Differential Refinement Logic

Sarah M. Loos and André Platzer
Computer Science Department
Carnegie Mellon University
Verified Cyber-Physical Systems
Verified Cyber-Physical Systems

\[ x(i) \]

\[ x(j) \]

\[ x(k) \]

\[ x(m) \]
Verified Cyber-Physical Systems

[FM11, ITSC11, ICCPS12, HSCC13, ITSC13]
Differential Refinement Logic (dRL)

\[ \alpha \leq \beta \]

Proof Calculus

\[
\frac{\Gamma \vdash [\beta] \phi, \Delta}{\Gamma \vdash [\alpha] \phi, \Delta}
\]

Time-triggered vs. Event-triggered

\[ \text{time}^* \leq \text{event}^* \]

Verified Car Control
Differential Refinement Logic (dRL)

\[ \alpha \leq \beta \]

Proof Calculus

\[ \Gamma \vdash [\beta] \phi, \Delta \quad \Gamma \vdash \alpha \leq \beta, \Delta \]

\[ \Gamma \vdash [\alpha] \phi, \Delta \] 

Time-triggered vs. Event-triggered

\[ \text{time}^* \leq \text{event}^* \]

Verified Car Control
Refinement Relation

$\alpha \leq \beta$
Refinement Relation

\[ \alpha \leq \beta \]

\[ ((?\phi; a := \theta \cup a := -B); x'' = a \& \psi)^* \]

\[ ((?\phi; a := \ast \cup a := -B); x'' = a)^* \]
Refinement Relation

$\alpha \leq \beta$

$((\phi; a := \theta \cup a := -B); x'' = a \& \psi)^*$

$((\phi; a := * \cup a := -B); x'' = a)^*$
Refinement Relation

\[ \alpha \leq \beta \]

\[ \left( (\exists \phi; \ a := \theta \cup a := -B); \ x'' = a \land \psi \right)^* \]

\[ \left( (\exists \phi; \ a := \ast \cup a := -B); \ x'' = a \right)^* \]
Refinement Relation

\[ \alpha \leq \beta \]

\[
\text{[((} \phi; a := \theta \cup a := \neg B); x'' = a \& \psi \text{)]}^* \\
\leq \\
\text{[((} \phi; a := * \cup a := \neg B); x'' = a \text{)]}^*
\]
\[ \alpha \leq \beta \]

\[ (((?\phi; a := \theta \cup a := -B); x'' = a \& \psi)^*) \]

\[ \leq \]

\[ (((?\phi; a := \ast \cup a := -B); x'' = a)^*) \]
So, what does dRL look like exactly?

Syntax of a dRL formula:

\[ \phi, \psi ::= \theta_1 \leq \theta_2 \mid \neg \phi \mid \phi \land \psi \mid \forall x \phi \]
So, what does dRL look like exactly?

Syntax of a dRL formula:

\[
\phi, \psi ::= \theta_1 \leq \theta_2 \mid \neg \phi \mid \phi \land \psi \mid \forall x \phi \\
\mid [\alpha] \phi \mid \langle \alpha \rangle \phi
\]
So, what does dRL look like exactly?

Syntax of a dRL formula:

\[ \phi, \psi ::= \theta_1 \leq \theta_2 \mid \neg \phi \mid \phi \land \psi \mid \forall x \phi \]

\[ \mid [\alpha] \phi \mid \langle \alpha \rangle \phi \]

\[ \mid \alpha \leq \beta \]
So, what does dRL look like exactly?

Syntax of a dRL formula:

$$\phi, \psi ::= \theta_1 \leq \theta_2 \mid \neg \phi \mid \phi \land \psi \mid \forall x \phi$$
$$\mid [\alpha] \phi \mid \langle \alpha \rangle \phi$$
$$\mid \alpha \leq \beta$$

Syntax of a hybrid program:
So, what does dRL look like exactly?

Syntax of a dRL formula:

\[ \phi, \psi ::= \theta_1 \leq \theta_2 | \neg \phi | \phi \land \psi | \forall x \phi \\
| [\alpha] \phi | \langle \alpha \rangle \phi \\
| \alpha \leq \beta \]

Syntax of a hybrid program:

\[ \alpha, \beta ::= x := \theta | x' = \theta \land \psi | ?\psi \\
| \alpha \cup \beta | \alpha; \beta | \alpha^* \]
So, what does dRL look like exactly?

Syntax of a dRL formula:

\[ \phi, \psi ::= \theta_1 \leq \theta_2 \mid \neg \phi \mid \phi \land \psi \mid \forall x \phi \]

\[ \mid [\alpha] \phi \mid \langle \alpha \rangle \phi \]

\[ \mid \alpha \leq \beta \]

dRL extends \( \mathcal{DL} \) by adding refinement directly into the grammar of formulas

Syntax of a hybrid program:

\[ \alpha, \beta ::= x ::= \theta \mid x' = \theta \land \psi \mid ?\psi \]

\[ \mid \alpha \cup \beta \mid \alpha; \beta \mid \alpha^* \]
Hybrid Programs model cyber-physical systems

\[ \rho(\alpha) = \{ (v, w) : \text{when starting in state } v \text{ and then following transitions of } \alpha, \text{ state } w \text{ can be reached.} \} \]
Semantics of hybrid programs

\[ v \xrightarrow{x := \theta} w \iff v = w \text{ except for the value of } x \]
Semantics of hybrid programs

\[ x := \theta \quad \text{iff } v = w \text{ except for the value of } x \]

\[ ?\psi \quad \text{Iff } \psi \text{ holds in state } v \]
Semantics of hybrid programs

iff $v = w$ except for the value of $x$

Iff $\psi$ holds in state $v$

If $y(t)$ solves $x' = \theta$
Semantics of hybrid programs

\[ x := \theta \quad \text{iff } v = w \text{ except for the value of } x \]

\[ ?\psi \quad \text{Iff } \psi \text{ holds in state } v \]

\[ x' = \theta \quad \text{If } y(t) \text{ solves } x' = \theta \]

\[ \alpha; \beta \]
Semantics of hybrid programs

Iff $v = w$ except for the value of $x$

Iff $\psi$ holds in state $v$

If $y(t)$ solves $x' = \theta$
Semantics of box modality

Box Modality:

\[ v \models [\alpha] \phi \]
Semantics of box modality

Box Modality:

\[ v \models [\alpha] \phi \]
Semantics of refinement

Refinement Relation:

\[ v \models \alpha \leq \beta \]
Semantics of refinement

Refinement Relation:

\[ v \models \alpha \leq \beta \]
Semantics of refinement

Refinement Relation:

\[ v \models \alpha \leq \beta \]
Semantics of refinement

Refinement Relation:

\[ v \models \alpha \leq \beta \]
Differential Refinement Logic (dRL)

\[ \alpha \leq \beta \]

### Proof Calculus

\[
\frac{\Gamma \vdash [\beta] \phi, \Delta \quad \Gamma \vdash \alpha \leq \beta, \Delta}{\Gamma \vdash [\alpha] \phi, \Delta} ([\leq])
\]

### Time-triggered vs. Event-triggered

\[ \text{time}^* \leq \text{event}^* \]

### Verified Car Control
Differential Refinement Logic (dRL)

\[ \alpha \leq \beta \]

Roadmap

Proof Calculus

\[
\frac{\Gamma \vdash [\beta] \phi, \Delta \quad \Gamma \vdash \alpha \leq \beta, \Delta}{\Gamma \vdash [\alpha] \phi, \Delta} ([\leq])
\]

Time-triggered vs. Event-triggered

\[ \text{time}^* \leq \text{event}^* \]

Verified Car Control
Combining refinement and box modality

\[
\Gamma \vdash [\beta] \phi, \Delta \quad \Gamma \vdash \alpha \leq \beta, \Delta \\
\Gamma \vdash [\alpha] \phi, \Delta \quad ([\leq])
\]
Combining refinement and box modality

\[
\begin{align*}
\frac{\Gamma \vdash [\beta] \phi, \Delta}{\Gamma \vdash [\alpha] \phi, \Delta} & \quad \frac{\Gamma \vdash \alpha \leq \beta, \Delta}{(\leq)} \\
\end{align*}
\]

\[\nu \]

\[\nu \models G, \nu \not\models D\]

for all \(G \in \Gamma, D \in \Delta\)
Combining refinement and box modality

\[
\frac{\Gamma \vdash [\beta] \phi, \Delta \quad \Gamma \vdash \alpha \leq \beta, \Delta}{\Gamma \vdash [\alpha] \phi, \Delta} ([\leq])
\]

\[
\nu \models G, \nu \not\models D
\]

for all \( G \in \Gamma, D \in \Delta \)
Combining refinement and box modality

\( \Gamma \vdash [\beta] \phi, \Delta \)
\( \Gamma \vdash \alpha \leq \beta, \Delta \)
\( \Gamma \vdash [\alpha] \phi, \Delta \)

