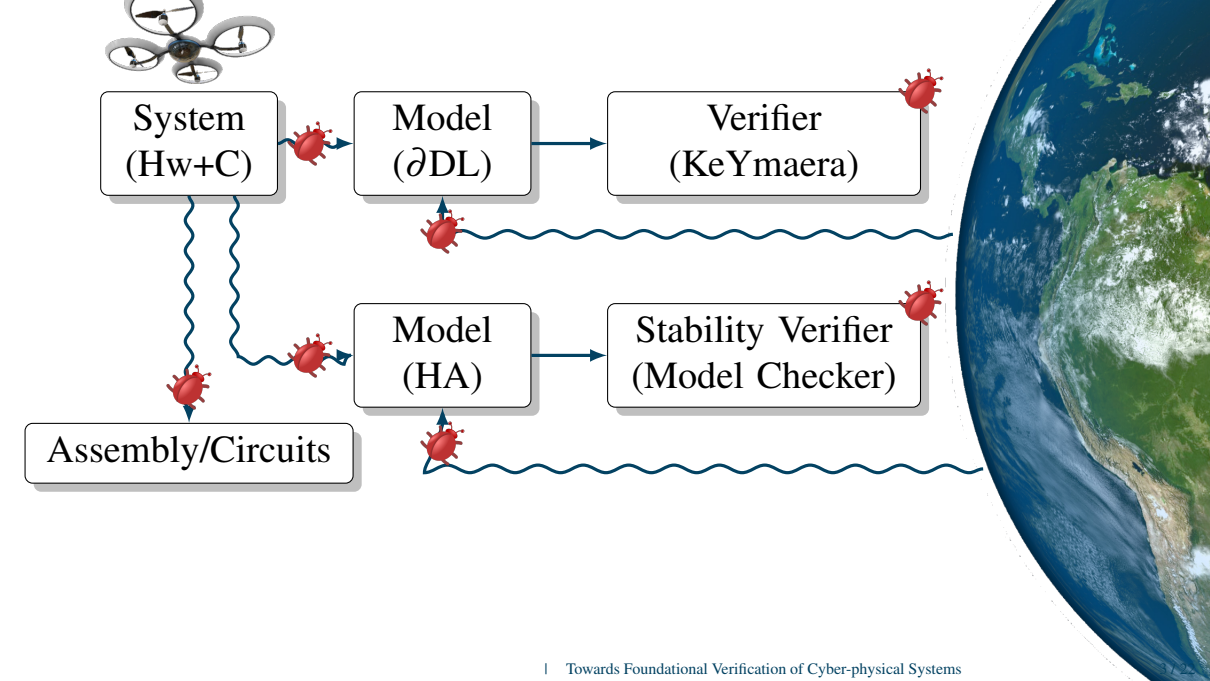

(HA)
(Model Checker)

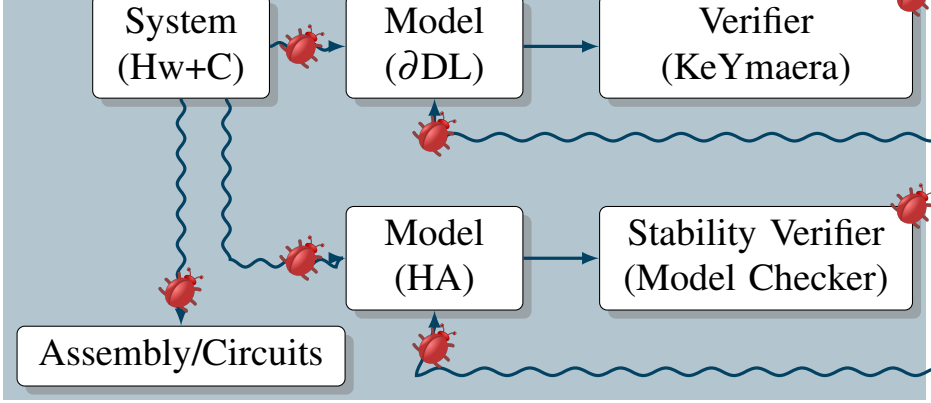

Model
(HA)
Stability Verifier
(Model Checker)

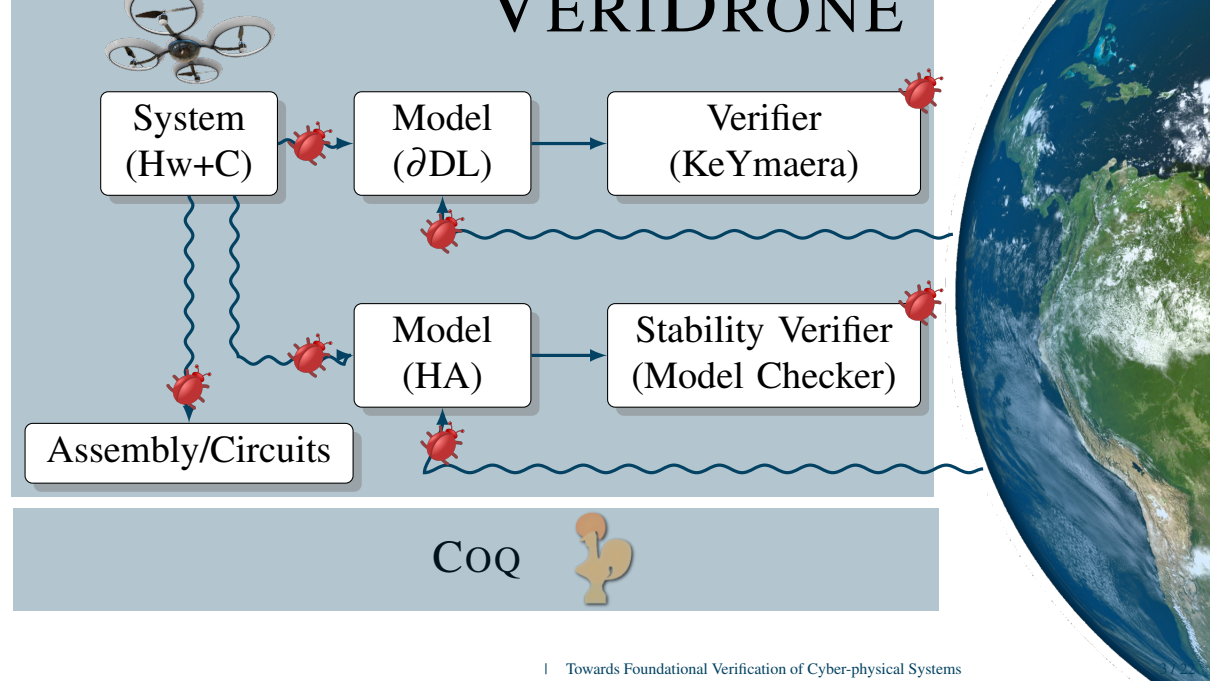

(HA)
(Model Checker)
Assembly/Circuits


(HA)
(Model Checker)
Assembly/Circuits


Compiler,
auto-coder
Assembly/Circuits
Model
(HA)
Stability Verifier
(Model Checker)


