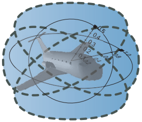
Intersymbolic AI

Interlinking Symbolic AI and Subsymbolic AI

André Platzer

Karlsruhe Institute of Technology



Outline

Outline

Artificial Intelligence: Symbolic AI vs. Subsymbolic AI

Symbolic AI Examples

- Logic
- Games
- Planning

Subsymbolic AI Examples

- Supervised ML
- Unsupervised ML
- Reinforcement learning

Symbolic AI

- symbolic roots
- parts carry meaning
- parts are interpretable
- whole composed of parts
- meaning is a function of the meaning of the parts

Symbolic AI Examples

- Logic
- Games
- Planning

Subsymbolic AI

- numerical roots
- parts carry no meaning
- (beyond role they play in the whole computation)
- indirect result via iterative optimization from data

Subsymbolic AI Examples

- Supervised ML
- Unsupervised ML
- Reinforcement learning

Artificial Intelligence: Symbolic AI vs. Subsymbolic AI

Symbolic AI

- symbolic roots
- parts carry meaning
- parts are interpretable
- whole composed of parts
- meaning is a function of the meaning of the parts

1. compositional
2. significance/meaning

Subsymbolic AI

- numerical roots
- parts carry no meaning
- (beyond role they play in the whole computation)
- indirect result via iterative optimization from data

1. summative
2. significance/effect

[0] Symbolic AI vs. Subsymbolic AI characterizes essential source of difference. Contrast: traditional vs. nontraditional AI, rule-based vs. learning-based, good old fashioned AI, neuro-symbolic AI ...

Artificial Intelligence: Symbolic AI vs. Subsymbolic AI

Symbolic AI

- symbolic roots
- parts carry meaning
- parts are interpretable
- whole composed of parts
- meaning is a function of the meaning of the parts

1. compositional
2. significance/meaning

Subsymbolic AI

- numerical roots
- parts carry no meaning
- (beyond role they play in the whole computation)
- indirect result via iterative optimization from data

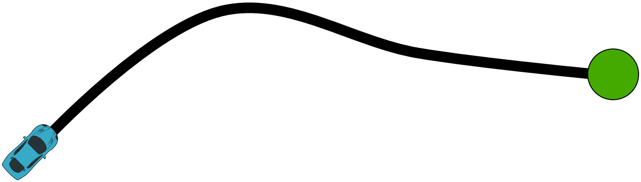
1. summative
2. significance/effect

Thesis and antithesis waiting for dialectic synthesis

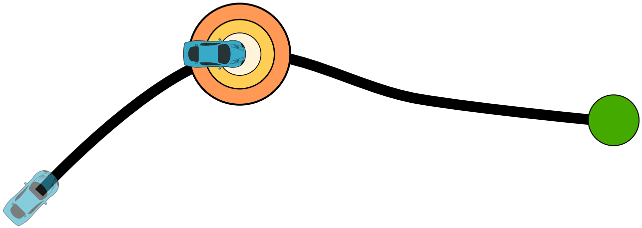
---

[0] Symbolic AI vs. Subsymbolic AI characterizes essential source of difference. Contrast: traditional vs. nontraditional AI, rule-based vs. learning-based, good old fashioned AI, neuro-symbolic AI . . .

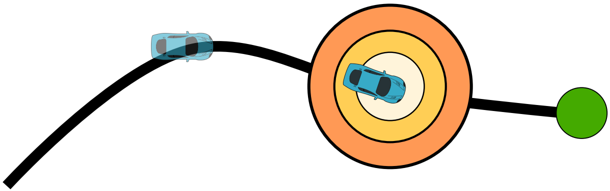
Intersymbolic AI = Symbolic AI + Subsymbolic AI

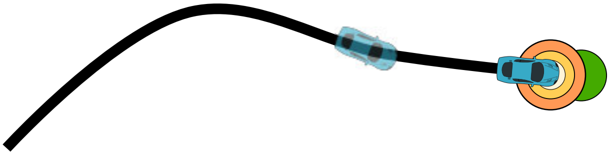


Intersymbolic AI = Symbolic AI + Subsymbolic AI



Intersymbolic AI = Symbolic AI + Subsymbolic AI





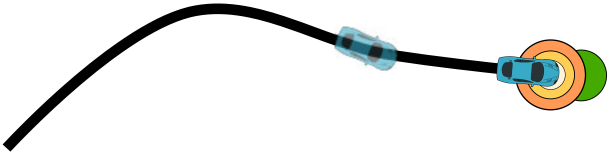
Intersymbolic AI = Symbolic AI + Subsymbolic AI

Symbolic AI

+ accurate & precise
+ self-explanatory
- manual formalization
+ clear correctness
- scaling hard

Subsymbolic AI

+ no explicit programming
- inexplicable
- data inefficiency
- no* clear correctness
+ computational scaling



Intersymbolic AI = Symbolic AI + Subsymbolic AI

Symbolic AI

+ accurate & precise
+ self-explanatory
- manual formalization
+ clear correctness
- scaling hard

Subsymbolic AI

+ no explicit programming
- inexplicable
- data inefficiency
- no* clear correctness
+ computational scaling

Symbolic AI Examples

- Logic
- Games
- Planning

Subsymbolic AI Examples

- Supervised ML
- Unsupervised ML
- Reinforcement learning

Intersymbolic AI = Symbolic AI + Subsymbolic AI

Symbolic AI

+ accurate & precise
+ self-explanatory
- manual formalization
+ clear correctness
- scaling hard

Subsymbolic AI

+ no explicit programming
- inexplicable
- data inefficiency
- no* clear correctness
+ computational scaling

Intersymbolic AI = Symbolic AI + Subsymbolic AI

Intersymbolic AI interlinks symbolic AI (compositional significance/meaning) with subsymbolic AI (summative significance/effect) to combine insights from both worlds by going between and across symbolic AI insights with subsymbolic AI helped by symbolic AI principles.

Intersymbolic AI = Symbolic AI + Subsymbolic AI

Symbolic AI

- \+ accurate & precise
- \+ self-explanatory
- \- manual formalization
- \+ clear correctness
- \- scaling hard

Subsymbolic AI

- \+ no explicit programming
- \- inexplicable
- \- data inefficiency
- \- no* clear correctness
- \+ computational scaling

Intersymbolic AI = Symbolic AI + Subsymbolic AI

Intersymbolic AI interlinks symbolic AI (compositional significance/meaning) with subsymbolic AI (summative significance/effect) to combine insights from both worlds by going between and across symbolic AI insights with subsymbolic AI helped by symbolic AI principles.

