Refining Constructive Hybrid Games

Brandon Bohrer and André Platzer

Logical Systems Lab
Computer Science Department
Carnegie Mellon University

FSCD'20

Why Refine Constructive Hybrid Games?

Refining

$\alpha \leq \beta \quad \alpha \cong \beta \qquad \alpha \cup \beta \leq \text{if}(P)\ \alpha\ \text{else}\ \beta$

Why Refine Constructive Hybrid Games?

Refining

$\alpha \leq \beta \quad \alpha \cong \beta \qquad \alpha \cup \beta \leq \text{if}(P)\ \alpha\ \text{else}\ \beta$

Constructive Differential Game Logic (CdGL)

[P]a[safe]	<α>safe	Witness a safe behavior of α
{P}a{safe}	[α]safe	Observe α, universal proof of "safe"
	$\alpha \leq \beta$	Compute β strategy from α strategy

Why Refine Constructive Hybrid Games?

Refining

$\alpha \leq \beta$ $\alpha \cong \beta$ $\alpha \cup \beta \leq$ if(P) α else β

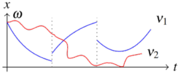
Constructive Differential Game Logic (CdGL)

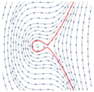
[P]a[safe] $\langle\alpha\rangle$safe Witness a safe behavior of α

{P}a{safe} $[\alpha]$safe Observe α, universal proof of "safe"

$\alpha \leq \beta$ Compute β strategy from α strategy

Hybrid





Why Refine Constructive Hybrid Games?

Refining

$\alpha \leq \beta \quad \alpha \cong \beta \qquad \alpha \cup \beta \leq$ if(P) α else β

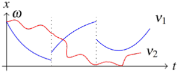
Constructive Differential Game Logic (CdGL)

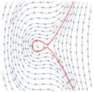
[P]a[safe] <α>safe Witness a safe behavior of α

{P}a{safe} [α]safe Observe α, universal proof of "safe"

$\alpha \leq \beta$ Compute β strategy from α strategy

Hybrid



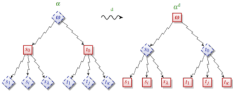


Games

<α>safe

[α]safe

α^d



Why Refine Constructive Hybrid Games?

Refining

$\alpha \leq \beta \quad \alpha \cong \beta \qquad \alpha \cup \beta \leq \text{if}(P)\ \alpha \text{ else } \beta$

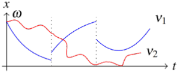
Constructive Differential Game Logic (CdGL)

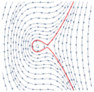
[P]a[safe] <α>safe Witness a safe behavior of α

{P}a{safe} [α]safe Observe α, universal proof of "safe"

$\alpha \leq \beta$ Compute β strategy from α strategy

Hybrid



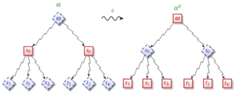


Games

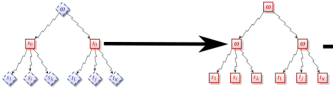
<α>safe

[α]safe

α^d



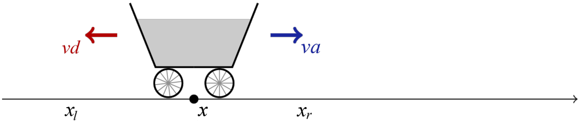
Enables



$\Gamma \vdash [\text{game}]P \quad \leq \quad \Gamma \vdash [\text{system}]P$

Strategies to Systems Synthesis

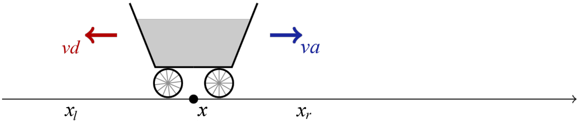
Constructive Hybrid Game: Push-pull



$$\text{safe} \equiv x_l < x_0 = x < x_r \to [\text{PP}](x = x_0)$$

$$\begin{aligned}\text{PP} \equiv \{&\{vd := -1 \cup vd := 1\};\\ &\{va := *; ?(-1 \le va \le 1)\}^d;\\ &\{x' = vd + va \,\&\, x_l \le x \le x_r\}\}^*\end{aligned}$$