System
(Hw+C)
Model
(∂DL)
Verifier
(KeYmaera)
Model
(HA)
Stability Verifier
(Model Checker)
Assembly/Circuits


System
(Hw+C)
Model
(∂DL)
Verifier
(KeYmaera)
Model
(HA)
Stability Verifier
(Model Checker)
Assembly/Circuits


VERIDRONE
System
(Hw+C)
Model
(∂DL)
Verifier
(KeYmaera)
Model
(HA)
Stability Verifier
(Model Checker)
Assembly/Circuits
Coq

Benefits of Foundational Verification [YCER11]

Compiler	Bugs
GCC	122
Clang/LLVM	181
CompCert	

Benefits of Foundational Verification [YCER11]

Compiler	Bugs
GCC	122
Clang/LLVM	181
CompCert	0†

Verified in Coq

† In verified code

Benefits of Foundational Verification [YCER11]

Compiler	Bugs
GCC	122
Clang/LLVM	181
CompCert	$0^{\dagger}$

Verified in Coq

Foundational verification

- Strong guarantees, **and**
- Expressive logic

Verification in Coq

Foundational verification

- Strong guarantees, **and**
- Expressive logic

Verification in Coq

Definitions

```
Def opt (c : c_prog) : c_prog :=
  ... c ...
```

Foundational verification

- Strong guarantees, **and**
- Expressive logic

Verification in Coq

```
Def opt (c : c_prog) : c_prog :=
  ...  c ...

Thm opt_sou                prog,
  c ∼ opt c.
```

Definitions

Theorems

Foundational verification

- Strong guarantees, **and**
- Expressive logic

Verification in Coq

```
Def opt (c : c_prog) : c_prog :=
  ... c ...

Thm opt_sound : ∀ c : c_prog,
  c ∼ opt c.
Proof.
  induction c.
  (* proof for each case *)
```

Definitions

Theorems

Foundational verification

- Strong guarantees, and
- Expressive logic

Verification in Coq

```
Def opt (c : c_prog) : c_prog :=
  ... c ...

Thm opt_sound : ∀ c : c_prog,
  c ∼ opt c.
Proof.
  induction c.
  (* proof for each case *)
Qed.
```

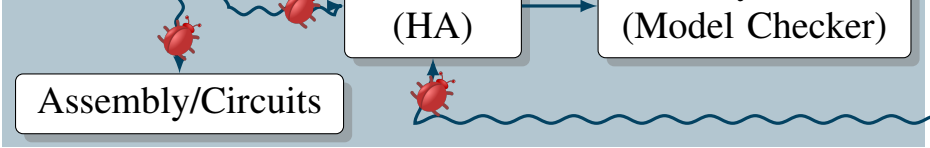
Definitions

Theorems

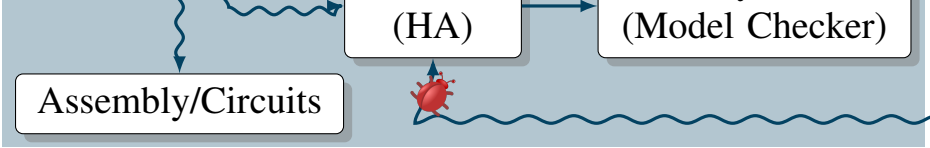
Interactive proof scripts

Foundational verification

- Strong guarantees, **and**
- Expressive logic

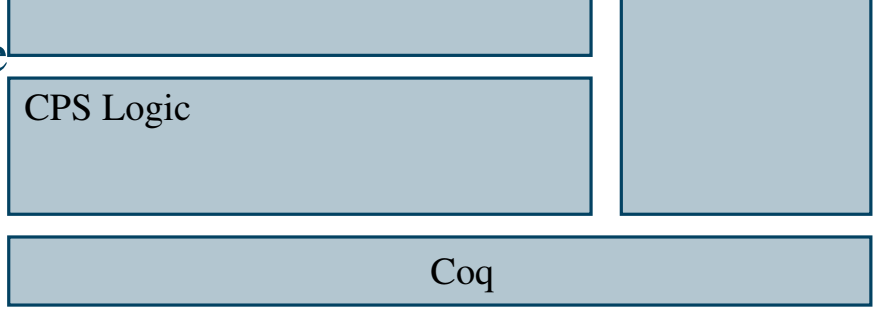


Coq

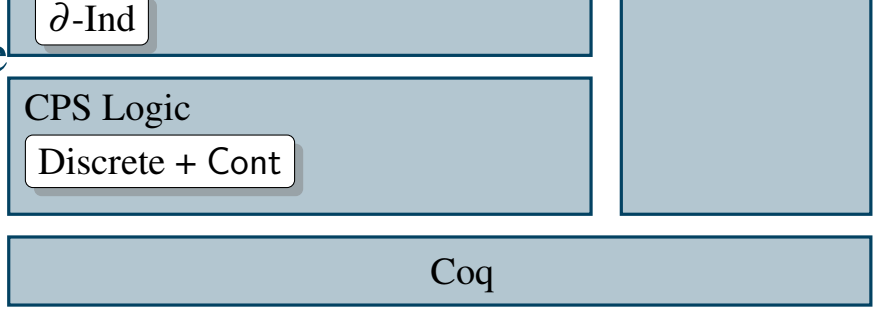


Coq

Outline

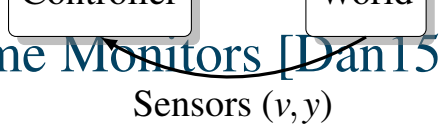


Outline

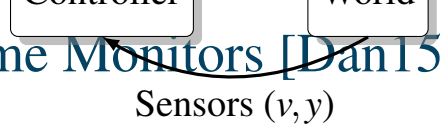


Case Study: Runtime Monitors [Dan15]

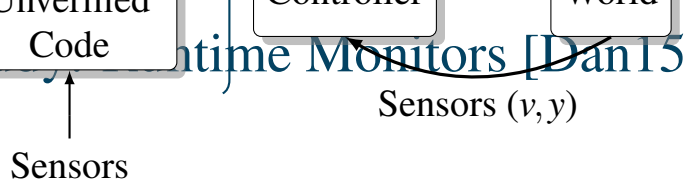
Case Study: Runtime Monitors [Dan15]



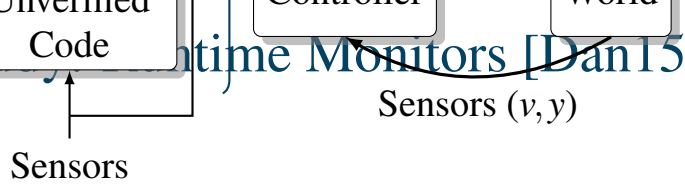
Case Study: Runtime Monitors [Dan15]