Analogy: Human Thought = Conscious Thought + Subconscious Thought

Intersymbolic AI = Symbolic AI + Subsymbolic AI

Symbolic AI

- \+ accurate & precise
- \+ self-explanatory
- \- manual formalization
- \+ clear correctness
- \- scaling hard

Subsymbolic AI

- \+ no explicit programming
- \- inexplicable
- \- data inefficiency
- \- no* clear correctness
- \+ computational scaling

Intersymbolic AI = Symbolic AI + Subsymbolic AI

Intersymbolic AI interlinks symbolic AI (compositional significance/meaning) with subsymbolic AI (summative significance/effect) to combine insights from both worlds by going between and across symbolic AI insights with subsymbolic AI helped by symbolic AI principles.

Example: symbolic AI understands system dynamics, subsymbolic AI learns control, symbolic AI safely uses learned control in dynamics

Outline

Intersymbolic AI Bestiarium

- Safe AI in CPS via reinforcement learning + hybrid systems theorem proving and proof-based synthesis AAAI'18
- Continuous invariant generation via symbolic proof, first integrals, Darboux + eigensystems, SOS & linear programming FMSD'22
- Loop invariant synthesis via AlphaZero for reinforcement learning on deep NN + theorem proving, nondeterministic programming NeurIPS'22
- Safe waypoint following via NN path tracking and MPC + hybrid systems theorem proving CSyL'22
- Control envelope synthesis via arithmetic simplification, refinement approximation + hybrid systems verification, game theory TACAS'24
- NNCS via NN verification + complete arithmetic linearization, hybrid systems theorem proving NeurIPS'24

3. Widely different symbolic AI + subsymbolic AI combinations!

Intersymbolic AI Bestiarium

- ✓ Safe AI in CPS via reinforcement learning + hybrid systems theorem proving and proof-based synthesis AAAI'18
- ✓ Continuous invariant generation via symbolic proof, first integrals, Darboux + eigensystems, SOS & linear programming FMSD'22
- ✓ Loop invariant synthesis via AlphaZero for reinforcement learning on deep NN + theorem proving, nondeterministic programming NeurIPS'22
- ✓ Safe waypoint following via NN path tracking and MPC + hybrid systems theorem proving CSyL'22
- ✓ Control envelope synthesis via arithmetic simplification, refinement approximation + hybrid systems verification, game theory TACAS'24
- ✓ NNCS via NN verification + complete arithmetic linearization, hybrid systems theorem proving NeurIPS'24

- ✓ Base technology: Dynamic logic proving LICS'12
2. Base technology: ModelPlex shielding FMSD'16
3. Widely different symbolic AI + subsymbolic AI combinations!

Intersymbolic AI Bestiarium

- ✓ Safe AI in CPS via reinforcement learning + hybrid systems theorem proving and proof-based synthesis — AAAI'18
- Continuous invariant generation via symbolic proof, first integrals, Darboux + eigensystems, SOS & linear programming — FMSD'22
- Loop invariant synthesis via AlphaZero for reinforcement learning on deep NN + theorem proving, nondeterministic programming — NeurIPS'22
- ✓ Safe waypoint following via NN path tracking and MPC + hybrid systems theorem proving — CSyL'22
- ✓ Control envelope synthesis via arithmetic simplification, refinement approximation + hybrid systems verification, game theory — TACAS'24
- ✓ NNCS via NN verification + complete arithmetic linearization, hybrid systems theorem proving — NeurIPS'24

---

1. Base technology: Dynamic logic proving — LICS'12
2. ✓ Base technology: ModelPlex shielding — FMSD'16
3. Widely different symbolic AI + subsymbolic AI combinations!

Intersymbolic AI Bestiarium

- Safe AI in CPS via reinforcement learning + hybrid systems theorem proving and proof-based synthesis AAAI'18
- Continuous invariant generation via symbolic proof, first integrals, Darboux + eigensystems, SOS & linear programming FMSD'22
- Loop invariant synthesis via AlphaZero for reinforcement learning on deep NN + theorem proving, nondeterministic programming NeurIPS'22
- Safe waypoint following via NN path tracking and MPC + hybrid systems theorem proving CSyL'22
- Control envelope synthesis via arithmetic simplification, refinement approximation + hybrid systems verification, game theory TACAS'24
- NNCS via NN verification + complete arithmetic linearization, hybrid systems theorem proving NeurIPS'24

✓ Base technology: Dynamic logic proving LICS'12

✓ Base technology: ModelPlex shielding FMSD'16

3. Widely different symbolic AI + subsymbolic AI combinations!

Outline

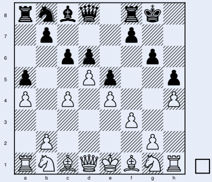
Logic

- AI by deductive proof
- Compositional meaning
- Clear interpretations
- Undeniable correctness
- Newell+Simon: Logic Theorist and General Problem Solver
- $(\text{rain} \to \text{wet}) \land \neg\text{wet} \to \neg\text{rain}$

- big SAT problems
- Java sort bugs
- FAA's ACAS X flight system
- 4 color, solvable odd order group, sphere packing

Games & Planning

- Actions have clear meaning
- Strategy composes actions



- Chinook wins checkers
- Deep Blue wins chess
- Checkers is solved

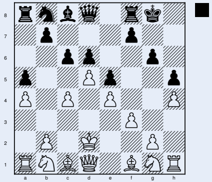
Logic

- AI by deductive proof
- Compositional meaning
- Clear interpretations
- Undeniable correctness
- Newell+Simon: Logic Theorist and General Problem Solver
- $(\mathrm{rain} \to \mathrm{wet}) \land \neg\mathrm{wet} \to \neg\mathrm{rain}$

- big SAT problems
- Java sort bugs
- FAA's ACAS X flight system
- 4 color, solvable odd order group, sphere packing

