ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models

Stefan Mitsch  André Platzer

Computer Science Department, Carnegie Mellon University

RV’14, Sept. 24, 2014

Simplex for Hybrid System Models
Formal Verification in CPS Development

Real CPS

Proof

Reachability Analysis

Verification Results

safe

Verification results about models only apply if CPS fits to the model $\Rightarrow$ Verifiably correct runtime model validation
Formal Verification in CPS Development

Real CPS

Model $\alpha^*$

Control $\alpha_{\text{ctrl}}$

\[ v := v + 1 \]

Plant $\alpha_{\text{plant}}$

\[ x' = v \]

Verification Results

Proof

Reachability Analysis

...
Formal Verification in CPS Development

Real CPS

Model

Challenge

Verification results about models

only apply if CPS fits to the model

⇝ Verifiably correct runtime model validation

\[ \begin{align*}
\alpha_{\text{plant}} & : \text{Plant} \\
\alpha_{\text{ctrl}} & : \text{Control} \\
v := v + 1 & : \text{sense} \\
x' &= v & : \text{act}
\end{align*} \]
ModelPlex ensures that verification results about models apply to CPS implementations.

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

Contributions

Verification results transfer to CPS when validating model compliance

Compliance with model is characterizable in logic

Compliance formula transformed by proof to executable monitor
ModelPlex ensures that verification results about models apply to CPS implementations.

Contributions

- Verification results transfer to CPS when validating model compliance
- Compliance with model is characterizable in logic
- Compliance formula transformed by proof to executable monitor

Model adequate?  
Control safe?  
Until next cycle?
ModelPlex at Runtime

Stefan Mitsch, André Platzer—ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models
ModelPlex at Runtime

Compliance Monitor  Checks CPS for compliance with model at runtime
  - Model Monitor: model adequate?
  - Controller Monitor: control safe?
  - Prediction Monitor: until next cycle?

Fallback  Safe action, executed when monitor is not satisfied

Challenge  What conditions do the monitors need to check to be safe?
Is current CPS behavior included in the behavior of the model?

- CPS observed through sensors
- Model describes behavior of CPS between states

Detect non-compliance as soon as possible to initiate safe fallback actions
ModelPlex Approach

Is current CPS behavior included in the behavior of the model?

- CPS observed through sensors
- Model describes behavior of CPS between states

Detect non-compliance as soon as possible to initiate safe fallback actions
Is current CPS behavior included in the behavior of the model?

- CPS observed through sensors
- Model describes behavior of CPS between states

\[ \text{CPS observed through sensors} \quad \text{Model describes behavior of CPS between states} \]

\[ \text{Model} \models \alpha \subseteq \text{fits to} \quad \text{CPS} \]

Detect non-compliance as soon as possible to initiate safe fallback actions
When are two states linked through a run of model $\alpha$?
When are two states linked through a run of model $\alpha$?

A prior state characterized by $x^-$

$\{i-1\}$

Model $\alpha$

A posterior state characterized by $x^+$

$\{i\}$

Semantical: $(x^-, x^+) \in \rho(\alpha)$

reachability relation of $\alpha$
When are two states linked through a run of model $\alpha$?

**Offline**

Semantical: $(x^-, x^+) \in \rho(\alpha) 
\iff \text{Theorem}$

Logic ($d\mathcal{L}$): $(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)$

starting at $x = x^-$
exists a run of $\alpha$ to a
state where $x = x^+$
When are two states linked through a run of model $\alpha$?

- **Semantical:** $(x^-, x^+) \in \rho(\alpha) \iff$ Theorem
- **Logic (dL):** $(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)$ \iff dL proof
- **Real arithmetic:** $F(x^-, x^+) \iff$ check at runtime (efficient)

**Offline**

**a prior state characterized by** $x^-$

$\cap$

**Model** $\alpha$

$\Rightarrow$

**a posterior state characterized by** $x^+$

**Starting at** $x = x^-$

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

---

**Stefan Mitsch, André Platzer—ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models**
When are two states linked through a run of model $\alpha$?

