Software Science under Uncertainties: A Survey

Ichiro Hasuo

National Institute of Informatics & SOKENDAI

Research Director, ERATO MMSD Project

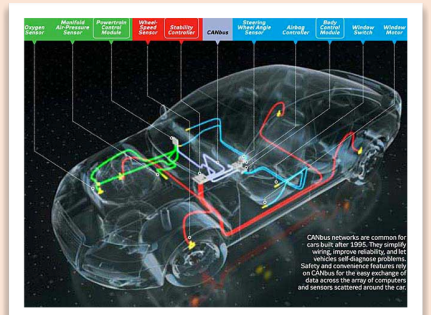
PPL 2021 (Category 4), 2021/03/11

Some slides are by
Satoshi Kura,
Toru Takisaka,
Masaki Waga,
Zhenya Zhang

情報システムの安全性と「不確かさ」

複雑な情報システム （例：自動運転車，1B LoC）

根本的な問い：
- 安全に動作するか？
- 設計にミスはないか？
- 実装にバグはないか？



ソフトウェア科学における形式検証 formal verification

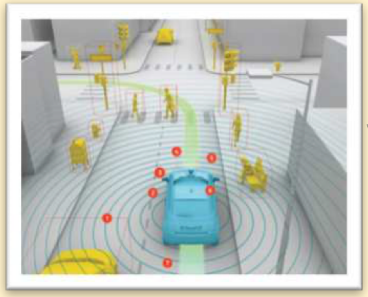
- バグ不在・安全性を数学の定理として論理的に証明
- 前提：
  プログラムは人の手で生み出される論理的実体なので，その振る舞いを数学的に記述（定義）できる

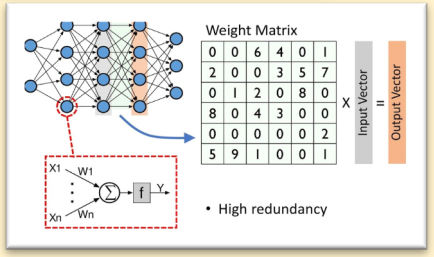
```
def factorial(num):
    """This is a recursive function that calls
   itself to find the factorial of given number"""
    if num == 1:
        return num
    else:
        return num * factorial(num - 1)
```

論理的記述の困難な情報システムの出現・普及

論理的記述 ＝ モデリング，数学的定義

例：
- 物理情報システム：
  物理的外界の完全な論理的記述は困難
- 機械学習システム：
  ノイジーなデータから数値最適化でルールを学習
  ➔ ルールの論理的記述は困難





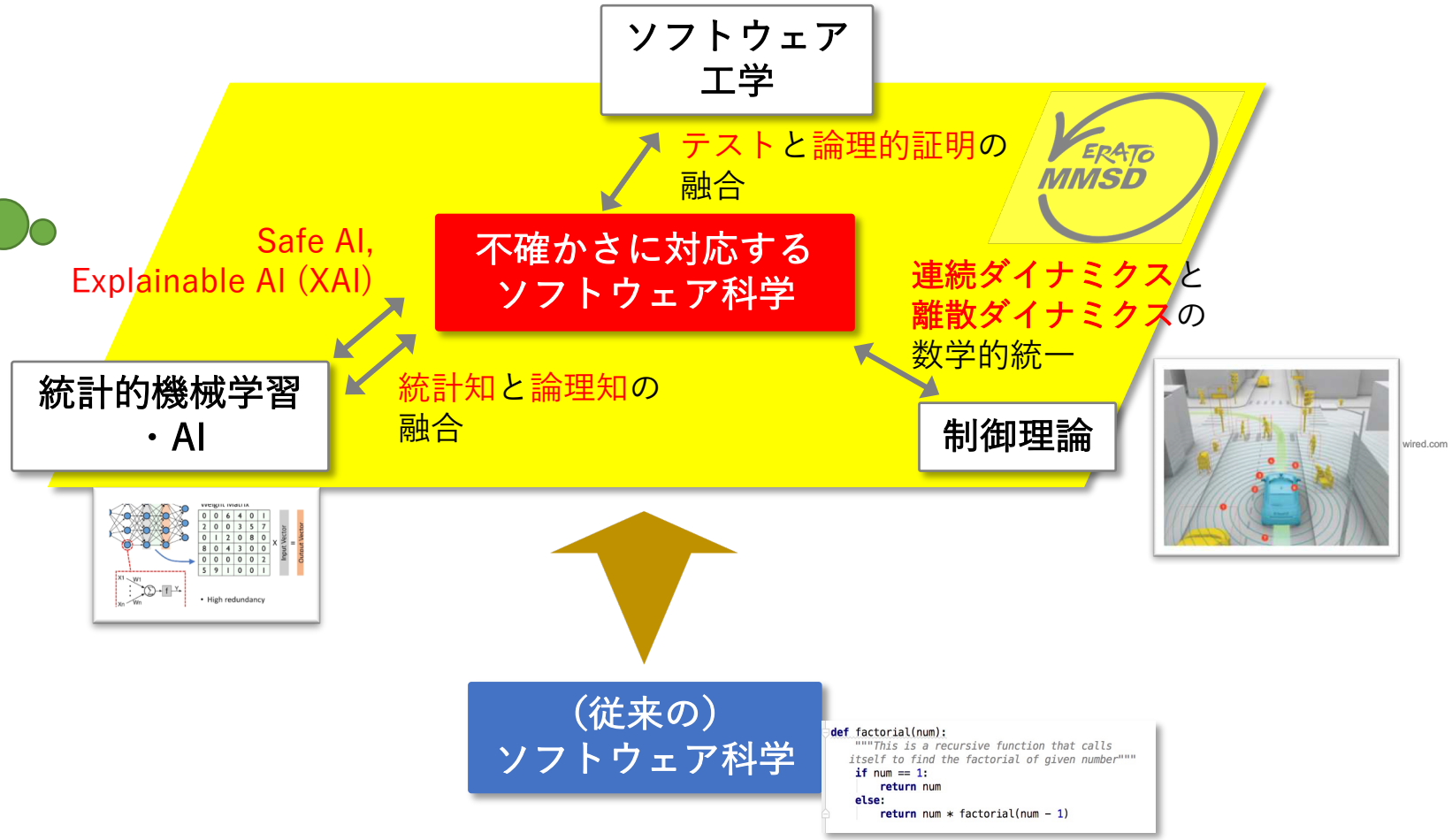
不確かさを内包する
情報システムの安全性保証
というチャレンジ

- ソフトウェア科学の新パラダイム
- テストの利用が不可避
  ➔ 統計知と論理知の融合
  （AI 研究の一大課題）
- 我々のスローガン：
  （不確かさのもとでも）
  論理的手法が必要かつ有効