Games & Planning

- Actions have clear meaning
- Strategy composes actions



- Chinook wins checkers
- Deep Blue wins chess
- Checkers is solved

Outline

Subsymbolic AI

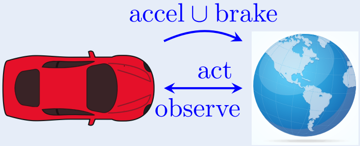
Neural Network

- nonlinear activation $f^{(n)} : \mathbb{R} \to \mathbb{R}$
- $x^{(n+1)} = f^{(n)}\left(A^{(n)} x^{(n)} + b^{(n)}\right)$
- optimize matrix $A^{(n)}$, vector $b^{(n)}$ $\frac{\partial}{\partial A^{(n)}_{i,j}}$ to fit expected $x^{(n+1)}$, $x^{(n)}$
- individual $A^{(n)}_{i,j}$ meaning unclear
- representative training data: lots
- distribution shifts: unpredictable

- image classification
- Generative adversarial networks
- LLM

Reinforcement Learning

- bridging ML & game theory
- learn to act from experience
- reinforce success / policy
- sample inefficient



- AlphaZero MCTS in DNN value
- for Chess, Shōgi, Go ...
- undecidable for continuous case

Outline

Intersymbolic AI = Symbolic AI + Subsymbolic AI

Symbolic AI

- + accurate & precise
- + self-explanatory
- - manual formalization
- + clear correctness
- - scaling hard

Subsymbolic AI

- + no explicit programming
- - inexplicable
- - data inefficiency
- - no* clear correctness
- + computational scaling

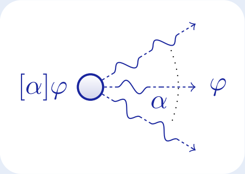
Intersymbolic AI = Symbolic AI + Subsymbolic AI

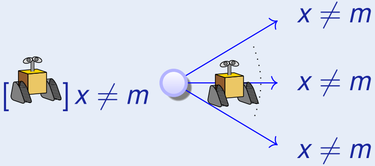
Intersymbolic AI interlinks symbolic AI (compositional significance/meaning) with subsymbolic AI (summative significance/effect) to combine insights from both worlds by going between and across symbolic AI insights with subsymbolic AI helped by symbolic AI principles.

Analogy: Human Thought = Conscious Thought + Subconscious Thought

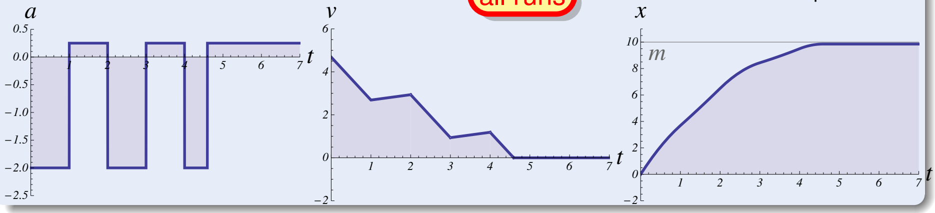
Outline

Concept (Differential Dynamic Logic) (JAR'08,LICS'12)





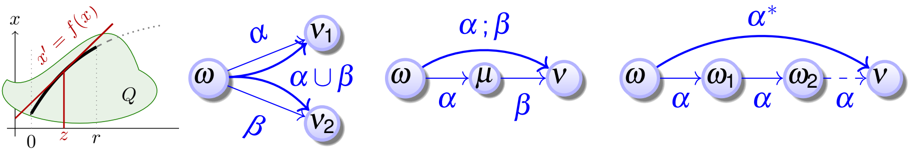
$$\underbrace{x \neq m \land b > 0}_{\text{init}} \rightarrow \Big[\underbrace{\big((\text{if}(SB(x,m))\quad a := -b)\ ;\ x' = v, v' = a\big)^*}_{\text{all runs}}\Big]\underbrace{x \neq m}_{\text{post}}$$



Differential Dynamic Logic dL

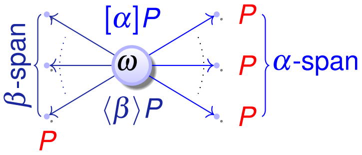
Definition (Hybrid program)

$$\alpha, \beta \ ::= \ x := e \mid ?Q \mid x' = f(x) \,\&\, Q \mid \alpha \cup \beta \mid \alpha ; \beta \mid \alpha^*$$



Definition (Differential dynamic logic) (JAR'08,LICS'12)

$$P, Q ::= e \geq \tilde{e} \mid \neg P \mid P \wedge Q \mid P \vee Q \mid P \rightarrow Q \mid \forall x\, P \mid \exists x\, P \mid [\alpha]P \mid \langle \alpha \rangle P$$



Differential Dynamic Logic dL: Axiomatization

equations of truth

$[:=]\ [x := e]P(x) \leftrightarrow P(e)$

$[?]\ [?Q]P \leftrightarrow (Q \rightarrow P)$

$[']\ [x' = f(x)]P \leftrightarrow \forall t{\geq}0\, [x := y(t)]P \qquad (y'(t) = f(y))$

$[\cup]\ [\alpha \cup \beta]P \leftrightarrow [\alpha]P \wedge [\beta]P$

$[;]\ [\alpha; \beta]P \leftrightarrow [\alpha][\beta]P$

$[^*]\ [\alpha^*]P \leftrightarrow P \wedge [\alpha][\alpha^*]P$

$\mathrm{K}\ [\alpha](P \rightarrow Q) \rightarrow ([\alpha]P \rightarrow [\alpha]Q)$

$\mathrm{I}\ [\alpha^*]P \leftrightarrow P \wedge [\alpha^*](P \rightarrow [\alpha]P)$

$\mathrm{C}\ [\alpha^*]\forall v{>}0\, (P(v) \rightarrow \langle\alpha\rangle P(v-1)) \rightarrow \forall v\, (P(v) \rightarrow \langle\alpha^*\rangle \exists v{\leq}0\, P(v))$