- **Offline**
  - **Semantical**: $(x^-, x^+) \in \rho(\alpha)$
  - **Logic (dL)**: $(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)$
  - **Real arithmetic**: $F(x^-, x^+)$

- **Theorem**: starting at $x = x^-$ exists a run of $\alpha$ to a state where $x = x^+$

- **check at runtime (efficient)**

**a prior state characterized by $x^-$**

**Model $\alpha$**

**a posterior state characterized by $x^+$**
Proof calculus of dL executes models symbolically.

Proof attempt:

\[(x = x^-) \rightarrow \langle \alpha(x) \rangle (x = x^+)\]
Proof calculus of $d\mathcal{L}$ executes models symbolically

Proof attempt:

$(x = x^-) \rightarrow \langle \text{climb} \cup \text{descend} \rangle (x = x^+)$

Monitor:

The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model $\Rightarrow$ execute at runtime

Model $\alpha$

Immediate detection of model violation $\Rightarrow$ Mitigates safety issues with safe fallback action
Proof calculus of $d\mathcal{L}$ executes models symbolically

Proof attempt

$(x = x^-) \rightarrow \langle\text{climb} \cup \text{descend}\rangle (x = x^+)$

$\langle\text{climb}\rangle (x = x^+) \lor \langle\text{descend}\rangle (x = x^+)$
Proof calculus of $d\mathcal{L}$ executes models symbolically.

Model $\alpha$

prior state $x^-$ $\rightarrow$ $\rightarrow$ $\rightarrow$ $i-1$ climb $\rightarrow$ $\rightarrow$ $\rightarrow$ $i$ posterior state $x^+$

descend

proof attempt

$(x = x^-) \rightarrow \langle \text{climb} \cup \text{descend} \rangle (x = x^+) \langle \text{climb} \rangle (x = x^+)$ $\forall$ $\langle \text{descend} \rangle (x = x^+)$

$F_1 (x^-, x^+)$ $\forall$ $F_2 (x^-, x^+)$
Proof calculus of $\mathcal{dL}$ executes models symbolically

$$\langle \text{climb} \cup \text{descend} \rangle (x = x^+)$$

$$\langle \text{climb} \rangle (x = x^+) \lor \langle \text{descend} \rangle (x = x^+)$$

$$F_1(x^-, x^+) \lor F_2(x^-, x^+)$$

Monitor: $F_1(x^-, x^+) \lor F_2(x^-, x^+)$

The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model $\Rightarrow$ execute at runtime
Proof calculus of $\mathcal{dL}$ executes models symbolically

Proof attempt:

$$\langle \alpha(x) \rangle (x = x^+) \rightarrow \langle \uparrow \rangle \phi \lor \langle \uparrow \rangle \phi \lor \langle \downarrow \rangle \phi$$

$$\bigvee F_1(x^-, x^+)$$

Monitor: $\bigvee F_1(x^-, x^+) \lor F_2(x^-, x^+)$

The subgoals that cannot be proved express all the conditions on the relations of variables imposed by the model $\rightsquigarrow$ execute at runtime
For typical models $\text{ctrl; plant}$ we can check earlier.
Early Compliance Checks for Controllers

Model \( \alpha \)

Model Monitor

prior state \( x^{-} \) → \( i-1 \) → Model \( \alpha \) → \( i \) → posterior state \( x^{+} \)

Semantical: reachability relation of \( \text{ctrl} \)

Theorem: \( x = x^{-} \) → \( \langle \text{ctrl}(x) \rangle \) \( \langle x = x^{+} \rangle \)

Logic (\( d_L \) proof): starting at \( x = x^{-} \) exists a run of \( \text{ctrl} \) to a state where \( x = x^{+} \)

Real arithmetic: Immediate detection of unsafe control before actuation → Safe execution of unverified implementations in perfect environments

Stefan Mitsch, André Platzer—ModelPlex: Verified Runtime Validation of Verified Cyber-Physical System Models
Early Compliance Checks for Controllers

Model $\alpha$ offline

prior state $x^-\rightarrow_{i-1}$

Controller Monitor before actuation
posterior state $x^+\rightarrow_i$

Semantical: $(x^-, x^+) \in \rho(\text{ctrl}) \iff \text{reachability relation of ctrl}$
Early Compliance Checks for Controllers