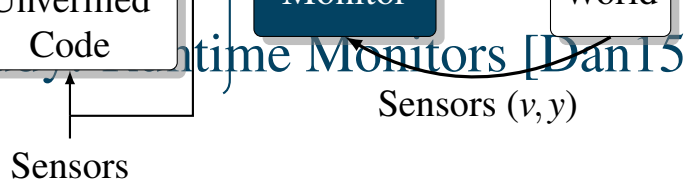
Case Study: Runtime Monitors [Dan15]



Case Study: Runtime Monitors [Dan15]



Case Study: Runtime Monitors [Dan15]



Case Study: Runtime Monitors [Dan15]

Case Study: Runtime Monitors [Dan15]

"Sampled-Data System"

Case Study: Runtime Monitors [Dan15]

"Always"

Case Study: Runtime Monitors [Dan15]

Initial condition

Case Study: Runtime Monitors [Dan15]

$$\text{Init} \wedge \Box\, \text{Sys}_{\Delta}\Big(\underbrace{\text{Monitor}}_{\text{Monitor}}, \underbrace{\text{World}}_{\text{World}}\Big) \vdash \Box P$$

Boundness, stability, robustness, etc.

A Flavor of Customizable Verification

$$\frac{}{P \wedge \Box \mathsf{Sys}_{\Delta}(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

Custom proof rules

A Flavor of Customizable Verification

′ = after the transition

(Discrete)

$$\frac{P \wedge D \wedge 0 \leq \tau' \leq \Delta \vdash P'}{P \wedge \Box \mathsf{Sys}_{\Delta}(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

A Flavor of Customizable Verification

(Discrete)

$$\frac{P \wedge D \wedge 0 \leq \tau' \leq \Delta \vdash P'}{P \wedge \Box \mathsf{Sys}_{\Delta}(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

A Flavor of Customizable Verification

(Continuous) $P \wedge \mathsf{Cont}(W \wedge \dot{\tau} = -1) \wedge 0 \leq \tau' \vdash P'$

(Discrete)

$$\frac{P \wedge D \wedge 0 \leq \tau' \leq \Delta \vdash P'}{P \wedge \Box \mathsf{Sys}_{\Delta}(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

Continuous evolution

Derivative

A Flavor of Customizable Verification

(Continuous)

(Discrete)

$$\frac{P \wedge \mathsf{Cont}\,(W \wedge \dot{\tau} = -1) \wedge 0 \leq \tau' \vdash P' \qquad P \wedge D \wedge 0 \leq \tau' \leq \Delta \vdash P'}{P \wedge \Box \mathsf{Sys}_{\Delta}\,(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

A Flavor of Customizable Verification

$$\frac{P \wedge \mathsf{Cont}\,(W \wedge \dot{\tau} = -1) \wedge 0 \leq \tau' \vdash P' \qquad P \wedge D \wedge 0 \leq \tau' \leq \Delta \vdash P'}{P \wedge \Box \mathsf{Sys}_{\Delta}\,(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

A Flavor of Customizable Verification

$$\frac{P \wedge \mathsf{Cont}\,(W \wedge \dot{\tau} = -1) \wedge 0 \leq \tau' \vdash P' \qquad P \wedge D \wedge 0 \leq \tau' \leq \Delta \vdash P'}{P \wedge \Box \mathsf{Sys}_{\Delta}\,(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

A Flavor of Customizable Verification

$$\frac{P \wedge \mathsf{Cont}\,(W \wedge \dot{\tau} = -1) \wedge 0 \leq \tau' \vdash P' \qquad P \wedge D \wedge 0 \leq \tau' \leq \Delta \vdash P'}{P \wedge \Box \mathsf{Sys}_{\Delta}\,(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

A Flavor of Customizable Verification

Def

$$\frac{P \wedge \mathsf{Cont}(W \wedge \dot{\tau} = -1) \wedge 0 \leq \tau' \vdash P' \qquad P \wedge D \wedge 0 \leq \tau' \leq \Delta \vdash P'}{P \wedge \Box \mathsf{Sys}_{\Delta}(D, W) \vdash \Box P}\ \textsc{Sys-Ind}$$

Thm

Def

Aside: Simplified Quadcopter Dynamics

Linear dynamics

World dynamics

Angular thrus

$$W_{QC} \triangleq \left(\begin{array}{lll} C_{\theta\phi} & \rightarrow & \dot{\mathbf{x}} = \mathbf{v_x} \wedge \dot{\mathbf{y}} = \mathbf{v_y} \wedge \dot{\mathbf{z}} = \mathbf{v_z} \\ & \wedge & \dot{\mathbf{v_x}} = \mathbf{T} \cos\phi \sin\theta \\ & \wedge & \dot{\mathbf{v_y}} = -\mathbf{T} \sin\phi \\ & \wedge & \dot{\mathbf{v_z}} = \mathbf{T} \cos\phi \cos\theta - g \\ & \wedge & \dot{\phi} = 0 \wedge \dot{\theta} = 0 \wedge \dot{\mathbf{T}} = 0 \end{array} \right)$$

Pitch

Roll

$$C_{\theta\phi} \triangleq \quad |\theta| \leq 30° \wedge |\phi| \leq 30° \wedge 0 \leq \mathbf{T}$$

Simple dynamics if "mostly level" ($< 30°$)

Aside: Simplified Quadcopter Dynamics

$$W_{OC} \triangleq \left(\begin{array}{lll} C_{\theta\phi} & \rightarrow & \dot{\mathbf{x}} = \mathbf{v_x} \wedge \dot{\mathbf{y}} = \mathbf{v_y} \wedge \dot{\mathbf{z}} = \mathbf{v_z} \\ & \wedge & \dot{\mathbf{v_x}} = \mathbf{T}\cos\phi\sin\theta \\ & \wedge & \dot{\mathbf{v_y}} = -\mathbf{T}\sin\phi \\ & \wedge & \dot{\mathbf{v_z}} = \mathbf{T}\cos\phi\cos\theta - \\ & \wedge & \dot{\phi} = 0 \wedge \dot{\theta} = 0 \wedge \dot{\mathbf{T}} = 0 \end{array} \right)$$

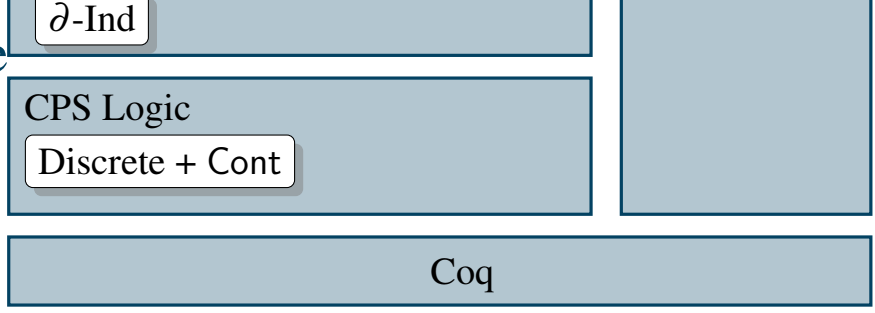
"Small-angle constraint"