\( v \models G, v \not\models D \)

for all \( G \in \Gamma, D \in \Delta \)
Combining refinement and box modality

\[ \Gamma \vdash [\beta] \phi, \Delta \]
\[ \Gamma \vdash \alpha \leq \beta, \Delta \]
\[ \Gamma \vdash [\alpha] \phi, \Delta \]

\( v \models G, v \not\models D \)

for all \( G \in \Gamma, D \in \Delta \)
Combining refinement and box modality

\[
\begin{align*}
\Gamma \vdash [\beta]\phi, \Delta & \quad \Gamma \vdash \alpha \leq \beta, \Delta \quad ([\leq]) \\
\Gamma \vdash [\alpha]\phi, \Delta & \quad (\text{for all } \Gamma, \Delta)
\end{align*}
\]

\[v \models G, v \not\models D\]

for all \( G \in \Gamma, D \in \Delta \)
Combining refinement and box modality

\[ \Gamma \vdash [\beta] \phi, \Delta \quad \Gamma \vdash \alpha \leq \beta, \Delta \]

\[ \Gamma \vdash [\alpha] \phi, \Delta \] (\([\leq] \))

\[ v \models G, v \not\models D \]

for all \( G \in \Gamma, D \in \Delta \)
Sequential Composition

\[
\begin{align*}
\Gamma \vdash & \alpha_1 \leq \alpha_2, \Delta \\
\frac{\Gamma \vdash (\beta_1 \leq \beta_2), \Delta}{\Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta} (\;)
\end{align*}
\]

\[
\alpha_1; \beta_1
\]
Sequential Composition

\[
\frac{\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \quad \Gamma \vdash (\beta_1 \leq \beta_2), \Delta}{\Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta}
\]

\[
\alpha_1; \beta_1
\]

\[
\alpha_2; \beta_2?
\]
Sequential Composition

\[
\frac{\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \quad \Gamma \vdash (\beta_1 \leq \beta_2), \Delta}{\Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta}
\]

\(\alpha_1; \beta_1\)
Sequential Composition

\[
\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \quad \Gamma \vdash (\beta_1 \leq \beta_2), \Delta \\
\implies \quad \Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta
\]
Sequential Composition

\[ \Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \]

\[ \Gamma \vdash (\beta_1 \leq \beta_2), \Delta \]

\[ \Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta \]
Sequential Composition

\[
\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \quad \Rightarrow \quad \Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta
\]

\[
\Gamma \vdash (\beta_1 \leq \beta_2), \Delta \quad \Rightarrow \quad (\;)
\]
Sequential Composition

\[
\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \\
\Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta
\]

\[
\Gamma \vdash (\beta_1 \leq \beta_2), \Delta
\]

(;)
Sequential Composition

\[
\frac{\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \quad \Gamma \vdash [\alpha_1] (\beta_1 \leq \beta_2), \Delta}{\Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta}
\]

\[
\alpha_1; \beta_1
\]

\[
\alpha_2
\]

\[
\beta_2
\]

\[
\beta_1
\]
Sequential Composition

\[
\frac{\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \quad \Gamma \vdash [\alpha_1] (\beta_1 \leq \beta_2), \Delta}{\Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta}
\]
Sequential Composition

\[
\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta \quad \Gamma \vdash [\alpha_1] (\beta_1 \leq \beta_2), \Delta \\
\Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta
\]

Diagram:

- Node v with \(\alpha_1\) and \(\alpha_2\)
- Node u with \(\beta_1\)
- Node w with \(\beta_2\)
- \(\alpha_1; \beta_1\) from v to u
- \(\alpha_2; \beta_2\) from u to w
- \(\beta_1\) from u to w
Differential Equations

\[(x' = 1) \overset{?}{\leq} (x' = 9)\]
Differential Equations

\[ (x' = 1) \overset{?}{\leq} (x' = 9) \]

\[ x \in [x_0, \infty) \]
Differential Equations

\( (x' = 1) \leq (x' = 9) \)

\( x \in [x_0, \infty) \) \hspace{2cm} \( x \in [x_0, \infty) \)
(x' = 1) = (x' = 9)

x ∈ [x_0, ∞)  \quad x ∈ [x_0, ∞)
\[ \Gamma \vdash \forall x \left( \frac{\theta_1}{\|\theta_1\|} = \frac{\theta_2}{\|\theta_2\|} \land (\|\theta_1\| = 0 \iff \|\theta_2\| = 0) \right), \Delta \]

\[ \Gamma \vdash (x' = \theta_1) = (x' = \theta_2), \Delta \]
\[
\Gamma \vdash \forall x \left( \frac{\theta_1}{\|\theta_1\|} = \frac{\theta_2}{\|\theta_2\|} \land (\|\theta_1\| = 0 \iff \|\theta_2\| = 0) \right), \Delta \\
\Gamma \vdash (x' = \theta_1) = (x' = \theta_2), \Delta
\]
\[
\Gamma \vdash \forall x \left( \frac{\theta_1}{||\theta_1||} = \frac{\theta_2}{||\theta_2||} \land (||\theta_1|| = 0 \iff ||\theta_2|| = 0) \right), \Delta \\
\Gamma \vdash (x' = \theta_1) = (x' = \theta_2), \Delta
\]
\[ \Gamma \vdash \forall x \left( \frac{\theta_1}{\|\theta_1\|} = \frac{\theta_2}{\|\theta_2\|} \land (\|\theta_1\| = 0 \iff \|\theta_2\| = 0) \right), \Delta \]

\[ \Gamma \vdash (x' = \theta_1) = (x' = \theta_2), \Delta \]
Roadmap

Differential Refinement Logic (dRL)

\[ \alpha \leq \beta \]

Proof Calculus

\[
\frac{\Gamma \vdash \beta \phi, \Delta \quad \Gamma \vdash \alpha \leq \beta, \Delta}{\Gamma \vdash [\alpha]\phi, \Delta} \quad ([\leq])
\]

Time-triggered vs.
Event-triggered

time* ≤ event*

Verified Car
Control

\[
\begin{array}{c}
\end{array}
\]

\[
\begin{array}{c}
\end{array}
\]
Differential Refinement Logic (dRL)

\[ \alpha \leq \beta \]

Proof Calculus

\[
\frac{
\Gamma \vdash [\beta]\phi, \Delta \\
\Gamma \vdash \alpha \leq \beta, \Delta \\
\Gamma \vdash [\alpha]\phi, \Delta
}{
\Gamma \vdash [\leq] \phi, \Delta
}
\]

Time-triggered vs. Event-triggered

\[ \text{time}^* \leq \text{event}^* \]

Verified Car Control
Two Modeling Paradigms

- Time-triggered
  - Discrete sensing

- Event-triggered
  - Continuous sensing
Two Modeling Paradigms

Time-triggered
- Discrete sensing

Event-triggered
- Continuous sensing
Two Modeling Paradigms

Time-triggered
- Discrete sensing
- Realistic, easy to implement
- Difficult to design controllers
- Challenging to verify

Event-triggered
- Continuous sensing
- Unrealistic, hard to implement
- Easier to design controllers
- Easier to verify
Two Modeling Paradigms

Time-triggered

- Discrete sensing
- Realistic, easy to implement
- Difficult to design controllers
- Challenging to verify

≤

Event-triggered

- Continuous sensing
- Unrealistic, hard to implement
- Easier to design controllers
- Easier to verify
**Differential Refinement Logic (dRL)**

\[ \alpha \leq \beta \]

**Proof Calculus**

\[ \frac{\Gamma \vdash \beta \phi, \Delta \quad \Gamma \vdash \alpha \leq \beta, \Delta}{\Gamma \vdash [\alpha] \phi, \Delta} \]

**Time-triggered vs. Event-triggered**

\[ \text{time}^* \leq \text{event}^* \]

**Verified Car Control**

Roadmap
Roadmap

Differential Refinement Logic (dRL)

$\alpha \leq \beta$

**Proof Calculus**

$\Gamma \vdash [\beta] \phi, \Delta$

$\Gamma \vdash \alpha \leq \beta, \Delta$

$\Gamma \vdash [\alpha] \phi, \Delta$ ([≤])

**Time-triggered vs. Event-triggered**

$\text{time}^* \leq \text{event}^*$

**Verified Car Control**
## Local Lane Control using Refinement

<table>
<thead>
<tr>
<th>Proof statistics for local lane controller, with and without refinement</th>
<th>Interactive Steps</th>
<th>Computation Time (seconds)</th>
<th>Proof Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time-triggered [FM11]</strong></td>
<td><strong>656</strong></td>
<td><strong>329.8</strong></td>
<td><strong>924</strong></td>
</tr>
<tr>
<td><strong>Event-triggered</strong></td>
<td>4</td>
<td>73.3</td>
<td>140</td>
</tr>
<tr>
<td>Controllers satisfy refinement</td>
<td>0</td>
<td>0.6</td>
<td>16</td>
</tr>
<tr>
<td>“Brake” for epsilon time</td>
<td>0</td>
<td>2.7</td>
<td>30</td>
</tr>
<tr>
<td>“Accelerate” for epsilon time</td>
<td>79</td>
<td>8.4</td>
<td>126</td>
</tr>
<tr>
<td><strong>Time-triggered (dRL)</strong></td>
<td><strong>83</strong></td>
<td><strong>85.0</strong></td>
<td><strong>312</strong></td>
</tr>
</tbody>
</table>
Contributions

Differential Refinement Logic

- Maintains a modular and hierarchical proof structure
- Abstracts implementation-specific designs
- Leverages iterative system design
- Prove time-triggered model refines event-triggered
- Encouraging evidence of reduced user interaction and computation time
Appendix
We have proved that the refinement relation can be embedded in \(dL\). As a result, \(dL\) and \(dRL\) are equivalent in terms of expressibility and provability.