Prior state $x^{-}$

Offline

Semantical: $(x^{-}, x^{+}) \in \rho(\text{ctrl})$

Logic ($dL$): $(x = x^{-}) \rightarrow \langle \text{ctrl}(x) \rangle (x = x^{+})$

Controller Monitor before actuation

Posterior state $x^{+}$

starting at $x = x^{-}$

exists a run of ctrl to a state where $x = x^{+}$
Early Compliance Checks for Controllers

Model $\alpha$

Controller Monitor before actuation

prior state $x^-$

$i-1$

ctrl

plant

posterior state $x^+$

$i$

Semantical: $(x^-, x^+) \in \rho(\text{ctrl})$

$
\iff$

Theorem

Logic ($d\mathcal{L}$): $(x = x^-) \Rightarrow (\langle \text{ctrl}_{(x)} \rangle (x = x^+))$

Real arithmetic: $F(x^-, x^+)$

starting at $x = x^-$ exists a run of $\text{ctrl}$ to a state where $x = x^+$
Early Compliance Checks for Controllers

### Controller Monitor

**Immediat detection of unsafe control before actuation**

\[ \implies \text{Safe execution of unverified implementations in perfect environments} \]

---

**Semantical:**

\[ (x^-, x^+) \in \rho(\text{ctrl}) \]

\[ \Downarrow \text{Theorem} \]

**Logic** (\(dL\)):

\[ (x = x^-) \implies (\text{ctrl}(x) \, (x = x^+)) \]

\[ \Downarrow dL \text{ proof} \]

**Real arithmetic:**

\[ F(x^-, x^+) \]

---
Safe despite evolution with disturbance?
Compliance Checks despite Disturbance

Prior state $x^{-i-1}$ → $x^{+i}$

Model $\alpha$

ctrl → plant

Prediction Monitor before actuation

Posterior state $x^{+i}$

Offline ($x^{-i-1}$ → $ctrl(\alpha)(x^{-i-1}$) ∨ $plant(\alpha)$ $\phi$)

Invariant state $\phi$ implies safety (known from safety proof)

Logic ($\mathcal{DL}$):

$\downarrow \mathcal{DL}$ proof $F(x^{-i-1}, x^{+i})$

Real arithmetic:

Proactive detection of unsafe control before actuation despite disturbance

Safety in realistic environments
Compliance Checks despite Disturbance

plant of the form \( x' = \theta & H \)

Prior state \( x^- \)  \( i \) .ctrl  \( i \)  plant

Prediction Monitor
before actuation
posterior state \( x^+ \)

Model \( \alpha \)
Compliance Checks despite Disturbance

time bound $t := 0; \left( x' = \theta, \ t' = 1 & H \land t \leq \varepsilon \right)$

Model $\alpha$

prior state $x_{i-1}$\hspace{10cm} ctrl \hspace{10cm} plant\hspace{10cm} posterior state $x_i$

Prediction Monitor before actuation posterior state $x^+$

states reachable within $\varepsilon$ time

logic (\text{dL}): \overline{\overline{dL_{\text{proof}}F(x_{i-1}, x_i)}}

Real arithmetic:

Prediction Monitor with Disturbance
Proactive detection of unsafe control before actuation despite disturbance

Safety in realistic environments
Compliance Checks despite Disturbance

disturbance \( t := 0; \left( \theta - \delta \leq x' \leq \theta + \delta, \ t' = 1 & H \land t \leq \varepsilon \right) \)

Prior state \( x^- \) \( i-1 \) \( \rightarrow \) Ctrl \( \rightarrow \) Plant \( \oplus \) \( i \) \( \rightarrow \) Posterior state \( x^+ \)

Prediction Monitor before actuation

States reachable within \( \varepsilon \) time
Compliance Checks despite Disturbance

disturbance $t := 0; \left( \theta - \delta \leq x' \leq \theta + \delta, \ t' = 1 & H \land t \leq \varepsilon \right)$

prior state $x^{-}$  $\rightarrow$  Model $\alpha$  $\rightarrow$  post. state $x^{+}$

Prediction Monitor before actuation

states reachable within $\varepsilon$ time

Offline

Logic ($d\mathcal{L}$): $(x = x^{-}) \rightarrow \langle ctrl(x) \rangle \left( x = x^{+} \land [plant(x)] \varphi \right)$