不確かさに対応するソフトウェア科学

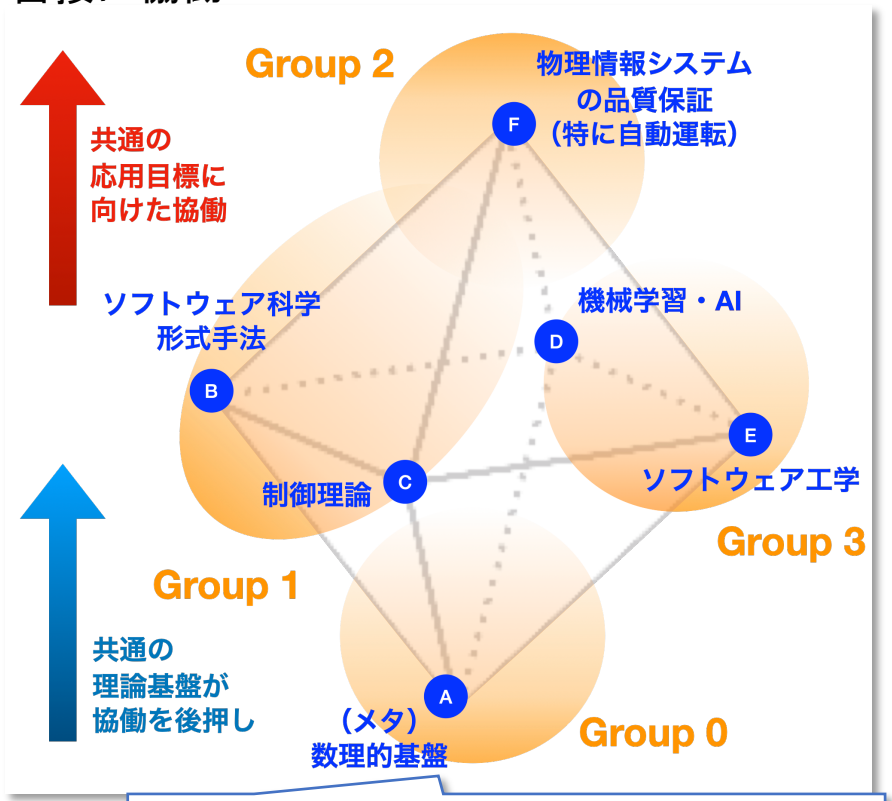
求ム 良い名前

こういう「分野」ってなんと呼べばいいでしょうか．．．



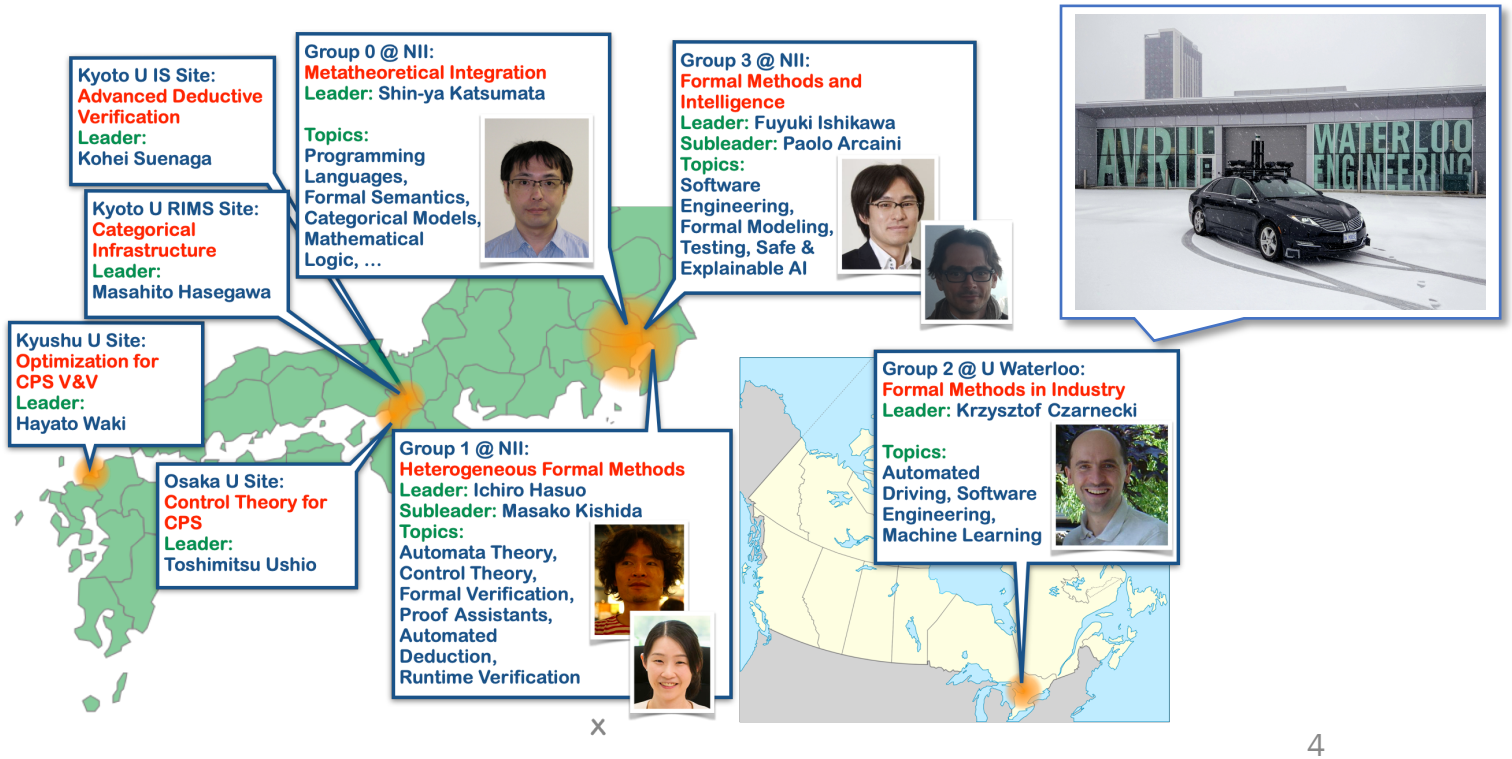
ERATO MMSD での取り組み (2016-2022)