However, we can analyze \(dRL\) on familiar (challenging) case studies. We can consider:

- Number of proof steps
- Computation time
- Qualitative difficulty to complete proof
- Proof structure
Semantics of hybrid programs

\[
\rho(x := \theta) = \{(v, w) : w = v \text{ except } [[x]]_{w} = [[\theta]]_{v}\}
\]

iff \( v = w \) except for the value of \( x \)

\[
\rho(?\psi) = \{(v, v) : v \models \psi\}
\]

Iff \( \psi \) holds in state \( v \)

\[
\rho(x' = \theta) = \{(\varphi(0), \varphi(t)) : \varphi(s) \models x' = \theta \text{ for all } 0 \leq s \leq t\}
\]

If \( y(t) \) solves \( x' = \theta \)
Semantics of hybrid programs

\[ \rho(\alpha; \beta) = \{ (v, w) : (v, u) \in \rho(\alpha), (u, w) \in \rho(\beta) \} \]
Combining refinement and diamond modality

\[
\Gamma \vdash [\beta] \phi, \Delta \\
\Gamma \vdash \alpha \leq \beta, \Delta \\
\Gamma \vdash [\alpha] \phi, \Delta \\
\]

\[
\Gamma \vdash \langle \alpha \rangle \phi, \Delta \\
\Gamma \vdash \alpha \leq \beta, \Delta \\
\Gamma \vdash \langle \beta \rangle \phi, \Delta \\
\]

\[
([\leq]) \\
(\langle \leq \rangle)
\]
Nondeterministic Assignment

\[ \Gamma \vdash (x := \theta) \leq (x := \ast), \Delta \]
Nondeterministic Assignment

\[
\Gamma \vdash (x := \theta) \leq (x := \ast), \Delta
\]

\[x := \theta \quad \Rightarrow \quad v[x := \theta]_v\]
Nondeterministic Assignment

\[ \Gamma \vdash (x := \theta) \leq (x := \ast), \Delta \]
Nondeterministic Assignment

\[ \Gamma \vdash (x := \theta) \leq (x := *) \]

\[ v \xrightarrow{x := \theta} v[x][\theta] \]

\[ v \xrightarrow{x := *} v[d_1] \]
\[ v \xrightarrow{x := *} v[d_2] \]
\[ v \xrightarrow{x := *} v[d_3] \]
Nondeterministic Repetition

\[
\Gamma \vdash (\alpha \leq \beta), \Delta \\
\frac{\Gamma \vdash \alpha^* \leq \beta^*, \Delta}{(unloop)}
\]
Nondeterministic Repetition

$\Gamma \vdash (\alpha \leq \beta), \Delta$

$\Gamma \vdash \alpha^* \leq \beta^*, \Delta$

(unloop)
Nondeterministic Repetition

\[ \Gamma \vdash (\alpha \leq \beta), \Delta \]

\[ \Gamma \vdash \alpha^* \leq \beta^*, \Delta \]

(\textit{unloop})
Nondeterministic Repetition

\[ \Gamma \vdash [\alpha ](\alpha \leq \beta), \Delta \]

\[ \frac{\Gamma \vdash \alpha^* \leq \beta^*, \Delta}{(\text{unloop})} \]
Nondeterministic Repetition

\[
\Gamma \vdash [\alpha ](\alpha \leq \beta), \Delta \\
\xi
\]

\[
\Gamma \vdash \alpha^* \leq \beta^*, \Delta \\
\text{(unloop)}
\]
Nondeterministic Repetition

\[
\Gamma \vdash [\alpha](\alpha \leq \beta), \Delta \\
\frac{\Gamma \vdash \alpha^* \leq \beta^*, \Delta}{(\text{unloop})}
\]
Nondeterministic Repetition

\[ \Gamma \vdash [\alpha^*](\alpha \leq \beta), \Delta \]

\[ \Gamma \vdash \alpha^* \leq \beta^*, \Delta \]

(unloop)
Nondeterministic Repetition

\[
\Gamma \vdash [\alpha^*](\alpha \leq \beta), \Delta \\
\Rightarrow \\
\Gamma \vdash \alpha^* \leq \beta^*, \Delta
\]
(unloop)
Nondeterministic Repetition

\[
\Gamma \vdash [\alpha^*](\alpha \leq \beta), \Delta \\
\quad \frac{\Gamma \vdash \alpha^* \leq \beta^*, \Delta}{(unloop)}
\]
Nondeterministic Repetition (KAT style)

\[
\Gamma \vdash (\alpha; \gamma) \leq \gamma, \Delta \\
\Gamma \vdash \beta \leq \gamma, \Delta \\
\hline
\Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta \\
\]

(\text{loop}_i)
Nondeterministic Repetition (KAT style)

\[
\Gamma \vdash (\alpha; \gamma) \leq \gamma, \Delta \\
\hline
\Gamma \vdash \beta \leq \gamma, \Delta
\hline
\Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta
\]

(loopp_l)
Nondeterministic Repetition (KAT style)

\[ \Gamma \vdash (\alpha; \gamma) \leq \gamma, \Delta \]
\[ \Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta \]
\[ \Gamma \vdash \beta \leq \gamma, \Delta \]

(loop$_l$)

\[ \nu \xrightarrow{\alpha} w_1 \xrightarrow{\ldots} w_2 \xrightarrow{\alpha} w_3 \xrightarrow{\beta} w_4 \]
Nondeterministic Repetition (KAT style)

\[ \Gamma \vdash (\alpha; \gamma) \leq \gamma, \Delta \]

\[
\frac{\Gamma \vdash \beta \leq \gamma, \Delta}
{\Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta}
\]

(\text{loop}_i)
Nondeterministic Repetition (KAT style)

\[
\Gamma \vdash (\alpha; \gamma) \leq \gamma, \Delta
\]

\[
\Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta
\]

(loopp)
Nondeterministic Repetition (KAT style)

\[
\begin{align*}
\Gamma \vdash (\alpha; \gamma) & \leq \gamma, \Delta \\
\Gamma \vdash [\alpha^*]\beta & \leq \gamma, \Delta \\
\Gamma \vdash \alpha^*; \beta & \leq \gamma, \Delta
\end{align*}
\]

(loop\textsubscript{l})
Nondeterministic Repetition (KAT style)

\[
\begin{align*}
\Gamma &\vdash (\alpha; \gamma) \leq \gamma, \Delta \\
\Gamma &\vdash [\alpha^*]\beta \leq \gamma, \Delta \\
\Gamma &\vdash \alpha^*; \beta \leq \gamma, \Delta \\
\end{align*}
\]

(loopp)
Nondeterministic Repetition (KAT style)

\[ \Gamma \vdash (\alpha; \gamma) \leq \gamma, \Delta \quad \Gamma \vdash [\alpha^*]\beta \leq \gamma, \Delta \]

\[ \Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta \]

(\textit{loop}_l)}
Nondeterministic Repetition (KAT style)

\[ \Gamma \vdash (\alpha; \gamma) \leq \gamma, \Delta \]

\[ \Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta \]

\[ \Gamma \vdash [\alpha^*]\beta \leq \gamma, \Delta \]

(loopy)

Diagram:

- \( v \) \( \xrightarrow{\alpha} \) \( w_1 \) \( \cdots \) \( w_2 \) \( \xrightarrow{\alpha} \) \( w_3 \) \( \xrightarrow{\beta} \) \( w_4 \)

- \( \gamma \)
Nondeterministic Repetition (KAT style)

$$\Gamma \vdash [\alpha^*](\alpha;\gamma) \leq \gamma, \Delta$$

$$\Gamma \vdash \alpha^*;\beta \leq \gamma, \Delta$$

$$(loop_i)$$

Diagram:

- From $v$ to $w_1$ with label $\alpha$.
- From $w_1$ to $\cdots$ to $w_2$ with label $\alpha$.
- From $w_2$ to $w_3$ with label $\alpha$.
- From $w_3$ to $w_4$ with label $\beta$.
- From $w_3$ to $w_3$ with label $\gamma$.
- From $w_3$ to $w_2$ with label $\gamma$.