Real arithmetic: $F(x^{-}, x^{+})$

Invariant state $\varphi$ implies safety
(known from safety proof)
Compliance Checks despite Disturbance

\[
\text{disturbance } t := 0; \left( \theta - \delta \leq x' \leq \theta + \delta, \ t' = 1 & H \land t \leq \varepsilon \right)
\]

**Model** $\alpha$

Prior state $x_{i-1}$ + $\text{ctrl}$ + $\text{plant}$

**Prediction Monitor with Disturbance**

Proactive detection of unsafe control before actuation despite disturbance

\[ \rightsquigarrow \text{Safety in realistic environments} \]

**Offline**

Logic ($d\mathcal{L}$): \( (x = x^-) \rightarrow \langle \text{ctrl}(x) \rangle \left( x = x^+ \land [\text{plant}(x)] \varphi \right) \)

\[ \uparrow d\mathcal{L} \text{ proof} \]

Real arithmetic:

\[ F(x^-, x^+) \]

Invariant state $\varphi$ implies safety (known from safety proof)
Evaluation

- Evaluated on hybrid system case studies

Water tank  Cruise control  Traffic control  Ground robots  Train control

- Model sizes: 5–16 variables
- Monitor sizes: 20–150 operations
  - with automated simplification to remove redundant checks
  - improvement potential: simplification for any monitor

**Theorem:** ModelPlex is decidable and monitor synthesis fully automated in important classes
ModelPlex ensures that proofs apply to real CPS

- Validate model compliance
- Characterize compliance with model in logic
- Prover transforms compliance formula to executable monitor
Stefan Mitsch
smitsch@cs.cmu.edu
www.cs.cmu.edu/~smitsch
Theorems

- State Recall (Online Monitoring)
- Model Monitor Correctness
- Controller Monitor Correctness
- Prediction Monitor Correctness
- Decidability and Computability
State Recall

V set of variables whose state we want to recall

\( \gamma_\mathbf{V}^- \equiv \wedge_{x \in \mathbf{V}} x = x^- \) characterizes a state prior to a run of \( \alpha \) (fresh variables \( x^- \) occur solely in \( \gamma_\mathbf{V}^- \) and recall this state)

\( \gamma_\mathbf{V}^+ \equiv \wedge_{x \in \mathbf{V}} x = x^+ \) characterizes the posterior states (fresh \( x^+ \))

Programs hybrid program \( \alpha, \alpha^* \) repeats \( \alpha \) arbitrarily many times

Assume all consecutive pairs of states \((\nu_{i-1}, \nu_i) \in \rho(\alpha)\) of \( n \in \mathbb{N}^+ \) executions, whose valuations are recalled with \( \gamma_i^\mathbf{V} \equiv \wedge_{x \in \mathbf{V}} x = x^i \) and \( \gamma_i^{-1} \) are plausible w.r.t. the model \( \alpha \), i.e., \( \models \wedge_{1 \leq i \leq n} (\gamma_i^{-1} \rightarrow \langle \alpha \rangle \gamma_i^\mathbf{V}) \) with \( \gamma_\mathbf{V}^- = \gamma_\mathbf{V}^0 \) and \( \gamma_\mathbf{V}^+ = \gamma_\mathbf{V}^n \).

Then the sequence of states originates from an \( \alpha^* \) execution from \( \gamma_\mathbf{V}^0 \) to \( \gamma_\mathbf{V}^n \), i.e., \( \models \gamma_\mathbf{V}^- \rightarrow \langle \alpha^* \rangle \gamma_\mathbf{V}^+ \).
Model Monitor Correctness

\[ \models \phi \rightarrow [\alpha^*] \psi \quad \alpha^* \text{ is provably safe} \]

Definitions

Let \( V_m = BV(\alpha) \cup FV(\psi) \); let \( \nu_0, \nu_1, \nu_2, \nu_3 \ldots \in \mathbb{R}^n \) be a sequence of states, with \( \nu_0 \models \phi \) and that agree on \( \Sigma \setminus V_m \), i.e., \( \nu_0|\Sigma \setminus V_m = \nu_k|\Sigma \setminus V_m \) for all \( k \).