Instantaneous change

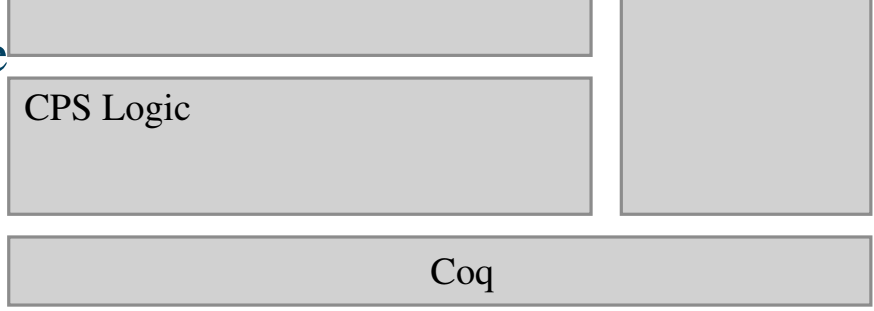
$$C_{\theta\phi} \triangleq \quad |\theta| \leq 30° \wedge |\phi| \leq 30° \wedge 0 \leq \mathbf{T}$$

† Approximation justified by fast angular dynamics and the small angle constraint [GHH+11].

Outline

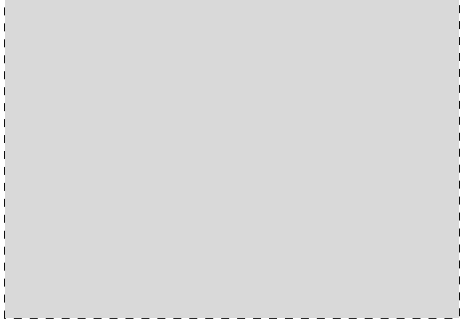


Outline



Composing Monitors [Dan16a]

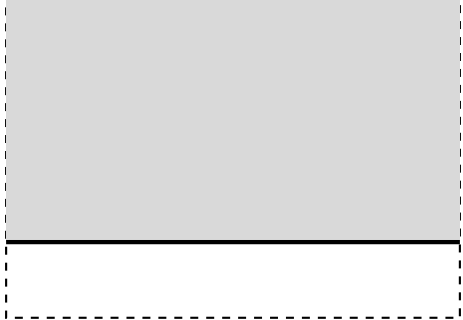
Composing Monitors [Dan16a]



- Spatial transformation

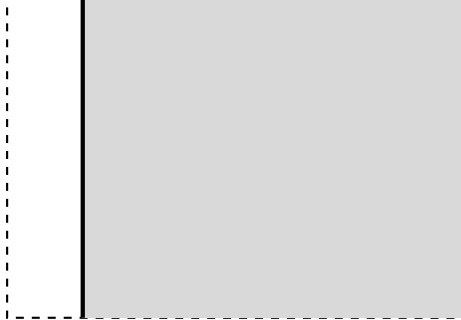
Composing Monitors [Dan16a]

- Spatial transformation



Composing Monitors [Dan16a]

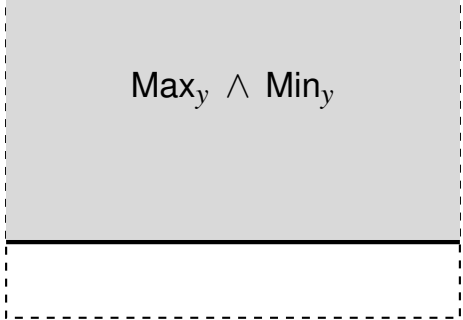
- Spatial transformation



Composing Monitors [Dan16a]

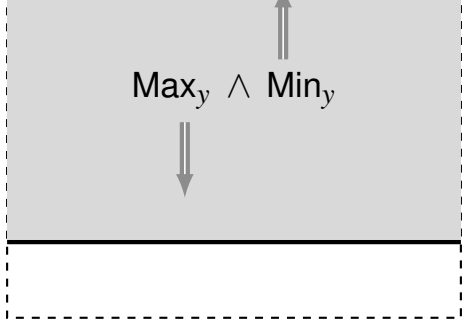
- Spatial transformation

Composing Monitors [Dan16a]



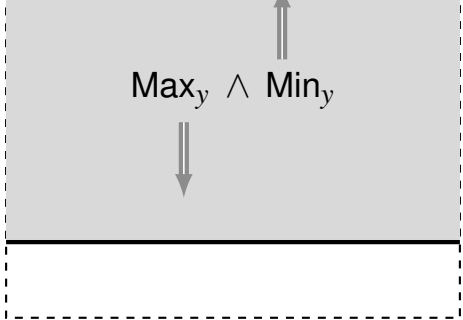
- Spatial transformation
- Conjunctive composition

Composing Monitors [Dan16a]



- Spatial transformation
- Conjunctive composition

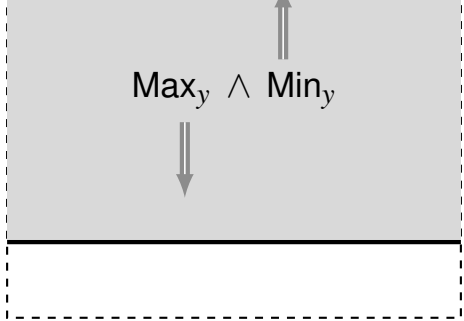
Composing Monitors [Dan16a]



Monitors might not be compatible!

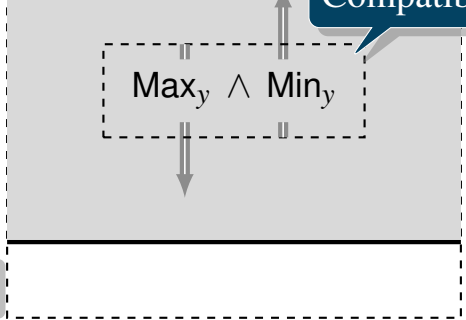
- Spatial transformation
- Conjunctive composition

Composing Monitors [Dan16a]



- Spatial transformation
- Conjunctive composition

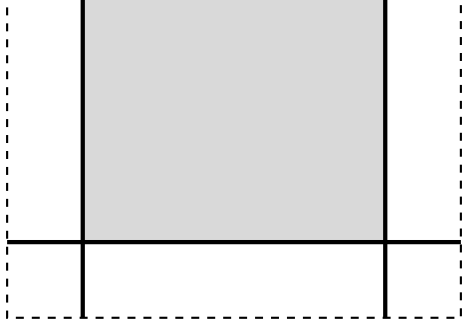
Composing Monitors [Dan16a]