Symbols:
- $\Gamma$:
- $\alpha$:
- $\beta$:
- $\gamma$:
- $\Delta$:
Nondeterministic Repetition (KAT style)

\[ \Gamma \vdash [\alpha^*](\alpha; \gamma) \leq \gamma, \Delta \]
\[ \Gamma \vdash [\alpha^*]\beta \leq \gamma, \Delta \]

(loopp_{l})

\[ \Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta \]
Nondeterministic Repetition (KAT style)

\[ \Gamma \vdash [\alpha^*](\alpha; \gamma) \leq \gamma, \Delta \]

\[ \Gamma \vdash [\alpha^*]\beta \leq \gamma, \Delta \]

\[ \Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta \]

(loopp)
Nondeterministic Repetition (KAT style)

\[ \Gamma \vdash [\alpha^*] (\alpha; \gamma) \leq \gamma, \Delta \]

\[ \Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta \] (loop)
Nondeterministic Repetition (KAT style)

\[ \Gamma \vdash \beta \leq \gamma, \Delta \quad \Gamma \vdash (\gamma; \alpha) \leq \gamma, \Delta \]

\[ \Gamma \vdash \beta; \alpha^* \leq \gamma, \Delta \]

(loopr)

Diagram:

\[ v \xrightarrow{\beta} w_1 \xrightarrow{\alpha} w_2 \xrightarrow{} \cdots \xrightarrow{} w_3 \xrightarrow{\alpha} w_4 \]
Nondeterministic Repetition (KAT style)

\[
\frac{\Gamma \vdash \beta \leq \gamma, \Delta \quad \Gamma \vdash (\gamma; \alpha) \leq \gamma, \Delta}{\Gamma \vdash \beta; \alpha^* \leq \gamma, \Delta}
\]

(loopr)

Diagram:

- Node $v$ to $w_1$ with label $\beta$
- $w_1$ to $w_2$ with label $\alpha$
- $w_2$ to $\ldots$ with label $\gamma$
- $\ldots$ to $w_3$ with label $\gamma$
- $w_3$ to $w_4$ with label $\alpha$
- $w_4$ to $v$ with label $\gamma$
Proof Tree
\[ H(x) \land I \vdash [\text{event}^*] \phi \quad H(x) \land I \vdash \text{time}^* \leq \text{event}^* \]

\[ H(x) \land I \vdash [\text{time}^*] \psi \]
Proof Tree

\[
\begin{array}{c}
H(x) \land I \vdash [event]H(x) \land I \\
H(x) \land I \vdash \text{events} \subseteq \text{event}\text{es}_E \\
H(x) \land I \vdash [event^*]\phi \\
H(x) \land I \vdash \text{time}^* \leq \text{event}^* \text{([\leq]})
\end{array}
\]
Proof Tree

Open goals
Proof Tree

Open goals

\[\text{Controllers satisfy refinement}\]

\[
\text{Safe}^{\rightarrow} \text{Safe}^{\rightarrow}
\]

\[
\text{Braking}^{\text{safe}} \text{ for time}^{\text{safe}}
\]

\[
\text{Accelerating}^{\text{safe}} \text{ for time}^{\text{safe}}\]

\[\text{H}(x) \land [\text{event}] \phi\]

\[\text{H}(x) \land \text{ctrl}_i \leq \text{ctrl}_{Ev}\]

[\text{U}]

\[\text{H}(x) \land [a := c \cup (a := \ast; ?\text{Safe}_c(x,a))] \leq (a := c \cup (a := \ast; ?\text{Safe}_c(x,a)))\]

\[\text{H}(x) \land [a := c \cup (a := \ast; ?\text{Safe}_c(x,a))] \leq (a := c \cup (a := \ast; ?\text{Safe}_c(x,a)))\]

\[\text{H}(x) \land I \land \text{ctrl}_i \leq \text{ctrl}_{Ev}\]

\[\text{H}(x) \land I \land \text{time} \leq \text{event}\]

\[\text{H}(x) \land I \land [\text{event}^{\ast}] \phi\]

\[\text{H}(x) \land I \land \text{time}^{\ast} \leq \text{event}^{\ast}([\leq])\]

\[\text{H}(x) \land I \land [\text{time}^{\ast}] \phi\]

\[\text{def of } S_0(x)\]

\[0 \leq t \leq E(x) \land x \leq S_0(x) \land f(x) \neq f(x) \]

\[0 \leq t \leq E(x) \land x \leq S_0(x) \land f(x) \neq f(x) \]

\[\text{Proof Tree}\]
“Braking” is safe for $\varepsilon$ time

$H(S_c(0)) \land 0 \leq t \leq \varepsilon \vdash H(S_c(t))$

“Accelerating” is safe for $\varepsilon$ time

$\text{Safe}_\varepsilon(S_a(0)) \land 0 \leq t \leq \varepsilon \vdash H(S_a(t))$

Controllers satisfy refinement

$\vdash \text{Safe}_\varepsilon \rightarrow \text{Safe}$

Event-triggered is safe

$H(x) \land I \vdash [\text{event}]H(x) \land I$

Time-triggered is safe

$H(x) \land I \vdash [\text{time}^*]\phi$
"Braking" is safe for \( \varepsilon \) time
\[ H(S_c(0)) \land 0 \leq t \leq \varepsilon \vdash H(S_c(t)) \]

"Accelerating" is safe for \( \varepsilon \) time
\[ \text{Safe}_\varepsilon(S_a(0)) \land 0 \leq t \leq \varepsilon \vdash H(S_a(t)) \]

Controllers satisfy refinement
\[ \vdash \text{Safe}_\varepsilon \rightarrow \text{Safe} \]

Event-triggered is safe
\[ H(x) \land I \vdash [\text{event}]H(x) \land I \]

Time-triggered is safe
\[ H(x) \land I \vdash [\text{time}^*]\phi \]
dRL Proof Rules: Partial Order

Reflexive:

\[ \Gamma \vdash \alpha \leq \alpha, \Delta \]

Transitive:

\[ \frac{\Gamma \vdash \alpha \leq \beta, \Delta \quad \Gamma \vdash \beta \leq \gamma, \Delta}{\Gamma \vdash \alpha \leq \gamma, \Delta} \] \quad (\leq_{trans})

Antisymmetric:

\[ \frac{\Gamma \vdash \alpha \leq \beta, \Delta \quad \Gamma \vdash \beta \leq \alpha, \Delta}{\Gamma \vdash \alpha = \beta, \Delta} \] \quad (\leq_{antisym})^{1}
dRL Proof Rules: KAT

\[
\frac{\Gamma \vdash \alpha \cup (\beta \cup \gamma) = (\alpha \cup \beta) \cup \gamma, \Delta}{(\cup_{\text{assoc}})} \quad \frac{\Gamma \vdash \alpha \cup \beta = \beta \cup \alpha, \Delta}{(\cup_{\text{comm}})}
\]

\[
\frac{\Gamma \vdash \alpha \cup \bot = \alpha, \Delta}{(\cup_{\text{id}})} \quad \frac{\Gamma \vdash (\alpha \cup \alpha) = \alpha, \Delta}{(\cup_{\text{idemp}})}
\]

\[
\frac{\Gamma \vdash \alpha; (\beta; \gamma) = (\alpha; \beta); \gamma, \Delta}{(\cdot_{\text{assoc}})} \quad \frac{\Gamma \vdash (?\top; \alpha) = \alpha, \Delta}{(\cdot_{\text{id-l}})} \quad \frac{\Gamma \vdash (\alpha; ?\top) = \alpha, \Delta}{(\cdot_{\text{id-r}})}
\]

\[
\frac{\Gamma \vdash \alpha; (\beta \cup \gamma) = ((\alpha; \beta) \cup (\alpha; \gamma)), \Delta}{(\text{dist-l})} \quad \frac{\Gamma \vdash (\alpha \cup \beta); \gamma = ((\alpha; \gamma) \cup (\beta; \gamma)), \Delta}{(\text{dist-r})}
\]

\[
\frac{\Gamma \vdash (\alpha; ?\bot) = ?\bot, \Delta}{(\cdot_{\text{annih-r}})} \quad \frac{\Gamma \vdash (?\bot; \alpha) = ?\bot, \Delta}{(\cdot_{\text{annih-l}})}
\]

\[
\frac{\Gamma \vdash (?\top \cup (\alpha; \alpha^*)) = \alpha^*, \Delta}{(\text{unroll}_l)} \quad \frac{\Gamma \vdash (?\top \cup (\alpha^*; \alpha)) = \alpha^*, \Delta}{(\text{unroll}_r)}
\]

\[
\frac{\Gamma \vdash \alpha^*(\alpha; \gamma) \leq \gamma, \Delta}{(\text{loop}_l)} \quad \frac{\Gamma \vdash [\alpha^*] \beta \leq \gamma, \Delta}{(\text{loop}_r)}
\]

\[
\frac{\Gamma \vdash \beta \leq \gamma, \Delta}{\Gamma \vdash \alpha^*; \beta \leq \gamma, \Delta}
\]

\[
\frac{\Gamma \vdash \beta \leq \gamma, \Delta}{\Gamma \vdash \beta; \alpha^* \leq \gamma, \Delta}
\]
dRL Proof Rules: Differential Equations