Model Monitor

\((\nu, \nu_{i+1}) \models \chi_m\) as \( \chi_m \) evaluated in the state resulting from \( \nu \) by interpreting \( x^+ \) as \( \nu_{i+1}(x) \) for all \( x \in V_m \), i.e.,

\[ \nu^{\nu_{i+1}(x)}|_{x^+} \models \chi_m \]

Correctness

If \((\nu_i, \nu_{i+1}) \models \chi_m\) for all \( i < n \) then we have \( \nu_n \models \psi \) where

\[ \chi_m \equiv \left( \phi|_{\text{const}} \rightarrow \langle \alpha \rangle \Upsilon^+_{V_m} \right) \]

and \( \phi|_{\text{const}} \) denotes the conditions of \( \phi \) that involve only constants that do not change in \( \alpha \), i.e.,

\( FV(\phi|_{\text{const}}) \cap BV(\alpha) = \emptyset \).
Controller Monitor Correctness

\( \models \phi \rightarrow [\alpha^*] \psi \) \( \alpha^* \) is provably safe with invariant \( \varphi \)

**Definitions** Let \( \alpha \) of the canonical form \( \alpha_{\text{ctrl}}; \alpha_{\text{plant}} \); let \( \nu \models \phi|_{\text{const}} \land \varphi \), as checked by \( \chi_m \); let \( \tilde{\nu} \) be a post-controller state.

**Controller Monitor** \((\nu, \tilde{\nu}) \models \chi_c \) as \( \chi_c \) evaluated in the state resulting from \( \nu \) by interpreting \( x^+ \) as \( \tilde{\nu}(x) \) for all \( x \in V_c \), i.e., \( \nu_{\tilde{\nu}(x)}^{x^+} \models \chi_c \)

**Correctness** If \((\nu, \tilde{\nu}) \models \chi_c \) where

\[
\chi_c \equiv \phi|_{\text{const}} \rightarrow \langle \alpha_{\text{ctrl}} \rangle T^+_{V_c}
\]

then we have that \((\nu, \tilde{\nu}) \in \rho(\alpha_{\text{ctrl}}) \) and \( \tilde{\nu} \models \varphi \).
$\models \phi \rightarrow [\alpha^*] \psi$ \hspace{1em} $\alpha^*$ is provably safe with invariant $\varphi$

**Definitions**

Let $V_p = BV(\alpha) \cup FV([\alpha] \varphi)$. Let $\nu \models \phi|_{\text{const}} \land \varphi$, as checked by $\chi_m$. Further assume $\tilde{\nu}$ such that $(\nu, \tilde{\nu}) \in \rho(\alpha_{\text{ctrl}})$, as checked by $\chi_c$.

**Prediction Monitor**

$(\nu, \tilde{\nu}) \models \chi_p$ as $\chi_p$ evaluated in the state resulting from $\nu$ by interpreting $x^+$ as $\tilde{\nu}(x)$ for all $x \in V_p$, i.e., $\nu_{x^+} \models \chi_p$

**Correctness**

If $(\nu, \tilde{\nu}) \models \chi_p$ where

$$\chi_p \equiv (\phi|_{\text{const}} \land \varphi) \rightarrow \langle \alpha_{\text{ctrl}} \rangle (\Upsilon_{V_p}^+ \land [\alpha_{\text{plant}}] \varphi)$$

then we have for all $(\tilde{\nu}, \omega) \in \rho(\alpha_{\text{plant}})$ that $\omega \models \varphi$
Decidability and Computability

Assumptions
- canonical models $\alpha \equiv \alpha_{\text{ctrl}}; \alpha_{\text{plant}}$ without nested loops
- with solvable differential equations in $\alpha_{\text{plant}}$
- disturbed plants $\alpha_{\delta_{\text{plant}}}$ with constant additive disturbance $\delta$