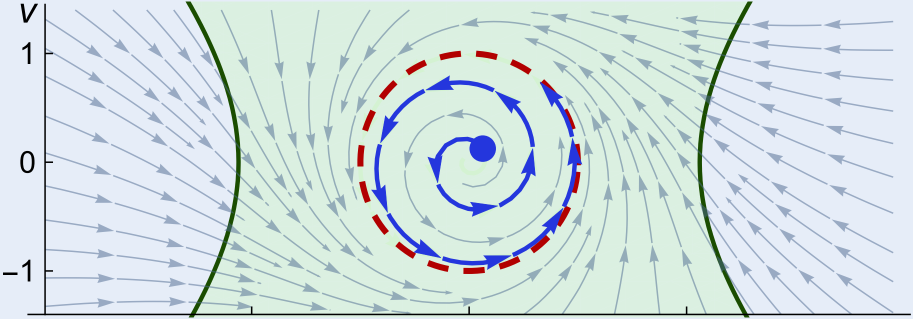
laws of logic of
laws of physics

LICS'12,JAR'17

Concept (Differential Dynamic Logic) (JAR'08,LICS'12,JACM'20)

$$u^2 \leq v^2 + \frac{9}{2} \rightarrow [u' = -v + \frac{u}{4}(1-u^2-v^2), v' = u + \frac{v}{4}(1-u^2-v^2)]\, u^2 \leq v^2 + \frac{9}{2}$$

$$u^2 + v^2 = 1 \rightarrow [u' = -v + \frac{u}{4}(1-u^2-v^2), v' = u + \frac{v}{4}(1-u^2-v^2)]\, u^2 + v^2 = 1$$

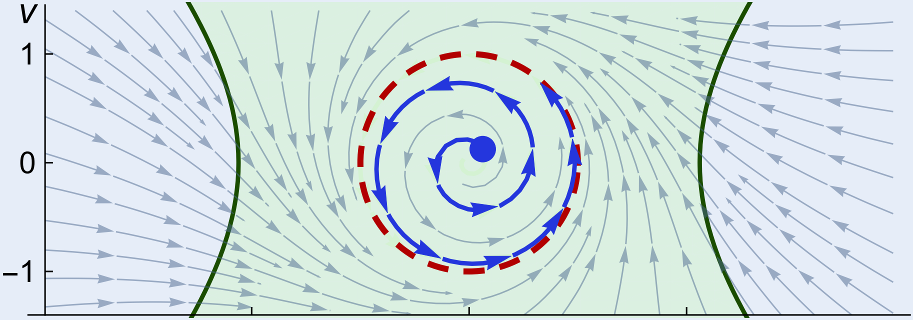


Analyzing ODEs via solutions undoes their descriptive power! Poincaré 1881

Concept (Differential Dynamic Logic) (JAR'08,LICS'12,JACM'20)

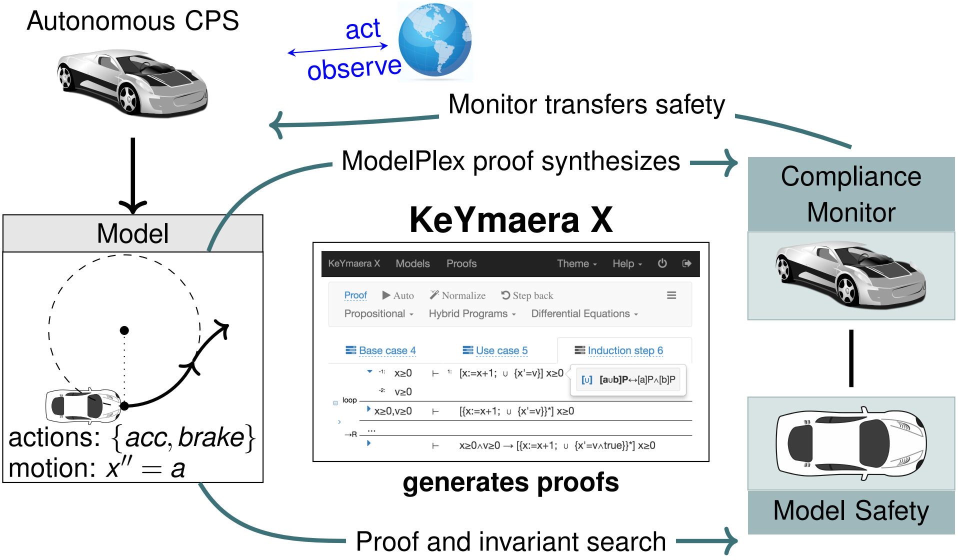
All true algebraic invariants are provable in dL with $e'^{*} = 0 \equiv e{=}0 \land (e')'^{*}{=}0$:

$$\text{DRI} \quad [x' = f(x)\,\&\,Q]e = 0 \leftrightarrow \big(Q \rightarrow e'^{*} = 0\big) \qquad \text{(e.g., } Q \text{ open)}$$