- Spati
- Conjunctive composition

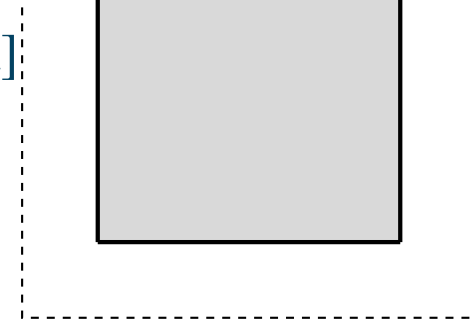
Formalize and build proof rules

Composing Monitors [Dan16a]



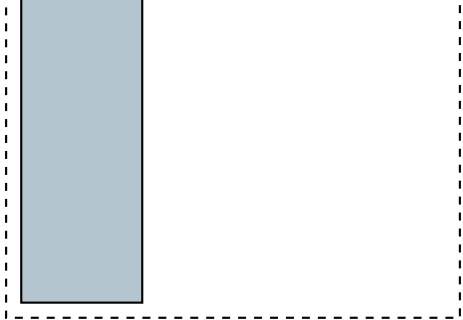
- Spatial transformation
- Conjunctive composition

Composing Monitors [Dan16a]



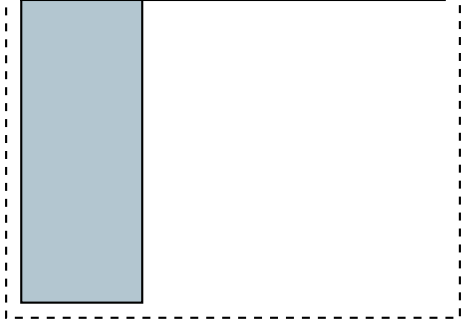
- Spatial transformation
- Conjunctive composition

Composing Monitors [Dan16a]



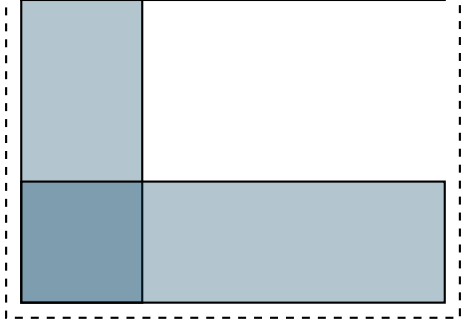
- Spatial transformation
- Conjunctive composition
- Disjunctive composition

Composing Monitors [Dan16a]



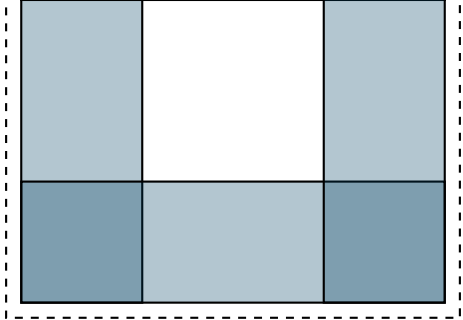
- Spatial transformation
- Conjunctive composition
- Disjunctive composition

Composing Monitors [Dan16a]



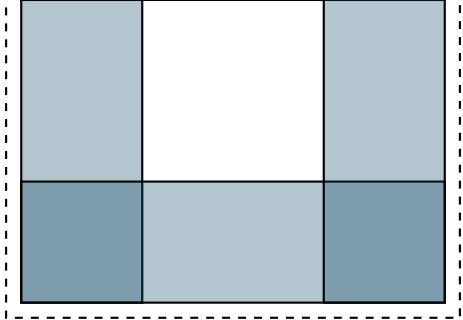
- Spatial transformation
- Conjunctive composition
- Disjunctive composition

Composing Monitors [Dan16a]



- Spatial transformation
- Conjunctive composition
- Disjunctive composition

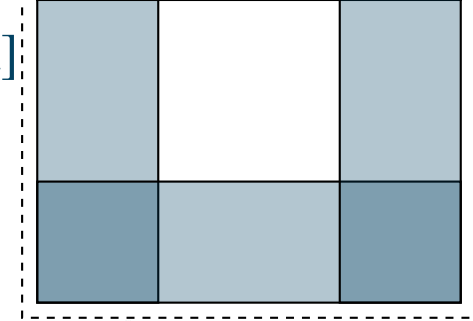
Composing Monitors [Dan16a]



- Spatial transformation
- Conjunctive composition
- Disjunctive composition

Composing Monitors [Dan16a]

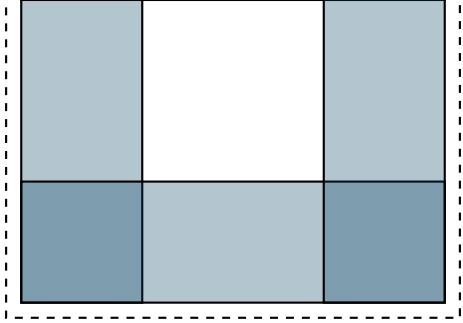
- Spatial transformation
- Conjunctive composition
- Disjunctive composition



Composing Monitors [Dan16a]

Thm

Thm

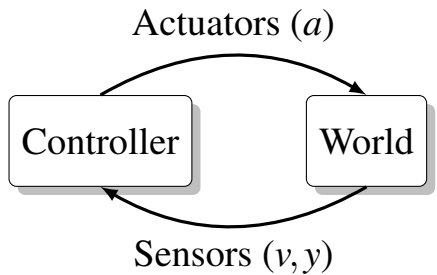


- Spatial transformation
- Conjunctive composition
- Disjunctive composition

Extending the Model: Robustness [Dan16b]

Tolerance to disturbances

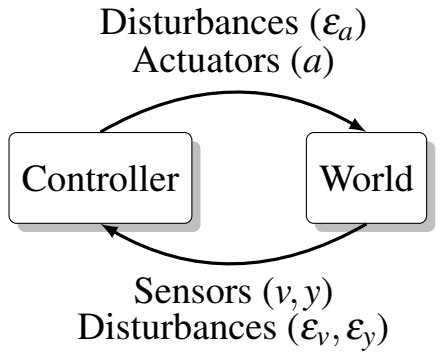
- Add disturbances
- Bound violations



Extending the Model: Robustness [Dan16b]

Tolerance to disturbances

- Add disturbances
- Bound violations

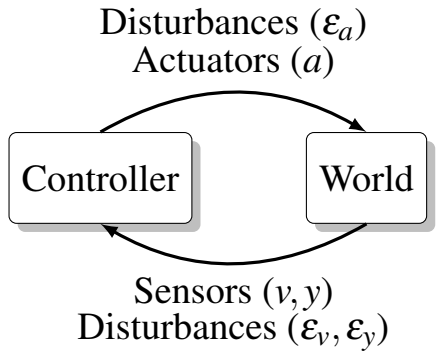


Extending the Model: Robustness [Dan16b]

