

S O K E N D A I

NII



物理情報システムの信頼性保証

ソフトウェア科学から、

制御理論・機械学習を包括する理論研究

蓮尾 一郎

国立情報学研究所 (NII) システム設計数理国際研究センター センター長・准教授

JST ERATO 蓮尾メタ数理システムデザインプロジェクト 研究総括

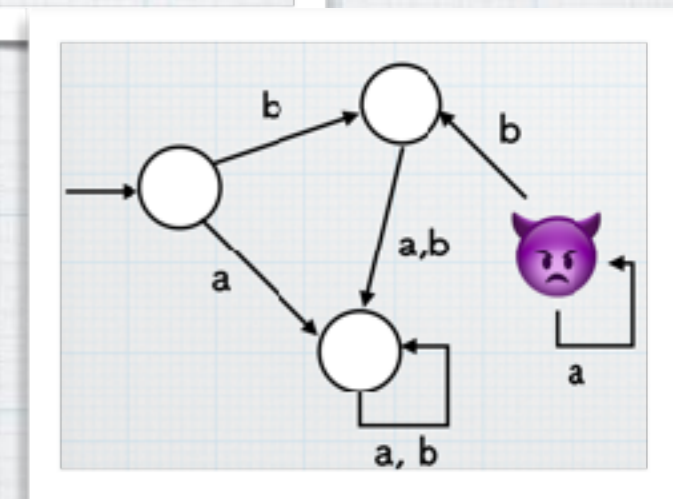
総合研究大学院大学 准教授

アウトライン



- * ERATO MMSD プロジェクト：
物理情報システムの信頼性保証支援
- * 形式手法 formal methods とは
 - * 定理証明, モデル検査
 - * 形式手法と制御理論
 - * 物理情報システムへの実世界応用へ
- * 技術紹介
 - * サーチベーステスト, 反例生成
 - * モニタリング
 - * 形式安全アーキテクチャ
 - * 形式仕様記述支援
 - * 機械学習システムの信頼性保証. 論理 vs 統計
- * プロジェクトの現状

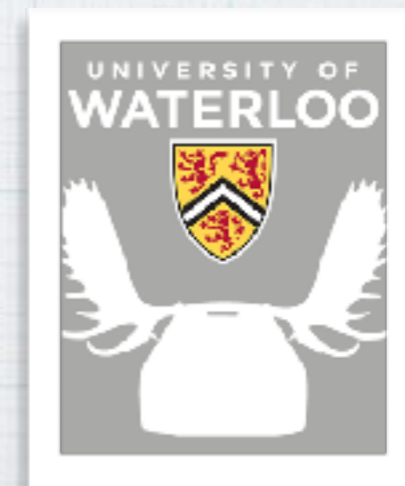
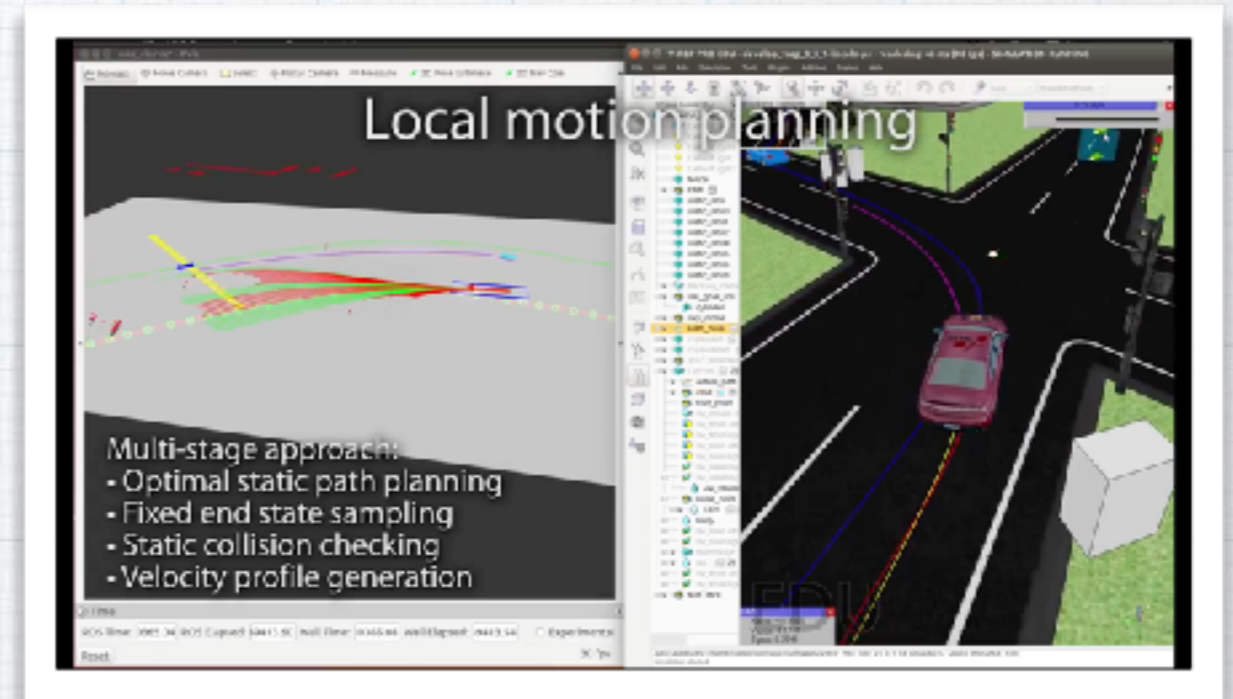
$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{[1]}}{B} \quad [\rightarrow E] \quad \frac{}{B \rightarrow C}^{[2]} \\
 \hline
 C \\
 \frac{}{A \rightarrow C}^{[\rightarrow I, 1]} \\
 \hline
 (B \rightarrow C) \rightarrow (A \rightarrow C) \quad [\rightarrow I, 2]
 \end{array}$$





ERATO MMSD 紹介

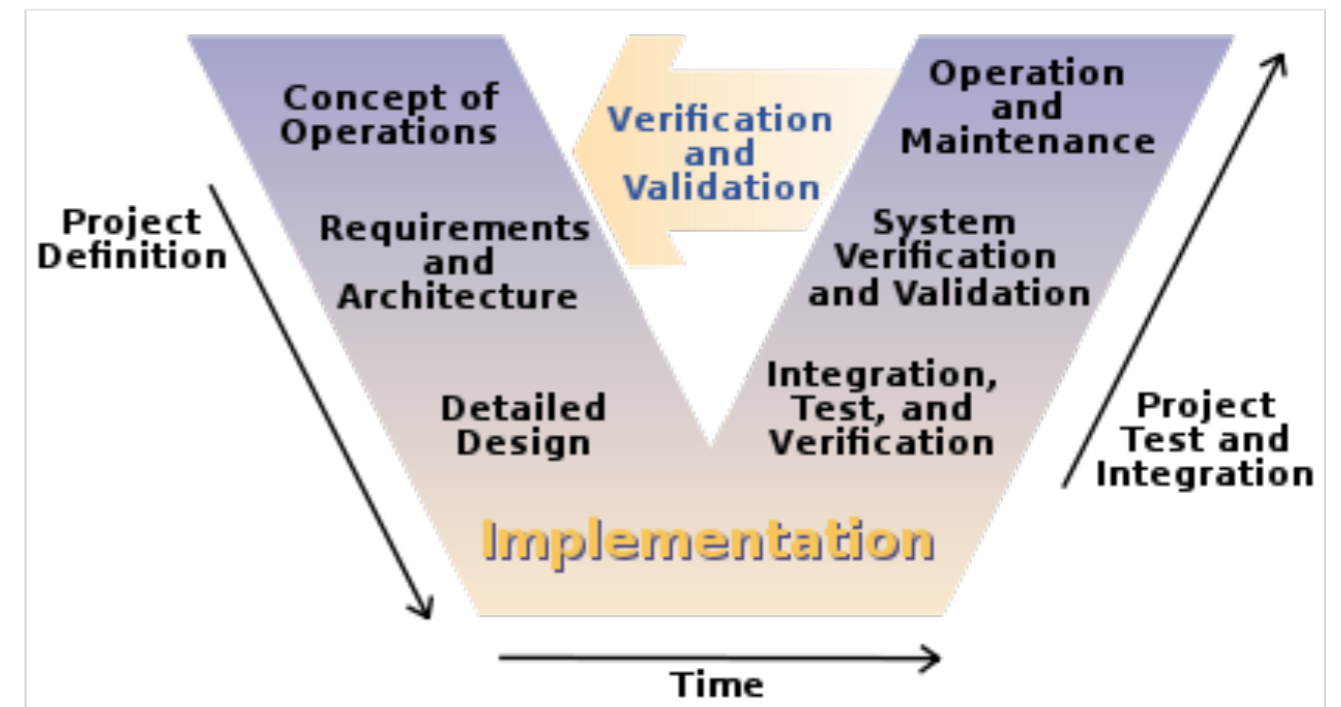
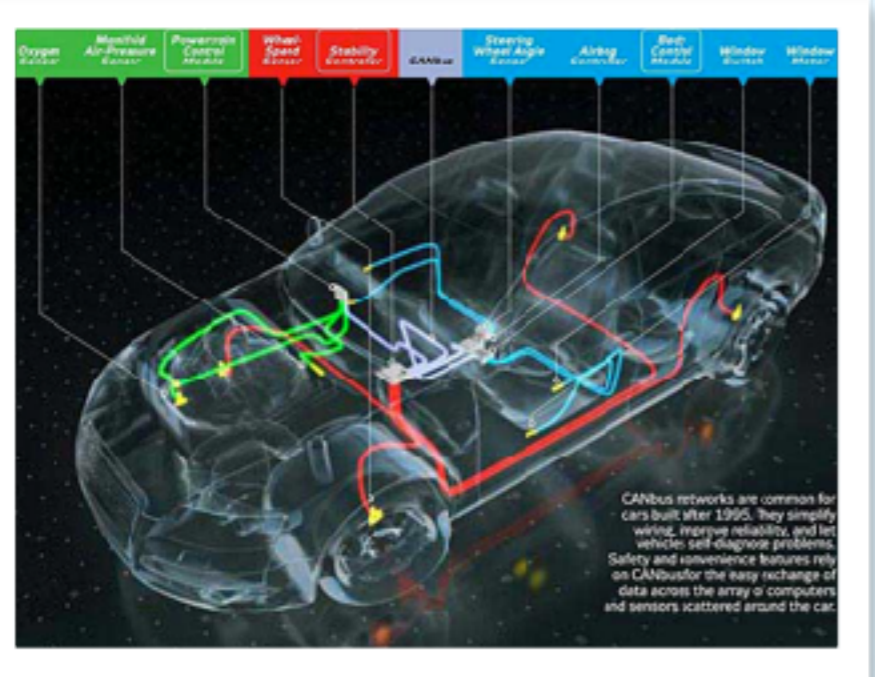
- * JST ERATO プロジェクト
 - * 2016/10-2022/03.
総勢50名規模の基礎研究プロジェクト
- * プロジェクト目標：工業製品の設計サポート
 - * 形式手法の拡張。ソフトウェアから物理情報システムへ
 - * 安全性・信頼性，
Verification & Validation.
「システムが期待通り動作するか」
 - * 対象：自動車，交通システム，生産システム
医療システム，…
→ IoT システム， Society 5.0
 - * 特に自動運転を戦略的ターゲットに。
U Waterloo と協働 www.autonomoose.net



今日の製造業における挑戦

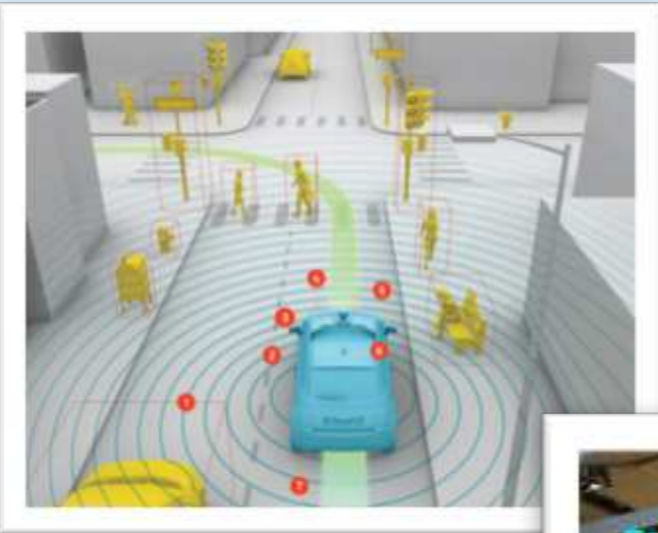
複雑さの爆発

http://www.popularmechanics.com/cars/how-to/a7386/how-it-works-the-computer-inside-your-car/



Clarus Concept of Operations. Publication No. FHWA-JPO-05-072, Federal Highway Administration (FHWA), 2005

新たな機能・応用



自動運転

wired.com

安全性保証と説明責任



オンデマンド
生産,
"Industrie 4.0"

automotiveit.eu

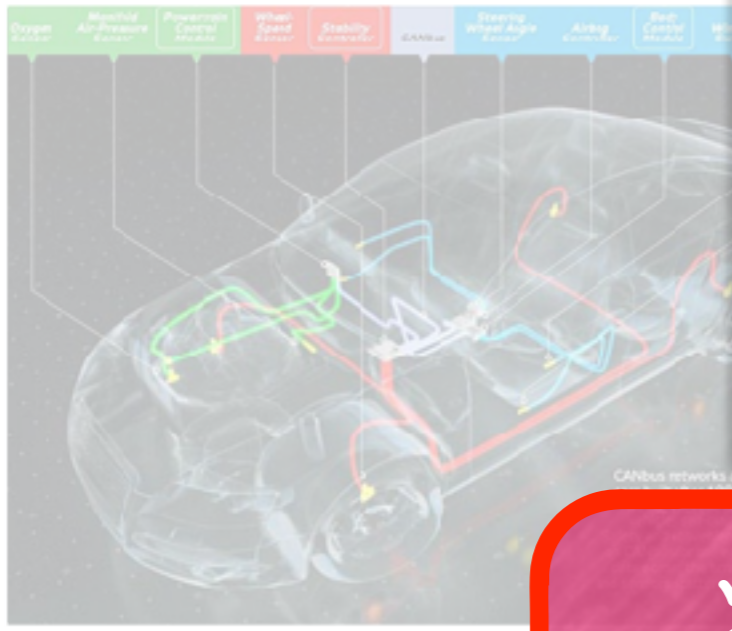


- * **性能・機能**が飛躍的に拡大, しかし
- * 工業製品の**複雑さ**も飛躍的に拡大



複雑さの爆発

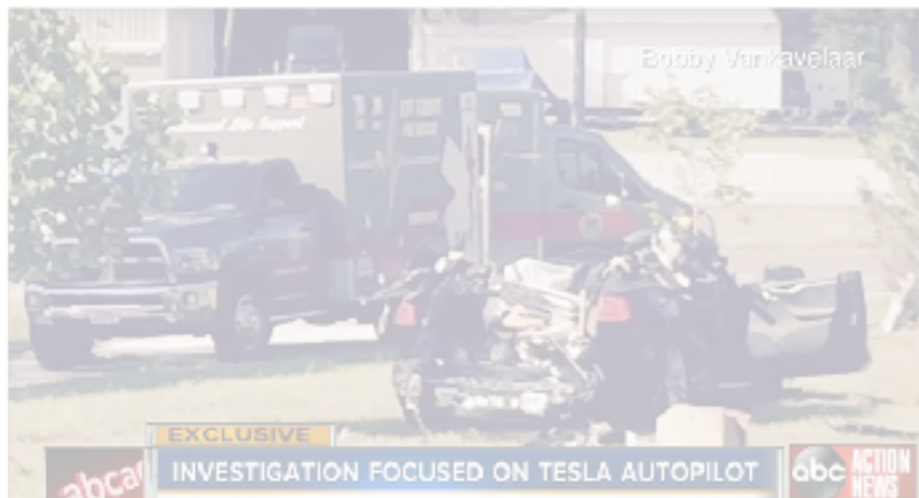
<http://www.popularmechanics.com/cars/how-to/a7386/how-it-works-the-computer-inside-your-car/>



現代の工業製品における
ソフトウェア制御が原因

ソフトウェアの品質向上手法を援用
→ **形式手法**の活用

安全性保証と説明責任



オンデマンド
生産,
"Industrie 4.0"
automotiveit.eu





ERATO MMSD: プロジェクト目標

* 数理的技法を用いた
 工業製品設計の支援。
 安全性, 信頼性



* 形式手法の拡張：
 ソフトウェアから
 物理情報システムへ

機械などの物理ダイナミクス
 (連続, アナログ)
 +
 計算機制御, ネットワーク
 (離散, デジタル)

$$A ::= \text{true} \mid \text{false} \mid A_1 \wedge A_2 \mid \neg A \mid a_1 < a_2 \mid \forall x \in {}^*N. A \mid \forall x \in {}^*\mathbb{R}. A$$

$F X \xrightarrow{\text{beh}_c} F Z$	$F X \xrightarrow{F f} F Y$
$c \uparrow$	$c \uparrow \quad \exists \quad \uparrow d$
$X \xrightarrow{\text{beh}_c} Z$	$X \xrightarrow{f} Y$
system behavior	simulation

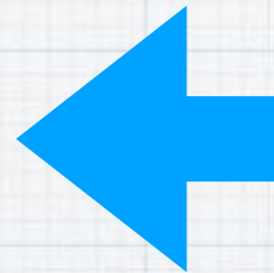
アウトライン



* ERATO MMSD プロジェクト：
 物理情報システムの信頼性保証支援

* 形式手法 formal methods とは

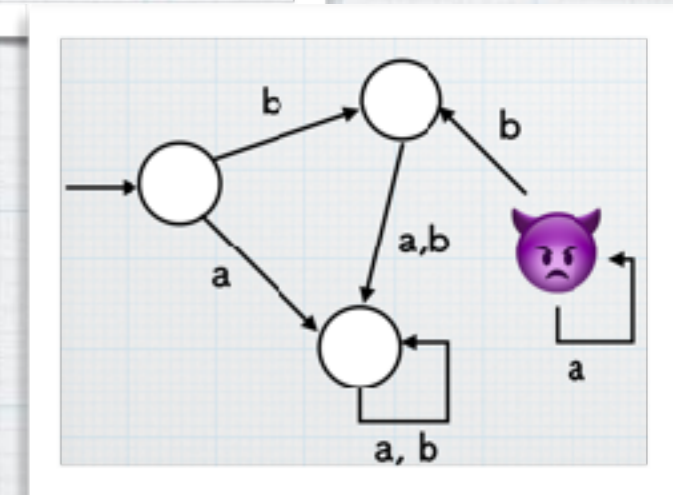
- * 定理証明, モデル検査
- * 形式手法と制御理論
- * 物理情報システムへの実世界応用へ



$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{(1)}}{B} \quad \frac{}{B \rightarrow C}^{(2)} \\
 \hline
 C \\
 \frac{}{A \rightarrow C}^{(1,1)} \\
 \hline
 (B \rightarrow C) \rightarrow (A \rightarrow C) \quad \text{[}\rightarrow\text{I, 2]}
 \end{array}$$

* 技術紹介

- * サーチベーステスト, 反例生成
- * モニタリング
- * 形式安全アーキテクチャ
- * 形式仕様記述支援
- * 機械学習システムの信頼性保証. 論理 vs 統計



* プロジェクトの現状

形式手法



- * (もともとは) ソフトウェアの品質保証のための、**数学的・記号論理的手法**の総体
- * 記号的であるため**計算機実装が可能**
- * 形式手法の例

計算機システムに対して多数の実績あり

- * IC design (Intel, ...)
- * device drivers (Microsoft)
- * light-weight FM (Facebook)
- * ...