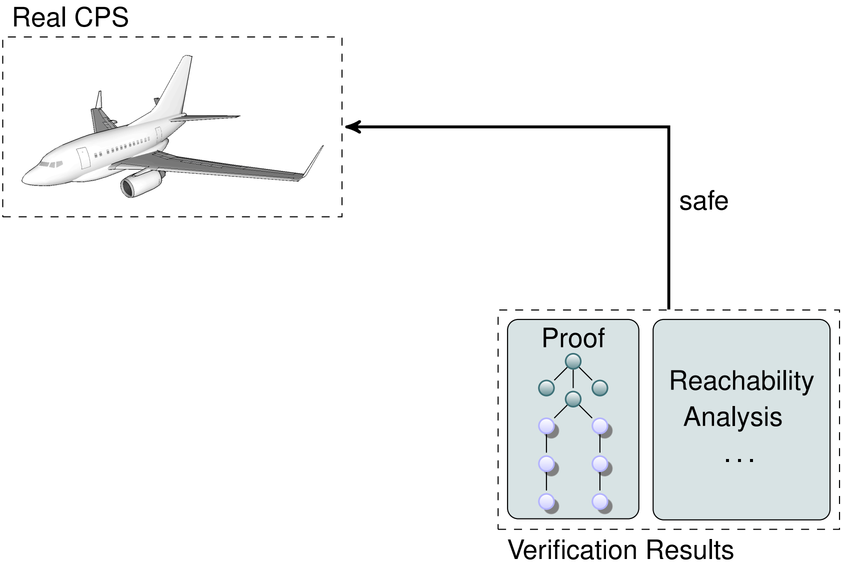


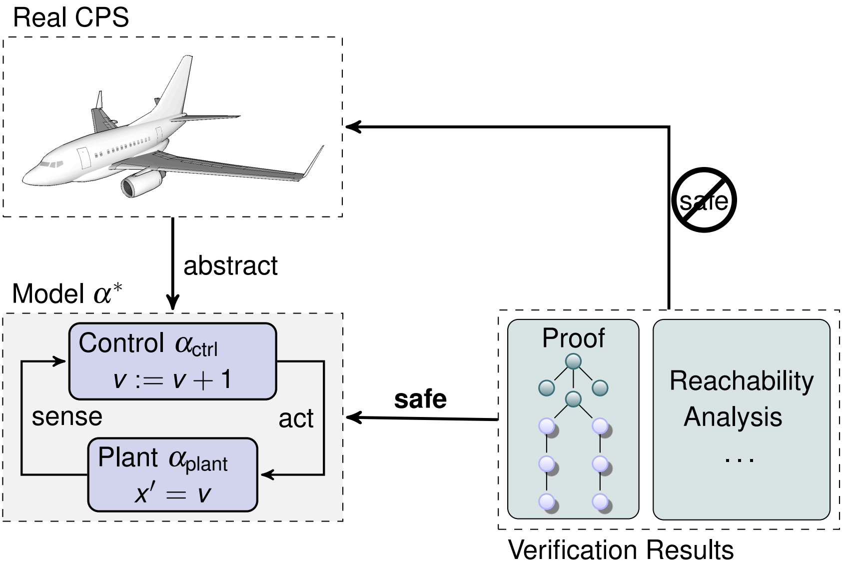
Analyzing ODEs via solutions undoes their descriptive power! Poincaré 1881

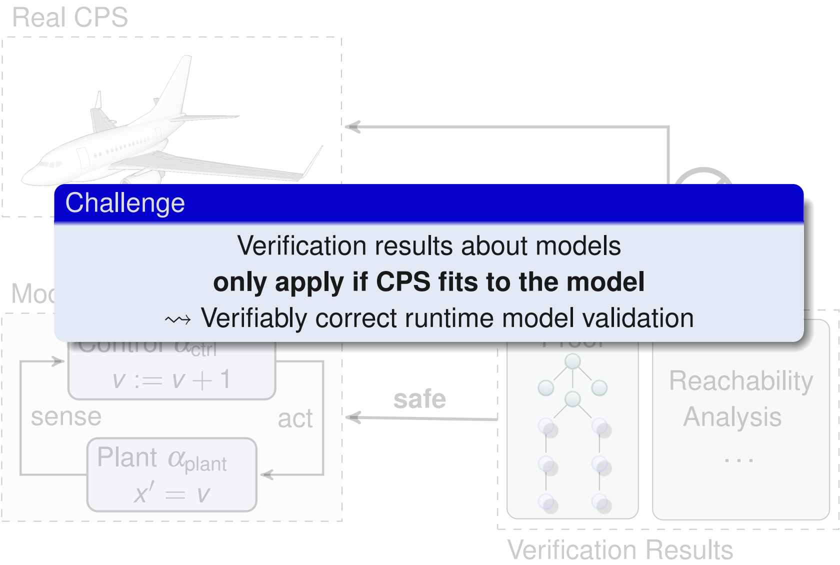
Approach: Safety Proofs for Autonomous CPS



Formal Verification in CPS Development




Real CPS
abstract
Model α^*
Control α_{ctrl}
$v := v + 1$
sense
act
Plant α_{plant}
$x' = v$
safe
safe
Proof
Reachability Analysis
...
Verification Results



Challenge

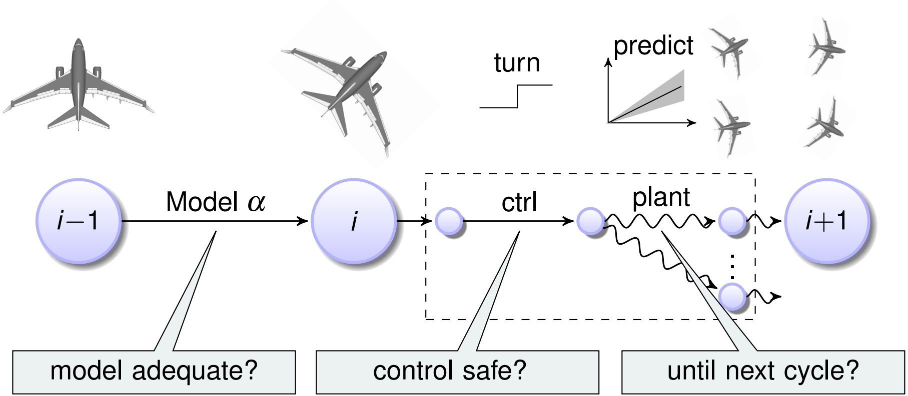
Verification results about models

only apply if CPS fits to the model

$\rightsquigarrow$ Verifiably correct runtime model validation

ModelPlex Runtime Model Validation

ModelPlex **ensures that verification results** about models **apply to CPS** implementations



RV'14,FMSD'16

ModelPlex Runtime Model Validation

ModelPlex **ensures that verification results** about models **apply to CPS** implementations

Insights

- Verification results about models transfer to the CPS when validating model compliance.
- Compliance with model is characterizable in logic dL.
- Compliance formula transformed by dL proof to monitor.
- Correct-by-construction provably correct model validation at runtime.

model adequate?

control safe?

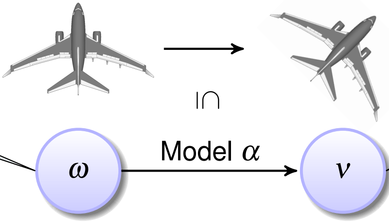
until next cycle?

RV'14,FMSD'16

Characterizing State Relations in Logic

When are two states linked through a run of model α?

a prior state characterized by x^-



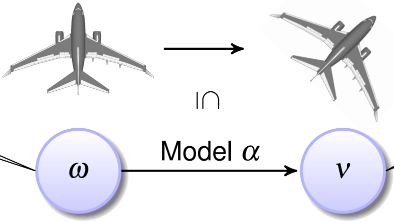
a posterior state characterized by x^+

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$ — reachability relation of α

RV'14,FMSD'16

When are two states linked through a run of model α?

a prior state characterized by x^-



a posterior state characterized by x^+

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dL: $(\omega, \nu) \models \langle \alpha \rangle (x = x^+)$

exists a run of α to a state where $x = x^+$

RV'14,FMSD'16

When are two states linked through a run of model α?