- **学際的研究体制**
  4つの学術分野が
  共通の理論基盤と応用目標を通じて
  密接に協働



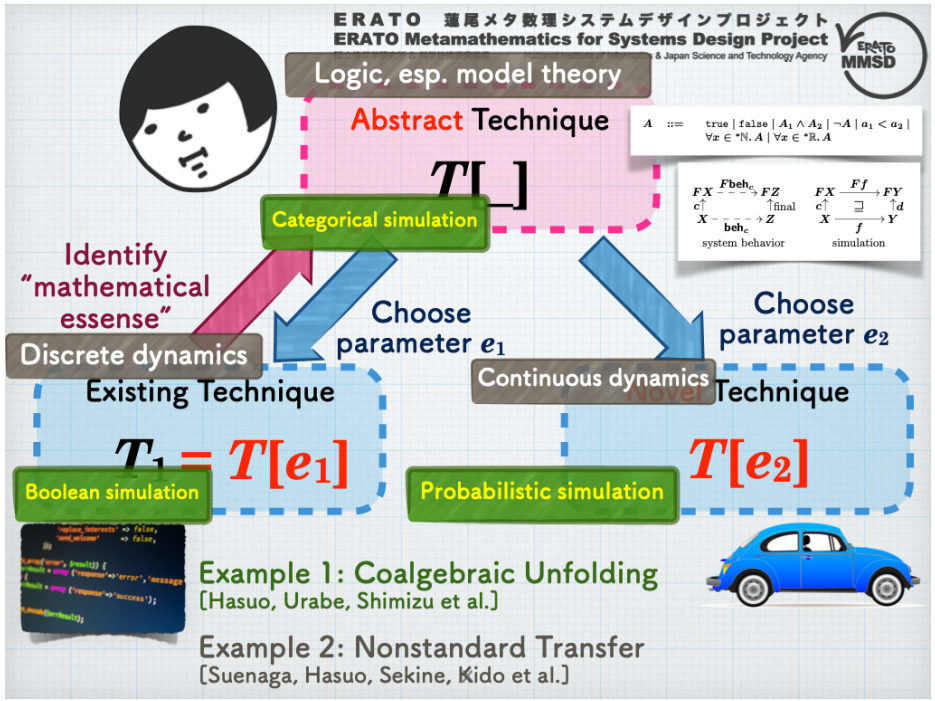
「メタ数理」の役割：
理論的研究の営為自体を理論的に解析
→ 理論研究者の応用進出を実現（次スライド参照）

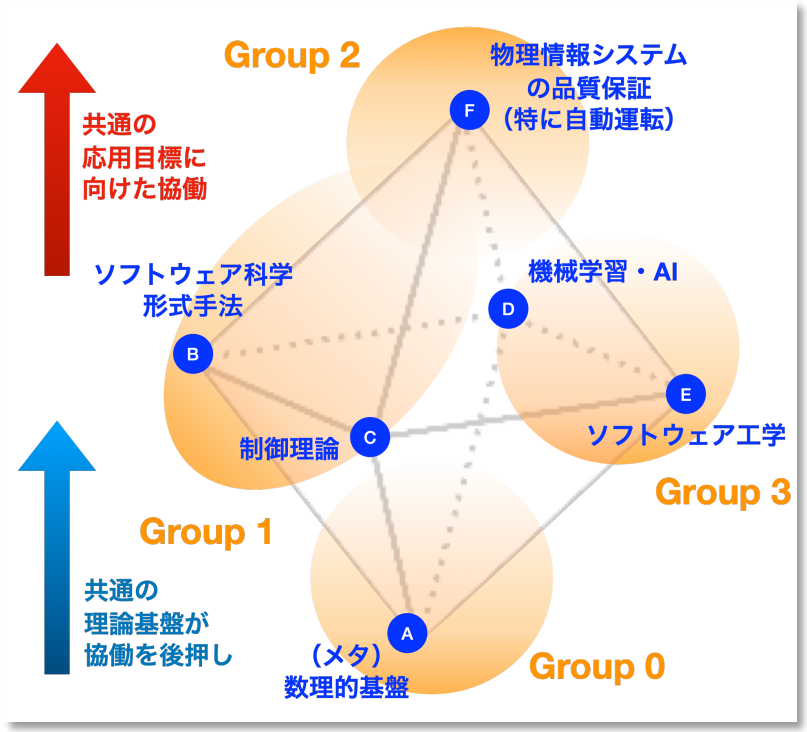
- **国際的研究体制**
  - グループの1つをカナダ Waterloo 大学に設置.
    実際の自動運転車を運用
  - 研究員の半数以上を海外から雇用



4

ERATO Metamathematics for Systems Design Project
「メタ理論」の直接・間接のご利益





- 直接のご利益：メタ数理的移転
  - 「理論の抽象的ひな形」を用いて手法を移転
  - 離散➔連続，定性➔定量の移転に特に効果的
  - 成功例： 圏論，論理学を経由した移転
    ➔ 確率的システムのモデル検査手法
    [Takisaka+, ATVA'18] [Komorida+, LICS'19]

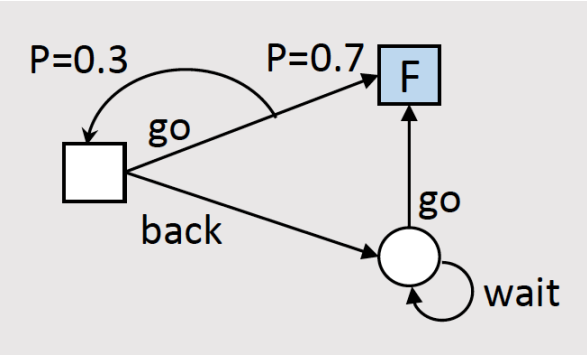
- 間接のご利益： 学際協働のインタプリタ
  - 一般に，理論➔応用の向きの移動のほうが逆よりも楽
  - メタ理論家の「内省」能力は（自らの研究活動を省みる）異分野をつなげるときに特に有効
  - 成功例： 自動運転車のゲーム理論的意思決定
    [Pruekprasert+, ITSC'19]
    自動運転安全性の形式検証 [Hasuo+, in preparation]

Outline

	モデル（➔ 実応用可能性）	不確かさの種類	解析手法
確率的モデル検査，確率的検証 [Kura+, TACAS'19] [Phalakarn et al., CAV'20] を例に	モデル要 (whitebox)	known unknowns	formal
実行時検証，モニタリング [Waga+, CAV'19] を例に	モデル不要	unknown unknowns	formal
サーチベーステスト＋論理・離散的構造 [Zhang+, CAV'19] を例に	blackbox モデルのみ要	unknown unknowns	testing
自動運転システムの安全性検証 [Kobayashi+, NFM'21] を例に	部分的 モデル要	unknown unknowns	formal

確率的形式検証

- 対象システム：
  確率的分岐のある状態遷移系や
  プログラム
  （右図）



Stochastic game (2.5 players).
(Subsumes MDP (1.5 pl.),
Markov chain (0.5 pl.))