形式検証 verification

* Input:

- * a system model \mathcal{M}
- * a specification φ

* Output: if $\mathcal{M} \models \varphi$ or not

- * w/ a proof, if yes
- * w/ a counterexample, if not

自動生成 synthesis

* Input:

- * a specification φ

* Output: a system \mathcal{M} such that $\mathcal{M} \models \varphi$

- * or: a parameter of a given (partial) model

形式仕様記述 specification

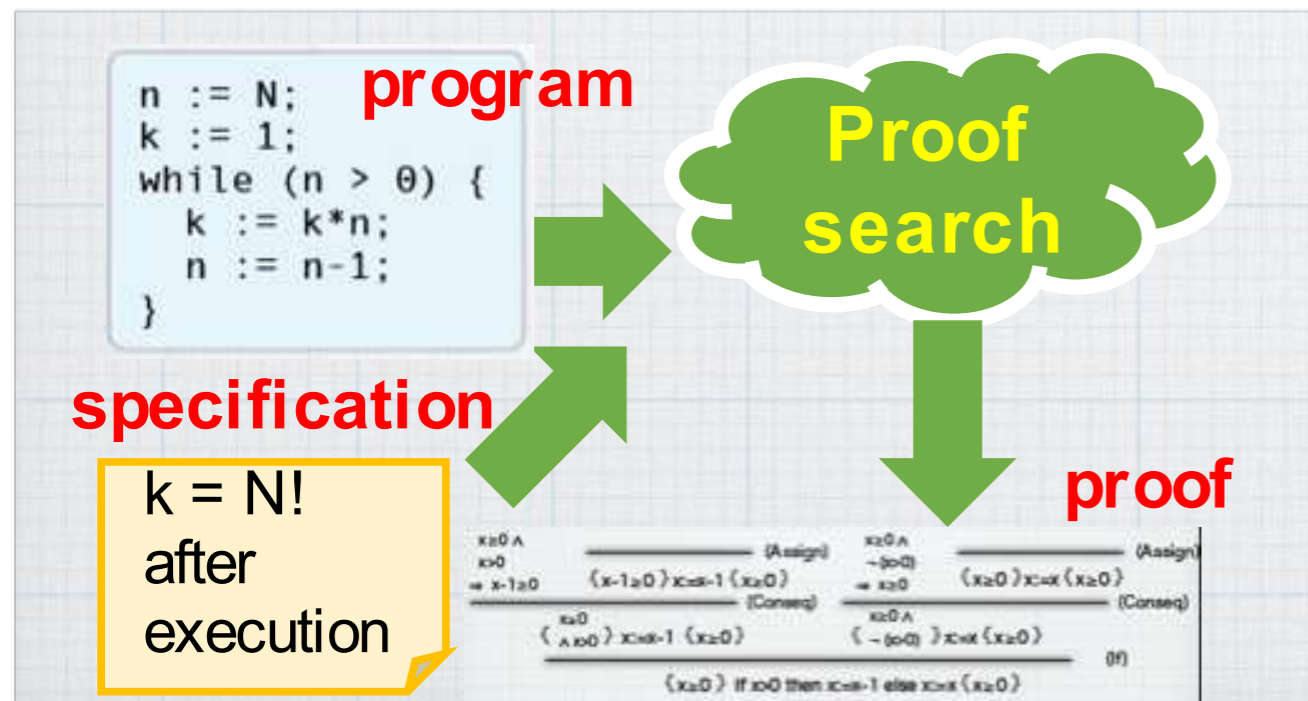
Expressing a property desired in a formal language

- * machine-representable
- * basis for verif. & synthesis

形式検証の例 1 : 定理証明

* 例: Hoare 論理による定理証明

- * イメージ: 紙とペンで証明を書くのと同じ
 - * しかし, 人力だと間違えるので, 計算機上で.
記号列で証明を表現 → 各ステップの正当性をプログラム (証明支援系) がチェック
 - * できれば証明の自動探索 (定理証明器)
- * (原理上は) 無限の入力空間をすべてカバーできる
 - * 証明では「i を入力 of 自然数とする」とかけますね
- * 数学的証明という, 強い品質保証



```

Theorem forall_exists : (forall P : Set->Prop,
    (forall x, ~(P x)) -> ~(exists x, P x)).
Proof.
  intros P.
  intros forall_x_not_Px.
  unfold not.
  intros exists_x_Px.
  destruct exists_x_Px as [ witness proof_of_Pwitness ].
  pose (not_Pwitness := forall_x_not_Px witness).
  unfold not in not_Pwitness.
  pose (proof_of_False := not_Pwitness proof_of_Pwitness).
  case proof_of_False.
Qed.
    
```

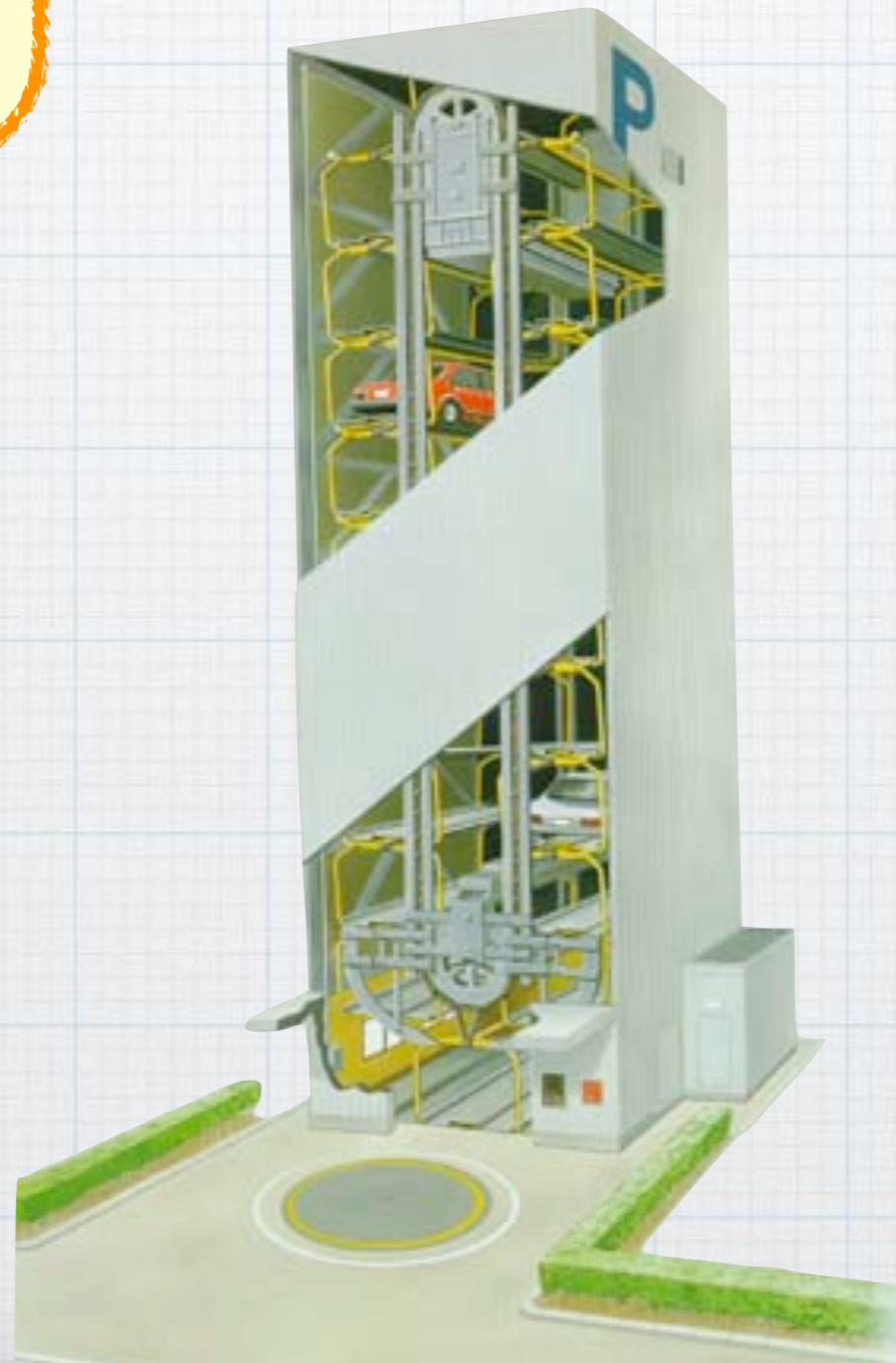
A Coq proof of $\forall x. \neg P(x) \rightarrow \neg \exists x. P(x)$

ビジネスでの証明

オレが責任持つっす！

絶対大丈夫っす！

この駐車場、
どうっすか！？



・・・
(ダメだこいつ)

ホントに？ なんで？

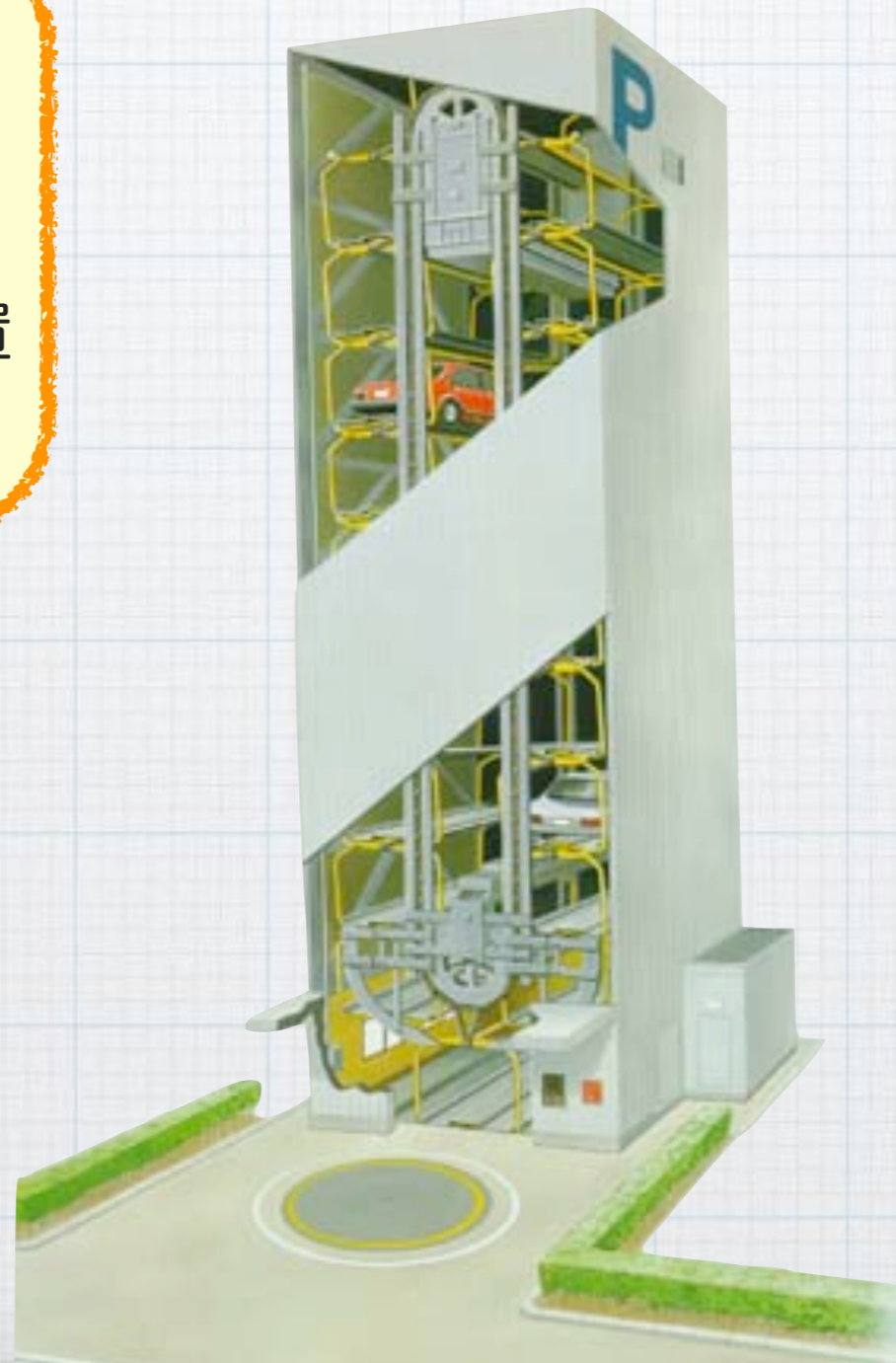
ゴンドラが衝突したり
りしない？ 大丈夫？

ビジネスでの証明

はい，大丈夫です。

なぜなら，任意の状態 s に対して，ゴンドラ g_1 の位置を x_1 とすると，...

この駐車場，
どうでしょう。



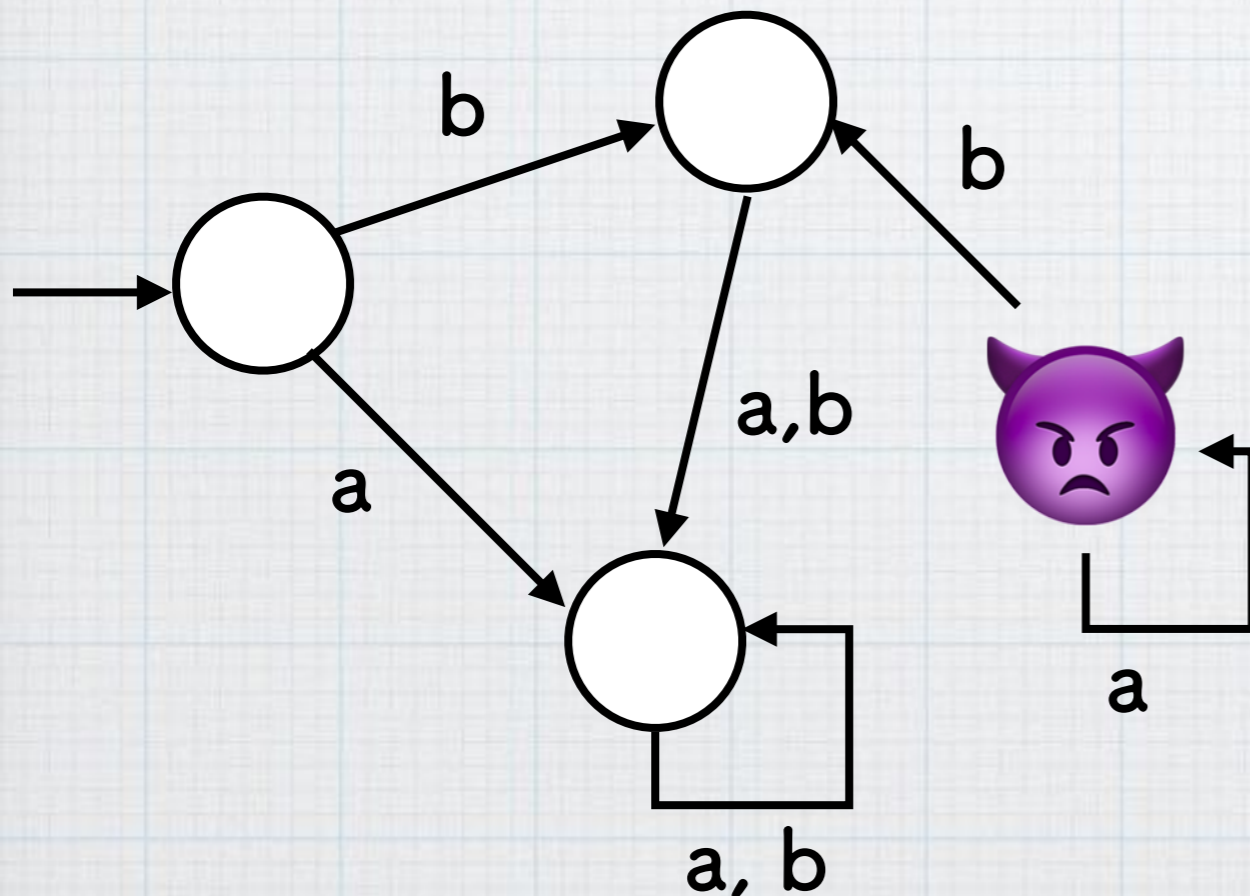
... (読んでる)
なるほど. それでは1
基いただこう.

ゴンドラが衝突した
りしない? 大丈夫?