Decidability
Monitor correctness is decidable, i.e., the formulas

\[
\begin{align*}
\chi_m &\rightarrow \langle \alpha \rangle \gamma^+_V \\
\chi_c &\rightarrow \langle \alpha_{\text{ctrl}} \rangle \gamma^+_V \\
\chi_p &\rightarrow \langle \alpha \rangle (\gamma^+_V \land [\alpha_{\delta_{\text{plant}}} \phi])
\end{align*}
\]

are decidable

Computability
Monitor synthesis is computable, i.e., the functions

\[
\begin{align*}
\text{synth}_m : \langle \alpha \rangle \gamma^+_V &\rightarrow \chi_m \\
\text{synth}_c : \langle \alpha_{\text{ctrl}} \rangle \gamma^+_V &\rightarrow \chi_c \\
\text{synth}_p : \langle \alpha \rangle (\gamma^+_V \land [\alpha_{\delta_{\text{plant}}} \phi]) &\rightarrow \chi_p
\end{align*}
\]

are computable
Water Tank Example: Monitor Conjecture

Variables

\[ \begin{align*}
 x & \quad \text{current level} \\
 m & \quad \text{maximum level} \\
 \varepsilon & \quad \text{control cycle} \\
 f & \quad \text{flow}
\end{align*} \]

Model and Safety Property

\( 0 \leq x \leq m \land \varepsilon > 0 \Rightarrow \phi \)  

\[ \phi : \begin{cases} 
 f := \ast; & \varepsilon \land \left( -1 \leq f \leq \frac{m-x}{\varepsilon} \right); \\
 t := 0; & (x' = f, \ t' = 1 \land x \geq 0 \land t \leq \varepsilon) \end{cases} \]

\( 0 \leq x \leq m \land \varepsilon > 0 \Rightarrow \psi \)

Model Monitor Specification Conjecture

\( \varepsilon > 0 \Rightarrow \langle \phi \mid_{\text{const}} : \begin{cases} 
 f := \ast; & \varepsilon \land \left( -1 \leq f \leq \frac{m-x}{\varepsilon} \right); \\
 t := 0; & (x' = f, \ t' = 1 \land x \geq 0 \land t \leq \varepsilon) \end{cases} \rangle \)  

\( \gamma_{V_m} \)
Proof Rules

\[
\begin{align*}
\langle \ast \rangle & \quad (\exists x) \exists X \langle x := X \rangle \phi & 1 \\
\langle x := \ast \rangle \phi & \quad (\exists r) \Gamma \vdash \phi(\theta), \exists x \phi(x), \Delta & 2 \\
\Gamma & \vdash \exists x \phi(x), \Delta \quad (\text{Wr}) & \quad \Gamma \vdash \phi, \Delta
\end{align*}
\]

1  \( X \) is a new logical variable
2  \( \theta \) is an arbitrary term, often a new (existential) logical variable \( X \).

Sequent Deduction

\[
\begin{align*}
\exists r, \text{Wr} & \quad \phi \vdash \langle f := F \rangle \langle ? - 1 \leq f \leq \frac{m - x}{\epsilon} \rangle \langle \text{plant} \rangle \gamma^+ & \text{w/o Opt. 1} \\
\phi & \vdash \exists F \langle f := F \rangle \langle ? - 1 \leq f \leq \frac{m - x}{\epsilon} \rangle \langle \text{plant} \rangle \gamma^+ \\
\langle \ast \rangle & \quad \phi \vdash \langle f := \ast \rangle ; \langle ? - 1 \leq f \leq \frac{m - x}{\epsilon} \rangle \langle \text{plant} \rangle \gamma^+ \quad \text{with Opt. 1 (anticipate } f = f^+ \text{ from } \gamma^+) \\
\exists r, \text{Wr} & \quad \phi \vdash \langle f := f^+ \rangle \langle ? - 1 \leq f \leq \frac{m - x}{\epsilon} \rangle \langle \text{plant} \rangle \gamma^+ \\
\ldots & \quad \end{align*}
\]
Water Tank Example: Differential Equations

Proof Rules

\[
\begin{align*}
&\exists T \geq 0 ((\forall 0 \leq \tilde{t} \leq T \langle x := y(\tilde{t}) \rangle H) \land \langle x := y(T) \rangle \phi) \\
\frac{(\langle' \rangle)}{\langle x' = \theta \land H \rangle \phi} \\
\end{align*}
\]