- 目標：
  - 到達可能性・実行時間に関する検証

* Qualitative questions
  * $\mathbf{Pr}(\mathrm{Reach}_C)$ =? **1** (almost sure reachability)
  * $\mathbf{Pr}(\mathrm{Reach}_C)$ >=? **α** (threshold reachability)
* Quantitative questions
  * $\mathbf{Pr}(\mathrm{Reach}_C)$ >= **??** (lowerbound, “verification”)
  * $\mathbf{Pr}(\mathrm{Reach}_C)$ <= **??** (upperbound, “refutation”)

  - 一般の時相論理式仕様の検証は
    到達可能性に帰着可能.
    [Baier & Katoen 2008, Chap. 10]

* Programs with
  * random assignment

x := **Gaussian**(0, 0.2)

  * probabilistic branching

if prob(0.2)

  * (also nondet. assignment & branching)

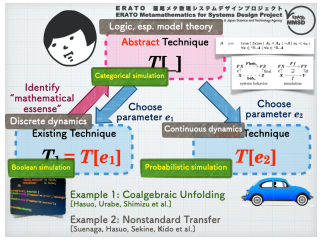
```
1  x := m
2  while x > 0 do
3     if prob(p) do
4        x := x − 1
5     else
6        x := x + 1
7     fi
8  od
```

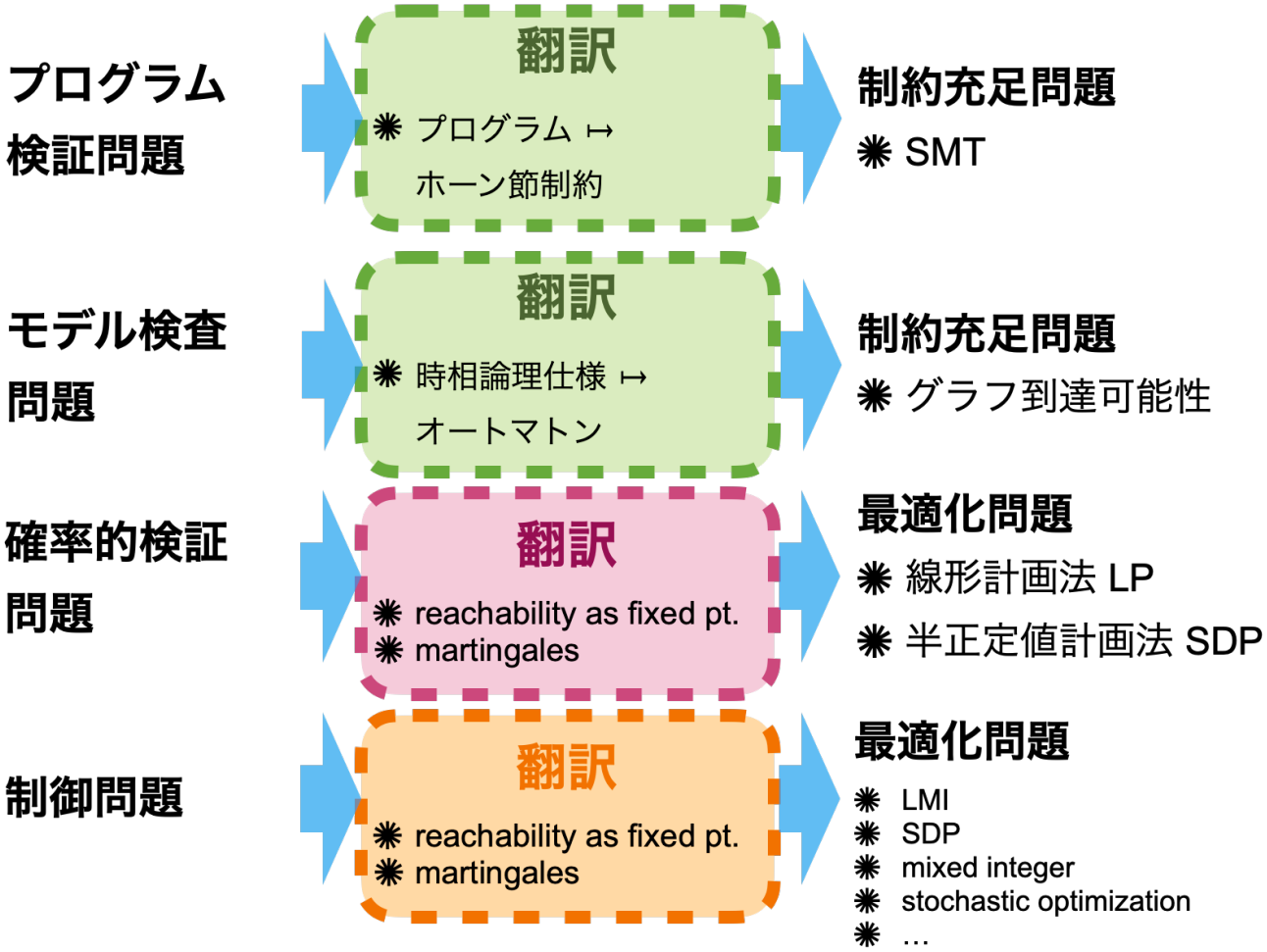
(say, $m = 16$ and $p = 0.2$)

Probabilistic programs (without conditioning)

ソフトウェア科学の仕事：「構造をときほぐす」

- 構造が複雑な検証問題から，
  構造が単純な制約充足問題・グラフ問題へ
- ソルバに渡すまでが仕事
- ポイント
  - 確率的検証でもスキームは同じ
    （「構造をときほぐす」）
  - 検証技法の発見は必要
    （ランク関数 ➔ ranking supermartingale，等）
    だが，束論・圏論のレベルでは
    しばしば同一視できる（メタ数理的移転）
  - 現実的課題，理論的に豊穣
    （fixed point theory + martingale 集中不等式など）
    ➔ 研究上やることがたくさん！
- （ついでに：
  制御理論もスキームは同じ）
- ERATO MMSD の成果群：
  [Urabe+, LICS'17] [Takisaka+, ATVA'18] [Kura+, TACAS'19] [Phalakarn+, CAV'20] 他