形式検証の例 2 : モデル検査

- * オートマトンのアルゴリズムによる「自動証明」
 - * 主に**グラフの到達可能性判定**に帰着
- * オートマトンは有限 → 数え上げによる自動証明が可能
- * 例 : 以下のオートマトンにて, 🍆 に至ることはない (*)



* (不正解)

仕様 (*) をすべての入力について確かめる
(入力は無限個あるので無理)

* (正解)

到達可能な領域を計算し (グラフ探索, 有限時間で飽和), 🍆 が含まれないことを確かめる

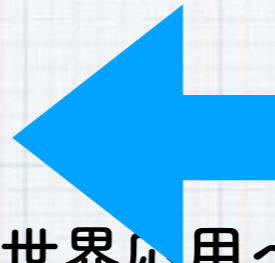
アウトライン



* ERATO MMSD プロジェクト：
 物理情報システムの信頼性保証支援

* 形式手法 formal methods とは

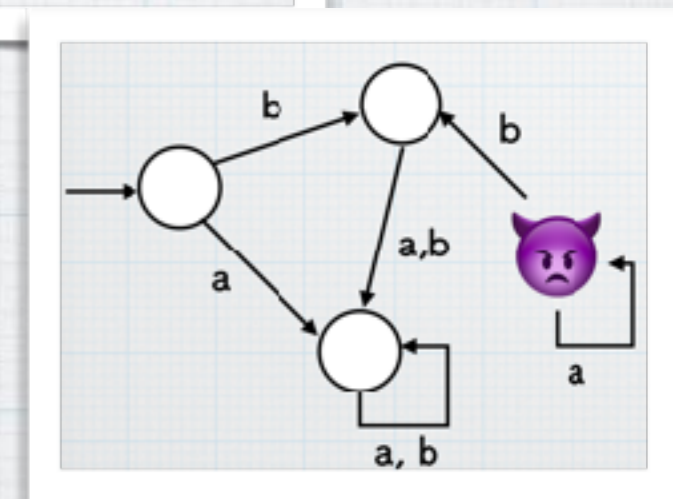
- * 定理証明, モデル検査
- * 形式手法と制御理論
- * 物理情報システムへの実世界応用へ



$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{(1)}}{B} \quad [\rightarrow E] \quad \frac{}{B \rightarrow C}^{(2)} \\
 \hline
 C \quad [\rightarrow E] \\
 \hline
 \frac{C}{A \rightarrow C} \quad [\rightarrow I, 1] \\
 \hline
 \frac{}{(B \rightarrow C) \rightarrow (A \rightarrow C)} \quad [\rightarrow I, 2]
 \end{array}$$

* 技術紹介

- * サーチベーステスト, 反例生成
- * モニタリング
- * 形式安全アーキテクチャ
- * 形式仕様記述支援
- * 機械学習システムの信頼性保証. 論理 vs 統計



* プロジェクトの現状

形式手法の歴史（超ダイジェスト）

ソフトウェア
の大規模化

ネットワー
ク，分散並列

物理システム
との融合

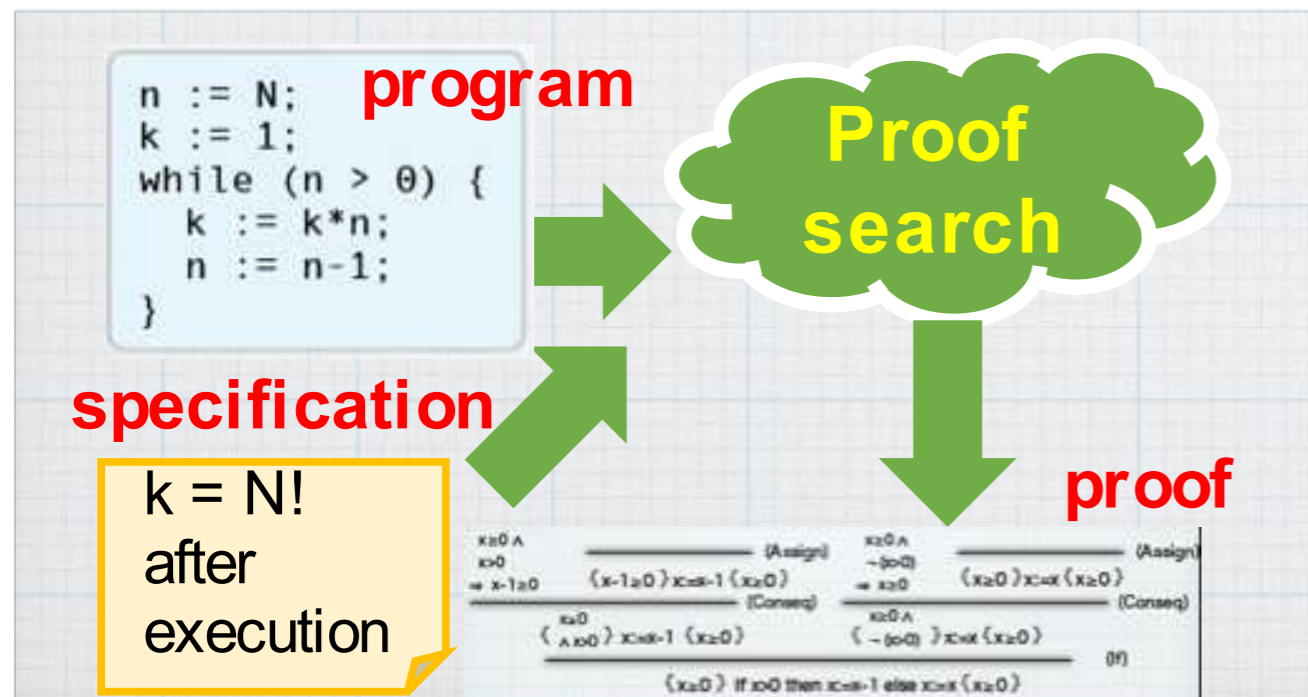
統計的機械学
習の能力向上

- * 1970頃 形式手法によるソフトウェア品質保証
（ソフトウェア検証）の研究開始。定理証明，モデル検査
- * 1990頃 ソフトウェア検証のツール化・実応用加速
定理証明： Isabelle, Coq, PVS, SMT solvers (Z3, ...), ...
モデル検査： SPIN, SMV/NuSMV, mCRL2, PRISM, Uppaal,
- * 2006 物理情報システム (Cyber-Physical Systems, CPS)
への応用開始。ソフトウェア検証 + 制御理論
（この経緯は [奥村, 研究技術計画 2017] に詳しい）
- * 2016 機械学習システムへの応用。
形式的推論と統計的推論

形式検証の例 1 : 定理証明 (再掲)

* 例: Hoare 論理による定理証明

- * イメージ: 紙とペンで証明を書くのと同じ
 - * しかし, 人力だと間違えるので, 計算機上で.
記号列で証明を表現 → 各ステップの正当性をプログラム (証明支援系) がチェック
 - * できれば証明の自動探索 (定理証明器)
- * (原理上は) 無限の入力空間をすべてカバーできる
 - * 証明では「i を入力 of 自然数とする」とかけますね
- * 数学的証明という, 強い品質保証



```

Theorem forall_exists : (forall P : Set->Prop,
    (forall x, ~(P x)) -> ~(exists x, P x)).
Proof.
  intros P.
  intros forall_x_not_Px.
  unfold not.
  intros exists_x_Px.
  destruct exists_x_Px as [ witness proof_of_Pwitness ].
  pose (not_Pwitness := forall_x_not_Px witness).
  unfold not in not_Pwitness.
  pose (proof_of_False := not_Pwitness proof_of_Pwitness).
  case proof_of_False.
Qed.
    
```

A Coq proof of $\forall x. \neg P(x) \rightarrow \neg \exists x. P(x)$

定理証明：物理情報システムへ

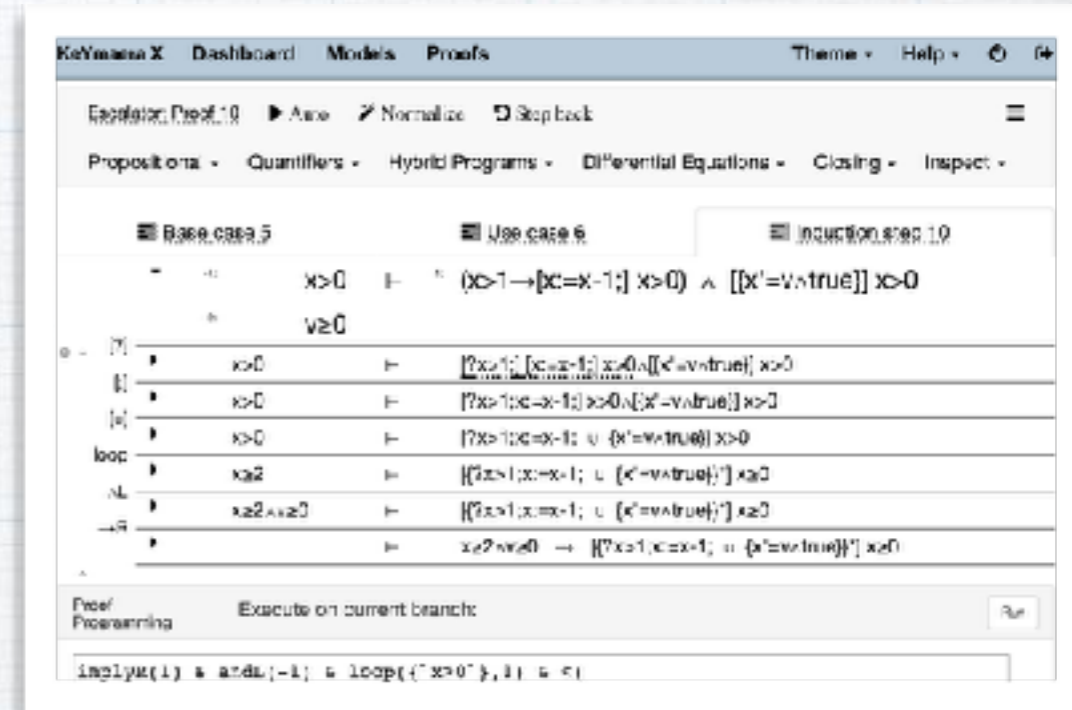
* 物理情報システムに適用可能な例：
 differential dynamic logic [Platzer]

- * André Platzer: Logical Foundations of Cyber-Physical Systems. Springer 2018
- * 本質的に：Hoare 論理 + 微分方程式
- * ツール：KeYmaera X

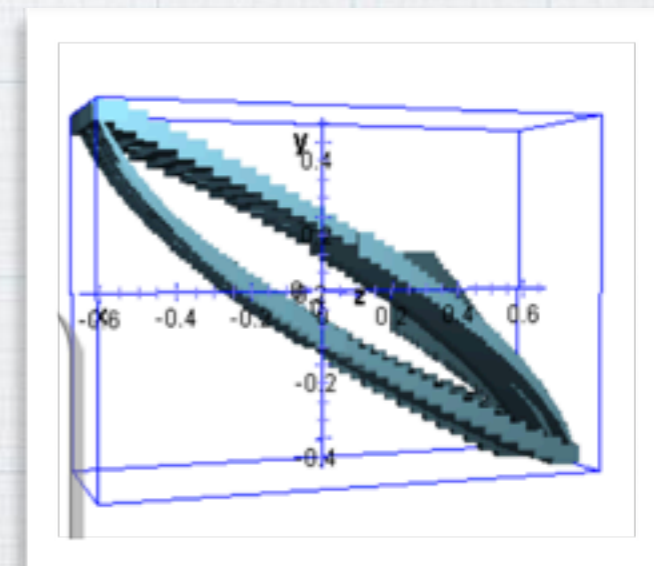
- * 自動定理証明系として（自動証明モード）
- * 証明支援系として（対話証明モード）

* 実用上のチャレンジ：論理体系が代数的

- * 厳密で，単純なホワイトボックスモデルが必要
 → 比較的単純なシステムの検証に応用が限られる
- * ブラックボックスコンポーネントや，外乱など，不確かさのモデリングが容易ではない



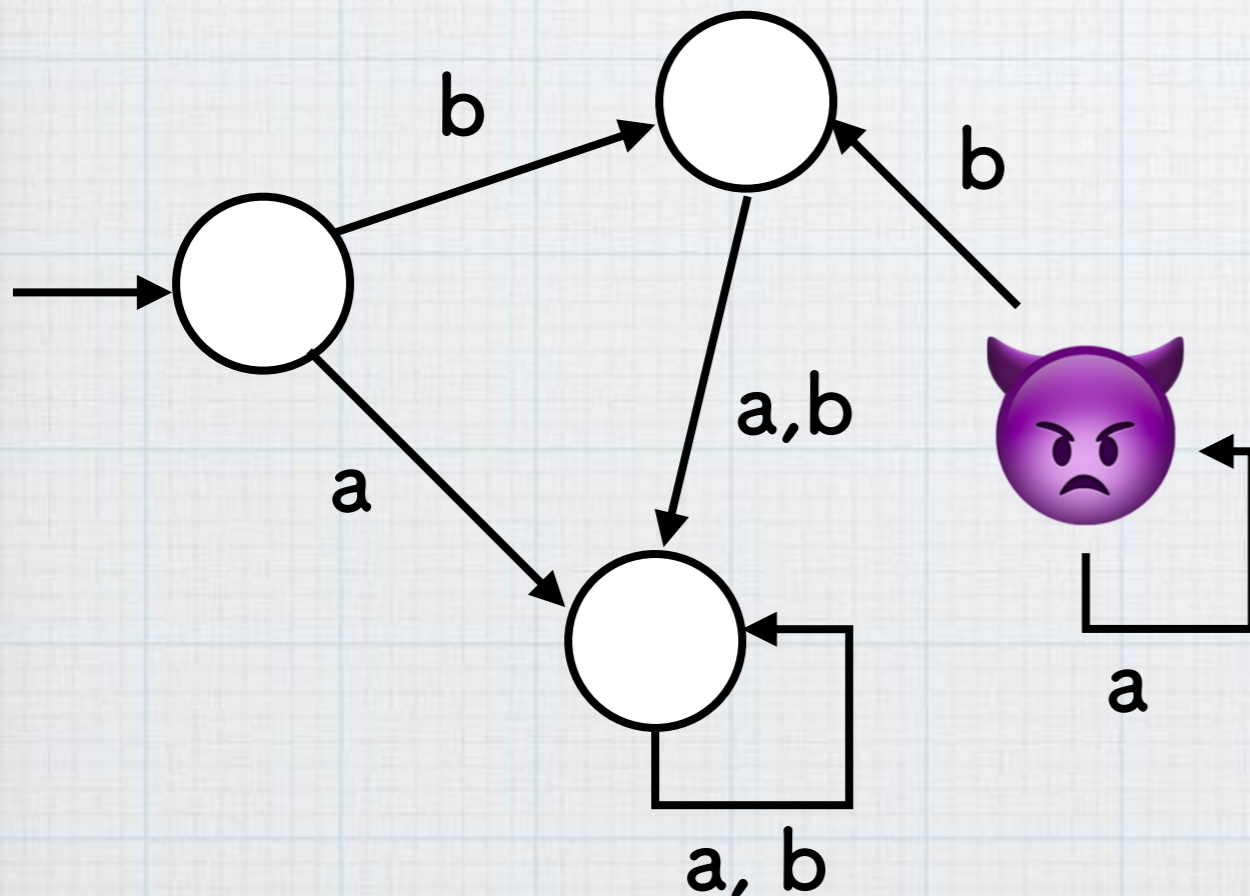
KeYmaera X スクリーンショット



参考：SpaceEx スクリーンショット
 制度保証付き数値計算による可達性解析

形式検証の例 2 : モデル検査 (再掲)

- * オートマトンのアルゴリズムによる「自動証明」
 - * 主に**グラフの到達可能性判定**に帰着
- * オートマトンは有限 → 数え上げによる自動証明が可能
- * 例 : 以下のオートマトンにて, 🍆 に至ることはない (*)



- * (不正解)

仕様 (*) をすべての入力について確かめる
(入力は無限個あるので無理)

- * (正解)

到達可能な領域を計算し (グラフ探索, 有限時間で飽和), 🍆 が含まれないことを確かめる



モデル検査：物理情報システムへ

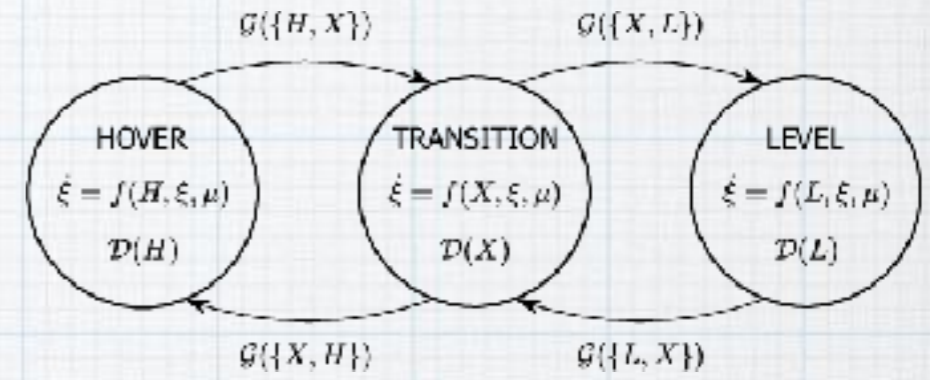
* 物理情報システムへの適用

* 一義的には、**連続量が現れた時点でアウト**

* 状態空間が無限になって、探索できない

* 例：ハイブリッド・オートマトン

(オートマトン+微分方程式) では、到達可能性が決定不能



ハイブリッド・オートマトンの例
 from [Casau+, CDC'11]

* 確率的オートマトンならばOK

* Markov chain, Markov decision process, ...

* 到達可能性判定の代わりに、到達確率を計算する

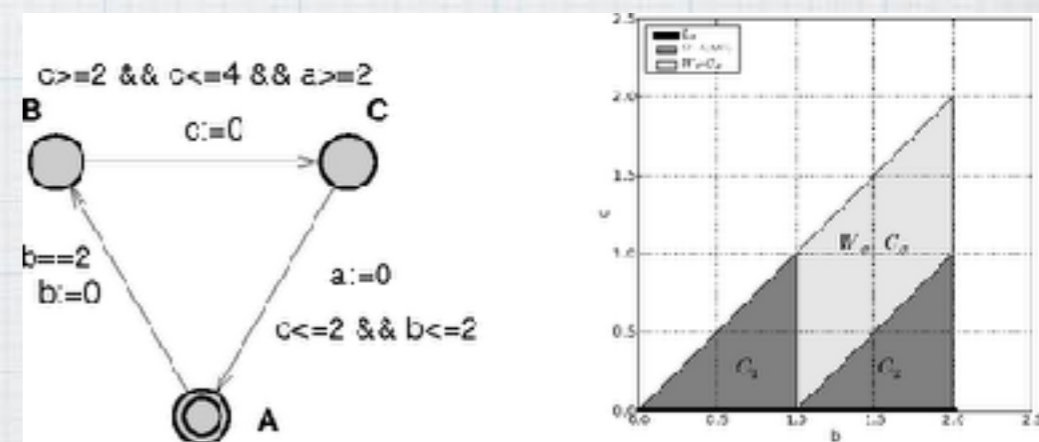
(線形計画法で解ける)

* 時間付きオートマトン timed automaton も OK

* 連続量の領域が、決まったテンプレート

(region, zone) で記述できる

* region, zone は有限個しかない



時間付きオートマトンと zone graph
 from [Daws+, FORMATS'06]