$\longrightarrow$

$\Cap$

a prior state characterized by x^-

$\omega \xrightarrow{\text{Model } \alpha} \nu$

a posterior state characterized by x^+

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dL: $(\omega, \nu) \models \langle \alpha \rangle (x = x^+)$

exists a run of α to a state where $x = x^+$

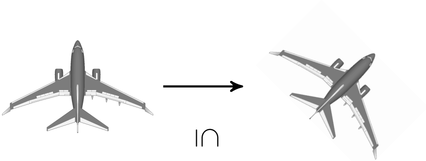
$\Updownarrow$ dL proof

Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

Logic reduces CPS safety to runtime monitor with offline proof



$\in$

dL proof $A \to [\alpha]S$

$\omega \xrightarrow{\text{Model } \alpha} \nu$

Init $\omega \in [\![A]\!]$ Safe $\nu \in [\![S]\!]$

Offline

Semantical: $(\omega, \nu) \in [\![\alpha]\!]$

$\Updownarrow$ Lemma

Logical dL: $(\omega, \nu) \models \langle\alpha\rangle(x = x^+)$

$\Uparrow$ dL proof

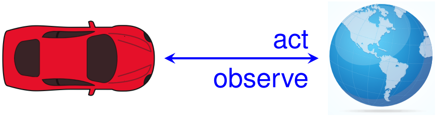
Arithmetical: $(\omega, \nu) \models F(x^-, x^+)$

check at runtime (efficient)

RV'14,FMSD'16

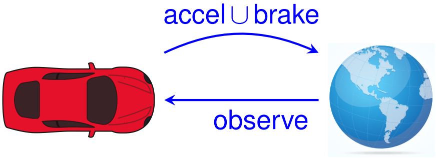
Outline

Learning to Act in a CPS



Reinforcement Learning learns from experience of trying actions

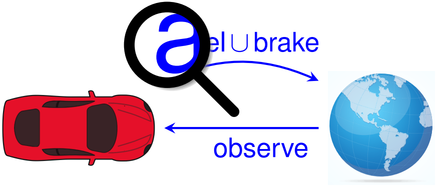
AAAI'18,ITC'18,TACAS'19,QEST'19



RL chooses an action, observes outcome, reinforces in policy if successful

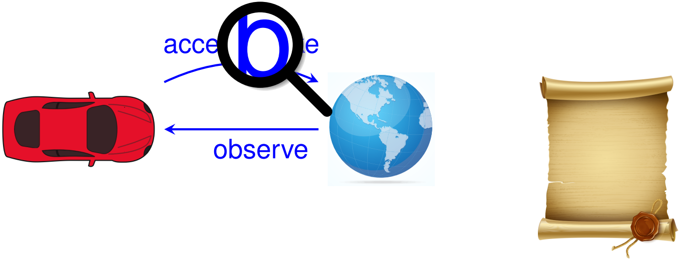
AAAI'18,ITC'18,TACAS'19,QEST'19

Learning to Act Safely in a CPS



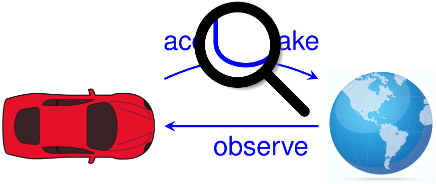
ModelPlex monitor inspects each decision, vetoes if unsafe

AAAI'18,ITC'18,TACAS'19,QEST'19



ModelPlex monitor gives early feedback about possible future problems.
No need to wait till disaster strikes and propagate back.

AAAI'18,ITC'18,TACAS'19,QEST'19



dL benefits from RL optimization.

RL benefits from dL safety signal.

AAAI'18,ITC'18,TACAS'19,QEST'19

Open-Loop NNV

*For **close** intruder approaching from left, advise "strong right".*

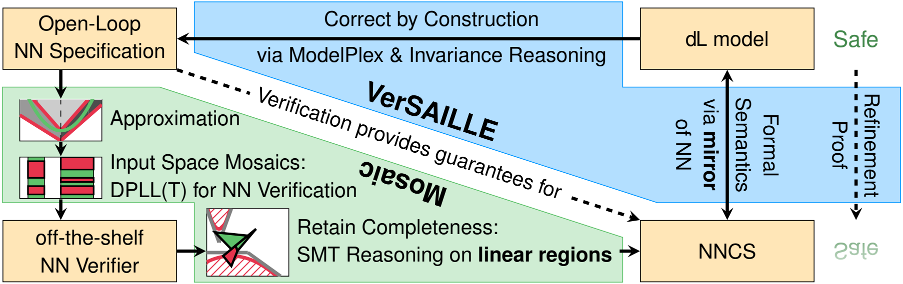
Ignores Feedback-Loop

Provably Safe Neural Network Controllers

Closed-Loop NNV

*Check whether there is a collision in the following **5s***

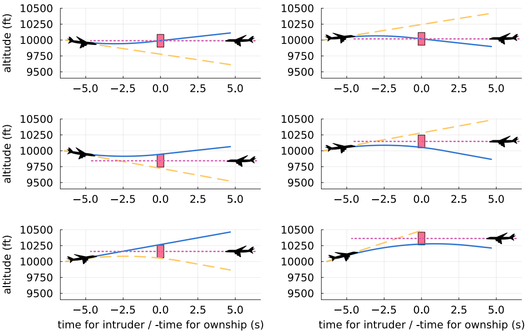
Insufficient Guarantees



NeurIPS'24

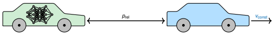
Airborne Collision Avoidance

(6×45 ReLU NNs)

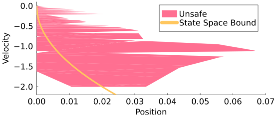


Adaptive Cruise Control

(4×64 ReLU NN)



- Training & verification
- Fallback for $\text{pos} \leq 0.1$



Result

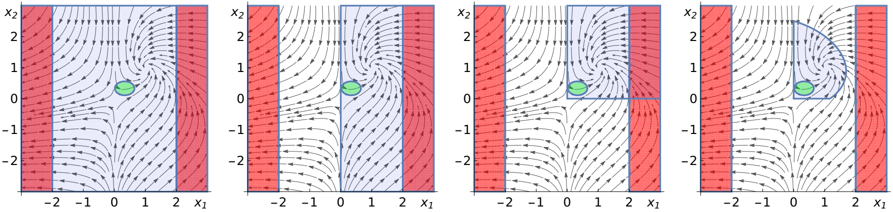
- 6/8 scenarios **unsafe**
- Other scenarios safe for **level intruder** (but crashes found for non-level intruder)