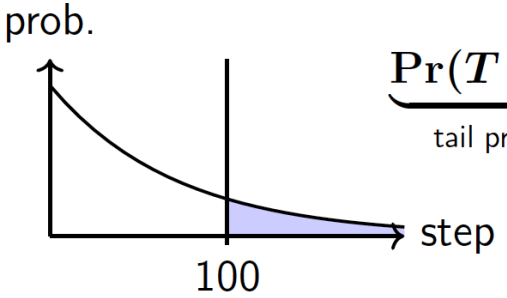
成果例１：
確率的プログラムの tail probability 検証

[Kura, Urabe & Hasuo, TACAS'19]

- 対象： 確率的プログラム（右図）
  （状態遷移系としては無限状態 ➔ 直接グラフ解析による検証は不可）
- 目標：
  stopping time の tail probability の guaranteed upper bound

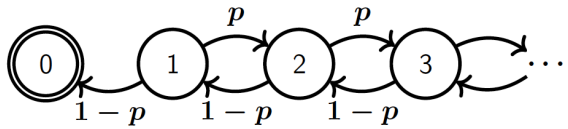
```
1  x := m
2  while x > 0 do
3     if prob(p) do
4        x := x − 1
5     else
6        x := x + 1
7     fi
8  od
```

(say, $m = 16$ and $p = 0.2$)



$\underbrace{\Pr(T \geq 100)}_{\text{tail probability}} \leq$??

"100" は所与の deadline

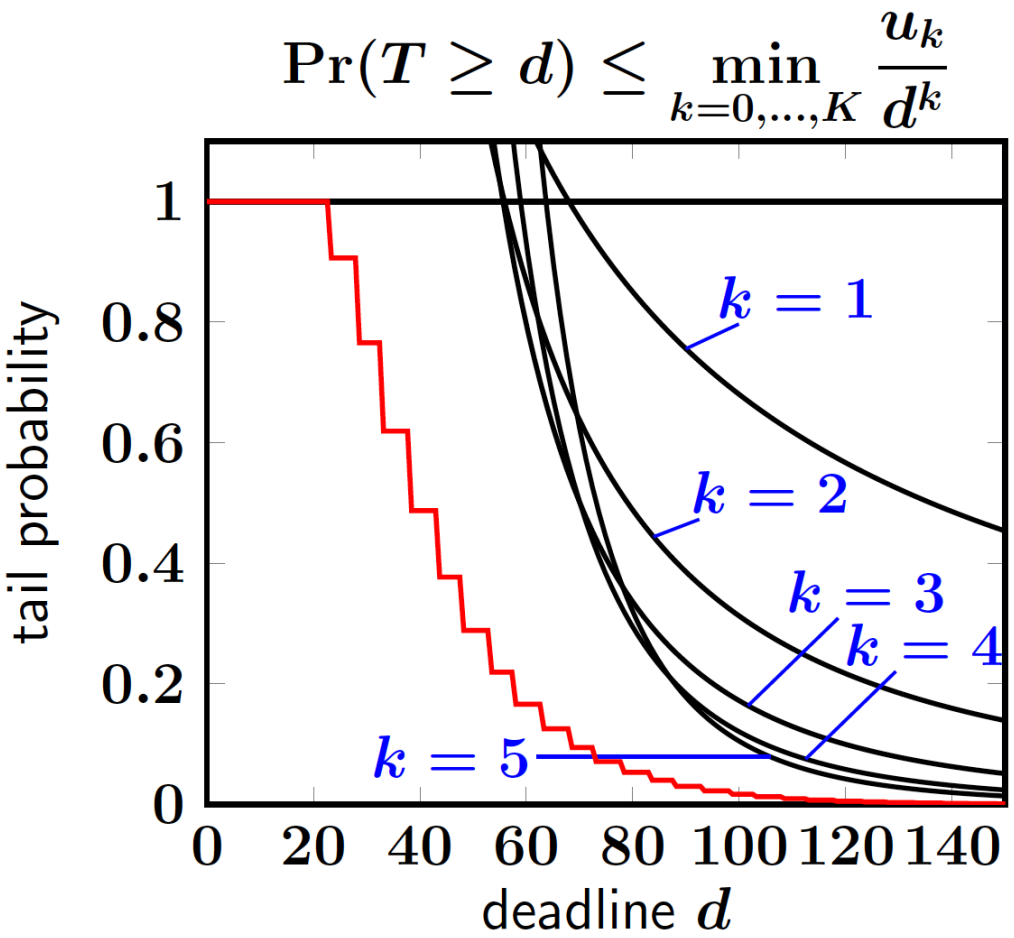


- 既存手法 [Chakarov & Sankaranarayanan, CAV'13]等：
  ranking supermartingale （〜 ランク関数の確率版） ＋ Markov 集中不等式 $\Pr(T \geq d) \leq \dfrac{\mathbf{E}[T]}{d}$
- 提案手法：
  - （平均＝1次モーメントだけでなく）高次モーメント $\mathbf{E}[T^k]$ $(k = 1, 2, \dots)$ を bound する ranking supermartingale
  - 高次 Markov 集中不等式 $\Pr(T \geq d) \leq \dfrac{\mathbf{E}[T^k]}{d^k}$ による bound の改善

成果例１：
確率的プログラムの tail probability 検証
[Kura, Urabe & Hasuo, TACAS'19]

- 実験結果
  - 対象プログラム：
    coupon collector's problem
  - 赤が真の tail probability
    （実行時間が d を超える確率）
  - k次モーメントまで考慮して
    supermartingale を線形計画法で
    合成
  - k が上がるほど，bound が tight になる
  - 線形計画法なので，高次化に伴う
    コスト増大も許容可能