形式手法の歴史（超ダイジェスト，再掲）

ソフトウェア
の大規模化

ネットワー
ク，分散並列

物理システム
との融合

統計的機械学
習の能力向上

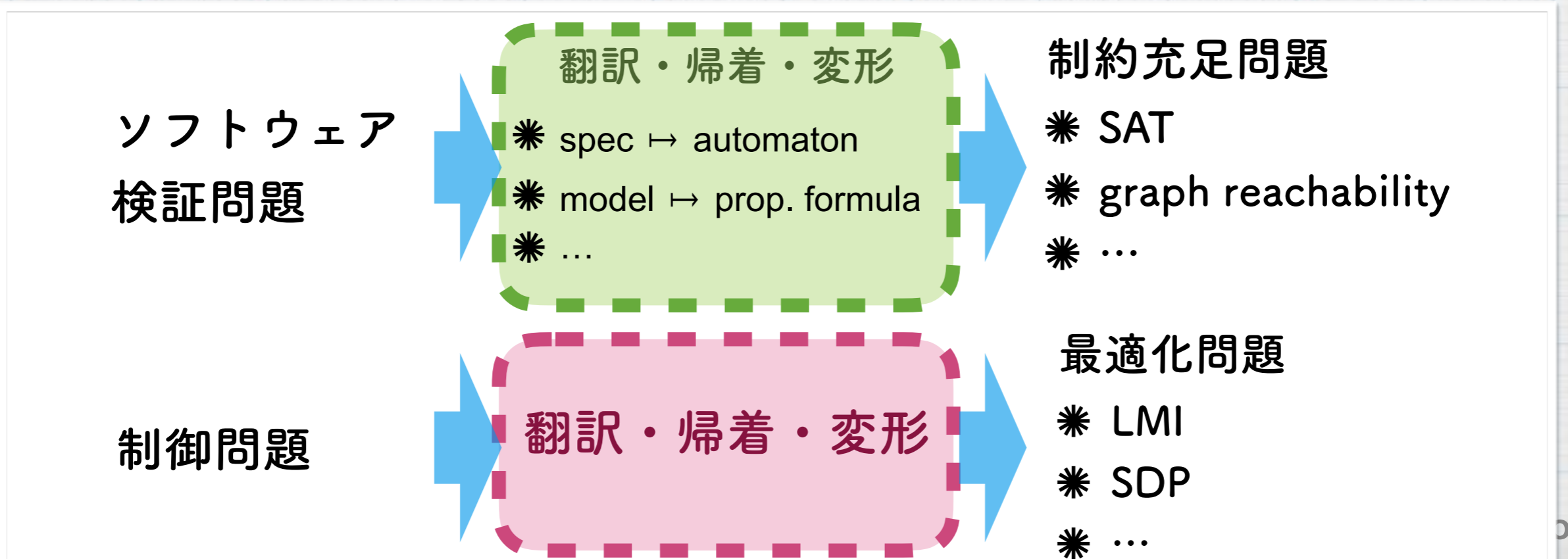
- * 1970頃 形式手法によるソフトウェア品質保証
（ソフトウェア検証）の研究開始。定理証明，モデル検査
- * 1990頃 ソフトウェア検証のツール化・実応用加速
定理証明： Isabelle, Coq, PVS, SMT solvers (Z3, ...), ...
モデル検査： SPIN, SMV/NuSMV, mCRL2, PRISM, Uppaal,
- * 2006 物理情報システム (Cyber-Physical Systems, CPS)
への応用開始。ソフトウェア検証 + 制御理論
（この経緯は [奥村, 研究技術計画 2017] に詳しい）
- * 2016 機械学習システムへの応用。
形式的推論と統計的推論

制御理論 と 形式手法・ソフトウェア科学

* 対比

制御理論		形式手法・ソフトウェア科学
連続空間・連続時間	対象システム	離散空間・離散時間
微分方程式	モデリング	オートマトン, プログラム
連続量最適化 LMI, SDP, ...	最終的な 解き方	グラフアルゴリズム SATソルバ, SMTソルバ
機械, 電気	主な応用対象	ソフトウェア

* しかしワークフローは共通. 「ダイナミクスを解きほぐす」





これまでの物理情報システム研究

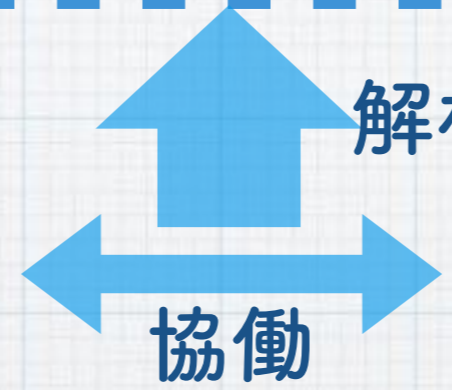
(特に安全性・信頼性の側面)



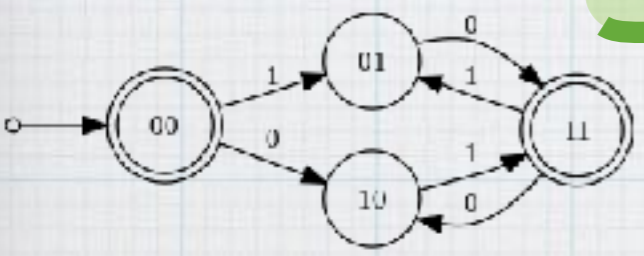
物理情報システム
(特に hybrid system)

形式手法

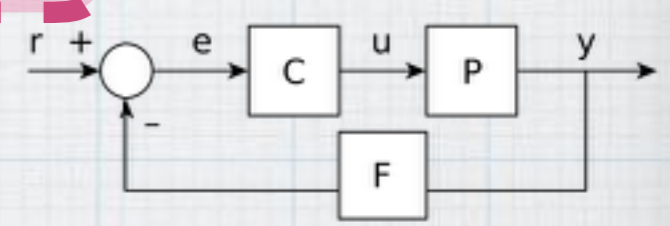
制御理論



$\square(p \Rightarrow \diamond q)$



$x' = f(x, u)$



追いつかなければ…??

- * 物理情報システム研究への取り組みでは**欧米が先行**
- * 米国：NSF の CPS 研究イニシアチブ (2006-)
- * 欧州：形式手法の定量的拡張 (確率, 時間, ...)



これまでの物理情報システム研究

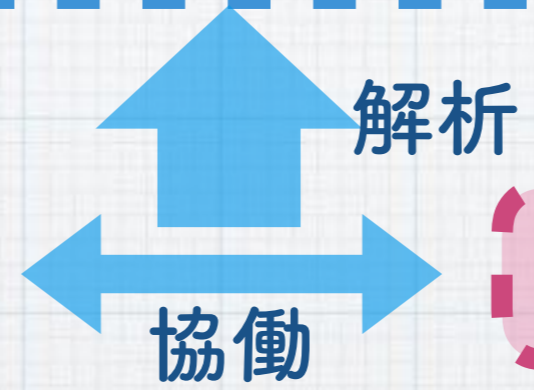
(特に安全性・信頼性の側面)



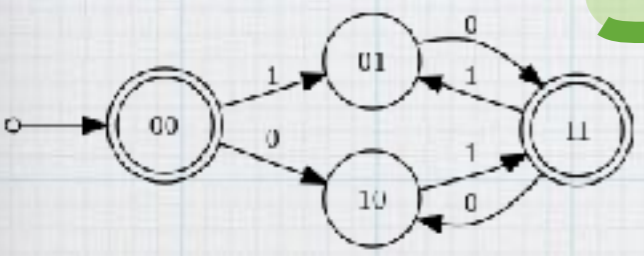
物理情報システム
(特に hybrid system)

形式手法

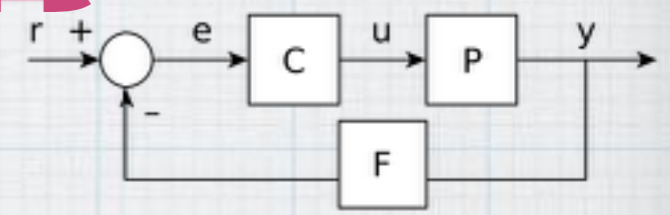
制御理論



$\square(p \Rightarrow \diamond q)$



$x' = f(x, u)$



* 課題：実システムに対するスケーラビリティ

- * ホワイトボックスモデルの完全な理解が前提
- * 結論の数学的正しさを絶対視
- * 不確かさを許容する余地が少ない → 統計的機械学習との相性の問題

可能？
役に立つ？

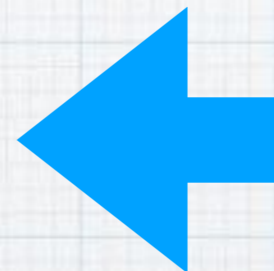
アウトライン



* ERATO MMSD プロジェクト：
 物理情報システムの信頼性保証支援

* 形式手法 formal methods とは

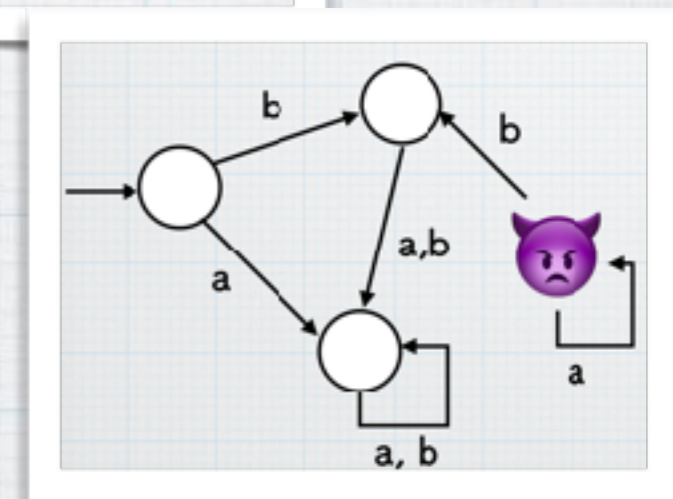
- * 定理証明, モデル検査
- * 形式手法と制御理論
- * 物理情報システムへの実世界応用へ



$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{[1]}}{B} \quad \frac{}{B \rightarrow C}^{[2]} \\
 \hline
 C \\
 \frac{}{A \rightarrow C}^{[1,1]} \\
 \hline
 (B \rightarrow C) \rightarrow (A \rightarrow C) \quad [1, 2]
 \end{array}$$

* 技術紹介

- * サーチベーステスト, 反例生成
- * モニタリング
- * 形式安全アーキテクチャ
- * 形式仕様記述支援
- * 機械学習システムの信頼性保証. 論理 vs 統計



* プロジェクトの現状

ERATO MMSD の研究開発の目標：
これらの柔軟なミックス → 実システムへ応用

ソフトウェア品質保証におけるスペクトル

形式手法

テスト test

モデル検査

自動定理証明

対話定理証明

低 コスト 高

ブラックボックス
モデルを許容

ホワイトボックスモデルを要求

高 自動化の度合い 低

自動

専門家を月・年
単位で拘束

低 保証の網羅性 高

低 保証の数学的厳密性 高

経験論的保証

数学的保証 → 「形式検証」

スケールダウンできない形式手法

- * 形式手法（特に形式検証）の多くは
スケールダウンできない
 - * ホワイトボックスモデルを要求。
全てのコンポーネントの形式モデル
 - * 既存の設計プロセスの刷新を要求
 - * 品質保証が非ゼロになるまでのコストが莫大
- * 物理情報システムへの適用例：
航空機（エアバス），宇宙開発（NASA），...
 - * 失敗のコストが莫大であり，
コストをかけてもペイする
 - * それでも数十年単位の取り組み
 - * 物理系の融合は困難
→ ソフトウェアの側面に注力

品質保証の度合い

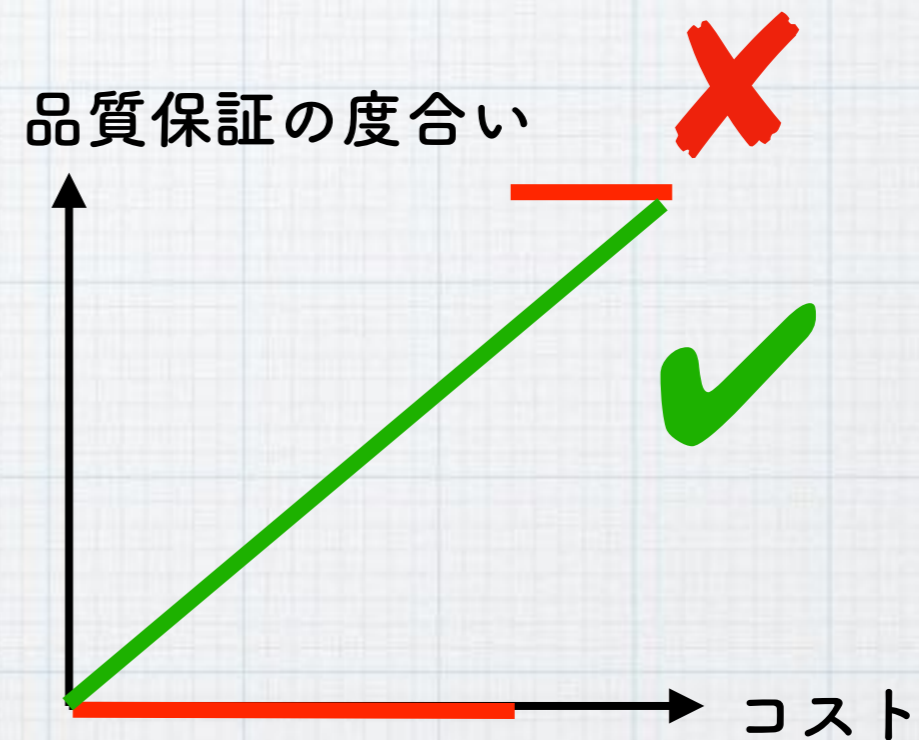
数学的証明 →
品質保証度は高い

コスト

ホワイトボックスモデル
の作成コスト

スケールダウンできる形式手法へ

- * 自動車などの実製品への応用では、
「スケールダウンできること」
が重要
- * 半分のコストしかかけられない
場合に、
- * 品質保証度合いがゼロではなく、
半分でも得られるように



アウトライン



* ERATO MMSD プロジェクト：
 物理情報システムの信頼性保証支援

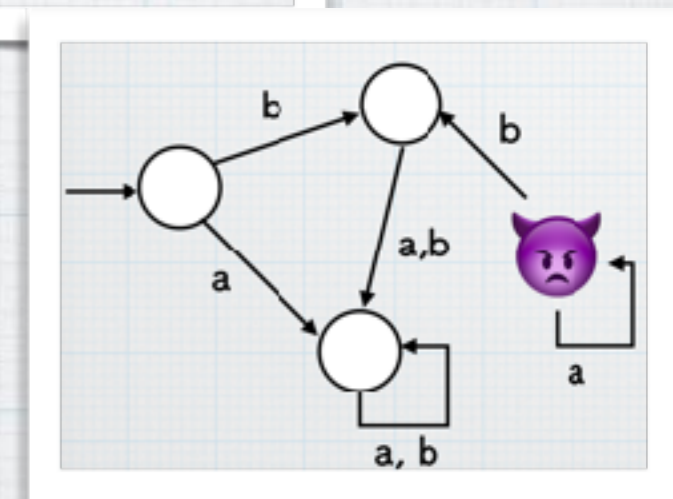
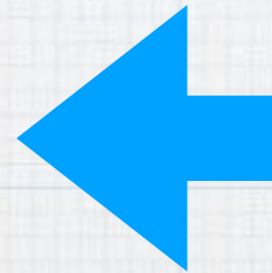
* 形式手法 formal methods とは

- * 定理証明, モデル検査
- * 形式手法と制御理論
- * 物理情報システムへの実世界応用へ

$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda} \quad [1]}{B} \quad [\rightarrow E] \quad \frac{}{B \rightarrow C} \quad [2]}{C} \quad [\rightarrow E] \\
 \frac{}{A \rightarrow C} \quad [\rightarrow I, 1] \\
 \frac{}{(B \rightarrow C) \rightarrow (A \rightarrow C)} \quad [\rightarrow I, 2]
 \end{array}$$

* 技術紹介

- * サーチベーステスト, 反例生成
- * モニタリング
- * 形式安全アーキテクチャ
- * 形式仕様記述支援
- * 機械学習システムの信頼性保証. 論理 vs 統計



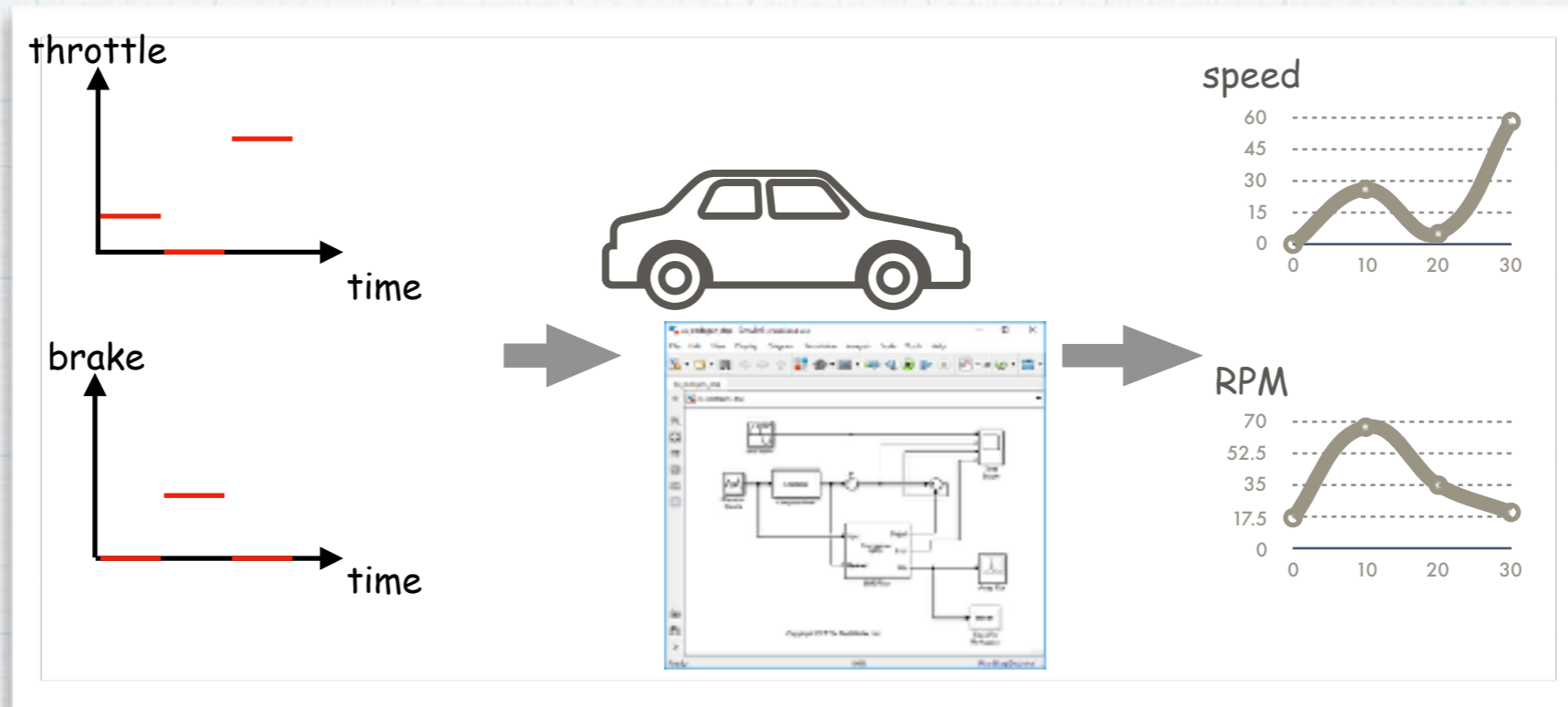
* プロジェクトの現状

強化学習によるサーチベーステスト： Simulink モデルの反例生成

[Akazaki & Hasuo, CAV'15]

[Zhang, Ernst, Sedwards, Arcaini & Hasuo, EMSOFT'18]

[Zhang, Hasuo & Arcaini, CAV'19] ...



* **ブラックボックス**テスト.