\[
\Gamma \vdash [x' = \theta \& H_1]H_2, \Delta \\
\Gamma \vdash (x' = \theta \& H_1) = (x' = \theta \& H_1 \land H_2), \Delta \tag{DC}
\]

\[
\Gamma \vdash \forall x (H_1 \rightarrow H_2), \Delta \\
\Gamma \vdash (x' = \theta \& H_1) \leq (x' = \theta \& H_2), \Delta \tag{DR}
\]

\[
\Gamma \vdash \forall x \left( \theta_1 \|\theta_2\| = \theta_2 \|\theta_1\| \land (\|\theta_1\|^2 = 0 \leftrightarrow \|\theta_2\|^2 = 0) \right), \Delta \\
\Gamma \vdash (x' = \theta_1) = (x' = \theta_2), \Delta \tag{match direction field}^2
\]

\[
\Gamma \vdash \forall x \left( \frac{\theta_1}{\|\theta_1\|} = \frac{\theta_2}{\|\theta_2\|} \land (\|\theta_1\| = 0 \leftrightarrow \|\theta_2\| = 0) \right), \Delta \\
\Gamma \vdash (x' = \theta_1) = (x' = \theta_2), \Delta \tag{mdf}^2
\]
dRL Proof Rules: Structural

\[ \frac{\Gamma \vdash \alpha \leq \gamma \land \beta \leq \gamma, \Delta}{\Gamma \vdash \alpha \cup \beta \leq \gamma, \Delta} \quad (\cup_l) \]

\[ \frac{\Gamma \vdash \alpha \leq \beta \lor \alpha \leq \gamma, \Delta}{\Gamma \vdash \alpha \leq \beta \cup \gamma, \Delta} \quad (\cup_r) \]

\[ \frac{\Gamma \vdash [\alpha^*](\alpha \leq \beta), \Delta}{\Gamma \vdash \alpha^* \leq \beta^*, \Delta} \quad (\text{unloop}) \]

\[ \frac{\Gamma \vdash \alpha_1 \leq \alpha_2, \Delta}{\Gamma \vdash (\alpha_1; \beta_1) \leq (\alpha_2; \beta_2), \Delta} \quad (;) \]

\[ \frac{\Gamma \vdash \phi \rightarrow \psi, \Delta}{\Gamma \vdash ?\phi \leq ?\psi, \Delta} \quad (?) \]

\[ \frac{\Gamma \vdash (x \leftarrow \theta) \leq (x \leftarrow \ast), \Delta}{\Gamma \vdash (x \leftarrow \theta) \leq (x \leftarrow \ast), \Delta} \quad (\leftarrow \ast) \]
\[
\frac{\Gamma \vdash \phi \rightarrow \psi, \Delta}{\Gamma \vdash \phi \leq \psi, \Delta}
\]

(?)

Iff \(\psi\) holds in state \(\nu\)

\[
\rho(\psi) = \{(\nu, \nu) : \nu \models \psi\}
\]
Differential Refinement

\[
\Gamma \vdash \forall x \left( H_1 \rightarrow H_2 \right), \Delta \\
\Gamma \vdash (x' = \theta \& H_1) \leq (x' = \theta \& H_2), \Delta
\]  

(DR)

\[
\rho(x' = \theta) = \{(\varphi(0), \varphi(t)) : \varphi(s) \models x' = \theta \text{ for all } 0 \leq s \leq t\}
\]

If \( y(t) \) solves \( x' = \theta \)

\[
\begin{array}{c}
u \\
\xrightarrow{x' = \theta} \\
x := y(t) \\
\rightarrow w
\end{array}
\]
dRL Proof Rules: Differential Equations

\[
\Gamma \vdash \left[ x' = \theta \land H_1 \right] H_2, \Delta \\
\Gamma \vdash (x' = \theta \land H_1) = (x' = \theta \land H_1 \land H_2), \Delta
\]

\[(DC)\]
Kleene Algebra with Tests (KAT)

- Kleene algebra with tests is a system for manipulating programs that are equivalent.
- KAT doesn’t have continuous dynamics, but we can see that it is still relevant to hybrid programs.
Verifying a specific local lane controller

\[ \text{l_lc} \equiv (\text{ctrl}; \text{dyn})^{*} \]

\[ \text{ctrl} \equiv \ell_{\text{ctrl}} \parallel f_{\text{ctrl}}; \]

\[ \ell_{\text{ctrl}} \equiv (a_{\ell} := *; \ ?(-B \leq a_{\ell} \leq A)) \]

\[ f_{\text{ctrl}} \equiv \text{brake} \cup \text{safe}_{\ast} \cup \text{stopped} \]

\[ \text{brake} \equiv (a_{f} := *; \ ?(-B \leq a_{f} \leq -b)) \]

\[ \text{safe}_{\ast} \equiv (?\text{Safe}_{\varepsilon}; \ a_{f} := *; \ ?(-B \leq a_{f} \leq A)) \]

\[ \text{stopped} \equiv (?(v_{f} = 0); \ a_{f} := 0) \]

\[ \text{Safe}_{\varepsilon} \equiv x_{f} + \frac{v_{f}^{2}}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2\varepsilon^{2}} + \varepsilon v_{f}\right) < x_{\ell} + \frac{v_{\ell}^{2}}{2B} \]

\[ \text{dyn} \equiv (t := 0; \ x'_{f} = v_{f}, \ v'_{f} = a_{f}, \ x'_{\ell} = v_{\ell}, \ v'_{\ell} = a_{\ell}, \ t' = 1 \]

& \ v_{f} \geq 0 \land v_{\ell} \geq 0 \land t \leq \varepsilon) \]
Verifying a specific local lane controller

\[ \text{llc}_{\theta} \equiv (\text{ctrl}_{\theta}; \text{dyn})^* \]

\[ \text{ctrl}_{\theta} \equiv \ell_{\text{ctrl}} \parallel f_{\text{ctrl}_{\theta}}; \]

\[ \ell_{\text{ctrl}} \equiv (a_{\ell} := *, \ ?(-B \leq a_{\ell} \leq A)) \]

\[ f_{\text{ctrl}_{\theta}} \equiv \text{brake} \cup \text{safe}_{\theta} \cup \text{stopped} \]

\[ \text{brake} \equiv (a_{f} := *, \ ?(-B \leq a_{f} \leq -b)) \]

\[ \text{safe}_{\theta} \equiv a_{f} := \theta(x_{f}, x_{\ell}, v_{f}, v_{\ell}) \]

\[ \text{stopped} \equiv (?(v_{f} = 0); a_{f} := 0) \]

\[ \text{Safe}_{\varepsilon} = x_{f} + \frac{v_{f}^2}{2b} + \left(\frac{A}{b} + 1\right)\left(\frac{A}{2} \varepsilon^2 + \varepsilon v_{f}\right) < x_{\ell} + \frac{v_{\ell}^2}{2B} \]

\[ \text{dyn} \equiv (t := 0; x_{f}' = v_{f}, v_{f}' = a_{f}, x_{\ell}' = v_{\ell}, v_{\ell}' = a_{\ell}, t' = 1 \& v_{f} \geq 0 \land v_{\ell} \geq 0 \land t \leq \varepsilon) \]
Additional dRL applications

- Designing proof search heuristics that exploit refinement to automatically create more hierarchical proof structures.
- Shifting the proof responsibility completely to determining refinement.
- Code synthesis – verifying that refinement relation is satisfied with each transformation step.
Event-triggered vs. Time-triggered

**Event-triggered**
- Continuous sensing
- Unrealistic, hard to implement
- Easier to design controllers
- Easier to verify

**Time-triggered**
- Discrete sensing
- Realistic, easy to implement
- Difficult to design controllers
- Challenging to verify

\[(\text{ctrl}; \text{dyn})^*\]

- discrete controller
- continuous dynamics
## Event-triggered vs. Time-triggered

<table>
<thead>
<tr>
<th>Event-triggered</th>
<th>Time-triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous sensing</td>
<td>Discrete sensing</td>
</tr>
<tr>
<td>Unrealistic, hard to implement</td>
<td>Realistic, easy to implement</td>
</tr>
<tr>
<td>Easier to design controllers</td>
<td>Difficult to design controllers</td>
</tr>
<tr>
<td>Easier to verify</td>
<td>Challenging to verify</td>
</tr>
</tbody>
</table>

\[
(ctrl; x' = \theta)^* 
\]

- **Discrete controller**
- **Continuous dynamics**
Event-triggered vs. Time-triggered

Event-triggered
- Continuous sensing
- Unrealistic, hard to implement
- Easier to design controllers
- Easier to verify

Time-triggered
- Discrete sensing
- Realistic, easy to implement
- Difficult to design controllers
- Challenging to verify