Constructive Hybrid Game: Push-pull

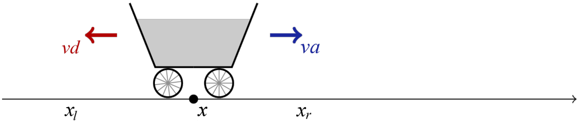


$$\text{safe} \equiv x_l < x_0 = x < x_r \rightarrow [\text{PP}](x = x_0)$$

Either speed

$$\text{PP} \equiv \{\{vd := -1 \cup vd := 1\};$$
$$\{va := *; ?(-1 \leq va \leq 1)\}^d;$$
$$\{x' = vd + va \,\&\, x_l \leq x \leq x_r\}\}^*$$

Constructive Hybrid Game: Push-pull



$$\text{safe} \equiv x_l < x_0 = x < x_r \rightarrow [\text{PP}](x = x_0)$$

Either speed

Limits

$$\text{PP} \equiv \{\{vd := -1 \cup vd := 1\};$$

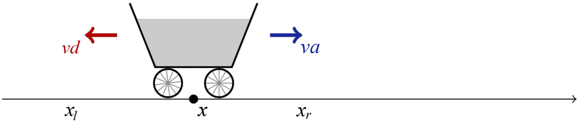
Speed

$$\{va := *; ?(-1 \leq va \leq 1)\}^d;$$

Switch

$$\{x' = vd + va \,\&\, x_l \leq x \leq x_r\}\}^*$$

Constructive Hybrid Game: Push-pull



$$\text{safe} \equiv x_l < x_0 = x < x_r \rightarrow [\text{PP}](x = x_0)$$

Either speed

Limits

$$\text{PP} \equiv \{\{vd := -1 \cup vd := 1\};$$

Speed

$$\{va := *; ?(-1 \le va \le 1)\}^d;$$

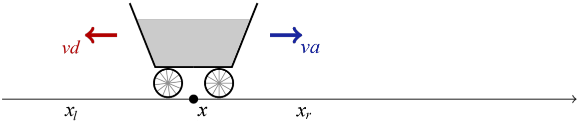
Switch

$$\{x' = vd + va \,\&\, x_l \le x \le x_r\}\}^*$$

Physics

Constraint

Constructive Hybrid Game: Push-pull



$$\text{safe} \equiv x_l < x_0 = x < x_r \rightarrow [PP](x = x_0)$$

Either speed

Limits

$$\begin{aligned}PP \equiv \{&\{vd := -1 \cup vd := 1\};\\ &\{va := *; ?(-1 \leq va \leq 1)\}^d;\\ &\{x' = vd + va \,\&\, x_l \leq x \leq x_r\}\}^*\end{aligned}$$

Speed

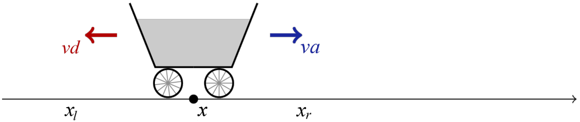
Switch

Physics

Constraint

Loop

Constructive Hybrid Game: Push-pull



$$\text{safe} \equiv x_l < x_0 = x < x_r \rightarrow [PP](x = x_0)$$

Either speed

Limits

$$PP \equiv \{\{vd := -1 \cup vd := 1\};$$

Speed

$$\{va := *; ?(-1 \le va \le 1)\}^d;$$

Switch

$$\{x' = vd + va \,\&\, x_l \le x \le x_r\}\}^*$$

Physics

Loop

Constraint

$$\alpha_{PP} = \{\{\{vd := -1; va := 1\}; \cup \{vd := 1; va := -1\}\}; x := *; x' := vd + va; ?x = x_0\}^*$$

Mirror

Types Give Constructive Semantics

$\ulcorner P \urcorner \quad : (\text{state} \Rightarrow \text{type} \quad)$

$\ulcorner \langle ?Q \rangle P \urcorner \, s \quad = \ulcorner Q \urcorner \, s * \ulcorner P \urcorner \, s$ — Prove test

$\ulcorner [?Q] P \urcorner \, s \quad = \ulcorner Q \urcorner \, s \Rightarrow \ulcorner P \urcorner \, s$ — Assume test

Types Give Constructive Semantics

$$\ulcorner P \urcorner : (\text{state} \Rightarrow \text{type})$$

$$\ulcorner \langle ?Q \rangle P \urcorner\ s = \ulcorner Q \urcorner\ s * \ulcorner P \urcorner\ s$$ — Prove test

$$\ulcorner \langle x := * \rangle P \urcorner\ s = \Sigma v : \mathbb{R}.\ \ulcorner P \urcorner\ (\text{set } s\ x\ v)$$ — Choose x

$$\ulcorner [?Q] P \urcorner\ s = \ulcorner Q \urcorner\ s \Rightarrow \ulcorner P \urcorner\ s$$ — Assume test

$$\ulcorner [x := *] P \urcorner\ s = \Pi v : \mathbb{R}.\ \ulcorner P \urcorner\ (\text{set } s\ x\ v)$$ — Receive x