入出力の対応を見ながら，反例となる入力信号を探索

* **Given: Simulink モデル M ，時相論理仕様 ϕ**

* 仕様 ϕ の例：

ギヤが4速になったらそれから3秒以内に時速 50 km/h 以上になる

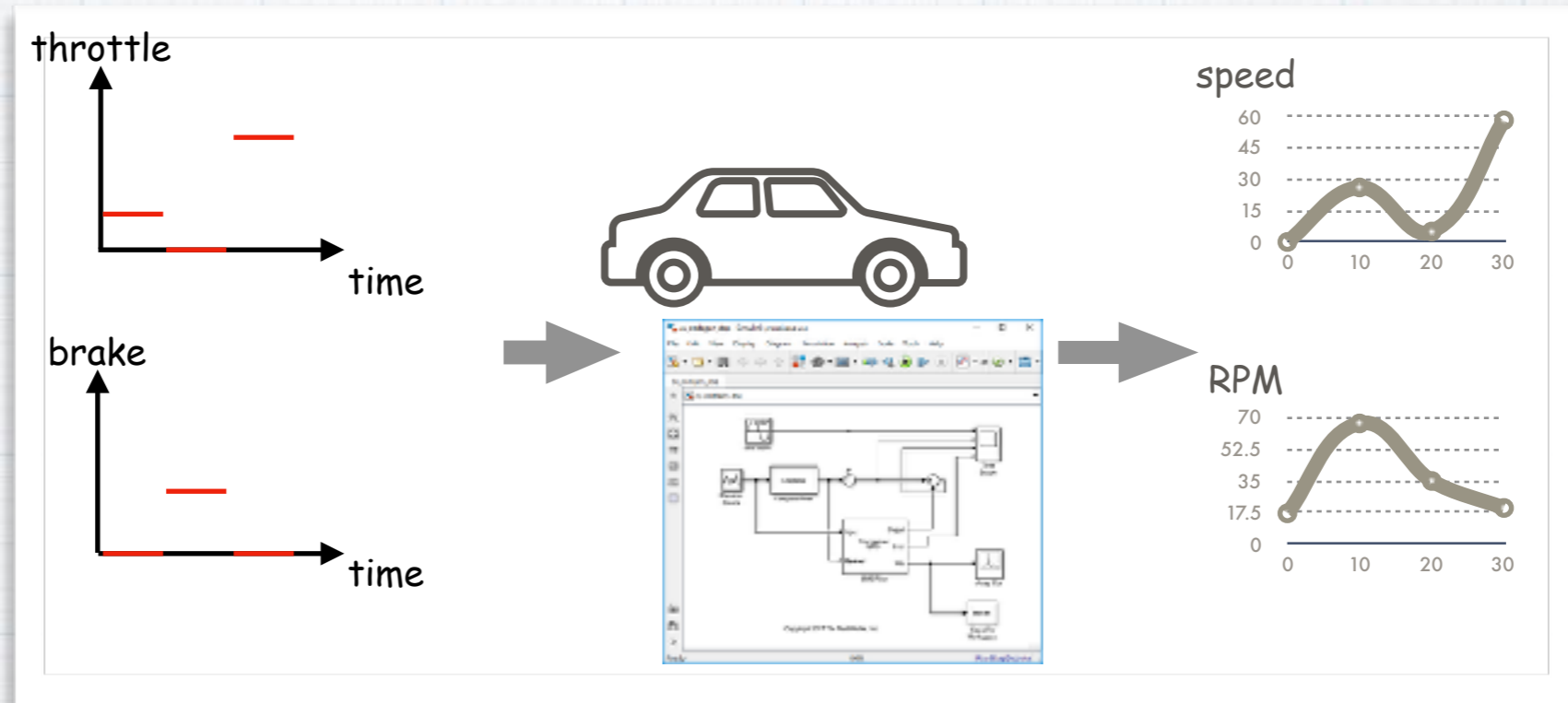
* **目標： 出力信号 $M(i)$ が ϕ をみたさないような，入力信号 i**

強化学習によるサーチベーステスト： Simulink モデルの反例生成

[Akazaki & Hasuo, CAV'15]

[Zhang, Ernst, Sedwards, Arcaini & Hasuo, EMSOFT'18]

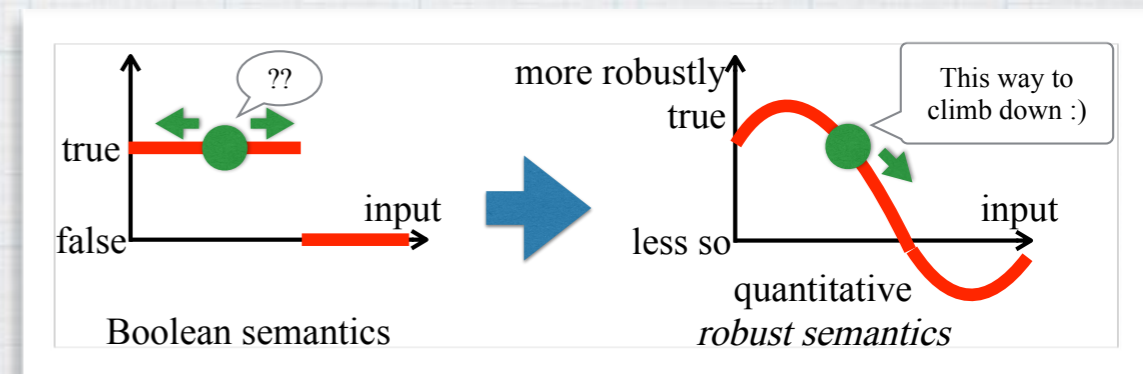
[Zhang, Hasuo & Arcaini, CAV'19] ...



* 主流のアプローチ [Fainekos & Pappas, TCS'09] :

* 強化学習, 確率的最適化で
 「あぶないところ」を狙う

* そのため, 時相論理式 ϕ の真偽値を,
 ブール値 (2値) から連続実数値に拡張
 → 勾配降下法

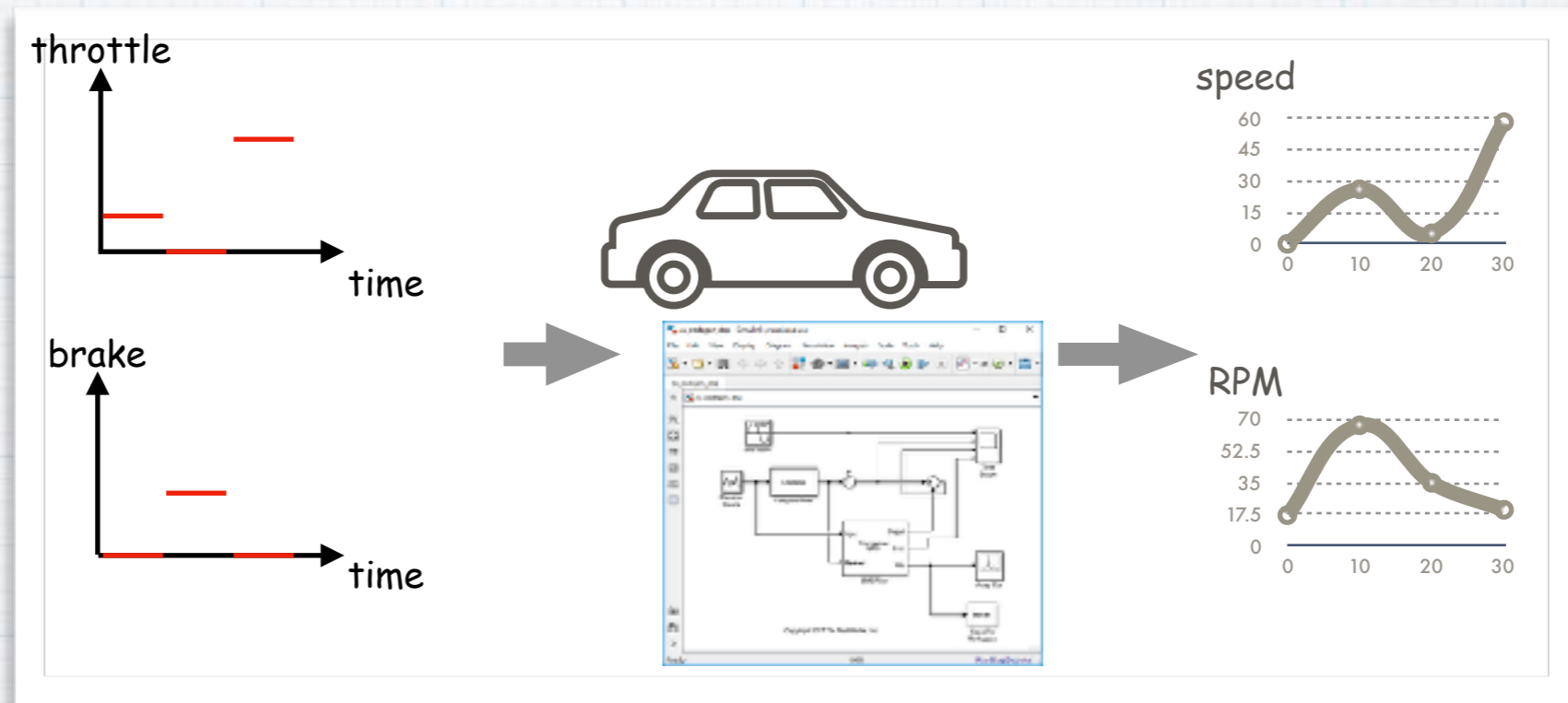


強化学習によるサーチベーステスト： Simulink モデルの反例生成

[Akazaki & Hasuo, CAV'15]

[Zhang, Ernst, Sedwards, Arcaini & Hasuo, EMSOFT'18]

[Zhang, Hasuo & Arcaini, CAV'19] ...



* ユースケース

* **ブラックボックステスト**. 効率的にバグを発見

* システムモデルは Simulink でなくてもよい.
Hardware-in-the-Loop でも可

* 注： 検証ではない. 「バグが存在しないこと」は結論できない

* **制御器生成**

強化学習によるサーチベーステスト：

Simulink モデルの反例生成

[Akazaki & Hasuo, CAV'15]

[Zhang, Ernst, Sedwards, Arcaini & Hasuo, EMSOFT'18]

[Zhang, Hasuo & Arcaini, CAV'19] ...

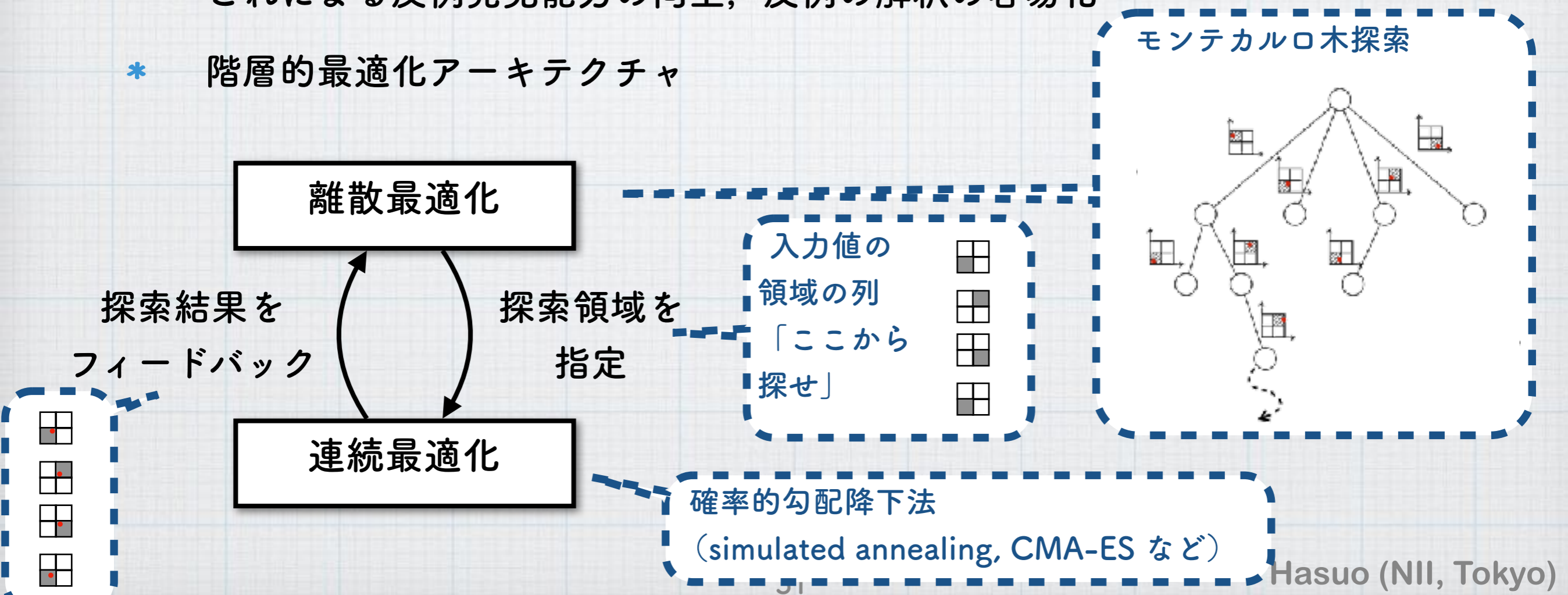
* Survey: [Kapinski+, IEEE Control Syst. '16]

J. Kapinski, J. V. Deshmukh, X. Jin, H. Ito, and K. Butts, "Simulation-based approaches for verification of embedded control systems: An overview of traditional and advanced modeling, testing, and verification techniques," IEEE Control Syst., vol. 36, no. 6, pp. 45–64, Dec. 2016.

* ERATO MMSD の貢献 [Zhang+, EMSOFT'18] を例に。 [Zhang+, CAV'19] も同種のアプローチ

* 確率的勾配降下法における **離散構造の利用**，
 これによる反例発見能力の向上，反例の解釈の容易化

* 階層的最適化アーキテクチャ



アウトライン



* ERATO MMSD プロジェクト：
 物理情報システムの信頼性保証支援

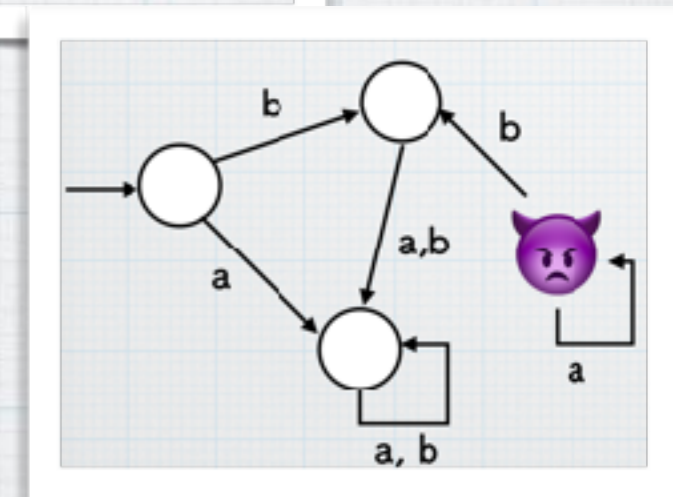
* 形式手法 formal methods とは

- * 定理証明, モデル検査
- * 形式手法と制御理論
- * 物理情報システムへの実世界応用へ

$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{(1)}}{B} \quad \frac{}{B \rightarrow C}^{(2)} \\
 \hline
 C \\
 \frac{}{A \rightarrow C}^{(1,1)} \\
 \hline
 (B \rightarrow C) \rightarrow (A \rightarrow C) \quad \text{[}\rightarrow\text{I, 2]}
 \end{array}$$

* 技術紹介

- * サーチベーステスト, 反例生成
- * モニタリング
- * 形式安全アーキテクチャ
- * 形式仕様記述支援
- * 機械学習システムの信頼性保証. 論理 vs 統計



* プロジェクトの現状

モニタリング：さまざまな定式化

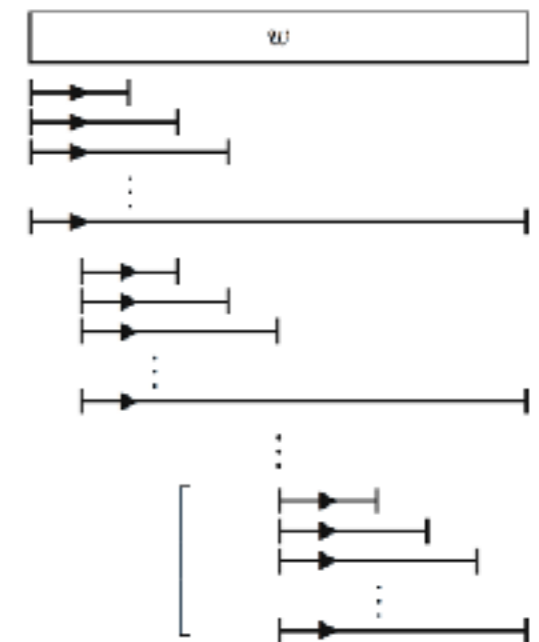
- * 関連トピック（しばしば同義語？）：
実行時検証，実行時監視，パターンマッチング，…
- * Given: 離散時間ログ $w = \text{abaaacb}\cdots\text{bbc}$
仕様 ϕ 「b の後，6 step 待っても c が現れず」
Answer: w の部分列で ϕ を満たすもの全体

- * ユースケース

- * 集めた大量のログの解析
「面白いところだけ見たい」
- * 実行時監視
「危ないときにアラート」

簡単？

… 時間的因果関係があり，
そう簡単ではない



モニタリング：さまざまな定式化

- * Given: 離散時間ログ $w = abaaacb \dots bbc$
仕様 ϕ 「b の後, 6 step 待っても c が現れず」

Answer: w の部分列で ϕ を満たすもの全体

- * Given: 連続時間ログ $w = (a, 0.12) (b, 1.28) \dots$
仕様 ϕ 「b の後, 6 秒待っても c が現れず」

Answer: w の部分列で ϕ を満たすもの全体

[Ulus, CAV'17] [Waga+, FORMATS'17] など

解は無数

(開始時刻は $t=1?$ $t=1.01?$ $t=1.001?$...)