$(\text{ctrl}; \quad x' = \theta \land H)^*$

discrete controller
continuous dynamics
## Event-triggered vs. Time-triggered

<table>
<thead>
<tr>
<th>Event-triggered</th>
<th>Time-triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous sensing</td>
<td>Discrete sensing</td>
</tr>
<tr>
<td>Unrealistic, hard to implement</td>
<td>Realistic, easy to implement</td>
</tr>
<tr>
<td>Easier to design controllers</td>
<td>Difficult to design controllers</td>
</tr>
<tr>
<td>Easier to verify</td>
<td>Challenging to verify</td>
</tr>
</tbody>
</table>

\[(\text{ctrl}; x' = \theta \& H)^*\]

- discrete controller
- ?
### Event-triggered vs. Time-triggered

<table>
<thead>
<tr>
<th>Event-triggered</th>
<th>Time-triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous sensing</td>
<td>Discrete sensing</td>
</tr>
<tr>
<td>Unrealistic, hard to implement</td>
<td>Realistic, easy to implement</td>
</tr>
<tr>
<td>Easier to design controllers</td>
<td>Difficult to design controllers</td>
</tr>
<tr>
<td>Easier to verify</td>
<td>Challenging to verify</td>
</tr>
</tbody>
</table>

\[(\text{ctrl}_t; x' = \theta \& t \leq \varepsilon)^*\]

- **discrete controller**
- **?**
Event-triggered vs. Time-triggered

Event-triggered

- Continuous sensing
- Unrealistic, hard to implement
- Easier to design controllers
- Easier to verify

Time-triggered

- Discrete sensing
- Realistic, easy to implement
- Difficult to design controllers
- Challenging to verify

\[(\text{ctrl}_t; x' = \theta \& t \leq \varepsilon)^*\]
Event-triggered vs. Time-triggered

**Event-triggered**
- Continuous sensing
- Unrealistic, hard to implement
- Easier to design controllers
- Easier to verify

\[
(\text{ctrl}_e; x' = \theta \land x + \frac{v^2}{2B} \leq S)^* \]

**Time-triggered**
- Discrete sensing
- Realistic, easy to implement
- Difficult to design controllers
- Challenging to verify

\[
(\text{ctrl}_t; x' = \theta \land t \leq \varepsilon)^* \]
Event-triggered vs. Time-triggered

Event-triggered

- Continuous sensing

$$\text{ctrl}_e; \ x' = \theta \land x + \frac{v^2}{2B} \leq S^*$$

Time-triggered

- Discrete sensing

$$\text{ctrl}_t; \ x' = \theta \land t \leq \epsilon^*$$
Event-triggered vs. Time-triggered

Event-triggered

- Continuous sensing

\[
\begin{align*}
&x + \frac{v^2}{2B} \leq S \\
(\text{ctrl}_e; \quad &x' = \theta \& \frac{v^2}{2B} \\
&x + \frac{v^2}{2B} \leq S)^*
\end{align*}
\]

Time-triggered

- Discrete sensing

\[
(\text{ctrl}_t; \quad x' = \theta \& \quad t \leq \varepsilon)^*
\]
Event-triggered vs. Time-triggered

**Event-triggered**

- Continuous sensing

\[
x + \frac{v^2}{2B} \leq S
\]

\[
(\text{ctrl}_e; \ x' = \theta \land \ x + \frac{v^2}{2B} \leq S)^*
\]

**Time-triggered**

- Discrete sensing

\[
t \leq \varepsilon
\]

\[
(\text{ctrl}_t; \ x' = \theta \land \ t \leq \varepsilon)^*
\]
Event-triggered vs. Time-triggered

Event-triggered
- Continuous sensing

\[
x + \frac{v^2}{2B} \leq S
\]

\((\text{ctrl}_e; \ x' = \theta \ \& \ \ x + \frac{v^2}{2B} \leq S)^*\)

Time-triggered
- Discrete sensing

\[
t \leq \varepsilon
\]

\((\text{ctrl}_t; \ x' = \theta \ \& \ \ t \leq \varepsilon)^*\)
Event-triggered vs. Time-triggered

**Event-triggered**

- Continuous sensing

\[ x + \frac{v^2}{2B} \leq S \]

\[(\text{ctrl}_e; \ x' = \theta \ \& \ x + \frac{v^2}{2B} \leq S)^*\]

**Time-triggered**

- Discrete sensing

\[ t \leq \varepsilon \]

\[(\text{ctrl}_t; \ x' = \theta \ \& \ t \leq \varepsilon)^*\]
Event-triggered vs. Time-triggered

**event-triggered**

\[(\text{?Safe}; a := *) \cup a := c; \quad x' = \theta \& E(x))^*\]

**time-triggered**

\[(\text{?Safe}_\varepsilon; a := *) \cup a := c; \quad x' = \theta \& t \leq \varepsilon)^*\]
dRL Proof Rules: Independence

\[
\vdash (x := \theta_1; y := \theta_2) = (y := \theta_2; x := \theta_1) \quad (\text{indep}_{:=})
\]

\[
\vdash (x' = \theta_1; y' = \theta_2) = (y' = \theta_2; x' = \theta_1) \quad (\text{indep}'_{:=})
\]

\[
\vdash (x := \theta_1; y' = \theta_2) = (y' = \theta_2; x := \theta_1) \quad (\text{indep}'_{:=})
\]
Motivation: Adaptive Cruise Control
Motivation: Adaptive Cruise Control

Low packet loss, small margin for error.
Motivation: Adaptive Cruise Control

Low packet loss, small margin for error.

High packet loss, large margin for error.
Efficiency Analysis of ACC

\[ \text{Eff}_{\text{assist}}(\mathcal{T}) \]

\[ \text{Eff}_{a_f}(\mathcal{T}) \]

\[ \text{Eff}(\mathcal{T}) \]
Modular Proof for Distributed Aircraft

To Prove:
Safe separation of aircraft.

\[ \forall i : A \quad \| x(i) - d(i) \| \leq r \]
\[ \forall i \neq j : A \quad \| d(i) - d(j) \| \geq 2r + p \]
\[ \forall i \neq j : A \quad \| x(i) - x(j) \| \geq p \]
“How can we provide people with cyber-physical systems they can bet their lives on?”
-- Jeanette Wing
Differential Dynamic Logic: Axiomatization

\[
\begin{align*}
[\cdot] & \quad [x := \theta] \phi(x) \leftrightarrow \phi(\theta) \\
[?] & \quad [?H] \phi \leftrightarrow (H \rightarrow \phi) \\
[\cdot'] & \quad [x' = f(x)] \phi \leftrightarrow \forall t \geq 0 [x := y(t)] \phi \quad (y'(t) = f(y)) \\
[\cup] & \quad [\alpha \cup \beta] \phi \leftrightarrow [\alpha] \phi \land [\beta] \phi \\
[;] & \quad [\alpha; \beta] \phi \leftrightarrow [\alpha][\beta] \phi \\
[*] & \quad [\alpha^*] \phi \leftrightarrow \phi \land [\alpha][\alpha^*] \phi \\
K & \quad [\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha] \phi \rightarrow [\alpha] \psi) \\
I & \quad [\alpha^*](\phi \rightarrow [\alpha] \phi) \rightarrow (\phi \rightarrow [\alpha^*] \phi)
\end{align*}
\]

[Platzer08]
Differential Refinement Logic (dRL)

- Proof rules
- Examples

Time-triggered vs. Event-triggered

Verified Car Control

Iterative System Design

\[ \alpha \leq \beta \]

\[
\begin{align*}
x & \coloneqq *; ?Event \\
x & \coloneqq *; ?Time \\
x & \coloneqq \theta
\end{align*}
\]
Differential Refinement Logic (dRL)

- Proof rules
- Examples

Time-triggered vs. Event-triggered

Verified Car Control

Iterative System Design

\( \alpha \leq \beta \)

\( x := \ast; \text{?Event} \quad x := \ast; \text{?Time} \quad x := \theta \)
Verifying a specific local lane controller

\[ \text{safe}_* \equiv (\text{?Safe}_\varepsilon; \ a_f := \ast; \ ?(-B \leq a_f \leq A)) \]
Verifying a specific local lane controller

\[
\text{safe}_* \equiv (\text{Safe}_\varepsilon; \ a_f := *; \ \varepsilon(-B \leq a_f \leq A))
\]

\[
\text{safe}_\theta \equiv \\
\quad a_f := K_p \left( (x_l - x_f) - \left( \frac{v^2}{2b} - \frac{\bar{v}^2}{2b} + \left( \frac{A}{b} + 1 \right) \left( \frac{A}{2} \varepsilon^2 + \varepsilon \bar{v} \right) \right) \right) \\
\quad + K_i(\bar{z}) + K_d(v_l - v_f)
\]
Verifying a specific local lane controller

\[
safe_* \equiv (\text{?Safe}_\varepsilon; \ a_f := \ast; \ ?(-B \leq a_f \leq A))
\]

\[
safe_\theta \equiv \\
\quad \ a_f := \theta
\]
Verifying a specific local lane controller

\[ \text{safe}_* \equiv (\text{Safe}_\epsilon; \ a_f := \ast; \ ?(\neg B \leq a_f \leq A)) \]

\[ \text{safe}_\theta \equiv a_f := \theta \]
Verifying a specific local lane controller

\[ \text{safe}_* \equiv (?\text{Safe}_\varepsilon; \ a_f := *; \ ?(-B \leq a_f \leq A)) \]

\[-B \leq \theta \leq A \quad \leq \quad (\theta > -b) \rightarrow \text{Safe}_\varepsilon \]

\[ \text{safe}_\theta \equiv a_f := \theta \]
Differential Refinement Logic (dRL)

\[ \alpha \leq \beta \]

- Proof rules
- Examples

Roadmap

Time-triggered vs. Event-triggered

Verified Car Control

Iterative System Design

\[ x := *; \text{?Event} \]
\[ x := *; \text{?Time} \]
\[ x := \theta \]
How Can We Prove Distributed Airspace?
Sensor limits on aircraft are local.
How Can We Prove Distributed Airspace?