Result

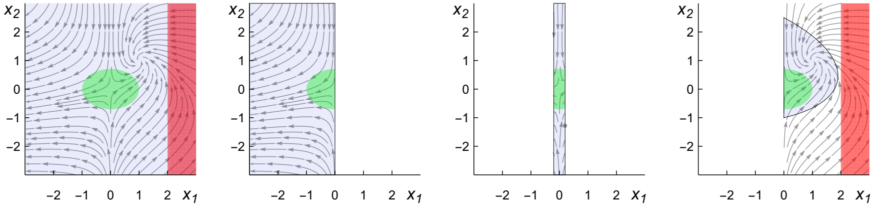
Using NN (+ fallback) ensures that two cars will **never** crash

NeurIPS'24

Differential Saturation: Refine candidate invariant until saturation



Differential Divide&Conquer: Divide space by invariant $p = 0$, and separately find invariants

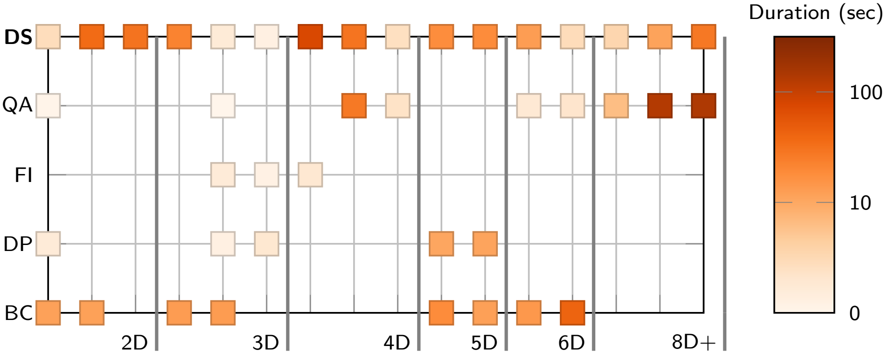


FM'19,FMSD'22

107/150 continuous safety verification benchmarks solved automatically

2–16 dim (non-)linear ODEs of varying syntactic complexity, topology

Differential saturation crucial for combining invariant generation primitives



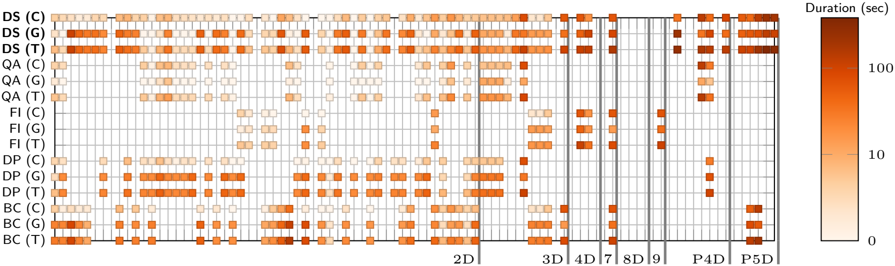
FM'19,FMSD'22

Pegasus: Experiments

107/150 continuous safety verification benchmarks solved automatically

2–16 dim (non-)linear ODEs of varying syntactic complexity, topology

Differential saturation crucial for combining invariant generation primitives

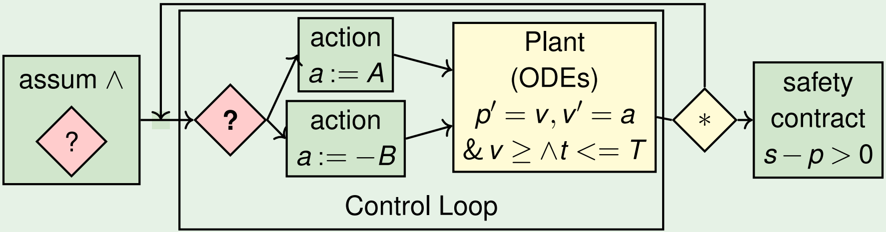


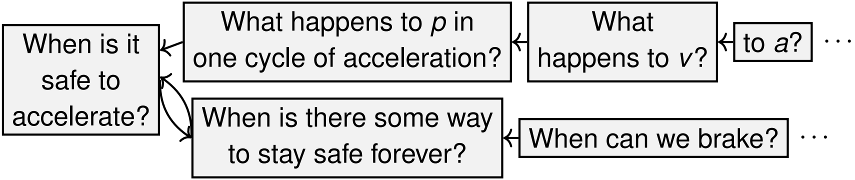
FM'19,FMSD'22

Control Envelope Synthesis via Angelic Refinements

Synthesize *all* safe control solutions (avoid one Überproperty). System has *interdependent control conditions* in a *nondeterministic*, *hybrid* environment.

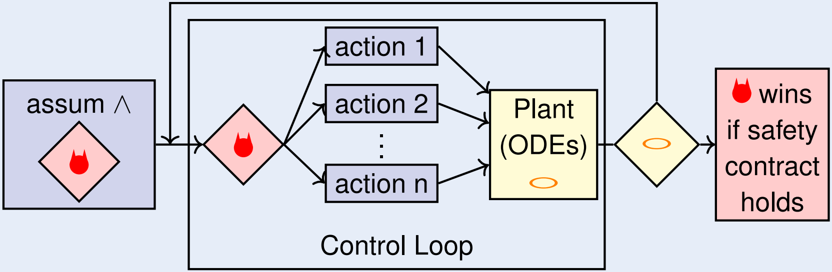
Example: When can a train accelerate while stopping before signal s?





TACAS'24

1. CESAR characterizes optimal solution in differential game logic.



invariant $I^0 \equiv \left[(\cup_i\, act_i; \mathrm{plant})^*\right]$ safe $\qquad$ guard $G_i^0 \equiv [act_i; \mathrm{plant}]\, I^0$.

2. CESAR obtains explicit solution formulas.

To the hybrid games solution characterization, apply *systematic refinements*. Then *symbolically execute* per the axioms of dGL.