→ 時間付きオートマトンの zone 構成で, 効率的に表現・計算可能

応用上重要:

仕様を書くのが一番むずかしい

「時間がかかりすぎると困る…
しかし, しきい値はどこに設定
すればよい?」

- * Given: 連続時間ログ $w = (a, 0.12) (b, 1.28) \dots$
パラメータ付き仕様 $\phi(p)$ 「b の後, p 秒待っても c が現れず」 「a は概ね p 秒周期で現れる」

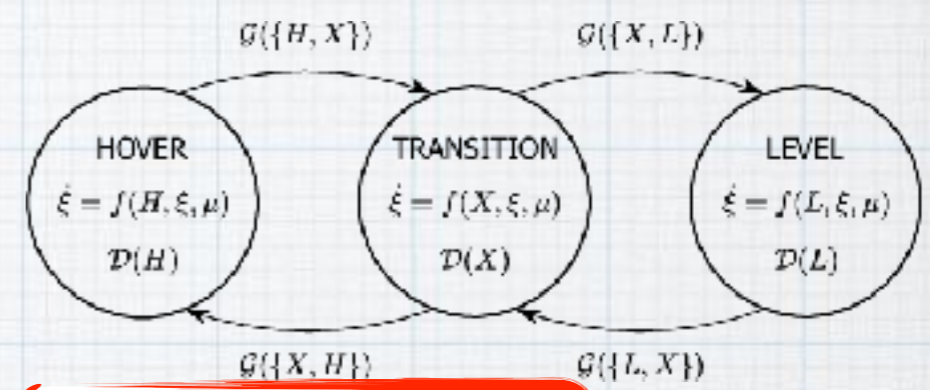
Answer: $(p, \{w \text{ の部分列で } \phi(p) \text{ を満たすもの}\})$ のペア全体

[Waga+, ICECCS'18] [Waga+, NFM'19] [Waga+, CAV'19] など

形式検証の例 2 : モデル検査

* 物理情報システムへの適用

- * 一義的には、**連続量が現れた時点でアウト**
 - * 状態空間が無限になって、探索できない
 - * 例：ハイブリッド・オートマトン
 (オートマトン+微分方程式) では、到達可能性が



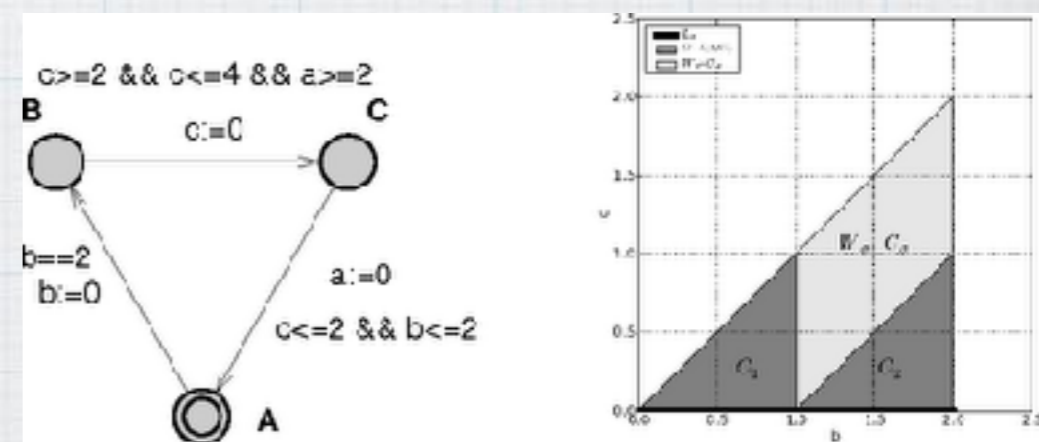
オートマトンの例
 [Daws+, CDC'11]

• 再掲

- * 確率的オートマトンならばOK
 - * Markov chain, Markov decision process, ...
 - * 到達可能性判定の代わりに、到達確率を計算する
 (線形計画法で解ける)

* 時間付きオートマトン timed automaton も OK

- * 連続量の領域が、決まったテンプレート
 (region, zone) で記述できる
- * region, zone は有限個しかない



時間付きオートマトンと zone graph
 from [Daws+, FORMATS'06]

モニタリング：ERATO MMSD の取り組み

- * Given: 離散時間ログ $w = \text{abaaacb}\cdots\text{bbc}$
仕様 ϕ 「b の後, 6 step 待っても c が現れず」

Answer: w の部分列で ϕ を満たすもの全体

- * Given: 連続時間ログ $w = (a, 0.12) (b, 1.28) \cdots$
仕様 ϕ 「b の後, 6 秒待っても c が現れず」

Answer: w の部分列で ϕ を満たすもの全体

[Ulus, CAV'17] [Waga+, FORMATS'17] など

- * Given: 連続時間ログ $w = (a, 0.12) (b, 1.28) \cdots$

パラメータ付き仕様 $\phi(p)$ 「b の後, p 秒待っても c が現れず」 「a は概ね p 秒周期で現れる」

Answer: $(p, \{w \text{ の部分列で } \phi(p) \text{ を満たすもの}\})$ のペア全体

[Waga+, ICECCS'18] [Waga+, NFM'19] [Waga+, CAV'19] など

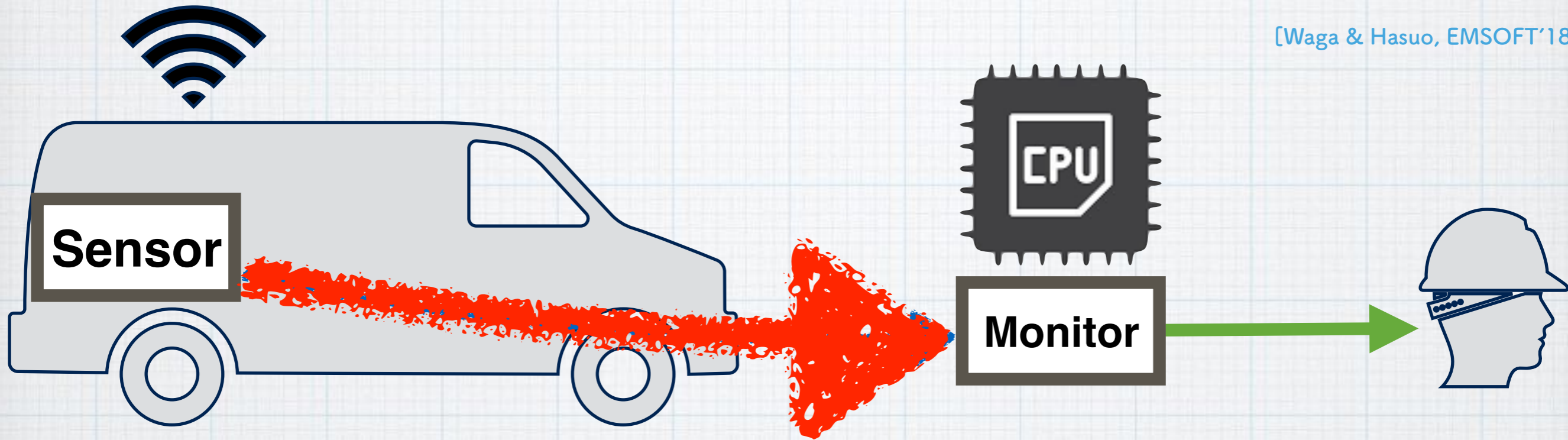
時間付きオートマトンの理論による
高速アルゴリズム. ラップトップで毎秒,
数M の長さのログを処理
[Waga+, FORMATS'17]
<https://github.com/maswag/monaa>
ルネサス RH850 実装もあり

パラメータ・時間付きオートマトンの理
論による高速アルゴリズム. ラップトップで
毎秒, 数十K の長さのログを処理
[Waga+, NFM'19 & CAV'19]
<https://github.com/maswag/symon>

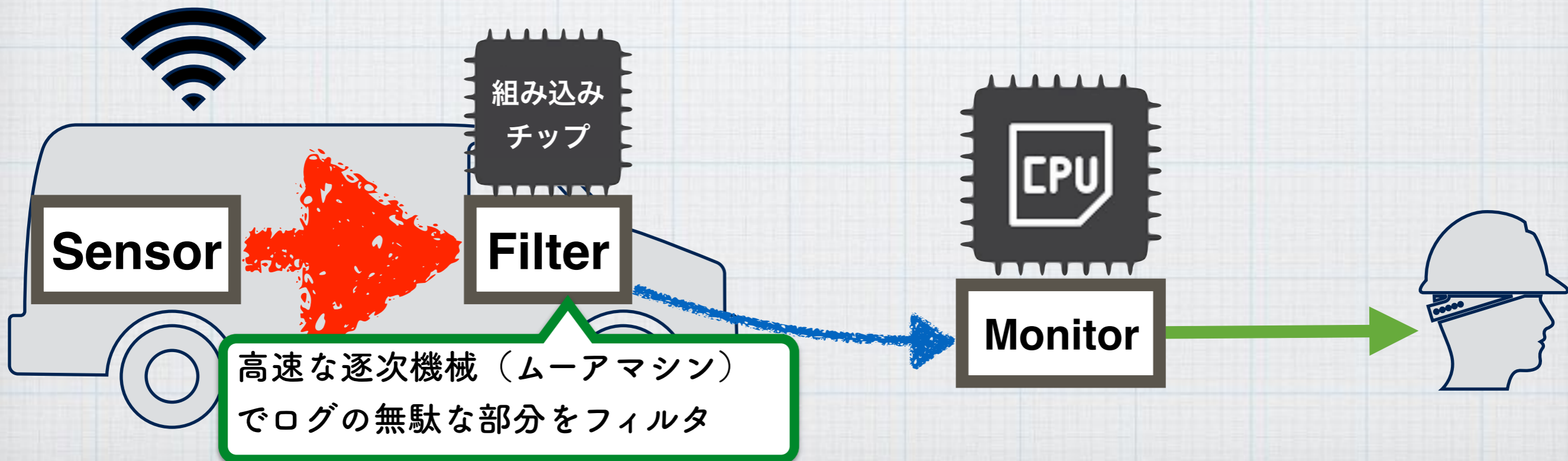
Naive Remote Monitor

フィルタリングによる前処理

[Waga & Hasuo, EMSOFT'18]



Contribution: Filter for Remote Monitor



アウトライン



* ERATO MMSD プロジェクト：
 物理情報システムの信頼性保証支援

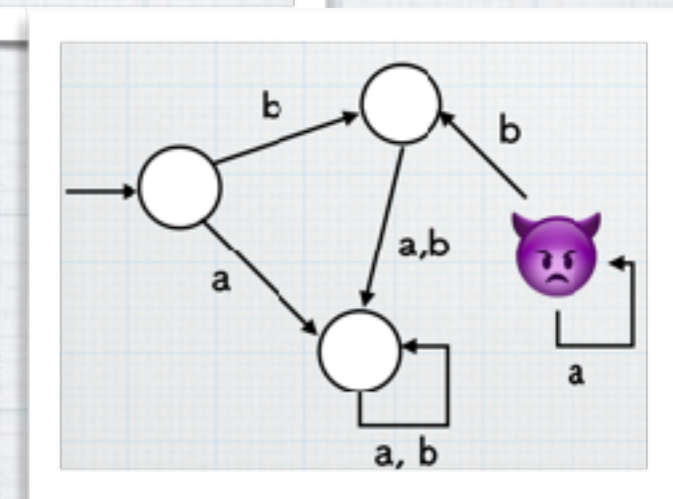
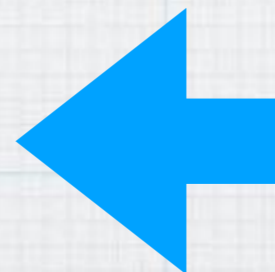
* 形式手法 formal methods とは

- * 定理証明, モデル検査
- * 形式手法と制御理論
- * 物理情報システムへの実世界応用へ

$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{[1]}}{B} \quad \frac{}{B \rightarrow C}^{[2]} \\
 \hline
 C \\
 \frac{}{A \rightarrow C}^{[1,1]} \\
 \hline
 (B \rightarrow C) \rightarrow (A \rightarrow C) \quad [1, 2]
 \end{array}$$

* 技術紹介

- * サーチベーステスト, 反例生成
- * モニタリング
- * 形式安全アーキテクチャ
- * 形式仕様記述支援
- * 機械学習システムの信頼性保証. 論理 vs 統計



* プロジェクトの現状



ERATO MMSD の研究開発の目標：
 これらの柔軟なミックス → 実システムへ応用

特に：定理証明をスケールダウンするためには？

(再掲)

形式手法

テスト test

モデル検査

自動定理証明

対話定理証明

低 コスト 高

ブラックボックス
モデルを許容

ホワイトボックスモデルを要求

高 自動化の度合い 低

自動

専門家を月・年
単位で拘束

低 保証の網羅性 高

低 保証の数学的厳密性 高

経験論的保証

数学的保証 → 「形式検証」

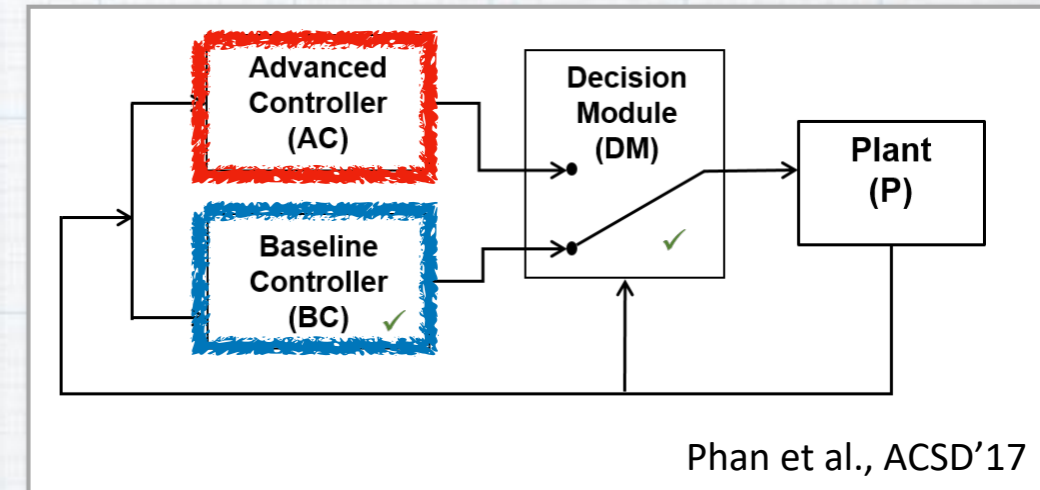
形式安全アーキテクチャ

- * 安全アーキテクチャの例：

simplex architecture (右図)

- * **通常系 (AC)** は
複雑, 性能重視, ブラックボックス

- * **保証系 (BC)** は
単純, 安全性重視, (できるだけ) ホワイトボックス



- * ポイント：**ACがブラックボックスであっても, システム全体を形式検証できる.**
BCの安全性及びDM (スイッチング戦略)の正しさが示せば十分
- * 仮に形式検証が不可能でも, システム設計・テストの系統的指針を与える
- * 留意点：冗長性, 独立性, 多様性, ...
- * ユースケース：
テスト以上の保証度合いが必要な, safety-critical system の品質保証

形式安全アーキテクチャ

- * ERATO MMSD の取り組み:

- * 安全アーキテクチャの
Event-B による形式化

- * Event-B: 形式モデリングツール

[Abrial, "The Event-B Book", 2010 CUP]

[Kobayashi+, ICFEM'18] など

状態遷移系を基礎とする。次をサポート

- * 安全性証明

- * SMT ソルバによる自動証明など

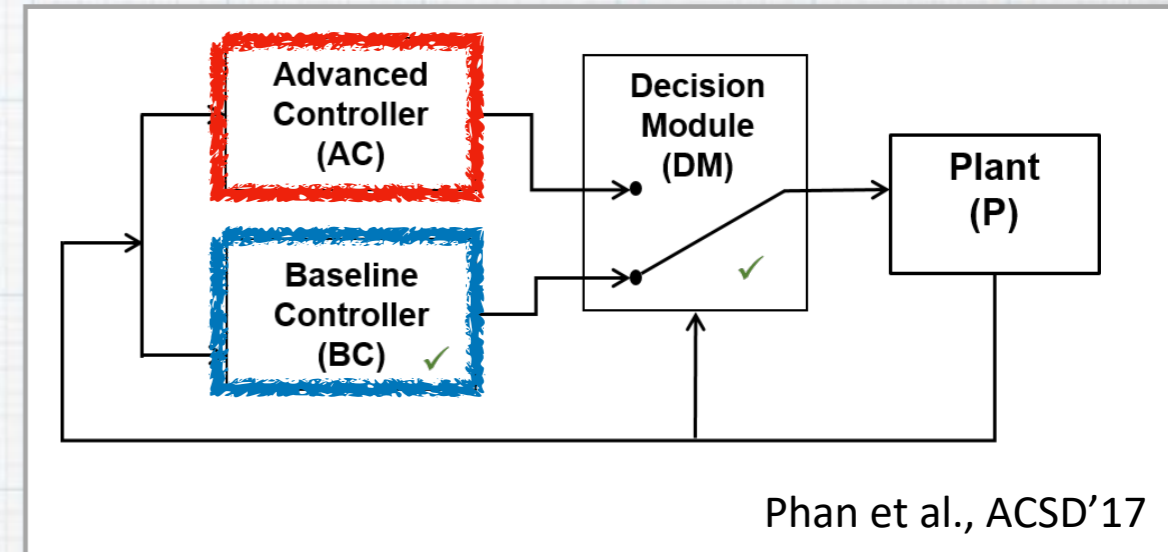
- * コンポーネント間の契約に基づく,

- assume-guarantee 型推論 → ブラックボックスを許容

- * 契約が履行されるかのチェックには, モニタリングが使える

- * モデルの段階的詳細化をサポート

- * 個別のアーキテクチャの体系的導出に使える



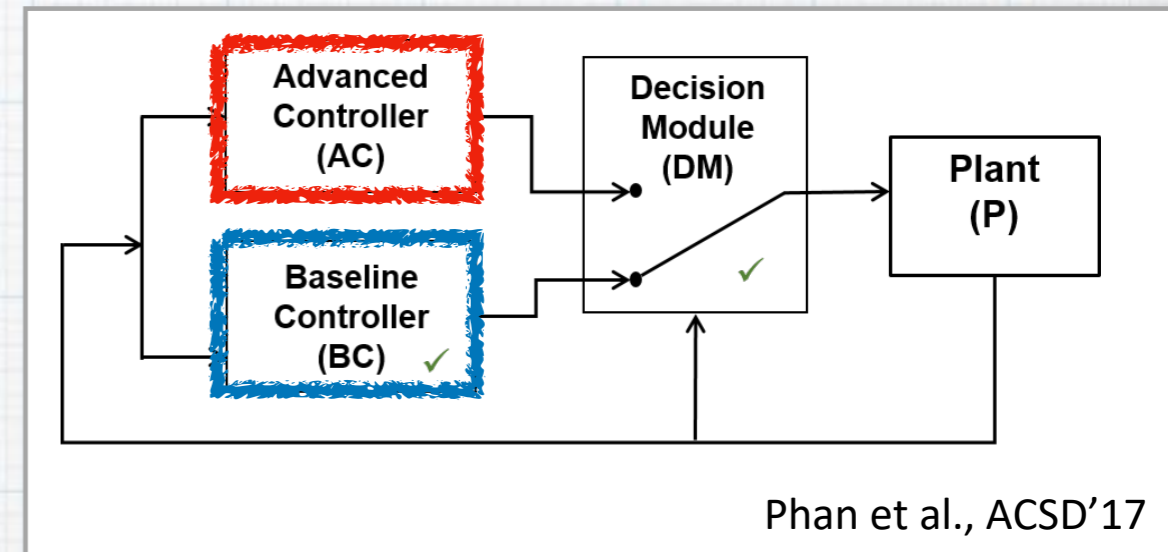
形式安全アーキテクチャ

- * ERATO MMSD の取り組み：

- * 安全アーキテクチャの **Event-B** による形式化

- * 現在の具体的な取組み

- * 自動運転システムのための、**一般的**安全アーキテクチャを構築，その信頼性を確認
(autonomoose と協働，自動運転独自のニーズを反映)
 - * 一般的安全アーキテクチャから，企業さまの個別ニーズに応える**個別**安全アーキテクチャを導出