Sensor limits on aircraft are local.
How Can We Prove Distributed Airspace?

Sensor limits on aircraft are local.
Sensor limits on aircraft are local.
Sometimes a maneuver may look safe locally...
Sensor limits on aircraft are local.
Sometimes a maneuver may look safe locally...
How Can We Prove Distributed Airspace?

Sensor limits on aircraft are local.
Sometimes a maneuver may look safe locally...
But is a terrible idea when implemented globally.
Sensor limits on aircraft are local.
Sometimes a maneuver may look safe locally...
But is a terrible idea when implemented globally.
Sensor limits on aircraft are local.
Sometimes a maneuver may look safe locally...
But is a terrible idea when implemented globally.
Sensor limits on aircraft are local.

Sometimes a maneuver may look safe locally...

But is a terrible idea when implemented globally.
Assumptions and Requirements

Requirements

• **Safety**: At all times, the aircraft must be separated by distance greater than $p$.
• Aircraft trajectories must always be **flyable**.
• An **arbitrary number** of aircraft may enter the maneuver at any time.

Assumptions

• Aircraft maintain constant velocity.
• Sensors are accurate and have no delay.
• Collision avoidance maneuvers are executed on the 2D plane.
Hybrid Dynamics

Aircraft are controlled by steering, through discrete changes in angular velocity $\omega$. 

![Graph showing hybrid dynamics with time $t$, angle $\omega$, displacement $d$, and pressure $p$.]
Big Disc Control

• Leaves maneuverability to pilot discretion.
• Requires large buffer disc.
• Requires aircraft to return to the center of the disc before completing avoidance maneuver.

[LoosRP13]
To Prove:

\[ \text{Init} \rightarrow [\text{BigDisc}]\text{Safe} \]
To Prove:

\[
\text{Init} \rightarrow [\text{BigDisc}]\text{Safe}
\]

\[
\text{Safe} \equiv \\
(\forall i, j : A \quad i \neq j \rightarrow \\
\|x(i) - x(j)\| \geq p)
\]
Big Disc Control

In the given text, the control system is defined as

\[ \text{BigDisc} \equiv (\text{Control} \cup \text{Plant})^* \]

where

\[ \text{Control} \equiv k := *_{A}; (\text{CA} \cup \text{NotCA}) \]

\[ \text{CA} \equiv \text{?(ca(k) = 1); (Steer} \cup \text{Exit)} \]

\[ \text{NotCA} \equiv \text{?(ca(k) = 0); (Steer} \cup \text{Flip} \cup \text{Enter)} \]

\[ \text{Steer} \equiv \omega(k) := *_{\mathbb{R}}; \text{?}\left(-\Omega(k) \leq \omega(k) \leq \Omega(k)\right) \]

\[ \text{Exit} \equiv \text{?(disc(k) = x(k)); ca(k) := 0} \]

\[ \text{Enter} \equiv \omega(k) := \text{side}(k) \cdot \Omega(k); \text{ca(k) := 1} \]

\[ \text{Flip} \equiv \text{side}(k) := -\text{side}(k) \]

\[ \text{Plant} \equiv \forall i : A \left( x(i)' = v(i) \cdot d(i), d(i)' = \omega(i) \cdot d(i)^\perp, \right. \]

\[ \left. \text{disc}(i)' = (1 - ca(i)) \cdot v(i) \cdot d(i) \quad \& \text{EvDom} \right) \]

\[ \text{EvDom} \equiv \forall j : A \]

\[ (j \neq i \land (\text{ca(i) = 0} \lor \text{ca(j) = 0})) \rightarrow \text{Sep}(i, j) \]

\[ \land ||\text{disc}(i) - (x(i) + \text{minr}(i) \cdot \text{side}(i) \cdot d(i)^\perp)|| \]

\[ \leq \text{minr}(i) \]

\[ \text{Sep}(i, j) \equiv ||\text{disc}(i) - \text{disc}(j)|| \geq 2\text{minr}(i) + 2\text{minr}(j) + p \]
Plant \equiv \forall i : A \left( x(i)' = v(i) \cdot d(i), \quad d(i)' = \omega(i) \cdot d(i)^\perp, \quad disc(i)' = (1 - ca(i)) \cdot v(i) \cdot d(i) \& EvDom \right)

Dubins Model for 2D motion

[1] Dubins57
Big Disc Control

The disc does not move when in a collision avoidance maneuver

\[ \text{Plant} \equiv \forall i : A \left( x(i)' = v(i) \cdot d(i), \ d(i)' = \omega(i) \cdot d(i)^\perp, \right. \]

\[ \left. disc(i)' = (1 - ca(i)) \cdot v(i) \cdot d(i) \& \text{EvDom} \right) \]
Plant ≡ \forall i : A \left( x(i)' = v(i) \cdot d(i), \; d(i)' = \omega(i) \cdot d(i)^+, \right.
\left. disc(i)' = (1 - ca(i)) \cdot v(i) \cdot d(i) & EvDom \right)
Big Disc Control

\[ \text{BigDisc} \equiv (\text{Control} \cup \text{Plant})^* \]

\[
\text{Plant} \equiv \forall i : A \left( x(i)' = v(i) \cdot d(i), \quad d(i)' = \omega(i) \cdot d(i)^\perp, \quad \text{disc}(i)' = (1 - ca(i)) \cdot v(i) \cdot d(i) & \text{EvDom} \right)\]
Big Disc Control

\[
\text{BigDisc} \equiv (\text{Control} \cup \text{Plant})^* \\
\text{Control} \equiv \begin{aligned}
  k &:= *_A; \\
  \text{CA} &\equiv ?(ca(k) = 1); (\text{Steer} \cup \text{Exit}) \\
  \text{NotCA} &\equiv ?(ca(k) = 0); (\text{Steer} \cup \text{Flip} \cup \text{Enter}) \\
  \text{Steer} &\equiv \omega(k) := *_{\mathbb{R}}; ?(-\Omega(k) \leq \omega(k) \leq \Omega(k)) \\
  \text{Exit} &\equiv ?(\text{disc}(k) = x(k)); ca(k) := 0 \\
  \text{Enter} &\equiv \omega(k) := \text{side}(k) \cdot \Omega(k); ca(k) := 1 \\
  \text{Flip} &\equiv \text{side}(k) := -\text{side}(k)
\end{aligned} \\
\text{Plant} \equiv \forall i: A \left( x(i)' = v(i) \cdot d(i), \\
  d(i)' = \omega(i) \cdot d(i)^\perp, \\
  \text{disc}(i)' = (1 - ca(i)) \cdot v(i) \cdot d(i) \& \text{EvDom} \right)
\]
Big Disc Control

$\text{Init} \rightarrow [\text{BigDisc}]\text{Safe}$

$$\text{BigDisc} \equiv (\text{Control } \cup \text{ Plant})^*$$

$$\text{Control} \equiv k := *_A; (\text{CA } \cup \text{ NotCA})$$

$$\text{CA} \equiv \left(?(ca(k) = 1); (\text{Steer } \cup \text{ Exit})\right)$$

$$\text{NotCA} \equiv \left(?(ca(k) = 0); (\text{Steer } \cup \text{ Flip } \cup \text{ Enter})\right)$$

$$\text{Steer} \equiv \omega(k) := *_{R}; ?(-\Omega(k) \leq \omega(k) \leq \Omega(k))$$

$$\text{Exit} \equiv ?(\text{disc(k) = x(k)}); ca(k) := 0$$

$$\text{Enter} \equiv \omega(k) := \text{side(k)} \cdot \Omega(k); ca(k) := 1$$

$$\text{Flip} \equiv \text{side}(k) := -\text{side}(k)$$

$$\text{Plant} \equiv \forall i : A \left(\begin{align*}
    x(i)' &= v(i) \cdot d(i), \\
    d(i)' &= w(i) \cdot d(i)^\perp, \\
    \text{disc}(i)' &= (1 - ca(i)) \cdot v(i) \cdot d(i) &\text{[EvDom]}$
\end{align*}\right)$$