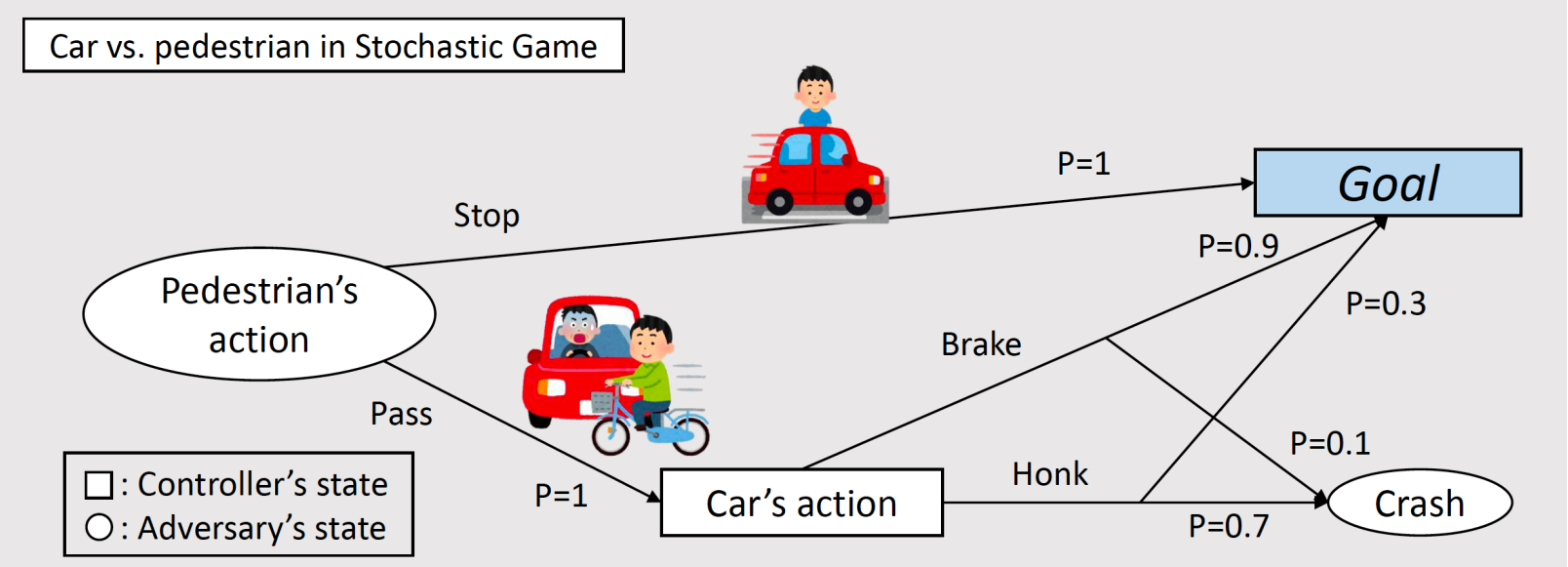
upper bound	execution time
$\mathbf{E}[T] \leq 68$	0.024 s
$\mathbf{E}[T^2] \leq 3124$	0.054 s
$\mathbf{E}[T^3] \leq 171932$	0.089 s
$\mathbf{E}[T^4] \leq 12049876$	0.126 s
$\mathbf{E}[T^5] \leq 1048131068$	0.191 s



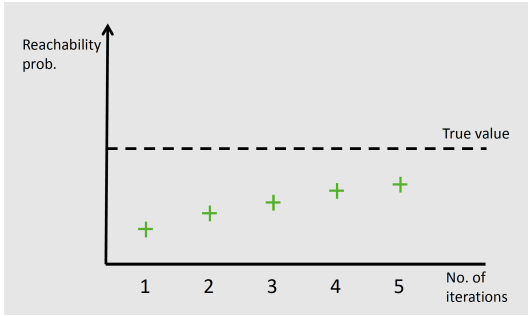
成果例２： 確率的ゲームの効率的解法

[Phalakarn, Takisaka, Haas & Hasuo, CAV'20]

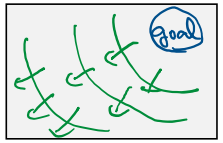
- 対象： 確率的ゲーム
  - オートマトン
    ＋ controller's *angelic* nondeterminism
    ＋ adversary's *demonic* nondeterminism
    ＋ probabilistic branching
  - "2.5-player game"
  - 多くの意思決定シナリオのモデル
  - （ここでは）有限状態に限る
  - 勝利条件：
    特定のポジションに到達（右図では "goal"）



- 目標： 最適戦略の合成，およびその場合のスコアの計算
  $\max_{\sigma:\text{Ctrl's str.}} \min_{\tau:\text{Adv's str.}} \Pr(\sigma, \tau \text{ のもとで Goal に到達})$
- 線形計画法で exact に解けるが計算量大
- 既存手法1： Value iteration (VI) による近似解法 [Condon, I&C '92]
  → Bellman operator による Kleene 反復（右図）

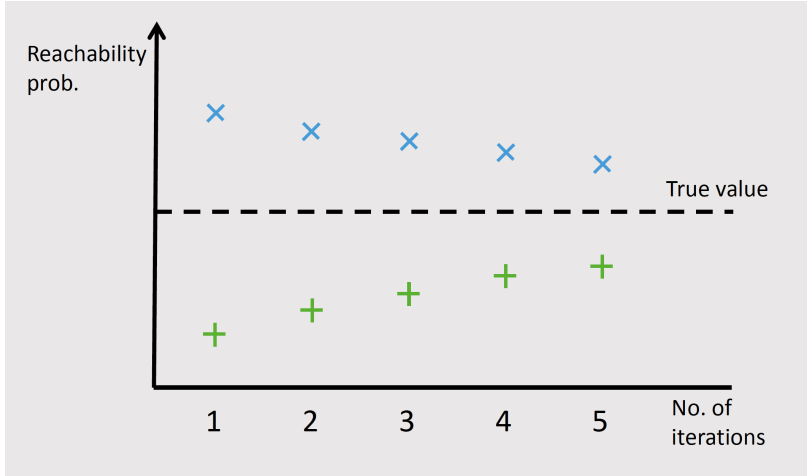


Goal から 到達可能性確率を逆伝播させていく

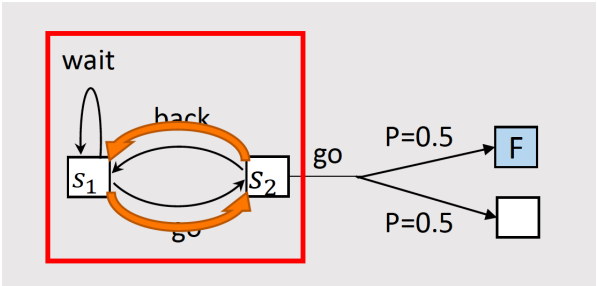


成果例２： 確率的ゲームの効率的解法
[Phalakarn, Takisaka, Haas & Hasuo, CAV'20]