Types Give Constructive Semantics

$$\ulcorner P \urcorner \quad : (\text{state} \Rightarrow \text{type} \quad)$$

$$\ulcorner \langle ?Q \rangle P \urcorner \; s \quad = \ulcorner Q \urcorner \; s * \ulcorner P \urcorner \; s$$ — Prove test

$$\ulcorner \langle x := * \rangle P \urcorner \; s \quad = \Sigma v : \mathbb{R}.\ \ulcorner P \urcorner \; (\text{set } s \; x \; v)$$ — Choose x

$$\ulcorner \langle \alpha \cup \beta \rangle P \urcorner \; s \quad = \ulcorner \langle \alpha \rangle P \urcorner \; s + \ulcorner \langle \beta \rangle P \urcorner \; s$$ — Choose branch

$$\ulcorner [?Q] P \urcorner \; s \quad = \ulcorner Q \urcorner \; s \Rightarrow \ulcorner P \urcorner \; s$$ — Assume test

$$\ulcorner [x := *] P \urcorner \; s \quad = \Pi v : \mathbb{R}.\ \ulcorner P \urcorner \; (\text{set } s \; x \; v)$$ — Receive x

$$\ulcorner [\alpha \cup \beta] P \urcorner \; s \quad = \ulcorner [\alpha] P \urcorner \; s * \ulcorner [\beta] P \urcorner \; s$$ — Can't choose

Types Give Constructive Semantics

$$\ulcorner P \urcorner \quad : (\text{state} \Rightarrow \text{type} \quad)$$

$$\begin{aligned}
\ulcorner \langle ?Q \rangle P \urcorner\ s &= \ulcorner Q \urcorner\ s * \ulcorner P \urcorner\ s \\
\ulcorner \langle x := * \rangle P \urcorner\ s &= \Sigma v : \mathbb{R}.\ \ulcorner P \urcorner\ (\text{set}\ s\ x\ v) \\
\ulcorner \langle \alpha \cup \beta \rangle P \urcorner\ s &= \ulcorner \langle \alpha \rangle P \urcorner\ s + \ulcorner \langle \beta \rangle P \urcorner\ s \\
\ulcorner \langle \alpha^d \rangle P \urcorner\ s &= \ulcorner [\alpha] P \urcorner\ s
\end{aligned}$$

Prove test

Choose x

Choose branch

Switch

$$\begin{aligned}
\ulcorner [?Q] P \urcorner\ s &= \ulcorner Q \urcorner\ s \Rightarrow \ulcorner P \urcorner\ s \\
\ulcorner [x := *] P \urcorner\ s &= \Pi v : \mathbb{R}.\ \ulcorner P \urcorner\ (\text{set}\ s\ x\ v) \\
\ulcorner [\alpha \cup \beta] P \urcorner\ s &= \ulcorner [\alpha] P \urcorner\ s * \ulcorner [\beta] P \urcorner\ s \\
\ulcorner [\alpha^d] P \urcorner\ s &= \ulcorner \langle \alpha \rangle P \urcorner\ s
\end{aligned}$$

Assume test

Receive x

Can't choose

Switch

Types Give Constructive Semantics

$$\ulcorner P \urcorner \quad : (\text{state} \Rightarrow \text{type} \quad)$$

$$\ulcorner \langle ?Q \rangle P \urcorner \; s \quad = \ulcorner Q \urcorner \; s * \ulcorner P \urcorner \; s$$ Prove test

$$\ulcorner \langle x := * \rangle P \urcorner \; s \quad = \Sigma v : \mathbb{R}. \; \ulcorner P \urcorner \; (\text{set } s \; x \; v)$$ Choose x

$$\ulcorner \langle \alpha \cup \beta \rangle P \urcorner \; s \quad = \ulcorner \langle \alpha \rangle P \urcorner \; s + \ulcorner \langle \beta \rangle P \urcorner \; s$$ Choose branch

$$\ulcorner \langle \alpha^d \rangle P \urcorner \; s \quad = \ulcorner [\alpha] P \urcorner \; s$$ Switch