$$\text{EvDom} \equiv \forall j : A \left((j \neq i \land (ca(i) = 0 \lor ca(j) = 0)) \rightarrow \text{Sep}(i, j) \right)$$

$$\land ||\text{disc}(i) - (x(i) + \text{minr}(i) \cdot \text{side}(i) \cdot d(i)^\perp)|| \leq \text{minr}(i)$$

$$\text{Sep}(i, j) \equiv ||\text{disc}(i) - \text{disc}(j)|| \geq 2\text{minr}(i) + 2\text{minr}(j) + p$$
Big Disc Control

\[ \text{BigDisc} \equiv (\text{Control} \cup \text{Plant})^* \]
\[ \text{Control} \equiv k := \ast_A; (\text{CA} \cup \text{NotCA}) \]
\[ \text{CA} \equiv ?(ca(k) = 1); (\text{Steer} \cup \text{Exit}) \]
\[ \text{NotCA} \equiv ?(ca(k) = 0); (\text{Steer} \cup \text{Flip} \cup \text{Enter}) \]
\[ \text{Steer} \equiv \omega(k) := \ast_R; ?(-\Omega(k) \leq \omega(k) \leq \Omega(k)) \]
\[ \text{Exit} \equiv ?(\text{disc}(k) = x(k)); ca(k) := 0 \]
\[ \text{Enter} \equiv \omega(k) := \text{side}(k) \cdot \Omega(k); ca(k) := 1 \]
\[ \text{Flip} \equiv \text{side}(k) := -\text{side}(k) \]
\[ \text{Plant} \equiv \forall i : A \left( x(i)' = v(i) \cdot d(i), \ d(i)' = \omega(i) \cdot d(i)^\perp, \right. \]
\[ \left. \quad \text{disc}(i)' = (1 - ca(i)) \cdot v(i) \cdot d(i) & \text{EvDom} \right) \]
\[ \text{EvDom} \equiv \forall j : A \]
\[ (((j \neq i \land ca(i) = 0 \lor ca(j) = 0)) \rightarrow \text{Sep}(i, j) \]
\[ \land \|\text{disc}(i) - (x(i) + \text{minr}(i) \cdot \text{side}(i) \cdot d(i)^\perp)\| \]
\[ \leq \text{minr}(i) \]
\[ \text{Sep}(i, j) \equiv \|\text{disc}(i) - \text{disc}(j)\| \geq 2\text{minr}(i) + 2\text{minr}(j) + p \]
Big Disc Control

\[
\text{BigDisc} \equiv (\text{Control} \cup \text{Plant})^* \\
\text{Control} \equiv k := *_A; (\text{CA} \cup \text{NotCA}) \\
\quad \text{CA} \equiv ?(ca(k) = 1); (\text{Steer} \cup \text{Exit}) \\
\quad \text{NotCA} \equiv ?(ca(k) = 0); (\text{Steer} \cup \text{Flip} \cup \text{Enter}) \\
\text{Steer} \equiv \omega(k) := *_F; ?(-\Omega(k) \leq \omega(k) \leq \Omega(k)) \\
\text{Exit} \equiv ?(\text{disc}(k) = x(k)); ca(k) := 0 \\
\text{Enter} \equiv \omega(k) := \text{side}(k) \cdot \Omega(k); ca(k) := 1 \\
\text{Flip} \equiv \text{side}(k) := -\text{side}(k) \\
\text{Plant} \equiv \forall i : A \left(x(i)' = v(i) \cdot d(i), \ d(i)' = \omega(i) \cdot d(i)^\perp, \right. \\
\left. d \cdot v(i)' = (1 - c(i)) \cdot v(i) \cdot d(i) \cup \text{EvDom} \right) \\
\text{EvDom} \equiv \forall j : A \\
\quad ((j \neq i \land (ca(i) = 0 \lor ca(j) = 0)) \rightarrow \text{Sep}(i, j) \\
\quad \land \left\|\text{disc}(i) - (x(i) + \text{minr}(i) \cdot \text{side}(i) \cdot d(i)^\perp)\right\| \\
\quad \leq \text{minr}(i)) \\
\text{Sep}(i, j) \equiv \left\|\text{disc}(i) - \text{disc}(j)\right\| \geq 2\text{minr}(i) + 2\text{minr}(j) + p
\]
• Deterministic control makes it well suited for UAVs.
• Smaller discs allow aircraft to fly closer together.
• Aircraft may exit maneuver as soon as it is safe to do so.

[PallottinoSBF07, LoosRP13]
Small Discs Control

\[ \text{SmallDiscs} \equiv (\text{Control} \cup \text{Plant})^* \]

Control \equiv k := *_A; (\text{CA} \cup \text{NotCA})

CA \equiv ?(ca(k) = 1); (\text{Exit} \cup \text{Skip})

NotCA \equiv ?(ca(k) = 0); (\text{Steer} \cup \text{Flip} \cup \text{Enter})

Skip \equiv ?true

Steer \equiv \omega(k) := *_R; ?(-\Omega(k) \leq \omega(k) \leq \Omega(k))

Exit \equiv ca(k) := 0

Enter \equiv (\omega(k) := \text{side}(k) \cdot \Omega(k)); ca(k) := 1

Flip \equiv ?(\forall j : A \ (j \neq k \rightarrow \text{FlipSep}(j, k)));

\[ \text{side}(k) := -\text{side}(k) \]

\[ \text{FlipSep}(i, j) \equiv \left\| (x(i) + \text{minr}(i) \cdot \text{side}(i) \cdot d(i)^\perp) \right. \]
\[ \left. - (x(j) - \text{minr}(j) \cdot \text{side}(j) \cdot d(j)^\perp) \right\| \]
\[ \geq \text{minr}(i) + \text{minr}(j) + p \]

Plant \equiv \forall i : A \left( x(i)' = v(i) \cdot d(i), \ d(i)' = \omega(i) \cdot d(i)^\perp \right.
\[ \left. \& \forall j : A \ ((j \neq i \wedge (ca(i) = 0 \vee ca(f) = 0)) \rightarrow \text{Sep}(i, j)) \right) \]

\[ \text{Sep}(i, j) \equiv \left\| (x(i) + \text{minr}(i) \cdot \text{side}(i) \cdot d(i)^\perp) \right. \]
\[ \left. - (x(j) + \text{minr}(j) \cdot \text{side}(j) \cdot d(j)^\perp) \right\| \]
\[ \geq \text{minr}(i) + \text{minr}(j) + p \]
Small Discs Control

SmallDiscs ≡ (Control ∪ Plant)*
Control ≡ k := *A; (CA ∪ NotCA)
    CA ≡?(ca(k) = 1); (Exit ∪ Skip)
NotCA ≡?(ca(k) = 0); (Steer ∪ Flip ∪ Enter)
Skip = ?true
Steer = ω(k) := *R; ?(−Ω(k) ≤ ω(k) ≤ Ω(k))
Exit = ca(k) := 0
Enter = (ω(k) := side(k) · Ω(k)); ca(k) := 1
Flip =?(∀j : A (j ≠ k → FlipSep(j, k)));
    side(k) := −side(k)
FlipSep(i, j) ≡∥((x(i) + minr(i) · side(i) · d(i)⊥)
    + (x(i) − minr(i) · side(i) · d(i)⊥))∥
    ≥ mr(i) + mr(j) + p
Plant ≡ ∀i : A (x(i)' = v(i) · d(i), d(i)' = ω(i)d(i)⊥
    & ∀j : A ((j ≠ i ∧ (ca(i) = 0 ∨ ca(j) = 0))
    → Sep(i, j))
Sep(i, j) ≡∥((x(i) + minr(i) · side(i) · d(i)⊥)
    − (x(j) + minr(j) · side(j) · d(j)⊥))∥
    ≥ minr(i) + minr(j) + p

✔ Verified in KeYmaeraD
Conclusions

Challenges
- CPS needs verification
- Infinite, continuous, and evolving state space, $\mathbb{R}^\infty$
- Continuous dynamics
- Discrete control decisions
- Distributed dynamics
- Arbitrary number of aircraft
- Emergent behaviors

Contributions
- Theorem proving is powerful for verifying distributed dynamics
- Non-linear flight paths and flyable maneuvers
- Compositionality – using small problems to solve the big ones
- Hierarchical proofs
- Undergraduates can understand and verify hybrid systems!
Theorem (Continuous Relative Completeness) (J.Autom.Reas. 2008)

\( d\mathcal{L} \) calculus is a sound & complete axiomatization of hybrid systems relative to differential equations.

Proof 15pp

Theorem (Discrete Relative Completeness) (LICS’12)

\( d\mathcal{L} \) calculus is a sound & complete axiomatization of hybrid systems relative to discrete dynamics.

Proof +10pp


Akshay Rajhans, Ajinkya Bhave, Sarah M. Loos, Bruce H. Krogh, André Platzer, and David Garlan. Using parameters in architectural views to support heterogeneous design and verification. In the IEEE Conference on Decision and Control and European Control Conference. 2011.