Def

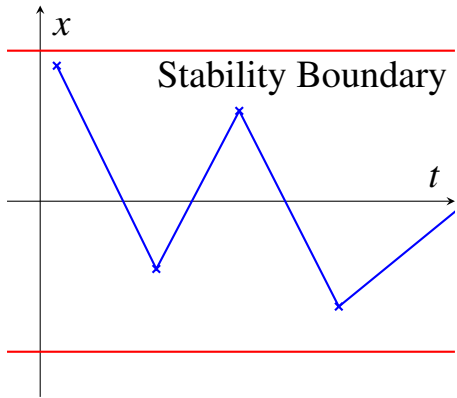
Tolerance to disturbances

- Add disturbances
- Bound violations



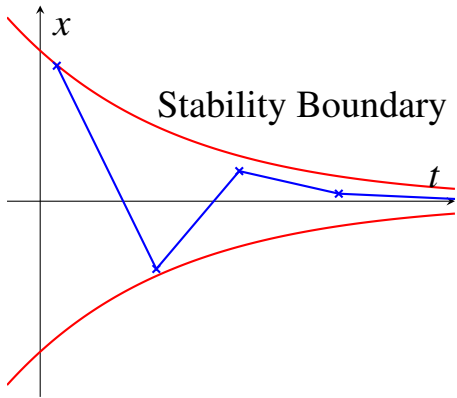
More Properties: Stability [Mat16]

- Boundedness over time



More Properties: Stability [Mat16]

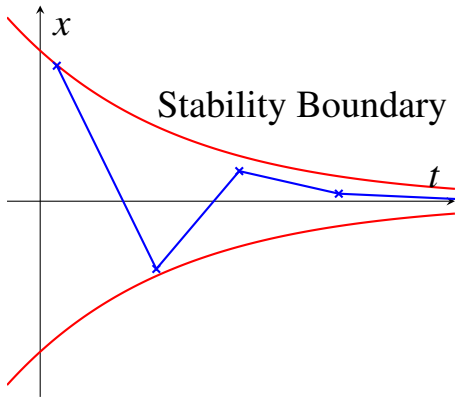
- Boundedness over time
- Convergence to a goal



More Properties: Stability [Mat16]

- Boundedness over time
- Convergence to a goal

- 100 years of control theory!



More Properties: Stability [Mat16]

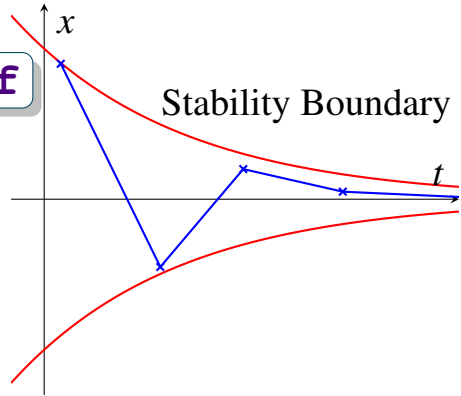
- Boundedness over time
- Convergence to a goal
- 100 years of control theory!

Def

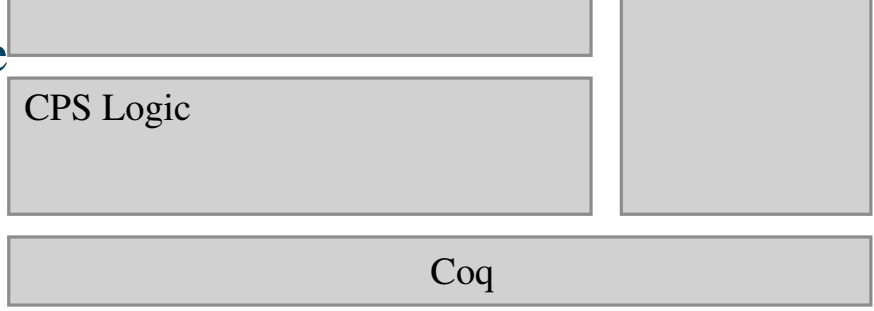
Def

Thm

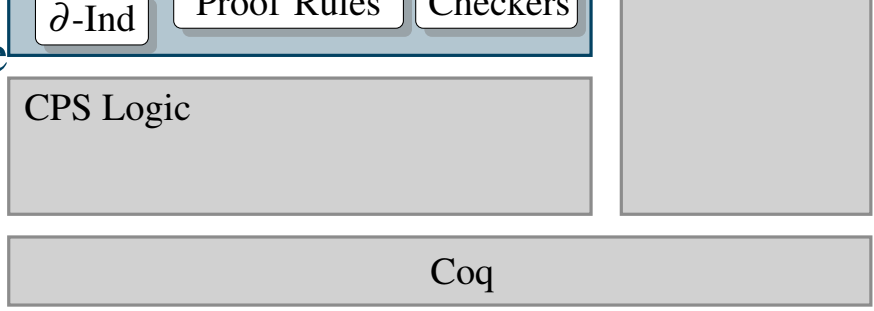
Thm



Outline



Outline



External Automation

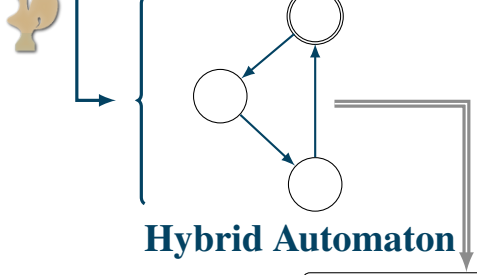
Connect to other tools

External Automation

Connect to other tools

- Formalize in Coq

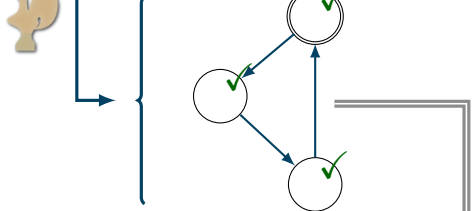
External Automation



Connect to other tools

- Formalize in Coq
- Leverage automation

External Automation

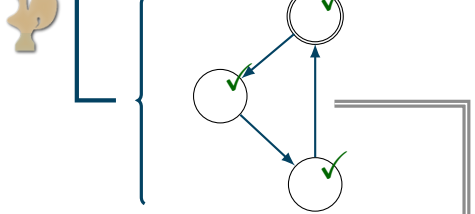


Connect to other tools

- Formalize in Coq
- Leverage automation

Model Checker
(e.g. SpaceEx)

External Automation

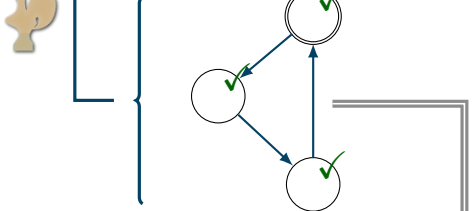


Connect to other tools

- Formalize in Coq
- Leverage automation

Model Checker
(e.g. SpaceEx)

External Automation



Connect to other tools

- Formalize in Coq
- Leverage automation

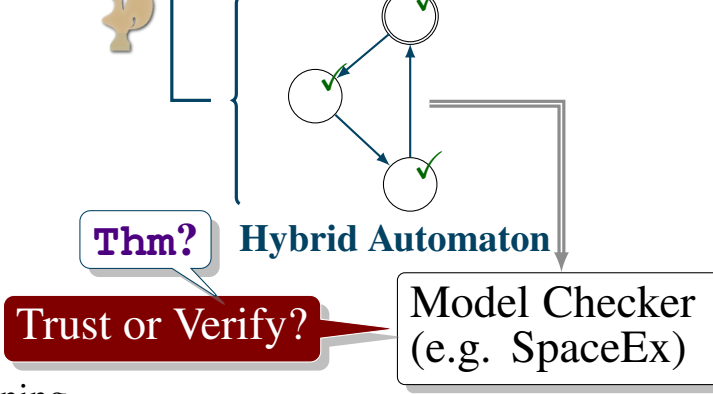
Trust or Verify?