1. \( T \) and \( \tilde{t} \) are fresh logical variables and \( \langle x := y(T) \rangle \) is the discrete assignment belonging to the solution \( y \) of the differential equation with constant symbol \( x \) as symbolic initial value

2. iff \( \phi \equiv \text{QE}(\phi) \), \( \phi \) is a first-order real arithmetic formula, \( \text{QE}(\phi) \) is an equivalent quantifier-free formula

Sequent Deduction

\[
\begin{align*}
\phi \vdash F = f^+ \land x^+ = x + Ft^+ \land t^+ \geq 0 \land x \geq 0 \land \varepsilon \geq t^+ \geq 0 \land Ft^+ + x \geq 0 \\
\text{QE} \\
\phi \vdash \forall 0 \leq \tilde{t} \leq T (x + f^+ \tilde{t} \geq 0 \land \tilde{t} \leq \varepsilon) \land F = f^+ \land x^+ = x + Ft^+ \land t^+ = t^+ \\
\exists r, \text{Wr} \\
\phi \vdash \exists T \geq 0 ((\forall 0 \leq \tilde{t} \leq T (x + f^+ \tilde{t} \geq 0 \land \tilde{t} \leq \varepsilon)) \land F = f^+ \land (x^+ = x + FT \land t^+ = T)) \\
(\langle' \rangle) \phi \vdash \langle f := F; t := 0 \rangle \{x' = f, t' = 1 \land x \geq 0 \land t \leq \varepsilon\} \gamma^+
\end{align*}
\]
### Evaluation

<table>
<thead>
<tr>
<th>Case Study</th>
<th>Model</th>
<th>Monitor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dim. proof size (branches)</td>
<td>dim. steps (open seq.) w/ Opt. 1 auto</td>
</tr>
<tr>
<td><strong>χ_m</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water tank</td>
<td>5 38 (4)</td>
<td>3 16 (2)</td>
</tr>
<tr>
<td>Cruise control</td>
<td>11 969 (124)</td>
<td>7 127 (13)</td>
</tr>
<tr>
<td>Speed limit</td>
<td>9 410 (30)</td>
<td>6 487 (32)</td>
</tr>
<tr>
<td><strong>χ_c</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water tank</td>
<td>5 38 (4)</td>
<td>1 12 (2)</td>
</tr>
<tr>
<td>Cruise control</td>
<td>11 969 (124)</td>
<td>7 83 (13)</td>
</tr>
<tr>
<td>Ground robot</td>
<td>14 3350 (225)</td>
<td>11 94 (10)</td>
</tr>
<tr>
<td>ETCS safety</td>
<td>16 193 (10)</td>
<td>13 162 (13)</td>
</tr>
<tr>
<td><strong>χ_p</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water tank</td>
<td>8 80 (6)</td>
<td>1 135 (4)</td>
</tr>
</tbody>
</table>

- **Theorem:** ModelPlex is decidable and monitor synthesis can be automated in important classes.
Algorithm 1: ModelPlex monitor synthesis

input: A hybrid program $\alpha$, a set of variables $\mathcal{V} \subseteq BV(\alpha)$, an initial condition $\phi$ such that $\models \phi \rightarrow [\alpha^*]\psi$.

output: A monitor $\chi_m$ such that $\models \chi_m \equiv \phi|_{\text{const}} \rightarrow \langle \alpha \rangle \Upsilon^+$.

begin

$S \leftarrow \emptyset$

$\Upsilon^+ \leftarrow \bigwedge_{x \in \mathcal{V}} x = x^+$ with fresh variables $x_i^+$ // Monitor conjecture

$G \leftarrow \{ \models \phi|_{\text{const}} \rightarrow \langle \alpha \rangle \Upsilon^+ \}$

while $G \neq \emptyset$ do // Analyze monitor conjecture

foreach $g \in G$ do

$G \leftarrow G - \{g\}$

if $g$ is first-order then

if $\not\models g$ then $S \leftarrow S \cup \{g\}$

else

$\tilde{g} \leftarrow$ apply dL proof rule to $g$

$G \leftarrow G \cup \{\tilde{g}\}$

end

end

$\chi_m \leftarrow \bigwedge_{s \in S} s$ // Collect open sequents