アウトライン



* ERATO MMSD プロジェクト：
 物理情報システムの信頼性保証支援

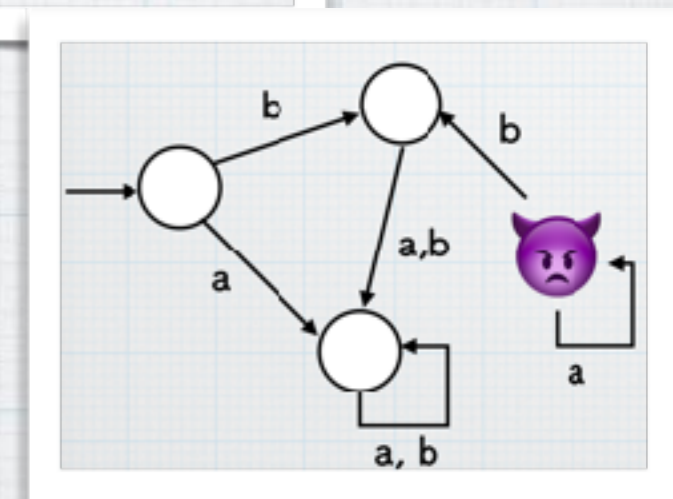
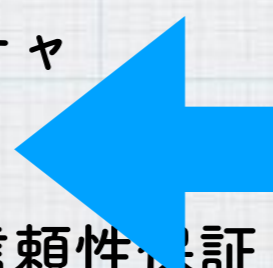
* 形式手法 formal methods とは

- * 定理証明, モデル検査
- * 形式手法と制御理論
- * 物理情報システムへの実世界応用へ

$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{(1)}}{B} \quad \frac{}{B \rightarrow C}^{(2)} \\
 \hline
 C \\
 \frac{}{A \rightarrow C}^{(1,1)} \\
 \hline
 (B \rightarrow C) \rightarrow (A \rightarrow C) \quad \text{[}\rightarrow\text{I, 2]}
 \end{array}$$

* 技術紹介

- * サーチベーステスト, 反例生成
- * モニタリング
- * 形式安全アーキテクチャ
- * 形式仕様記述支援
- * 機械学習システムの信頼性保証. 論理 vs 統計



* プロジェクトの現状

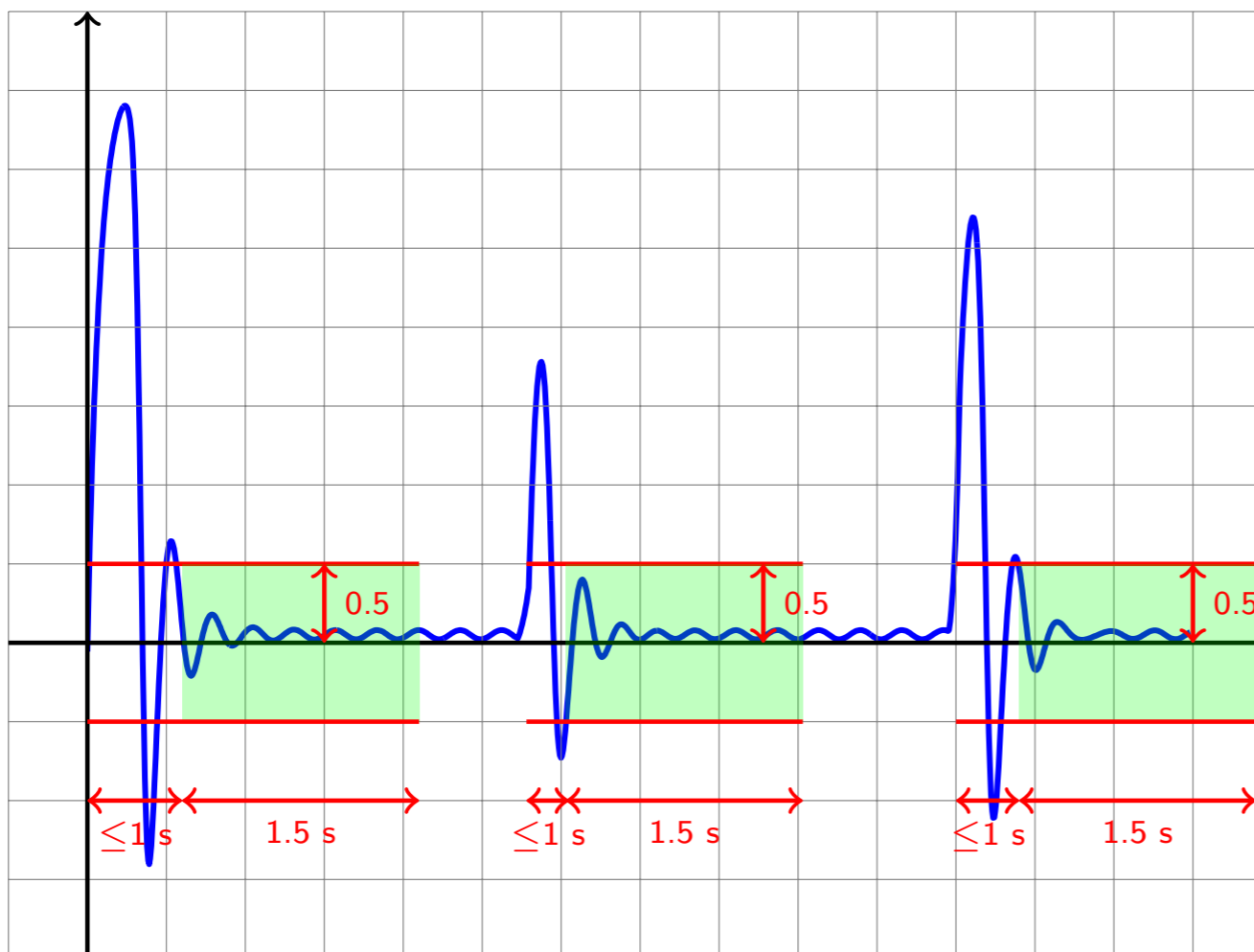
形式仕様記述を支援する対話型ツール

Slide by Alexandre Donzé

STL Examples

Always $|x| > 0.5 \Rightarrow$ after 1 s, $|x|$ settles under 0.5 for 1.5 s

$$\varphi := G(x[t] > .5 \rightarrow F_{[0,.6]} (G_{[0,1.5]} x[t] < 0.5))$$



- いきなり正しいものを書くのは困難
- しかし、正解はエンジニアさんの頭の中!
- → 形式仕様記述を支援する対話型ツール
- 正例・反例を例示
cf. [Prabhakar+, RTSS'18]
- ユーザーのフィードバックを受けて、改訂をsuggest
- 研究推進中
予備的成果: [Arcaini+, Softw. Test., Verif. Reliab.'19, ICTSS'18, arxiv 1907.02133]
- 対面でのコンサルテーションも行っています

REGULAR EXPRESSION 2 matches, 262 steps (-0ms)

TEST STRING SWITCH TO UNIT TESTS

This is a <div> simple div</div> test

EXPLANATION

- ▼ / (\S*)\s*< / gm
 - ▼ 1st Capturing Group (\S*)
 - ▼ \S* matches any non-whitespace character (equal to [\r\n\t\f\v])
 - * Quantifier — Matches between zero and unlimited times, as many times as possible, giving back as needed (greedy)
 - ▼ \s* matches any whitespace character (equal to [\r\n\t\f\v])
 - * Quantifier — Matches between zero and unlimited times, as many times as possible, giving back as needed (greedy)
 - < matches the character < literally (case sensitive)
 - ▼ Global pattern flags
 - g modifier: global. All matches (don't return after first)

MATCH INFORMATION

Match 1

Full match	8-11	a <
Group 1.	8-9	a

Match 2

Full match	23-27	div<
Group 1.	23-26	div

QUICK REFERENCE

Search reference	
All Tokens	A single character of: a, b ... [abc]
★ Common Tokens ✓	A character except: a, b ... [^abc]
General Tokens	A character in the ranges: ... [a-z]
Anchors	A character not in the ra... [^a-z]
Meta Sequences	A character in the ran... [a-zA-Z]
* Quantifiers	Any single character .
Group Constructs	Any whitespace character \s
Character Classes	Any non-whitespace character \S
Flags/Modifiers	Any digit \d
	Any non-digit \D

SUBSTITUTION

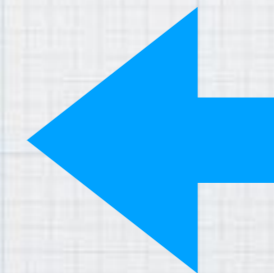
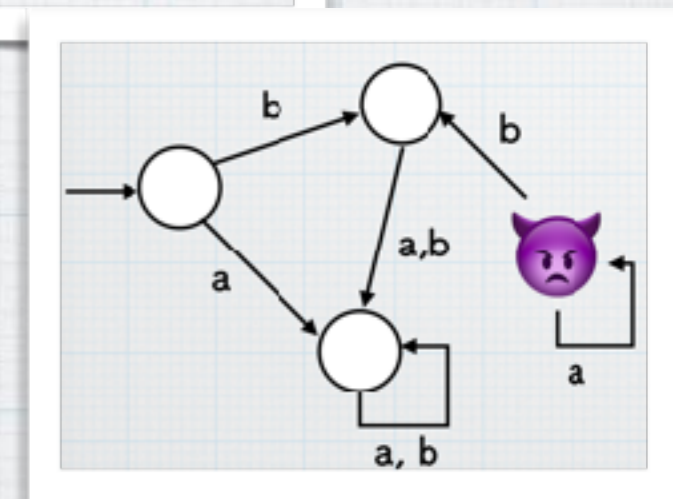
- * 試行 → 結果を表示
- * 「なぜそうなるか」の説明

アウトライン



- * ERATO MMSD プロジェクト：
物理情報システムの信頼性保証支援
- * 形式手法 formal methods とは
 - * 定理証明, モデル検査
 - * 形式手法と制御理論
 - * 物理情報システムへの実世界応用へ
- * 技術紹介
 - * サーチベーステスト, 反例生成
 - * モニタリング
 - * 形式安全アーキテクチャ
 - * 形式仕様記述支援
 - * 機械学習システムの信頼性保証. 論理 vs 統計
- * プロジェクトの現状

$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{(1)}}{B} \quad \frac{}{B \rightarrow C}^{(2)} \\
 \hline
 C \\
 \frac{}{A \rightarrow C}^{(1,1)} \\
 \hline
 (B \rightarrow C) \rightarrow (A \rightarrow C) \quad \text{[}\rightarrow\text{I, 2]}
 \end{array}$$

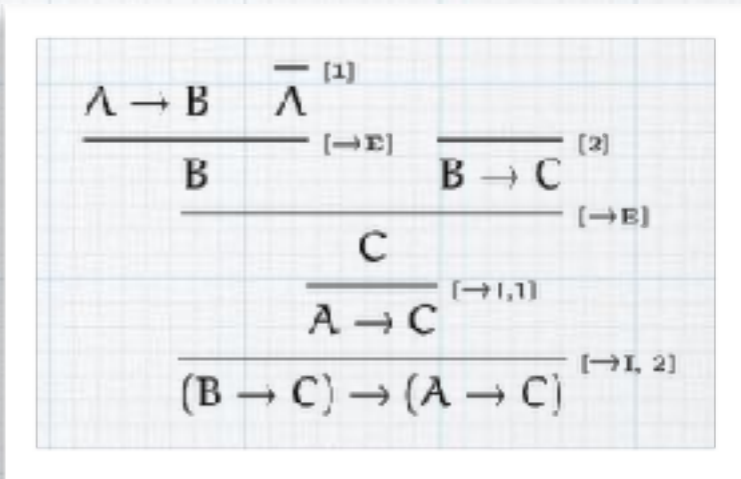


ところで：AI 研究の歴史（超ダイジェスト）

* 1980-1987 「論理的人工知能，エキスパートシステム」

* 与えられたルールによる論理的推論

- * 判断の過程が明らかにトレースできる
- * 「与えられたルール」の準備が大変（不可能）

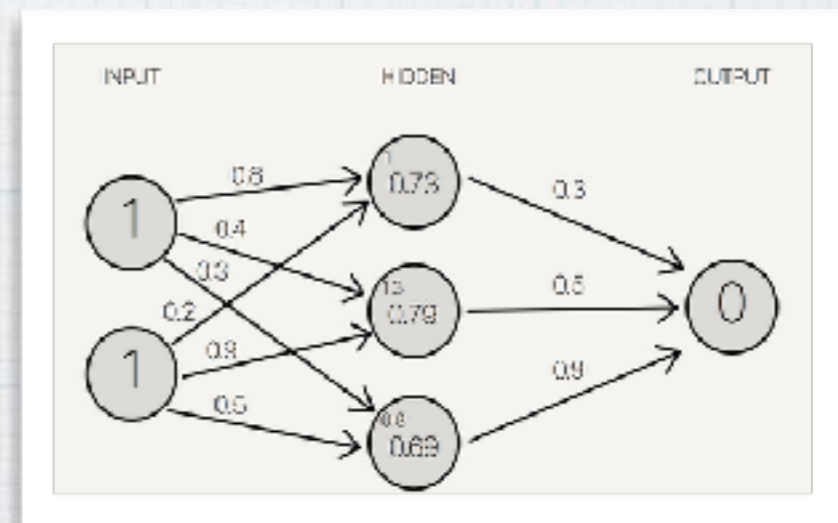


- * Prolog, 第5世代コンピュータ
- * 関連する数学分野：論理学，数学基礎論

* 2011-現在 「統計的機械学習」

* 大量のデータから自動で統計的に特徴量抽出

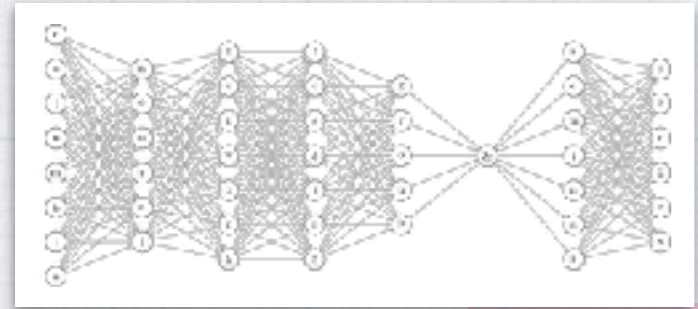
- * 驚くべきスケールビリティ。
画像処理，物体認識，自動翻訳，...
- * 判断の過程はブラックボックス



- * ディープニューラルネットワーク
- * 関連する数学分野：統計学，最適化



統計的機械学習 vs 演繹的形式推論



$$\frac{A \quad A \supset B}{B}$$

統計的機械学習		演繹的形式推論
データのノイズを許容	入力の誤り	公理は絶対 誤りは想定せず
保証されない	結論の正しさ	論理的に保証 (cf. 数学的証明)
高い データから自動で特徴量発見	スケーラビリティ	低い 公理の準備は人力 (cf. エキスパートシステム)
低い 判断の理由はパラメータ (重み)	説明可能性	高い 推論過程が証明として明示的

Key: 論理と統計的学習の間で,
システムレベルのタスク切り分けと構造化

機械学習システムの 安全性・信頼性保証

* 統計的 機械学習	帰納的 (データから学ぶ)	不確かさ (データのノイズ)	ブラックボックス
制御理論・ 形式手法	演繹的 (証明)	数学的・論理的 厳密性	ホワイトボックス

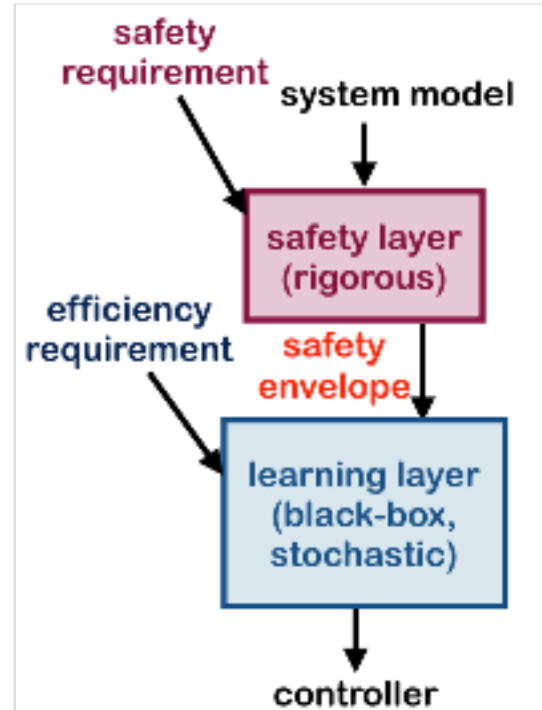
*

* 総当たり探索の代わりに,
データから傾向を読み取り
高速に証明や補題を探索

制御理論・
形式手法

統計的
機械学習

- * 不確かな機械学習コンポーネントを
包み込む "safety envelope"
- * 例: 自動運転の経路計画. safety
envelope の中で最適化



アウトライン



* ERATO MMSD プロジェクト：
 物理情報システムの信頼性保証支援

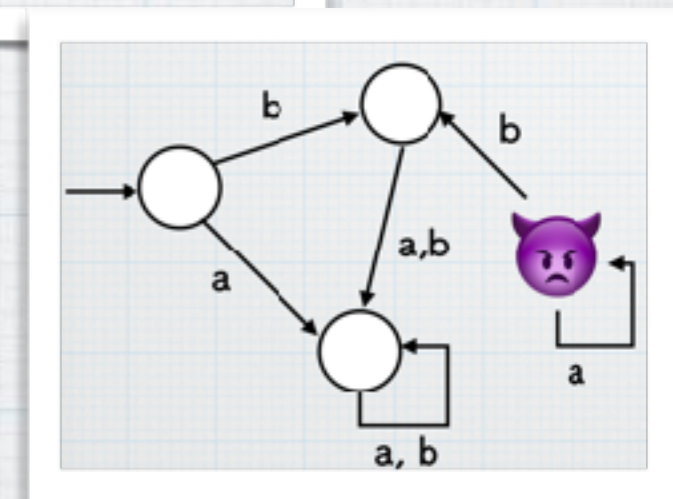
* 形式手法 formal methods とは

- * 定理証明, モデル検査
- * 形式手法と制御理論
- * 物理情報システムへの実世界応用へ

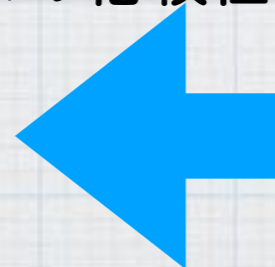
$$\begin{array}{c}
 \frac{\Lambda \rightarrow B \quad \overline{\Lambda}^{(1)}}{B} \quad \frac{}{B \rightarrow C}^{(2)} \\
 \hline
 C \\
 \frac{}{A \rightarrow C}^{(1,1)} \\
 \hline
 (B \rightarrow C) \rightarrow (A \rightarrow C) \quad \text{[}\rightarrow\text{I, 2]}
 \end{array}$$

* 技術紹介

- * サーチベーステスト, 反例生成
- * モニタリング
- * 形式安全アーキテクチャ
- * 形式仕様記述支援
- * 機械学習システムの信頼性保証. 論理 vs 統計



* プロジェクトの現状





ERATO MMSD 紹介

* JST ERATO プロジェクト

* 2016/10-2022/03.