$$\ulcorner [?Q] P \urcorner \; s \quad = \ulcorner Q \urcorner \; s \Rightarrow \ulcorner P \urcorner \; s$$ Assume test

$$\ulcorner [x := *] P \urcorner \; s \quad = \Pi v : \mathbb{R}. \ulcorner P \urcorner \; (\text{set } s \; x \; v)$$ Receive x

$$\ulcorner [\alpha \cup \beta] P \urcorner \; s \quad = \ulcorner [\alpha] P \urcorner \; s * \ulcorner [\beta] P \urcorner \; s$$ Can't choose

$$\ulcorner [\alpha^d] P \urcorner \; s \quad = \ulcorner \langle \alpha \rangle P \urcorner \; s$$ Switch

$$\ulcorner \alpha \leq_{[]} \beta \urcorner \; s \quad = \left(\Pi P : (\text{state} \Rightarrow \text{type} \;). \left(\ulcorner [\alpha] P \urcorner \; s \Rightarrow \ulcorner [\beta] P \urcorner \; s\right)\right)$$

Types Give Constructive Semantics

$$\ulcorner P \urcorner : (\text{state} \Rightarrow \text{type}_{i+1})$$

$$\ulcorner \langle ?Q \rangle P \urcorner\ s = \ulcorner Q \urcorner\ s * \ulcorner P \urcorner\ s$$ — Prove test

$$\ulcorner \langle x := * \rangle P \urcorner\ s = \Sigma v : \mathbb{R}.\ \ulcorner P \urcorner\ (\text{set } s\ x\ v)$$ — Choose x

$$\ulcorner \langle \alpha \cup \beta \rangle P \urcorner\ s = \ulcorner \langle \alpha \rangle P \urcorner\ s + \ulcorner \langle \beta \rangle P \urcorner\ s$$ — Choose branch

$$\ulcorner \langle \alpha^d \rangle P \urcorner\ s = \ulcorner [\alpha] P \urcorner\ s$$ — Switch

$$\ulcorner [?Q] P \urcorner\ s = \ulcorner Q \urcorner\ s \Rightarrow \ulcorner P \urcorner\ s$$ — Assume test

$$\ulcorner [x := *] P \urcorner\ s = \Pi v : \mathbb{R}.\ \ulcorner P \urcorner\ (\text{set } s\ x\ v)$$ — Receive x

$$\ulcorner [\alpha \cup \beta] P \urcorner\ s = \ulcorner [\alpha] P \urcorner\ s * \ulcorner [\beta] P \urcorner\ s$$ — Can't choose

$$\ulcorner [\alpha^d] P \urcorner\ s = \ulcorner \langle \alpha \rangle P \urcorner\ s$$ — Switch

$$\ulcorner \alpha \leq^i_{[]} \beta \urcorner\ s = \left(\Pi P : (\text{state} \Rightarrow \text{type}_i).\ \left(\ulcorner [\alpha] P \urcorner\ s \Rightarrow \ulcorner [\beta] P \urcorner\ s\right)\right)$$

Types Give Constructive Semantics

$$\ulcorner P \urcorner : (\text{state} \Rightarrow \text{type}_{i+1})$$

$$\ulcorner \langle ?Q \rangle P \urcorner\ s = \ulcorner Q \urcorner\ s * \ulcorner P \urcorner\ s$$ — Prove test

$$\ulcorner \langle x := * \rangle P \urcorner\ s = \Sigma v : \mathbb{R}.\ \ulcorner P \urcorner\ (\text{set } s\ x\ v)$$ — Choose x

$$\ulcorner \langle \alpha \cup \beta \rangle P \urcorner\ s = \ulcorner \langle \alpha \rangle P \urcorner\ s + \ulcorner \langle \beta \rangle P \urcorner\ s$$ — Choose branch

$$\ulcorner \langle \alpha^d \rangle P \urcorner\ s = \ulcorner [\alpha] P \urcorner\ s$$ — Switch

$$\ulcorner [?Q] P \urcorner\ s = \ulcorner Q \urcorner\ s \Rightarrow \ulcorner P \urcorner\ s$$ — Assume test

$$\ulcorner [x := *] P \urcorner\ s = \Pi v : \mathbb{R}.\ \ulcorner P \urcorner\ (\text{set } s\ x\ v)$$ — Receive x

$$\ulcorner [\alpha \cup \beta] P \urcorner\ s = \ulcorner [\alpha] P \urcorner\ s * \ulcorner [\beta] P \urcorner\ s$$ — Can't choose