Model Checker
(e.g. SpaceEx)

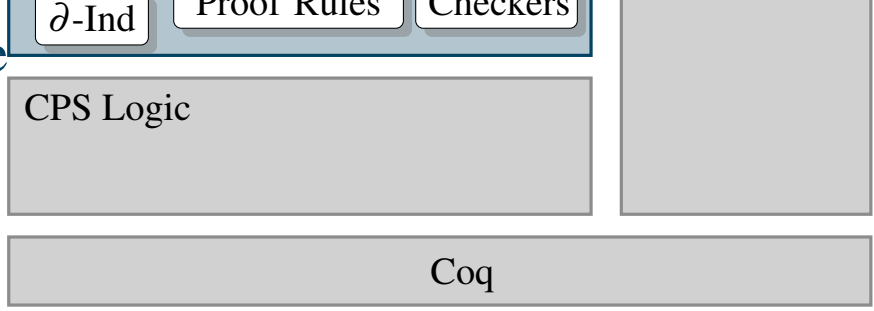
External Automation

Connect to other tools

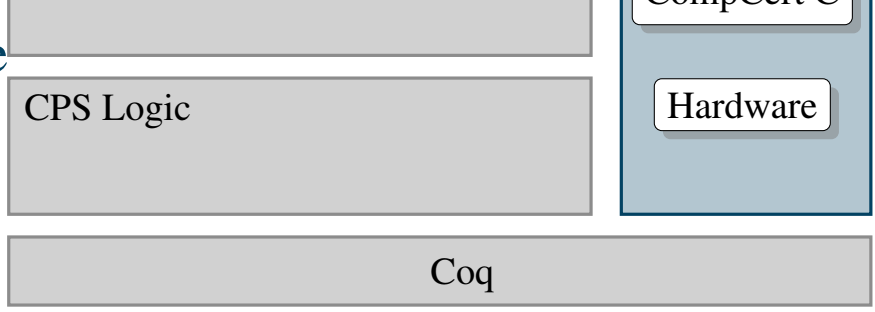
- Formalize in Coq
- Leverage automation
- Combine with other reasoning