総勢50名規模の基礎研究プロジェクト

* プロジェクト目標：工業製品の設計サポート

* 形式手法の拡張. ソフトウェアから物理情報システムへ

* 安全性・信頼性, Verification & Validation. 「システムが期待通り動作するか」

* 特に自動運転を戦略的ターゲットに. U Waterloo と協働 www.autonomoose.net

* 研究体制

* 国際的体制. 4つのグループの1つは @ U Waterloo.

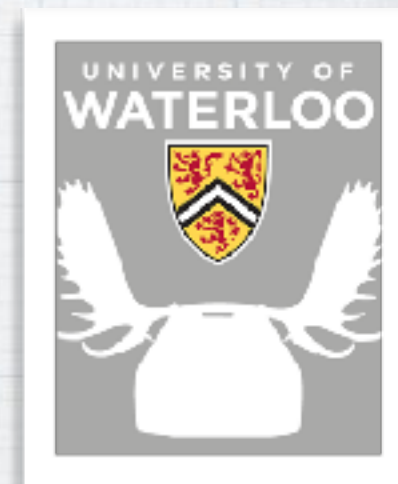
雇用する研究員15名余のうち, 外国人が半数以上

* 先端的・包括的学術研究を実システムに応用

* 学際的 "creative chaos" によるブレイクスルー

* Waterloo・京大・阪大・九大のチームと協働しつつ, リソースの大部分を NII に集約

* 多様な背景: 論理学・代数学から形式手法, 制御理論, 機械学習, ソフトウェア工学まで



Our Organization

International and multi-disciplinary. “creative chaos”



Kyoto U IS Site:
Advanced Deductive Verification
Leader:
Kohei Suenaga

Kyoto U RIMS Site:
Categorical Infrastructure
Leader:
Masahito Hasegawa

Group 0 @ NII:
Metatheoretical Integration
Leader: Shin-ya Katsumata

Topics:
Programming Languages,
Formal Semantics,
Categorical Models,
Mathematical Logic, ...



Group 3 @ NII:
Formal Methods and Intelligence
Leader: Fuyuki Ishikawa

Topics:
Software Engineering,
Formal Modeling,
Testing, Safe & Explainable AI



Kyushu U Site:
Optimization for CPS V&V
Leader:
Hayato Waki

Osaka U Site:
Control Theory for CPS
Leader:
Toshimitsu Ushio

Group 1 @ NII:
Heterogeneous Formal Methods
Leader: Ichiro Hasuo
Subleader: Masako Kishida

Topics:
Automata Theory,
Control Theory,
Formal Verification,
Proof Assistants,
Automated Deduction,
Runtime Verification



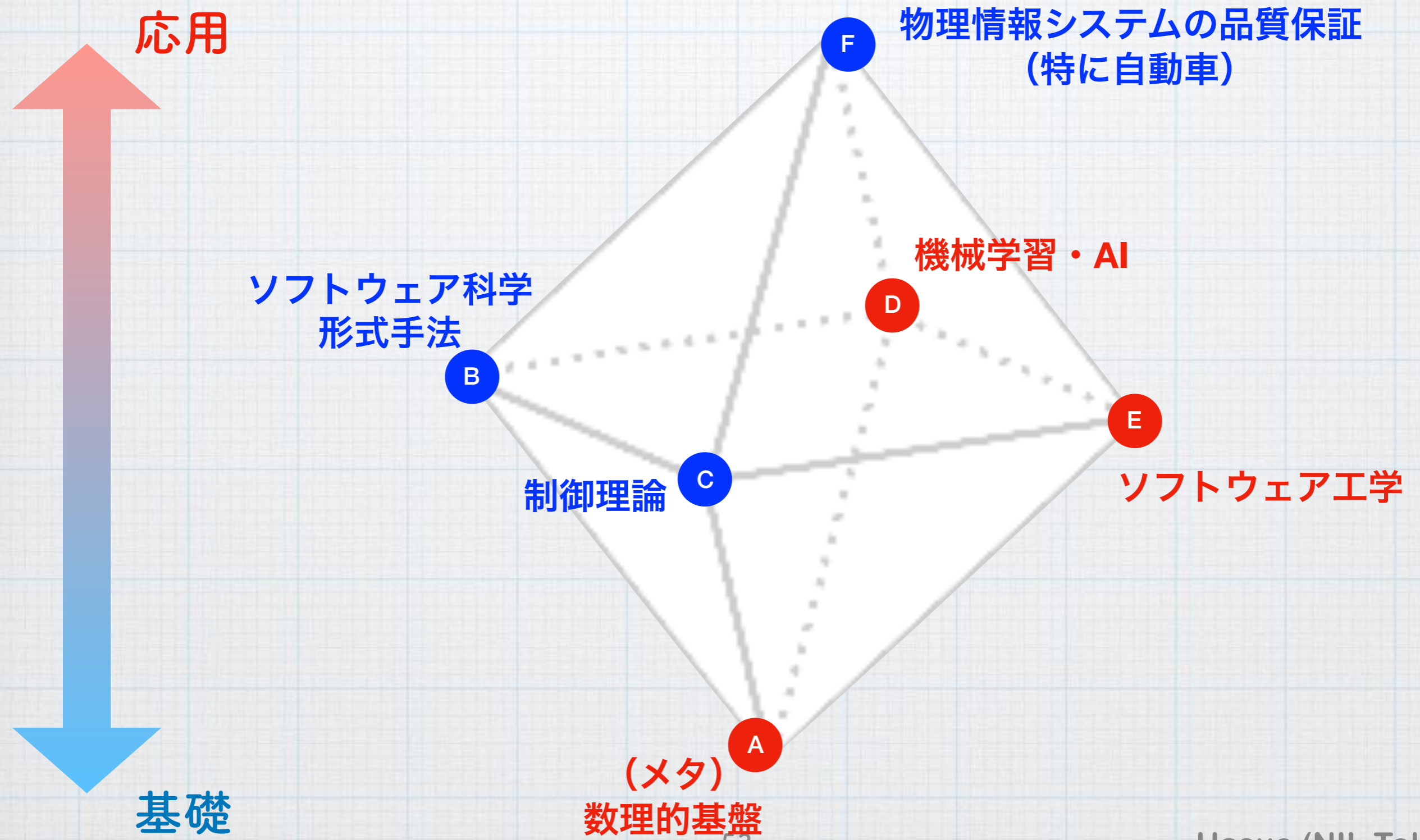
Group 2 @ U Waterloo:
Formal Methods in Industry
Leader: Krzysztof Czarnecki

Topics:
Automated Driving, Software Engineering,
Machine Learning





物理情報システムのための 分野横断：ERATO MMSDの場合



研究体制

* **G0: メタ理論的統合グループ (勝股)**

論理学, 代数学, 圏論,
 プログラミング言語理論, ...

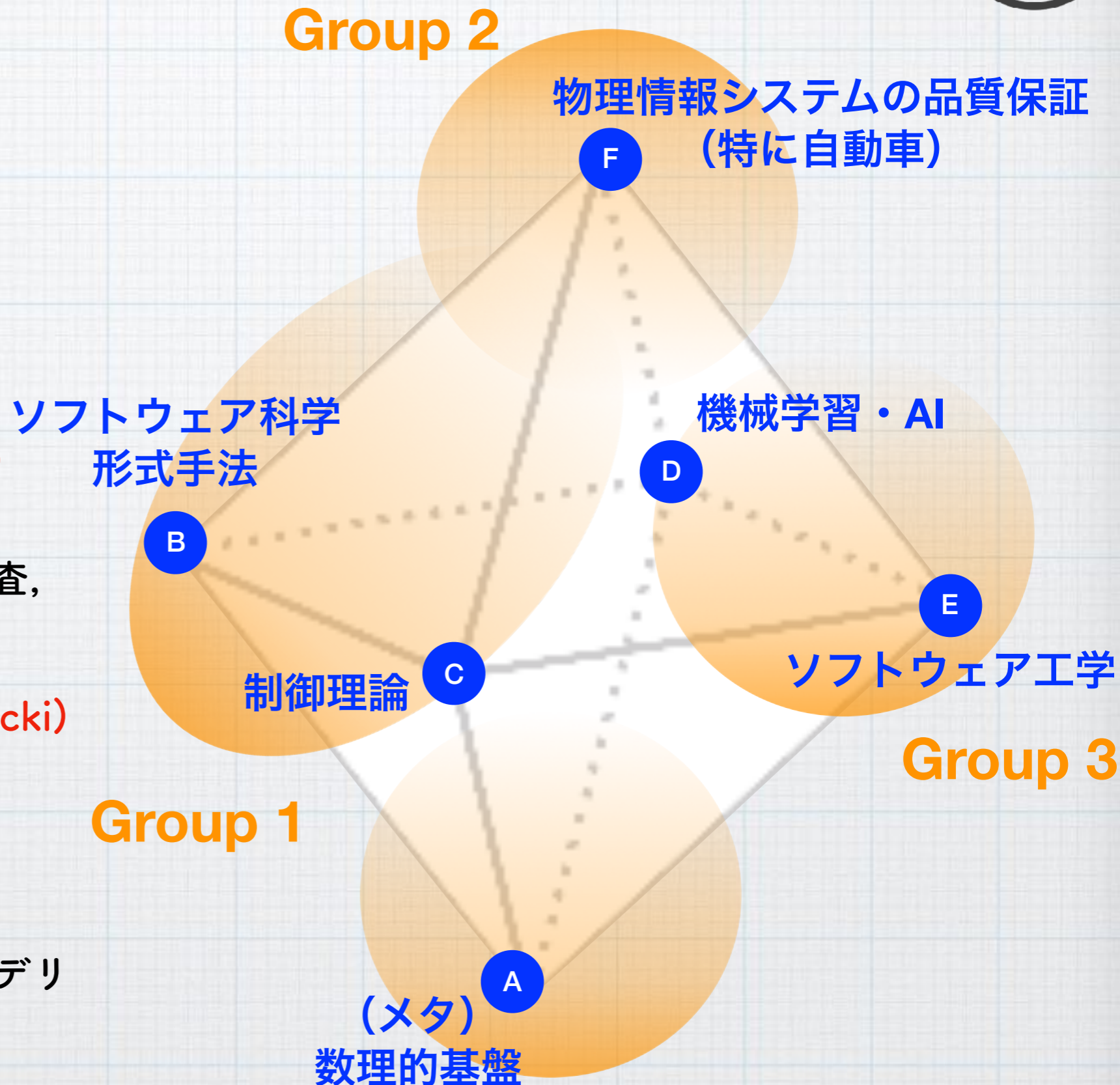
* **G1: ヘテロジニアス形式手法グループ (蓮尾・岸田)**

ソフトウェア科学,
 制御理論, 形式手法, モデル検査,
 自動定理証明, ...

* **G2: 産業応用グループ (Czarnecki)**
 自動運転システム

* **G3: インテリジェンス協働形式手法グループ (石川)**

ソフトウェア工学, テスト, モデリ
 ング, 要求工学, ...



学術研究の進捗状況

* グループ間の協働の進展

* 新たな応用が導く理論発展

- * モニタリング

- * 深層学習のための圏論的理論

- * 機械学習利用 → 効率的テスト

* 数理的基盤(G0)の重要性

- * 学術的&人的

* 国際的 visibility:

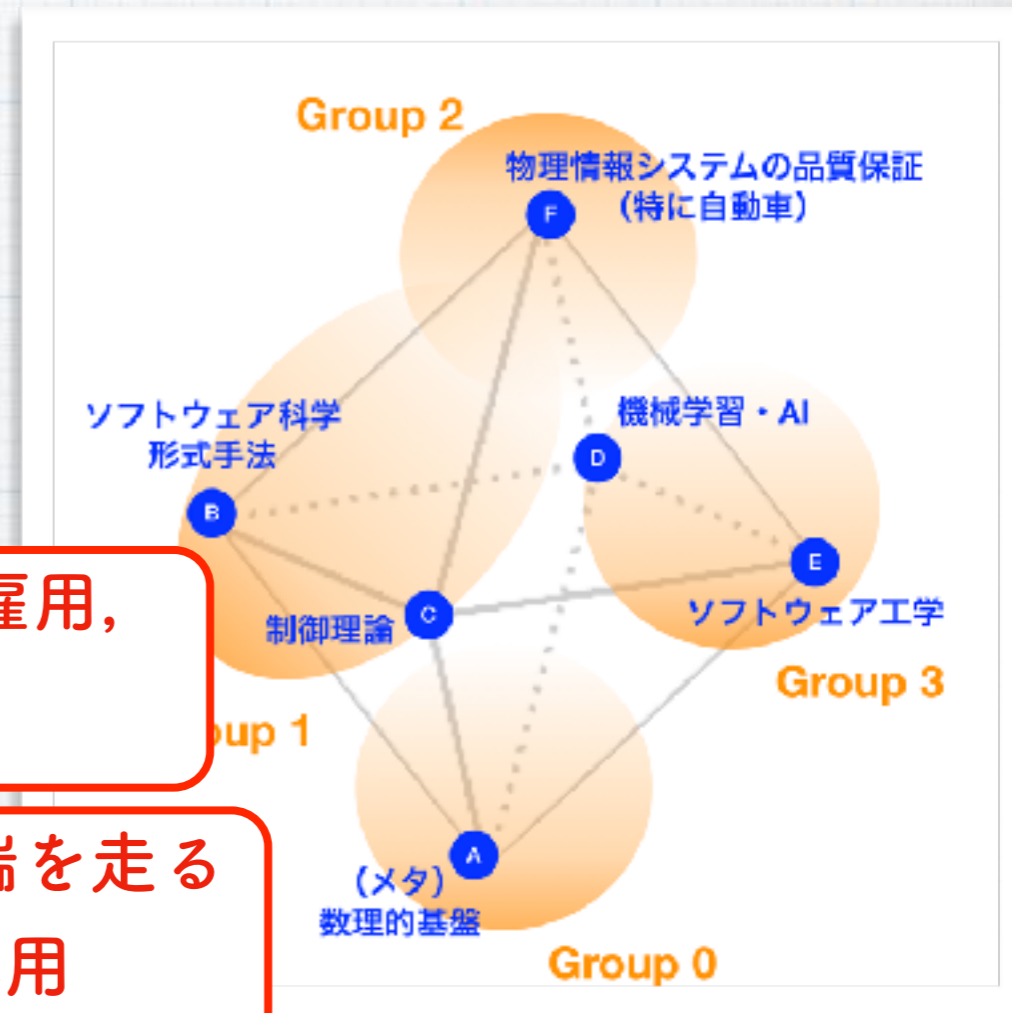
- * いわゆる「トップ国際会議」(CORE rank A, A*) 論文 > 30 報

- * 最優秀論文賞: ICECCS'18, FoSSaCS'19 (共にCORE rank A)

- * 理論計算機科学の最高峰国際会議 LICS'19 (CORE rank A*) では、
全採択数 60 報のうち 6 報でERATO MMSD 研究者が (共) 著者

聞き上手な理論家の雇用,
おすすめです

学術研究で世界的先端を走る
からこそ可能な産業応用

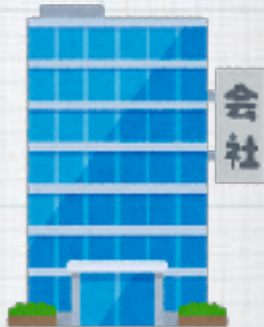
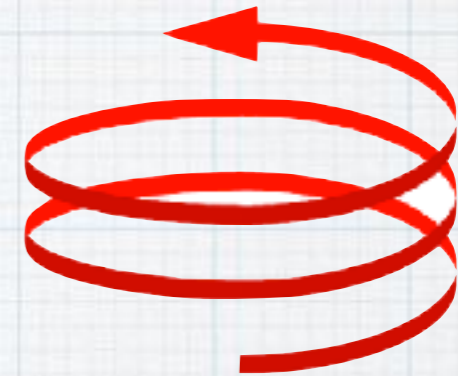


応用上の進捗状況

- * プロトタイプツール多数
 - * 実行時監視ツール
 - * **MONAA** [Waga+, FORMATS'17] <https://github.com/maswag/monaa>
 - * **SyMon** [Waga+, CAV'19] <https://github.com/MasWag/symon>
 - * サーチベーステストツール
 - * **FalStar** [Zhang+, EMSOFT'18] <https://github.com/eratommsd/falstar>
 - * その他： **確率的プログラム自動検証** ツール, **RNN2WFA** ツール (recurrent neural network を重み付きオートマトンに近似), ...
- * **U Waterloo の自動運転プロジェクト autonomoose** と協働,
形式手法の応用に向けて, 具体的トピックについて研究推進中
- * **国内の企業10社弱** と共同研究・学術指導・定期的議論
(自動車メーカー, 自動車部品メーカー, 総合電機メーカー, ソフトウェアベンダーなど)

産学協働の現状

- * 学術研究と産業応用は、研究推進の車の両輪。
テーマ：自動車に限らず、ロボット、生産システム
などから、機械学習システム、ソフトウェアまで
- * いくつかのツールを公開しています
- * 研究レベルのプロトタイプですが、ニーズに合わせて改良・適合が可能
- * 論文リストから辿っていただけます
- * 研究会を組織する予定です（企業の方対象、今年度上半期）
- * 定期セミナーで技術紹介，情報共有
- * 先日のシンポジウム「高信頼自動運転システムのための先進的研究
——数理的理論から，AI 協働，ソフトウェアプラットフォームへ」をきっかけに
（ビデオ・スライド公開中）



くわしくは，ERATO MMSD プロジェクトウェブページまで
<https://group-mmm.org/eratommsd>（「erato mmsd」で検索）

まとめ

- * Society 5.0 に向けた、物理情報システムの信頼性保証
- * 従来ソフトウェアを対象としていた形式手法を、制御理論と組み合わせ、物理情報システムへ適用
- * ブラックボックスモデル、統計的機械学習など、実用上の課題
→ テストとの有機的組み合わせ

- * ERATO MMSD プロジェクト：
 - * 先端的学術研究と産業応用の上向きスパイラル
 - * 「スケールダウンできる形式手法」
 - * 今すぐ使える技術多数。
既存の設計過程に寄り添う
(サーチベーステスト, モニタリング, ...)
 - * 設計過程の新パラダイムの提案も
(形式安全アーキテクチャ, ...)

- * ... 新しいチャレンジ → 分野間協働, それをささえる数学的基盤