$$\ulcorner [\alpha^d] P \urcorner\ s = \ulcorner \langle \alpha \rangle P \urcorner\ s$$ — Switch

$$\ulcorner \alpha \le^i_{[]} \beta \urcorner\ s = \left(\Pi P : (\text{state} \Rightarrow \text{type}_i).\ \left(\ulcorner [\alpha] P \urcorner\ s \Rightarrow \ulcorner [\beta] P \urcorner\ s\right)\right)$$

$$\text{R}[\cdot]\quad \frac{\Gamma \vdash [\alpha] P \qquad \Gamma \vdash \alpha \le^i_{[]} \beta}{\Gamma \vdash [\beta] P}\ 0 \qquad\qquad \text{R}\langle\cdot\rangle\quad \frac{\Gamma \vdash \langle \alpha \rangle P \qquad \Gamma \vdash \alpha \le^i_{\langle\rangle} \beta}{\Gamma \vdash \langle \beta \rangle P}\ 0$$

Refinements Subsume Game Algebra

$$\text{trans}\quad \frac{\Gamma \vdash \alpha \leq_{[]} \beta \quad \Gamma \vdash \beta \leq_{[]} \gamma}{\Gamma \vdash \alpha \leq_{[]} \gamma} \qquad \text{refl}\quad \Gamma \vdash \alpha \leq_{[]} \alpha$$

$$\cup\text{A}\quad \Gamma \vdash \{\alpha \cup \beta\} \cup \gamma \cong \alpha \cup \{\beta \cup \gamma\} \qquad \cup\text{c}\quad \Gamma \vdash \alpha \cup \beta \cong \beta \cup \alpha$$

$$;\text{d}_r\quad \Gamma \vdash \{\alpha \cup \beta\}; \gamma \cong \{\alpha; \gamma\} \cup \{\beta; \gamma\}$$

Refinements Resolve Strategic Choice

[∪]L1 $\quad \Gamma \vdash \alpha^d \leq_{[]} \{\alpha \cup \beta\}^d$ $\qquad$ [∪]L2 $\quad \Gamma \vdash \beta^d \leq_{[]} \{\alpha \cup \beta\}^d$

[:∗] $\quad \Gamma \vdash \{x := f\}^d \leq_{[]} \{x := *\}^d$

$$;\text{G} \quad \frac{\cdot \vdash \alpha_1 \leq_{[]} \alpha_2 \quad \cdot \vdash \beta_1 \leq_{[]} \beta_2}{\cdot \vdash \alpha_1;\beta_1 \leq_{[]} \alpha_2;\beta_2}$$

Refinements Resolve Strategic Choice

[∪]L1 $\Gamma \vdash \alpha^d \leq_{[]} \{\alpha \cup \beta\}^d$ [∪]L2 $\Gamma \vdash \beta^d \leq_{[]} \{\alpha \cup \beta\}^d$

[:∗] $\Gamma \vdash \{x := f\}^d \leq_{[]} \{x := *\}^d$

;G $$\frac{\cdot \vdash \alpha_1 \leq_{[]} \alpha_2 \quad \cdot \vdash \beta_1 \leq_{[]} \beta_2}{\cdot \vdash \alpha_1; \beta_1 \leq_{[]} \alpha_2; \beta_2}$$

;S $$\frac{\Gamma \vdash \boldsymbol{\alpha}_1 \leq_{[]} \alpha_2 \quad \Gamma \vdash [\boldsymbol{\alpha}_1]\beta_1 \leq_{[]} \beta_2}{\Gamma \vdash \boldsymbol{\alpha}_1; \beta_1 \leq_{[]} \alpha_2; \beta_2}$$ [1]

[1] $\boldsymbol{\alpha}_1$ is a hybrid system

ODEs are Solved or Abstracted

Assignment

ODE

$$\text{solve}\quad \frac{\Gamma \vdash t = 0 \wedge d \geq 0 \quad \Gamma \vdash [t := *; ?0 \leq t \leq d; x := sol]Q}{\Gamma \vdash \{t := d; x := sol; t' := 1; x' := f\} \leq_{[]} \{t' = 1, x' = f \,\&\, Q\}^d}\ ^1$$

$$\text{DC}\quad \frac{\Gamma \vdash [x' = f \,\&\, P]Q}{\Gamma \vdash \{x' = f \,\&\, P\} \cong \{x' = f \,\&\, P \wedge Q\}}$$

$$\text{DW}\quad \Gamma \vdash \{x := *; x' := f; ?Q\} \leq_{[]} \{x' = f \,\&\, Q\}$$