Outline



Outline



End-to-End Guarantees

$$\mathsf{Sys}_{\Delta}(\qquad D,\qquad W) \vdash P$$

Connect models & code

- Existing developments

End-to-End Guarantees

Floating point & execution

$$\mathsf{Sys}_\Delta(\ D,\ W) \vdash P$$

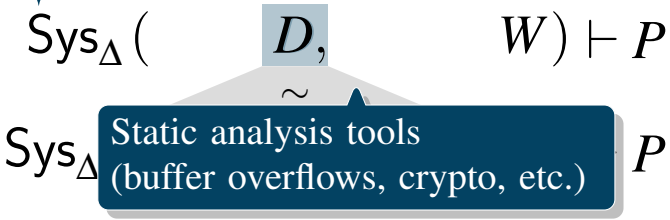
$$\sim$$

$$\mathsf{Sys}_\Delta(\ \texttt{C\{C-code\}},\ W) \vdash P$$

Connect models & code

- Existing developments

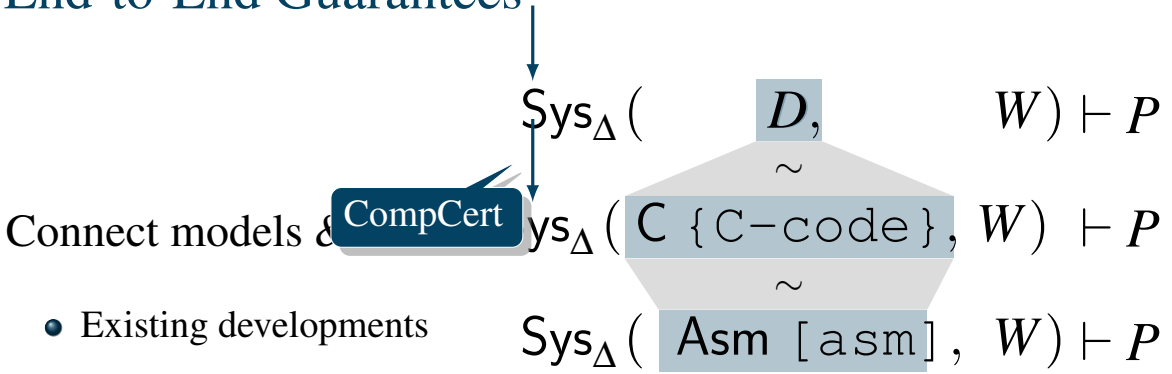
End-to-End Guarantees



Connect models & code

- Existing developments

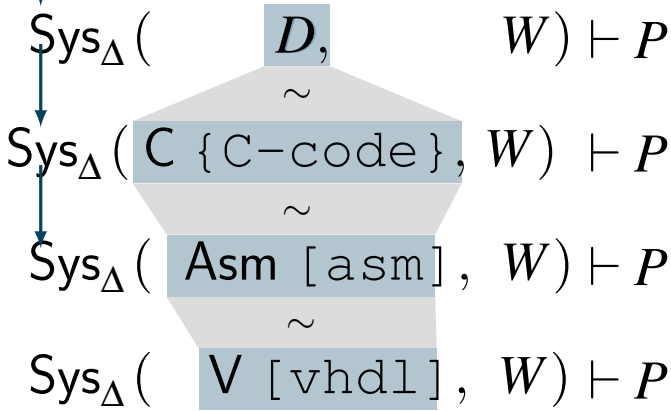
End-to-End Guarantees



End-to-End Guarantees

Connect models & code

- Existing developments



End-to-End Guarantees

Thm

Def

$$\mathsf{Sys}_\Delta(\ D,\ W) \vdash P$$

Def

$$\sim$$

$$\mathsf{Sys}_\Delta(\ \mathsf{C}\ \texttt{\{C-code\}},\ W) \vdash P$$

Connect models & code

Thm

Def

$$\sim$$

$$\mathsf{Sys}_\Delta(\ \mathsf{Asm}\ \texttt{[asm]},\ W) \vdash P$$

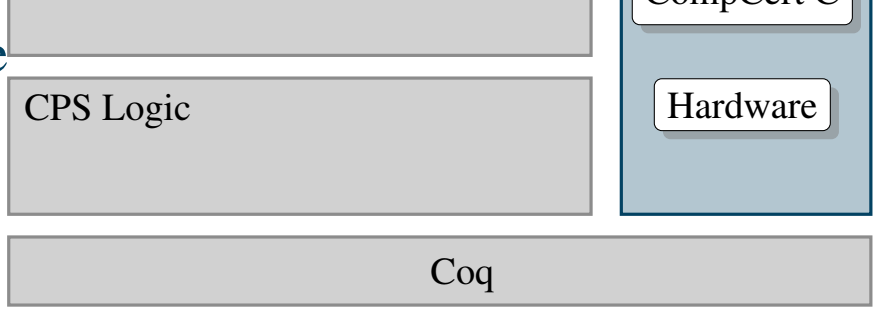
- Existing developments

Foundational proofs of *actual code*

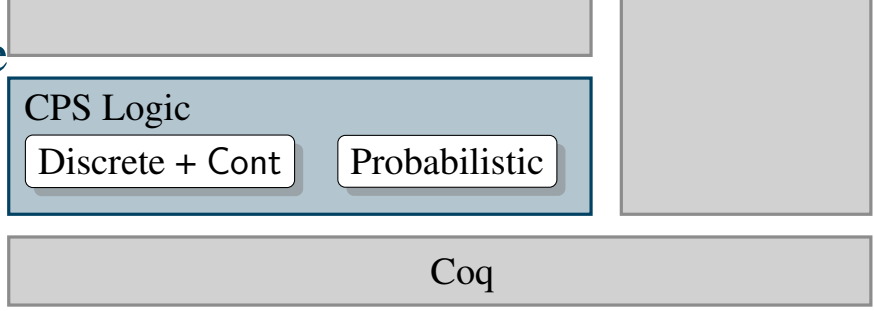
$$\sim$$

$$\mathsf{Sys}_\Delta(\ \mathsf{V}\ \texttt{[vhdl]},\ W) \vdash P$$

Outline

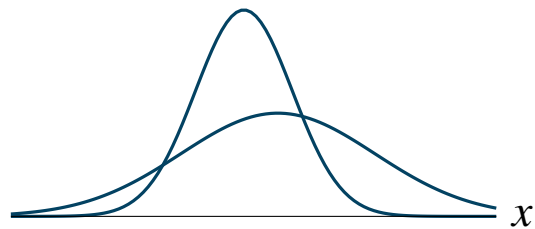


Outline



Uncertainty

- Probabilistic processes
- Minimize uncertainty



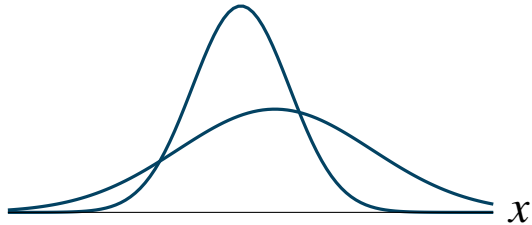
Uncertainty

fused

gps

barometer

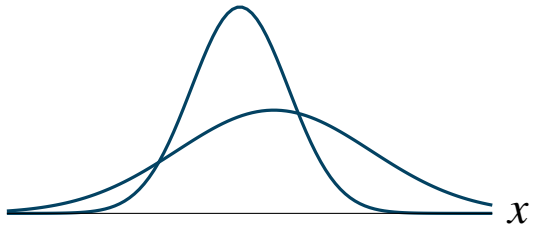
- Probabilistic processes
- Minimize uncertainty



Uncertainty

Safe (80%) Unsafe (20%)

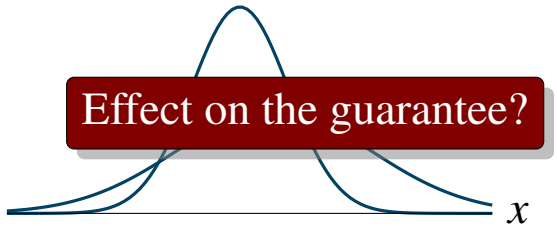
- Probabilistic processes
- Minimize uncertainty
- Decide with uncertainty



Uncertainty

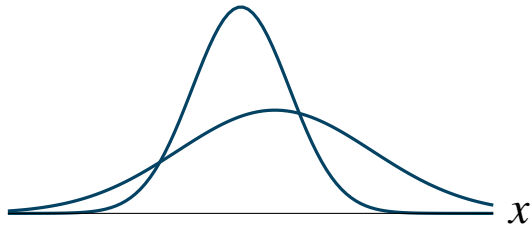
Safe (80%) Unsafe (20%)

- Probabilistic processes
- Minimize uncertainty
- Decide with uncertainty

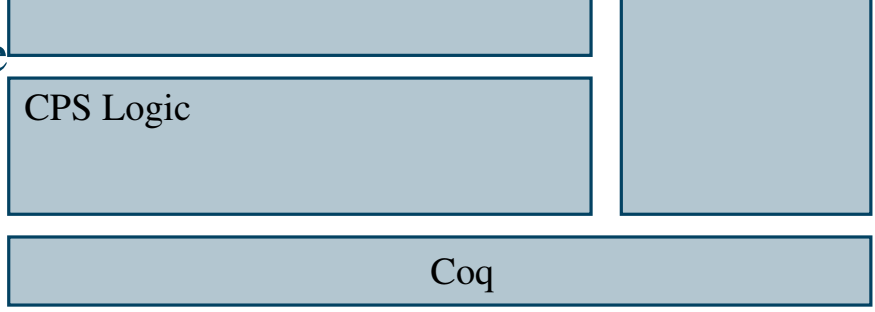


Uncertainty

- Probabilistic processes
- Minimize uncertainty
- Decide with uncertainty



Outline



Out

http://verifydone.ucsd.edu/

CPS Logic

Coq

References I

Daniel Ricketts and Gregory Malecha and Mario M. Alvarez and Vignesh Gowda and Sorin Lerner.

Towards Verification of Hybrid Systems in a Foundational Proof Assistant.

In *MEMOCODE '15*, 2015.

Daniel Ricketts and Gregory Malecha and Sorin Lerner.

Modular Reasoning about Cyber-physical Systems.

2016.

Daniel Ricketts and Gregory Malecha and Sorin Lerner.

Verifying Robustness of Cyber-Physical Systems.

2016.

References II

Jeremy H. Gillula, Gabriel M. Hoffmann, Huang Haomiao, Michael P. Vitus, and Claire J. Tomlin.

Applications of hybrid reachability analysis to robotic aerial vehicles.

The International Journal of Robotics Research, 30(3):335–354, 2011.

Matthew Chan and Daniel Ricketts and Sorin Lerner and Gregory Malecha.

Formal Verification of Stability Properties of Cyber-physical Systems, 2016.

Xuejun Yang, Yang Chen, Eric Eide, and John Regehr.

Finding and understanding bugs in C compilers.

PLDI '11, pages 283–294, New York, NY, USA, 2011. ACM.