TACAS'24

Outline

Intersymbolic AI Bestiarium

- Safe AI in CPS via reinforcement learning + hybrid systems theorem proving and proof-based synthesis AAAI'18
- Continuous invariant generation via symbolic proof, first integrals, Darboux + eigensystems, SOS & linear programming FMSD'22
- Loop invariant synthesis via AlphaZero for reinforcement learning on deep NN + theorem proving, nondeterministic programming NeurIPS'22
- Safe waypoint following via NN path tracking and MPC + hybrid systems theorem proving CSyL'22
- Control envelope synthesis via arithmetic simplification, refinement approximation + hybrid systems verification, game theory TACAS'24
- NNCS via NN verification + complete arithmetic linearization, hybrid systems theorem proving NeurIPS'24

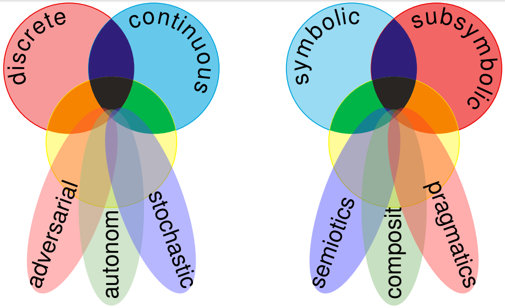
✓ Base technology: Dynamic logic proving LICS'12

✓ Base technology: ModelPlex shielding FMSD'16

3. Widely different symbolic AI + subsymbolic AI combinations!

Intersymbolic AI = Symbolic AI + Subsymbolic AI

Intersymbolic AI interlinks symbolic AI (compositional significance/meaning) with subsymbolic AI (summative significance/effect) to combine insights.



- Characterize complementary symbolic vs. subsymbolic AI
- “Conscious + subconscious thought”
- Many different flavors of technology combinations in intersymbolic AI
- Scientific diversity with common metaprinciple: Intersymbolic AI
- Future work abounds for equally wide range of scientists

André Platzer.
Intersymbolic AI: Interlinking Symbolic AI and Subsymbolic AI.
In Bernhard Steffen, editor, *ISoLA 2024*, volume 15222 of *LNCS*, pages 162–180. Springer, 2024.
doi:10.1007/978-3-031-75387-9_11.

André Platzer.
Logical Foundations of Cyber-Physical Systems.
Springer, Cham, 2018.
doi:10.1007/978-3-319-63588-0.

André Platzer.
Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics.
Springer, Heidelberg, 2010.
doi:10.1007/978-3-642-14509-4.

André Platzer.
Differential dynamic logic for hybrid systems.
J. Autom. Reas., 41(2):143–189, 2008.
doi:10.1007/s10817-008-9103-8.

André Platzer.
A complete uniform substitution calculus for differential dynamic logic.
J. Autom. Reas., 59(2):219–265, 2017.
doi:10.1007/s10817-016-9385-1.

André Platzer.
Logics of dynamical systems.
In LICS [20], pages 13–24.
doi:10.1109/LICS.2012.13.

Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
Form. Methods Syst. Des., 49(1-2):33–74, 2016.
Special issue of selected papers from RV'14.
doi:10.1007/s10703-016-0241-z.

André Platzer and Yong Kiam Tan.
Differential equation invariance axiomatization.
J. ACM, 67(1):6:1–6:66, 2020.
doi:10.1145/3380825.

Nathan Fulton and André Platzer.
Safe reinforcement learning via formal methods: Toward safe control through proof and learning.
In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*, pages 6485–6492. AAAI Press, 2018.

Andrew Sogokon, Stefan Mitsch, Yong Kiam Tan, Katherine Cordwell, and André Platzer.
Pegasus: Sound continuous invariant generation.
Form. Methods Syst. Des., 58(1):5–41, 2022.
Special issue for selected papers from FM'19.
`doi:10.1007/s10703-020-00355-z`.

Jonathan Laurent and André Platzer.
Learning to find proofs and theorems by learning to refine search strategies.
In Sanmi Koyejo, Shakir Mohamed, Alekh Agarwal, Danielle Belgrave, Kyunghyun Cho, and Alice Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, page 4843–4856. Curran Associates, Inc., 2022.

Qin Lin, Stefan Mitsch, André Platzer, and John M. Dolan.
Safe and resilient practical waypoint-following for autonomous vehicles.
IEEE Control. Syst. Lett., 6:1574–1579, 2022.
doi:10.1109/LCSYS.2021.3125717.

Aditi Kabra, Jonathan Laurent, Stefan Mitsch, and André Platzer.
CESAR: Control envelope synthesis via angelic refinements.
In Bernd Finkbeiner and Laura Kovács, editors, *TACAS*, volume 14570 of *LNCS*, pages 144–164. Springer, 2024.
doi:10.1007/978-3-031-57246-3_9.

Samuel Teuber, Stefan Mitsch, and André Platzer.
Provably safe neural network controllers via differential dynamic logic.
CoRR, abs/2402.10998, 2024.
arXiv:2402.10998.

André Platzer.
The complete proof theory of hybrid systems.
In LICS [20], pages 541–550.
doi:10.1109/LICS.2012.64.

Stefan Mitsch and André Platzer.
ModelPlex: Verified runtime validation of verified cyber-physical system models.
In Borzoo Bonakdarpour and Scott A. Smolka, editors, *RV*, volume 8734 of *LNCS*, pages 199–214. Springer, 2014.
doi:10.1007/978-3-319-11164-3_17.

Nathan Fulton and André Platzer.
Verifiably safe off-model reinforcement learning.
In Tomas Vojnar and Lijun Zhang, editors, *TACAS, Part I*, volume 11427 of *LNCS*, pages 413–430. Springer, 2019.
doi:10.1007/978-3-030-17462-0_28.

Samuel Teuber, Stefan Mitsch, and André Platzer.
Provably safe neural network controllers via differential dynamic logic.
In A. Globerson, L. Mackey, A. Fan, C. Zhang, D. Belgrave, J. Tomczak, and U. Paquet, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2024.

Andrew Sogokon, Stefan Mitsch, Yong Kiam Tan, Katherine Cordwell, and André Platzer.

Pegasus: A framework for sound continuous invariant generation.
In Maurice ter Beek, Annabelle McIver, and José N. Oliviera, editors, *FM*, volume 11800 of *LNCS*, pages 138–157. Springer, 2019.
doi:10.1007/978-3-030-30942-8_10.

Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on, Los Alamitos, 2012. IEEE.