[1] *sol* solves ODE, $\{t, t', x, x'\}$ not free in d

Game Proofs are Reified as Systems

$$(\text{Proof of } [\alpha]P \text{ or } \langle\alpha\rangle P) \leadsto \text{System}$$

First IH $\boldsymbol{\alpha}$

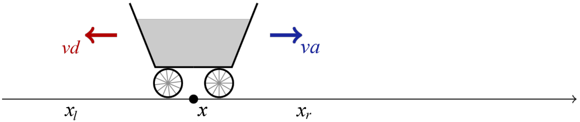
$$\langle{:=}\rangle\mathrm{I}\ \frac{\Gamma(x_0), x = f_x^{x_0} \vdash P}{\Gamma(x) \vdash \langle x := f\rangle P} \leadsto x := f; \boldsymbol{\alpha} \qquad \langle{:*}\rangle\mathrm{I}\ \frac{\Gamma(x_0), x = f_x^{x_0} \vdash P}{\Gamma(x) \vdash \langle x := *\rangle P} \leadsto x := f; \boldsymbol{\alpha}$$

$$\mathrm{dw}\ \frac{\Gamma(x_0), Q \vdash P}{\Gamma(x) \vdash [x' = f \,\&\, Q]P} \leadsto x := *; x' := f; ?Q; \boldsymbol{\alpha}$$

$$\mathrm{dc}\ \frac{\Gamma \vdash [x' = f \,\&\, Q]R \quad \Gamma \vdash [x' = f \,\&\, Q \land R]P}{\Gamma \vdash [x' = f \,\&\, Q]P} \leadsto \boldsymbol{\beta}$$

Second IH $\boldsymbol{\beta}$

Cart Proof Reifies Strategy



$$\text{safe} \equiv x_l < x_0 = x < x_r \rightarrow [\text{PP}](x = x_0)$$

$$\begin{aligned} \text{PP} \equiv \{ & \{vd := -1 \cup vd := 1\}; \\ & \{va := *; ?(-1 \leq va \leq 1)\}^d; \\ & \{x' = vd + va \,\&\, x_l \leq x \leq x_r\}\}^* \end{aligned}$$

$$\begin{aligned} \alpha_{PP} = \Big\{ & \{\{vd := -1; va := 1\} \\ & \cup \{vd := 1; va := -1\}\}; \\ & x := *; x' := vd + va; ?x = x_0 \Big\}^* \end{aligned}$$

Let $\mathcal{A}$ be standard mirroring strategy for PP, then $\mathcal{A} \rightsquigarrow \alpha_{PP}$

Theory

Let $\mathcal{A}$ be a proof of $(\Gamma \vdash [\alpha]P)$ and let $\mathcal{A} \rightsquigarrow \boldsymbol{\alpha}$. [1]

Theorem (Systemhood)

$\boldsymbol{\alpha}$ is a system, i.e., it does not contain dualities.

Theorem (Reification transfer)

$\Gamma \vdash [\boldsymbol{\alpha}]P$ is provable.

Theorem (Reification refinement)

$\Gamma \vdash \boldsymbol{\alpha} \leq_{[]} \alpha$ is provable.

[1] Recursively assume Γ free of duals β^d

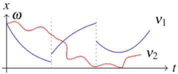
Refining

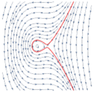
$\alpha \leq \beta \quad \alpha \cong \beta \qquad \alpha \cup \beta \leq \text{if}(P)\ \alpha \text{ else } \beta$

Constructive Differential Game Logic (CdGL)

[P]a[safe]	$\langle\alpha\rangle$safe	Witness a safe behavior of α
{P}a{safe}	$[\alpha]$safe	Observe α, universal proof of "safe"
	$\alpha \leq \beta$	Compute β strategy from α strategy

Hybrid



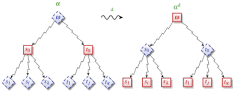


Games

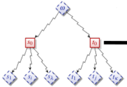
$\langle\alpha\rangle$safe

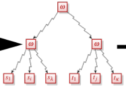
$[\alpha]$safe

α^d



Enables





$\Gamma \vdash [\text{game}]P \quad \leq \quad \Gamma \vdash [\text{system}]P$

Strategies to Systems

Synthesis