- VIの問題点： 真の解にどれだけ肉薄できているか不明
- 既存手法2： Bounded Value iteration (BVI) による近似解法 [McMahan+, ICML'05]等
  ➜ 真の解の上界についても Bellman operator で Kleene 反復
- BVIのチャレンジ： 上界の反復改良は最大不動点に収束.
  到達可能性確率（最小不動点）とは一致しないことがある



- より具体的に：
  ゲームにループがあると,
  上界の反復改良が自家撞着,
  高止まりして下がらない

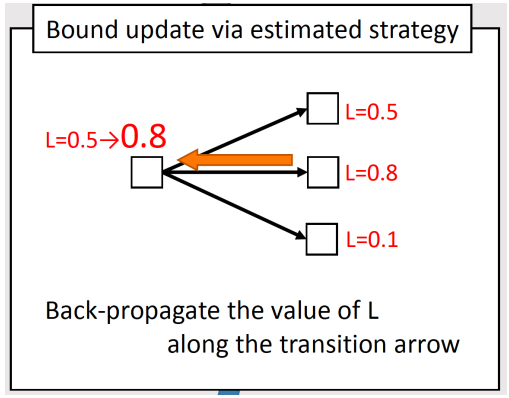


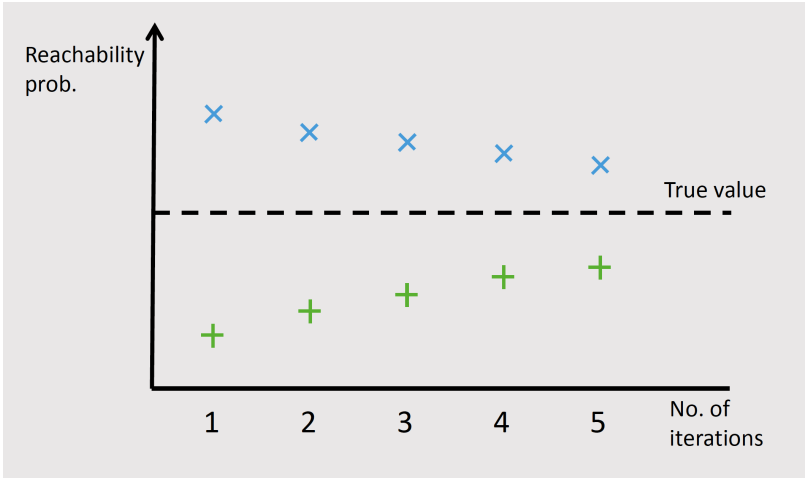
- 既存手法での解決法：
  ゲームにおける「ループ」をグラフアルゴリズムで発見して「つぶす」 [Kelmendi+, CAV'18]
  ➜ 計算コスト大

成果例２： 確率的ゲームの効率的解法

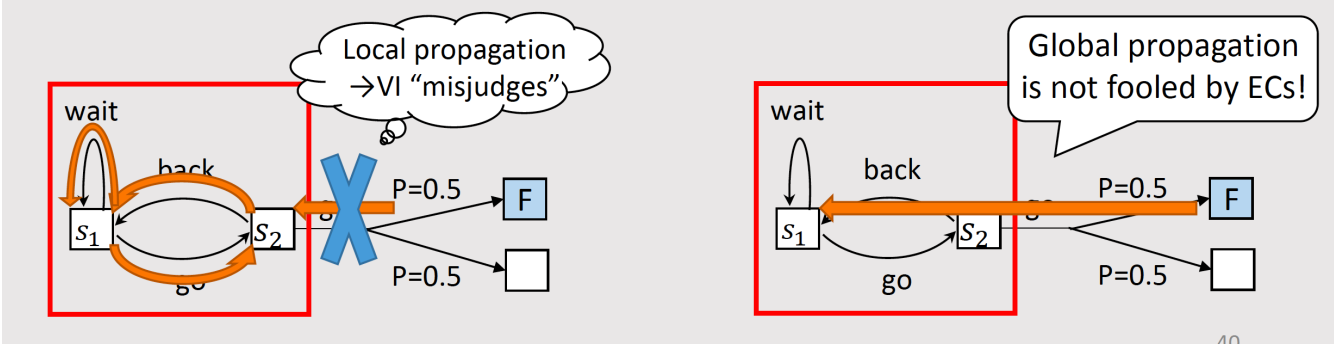
[Phalakarn, Takisaka, Haas & Hasuo, CAV'20]

- 提案手法：
  Bellman operator による局所伝播に，
  「ラフな大域伝播」を組み合わせる
- 下界については，地道に．
  Bellman update による局所伝播を
  Kleene 反復





- 上界については，局所伝播はループにつかまるので，もともとのゲームを抽象化した重み付きグラフを適切に構成して，そこで widest path problem を解く



成果例２： 確率的ゲームの効率的解法

[Phalakarn, Takisaka, Haas & Hasuo, CAV'20]

- 実験結果
  - ループをつぶす既存手法に対して数倍～数千倍の高速化

Experimental result

model	Param.	#states	#trans	#EC	[K+, Ver.1] itr	[K+, Ver.1] time(s)	[K+, Ver.2] itr	[K+, Ver.2] time(s)	[K+, learning] itr	[K+, learning] visit%	[K+, learning] time(s)	**Our alg.** itr	**Our alg.** time(s)
mdsm	3	62245	151143	1	121	3	121	4	17339	49.3	15	120	5
	4	335211	882765	1	125	15	125	47	91301	42.1	86	124	38
cloud	5	8842	60437	4421	7	7	7	1	167	6.9	14	7	<1
	6	34954	274965	17477	11	177	11	5	41	0.6	3	11	1
	7	139402	1237525	69701	11	19721	11	62	41	0.2	4	11	5
teamform	3	12475	15228	2754	2	<1	2	<1	972	49.0	137	2	<1
	4	96665	116464	19800	2	<1	2	<1	4154	34.6	9603	2	<1
	5	907993	1084752	176760	2	<1	2	<1			TO	2	<1
investor	50	211321	673810	29690	441	184	441	249			TO	364	48
	100	807521	2587510	114390	801	3318		OOM			TO	688	736
manyECs	500	1004	3007	502	6	7	6	7			TO	5	<1
	1000	2004	6007	1002	6	51	6	51			TO	5	<1
	5000	10004	30007	5002		SO		SO			TO	5	<1

[K+] Kelmendi, E., Kramer, J., Kretnsky, J., Weininger, M.: *Value iteration for simple stochastic games: stopping criterion and learning algorithm*. Proc. CAV 2018

Slow/failure sometimes

Stably fast

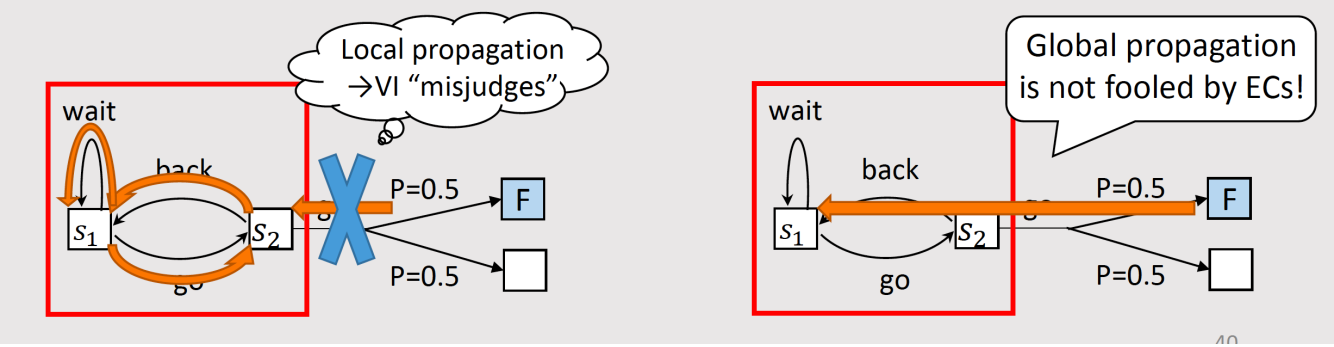
成果例２： 確率的ゲームの効率的解法

[Phalakarn, Takisaka, Haas & Hasuo, CAV'20]

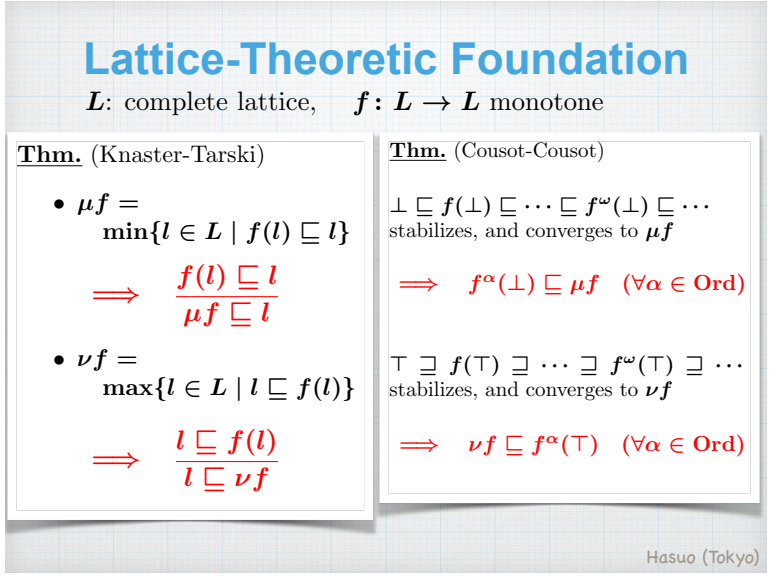
（うまく広報と連携しました.
滝坂さんありがとうございました）

成果例２： 確率的ゲームの効率的解法
[Phalakarn, Takisaka, Haas & Hasuo, CAV'20]

- 提案手法（再掲）： 上界については，局所伝播はループにつかまるので，もともとのゲームを抽象化した重み付きグラフを適切に構成して，そこで widest path problem を解く



- 教訓１：
不動点理論の
威力
（CSの基本！）



- 教訓２：

「大域解を局所反復で求める」が
CS の基本戦略だが，

時々「ラフな大域解を直接求める」
が有効！

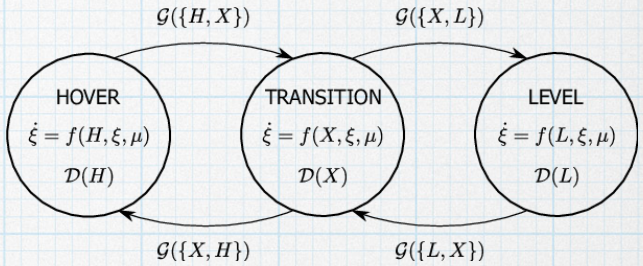
Outline

Messages:
- 確率的検証でもスキームは同じ
  （構造を解きほぐして制約充足・最適化問題に帰着）
- 特に，束論・圏論のレベルではしばしば同じになる
- 線形計画法に落とせば勝ち
- 現実的課題，理論的に豊穣 （不動点理論，集中不等式等）
  ➜ 研究上やることがたくさん

	モデル（➜ 実応用可能性）	不確かさの種類	解析手法
確率的モデル検査，確率的検証 [Kura+, TACAS'19] [Phalakarn et al., CAV'20] を例に	モデル要 (whitebox)	known unknowns	formal
実行時検証，モニタリング [Waga+, CAV'19] を例に	モデル不要	unknown unknowns	formal
サーチベーステスト＋論理・離散的構造 [Zhang+, CAV'19] を例に	blackbox モデルのみ要	unknown unknowns	testing
自動運転システムの安全性検証 [Kobayashi+, NFM'21] を例に	部分的モデル要	unknown unknowns	formal

（モデル検査＋連続量）の一般論

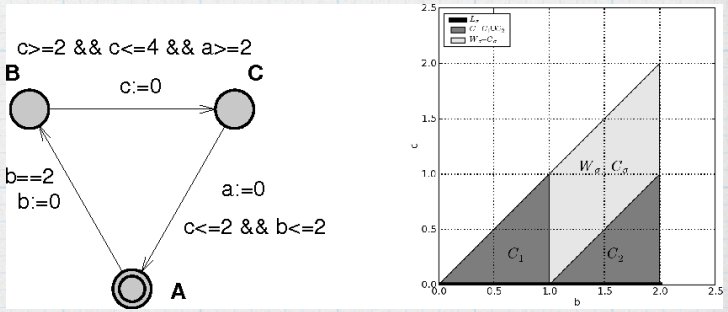
* 物理情報システムへの適用
  * 一義的には，連続量が現れた時点でアウト
    * 状態空間が無限になって，探索できない
    * 例： ハイブリッド・オートマトン
      （オートマトン＋微分方程式）では，到達可能性が決定不能



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]

×

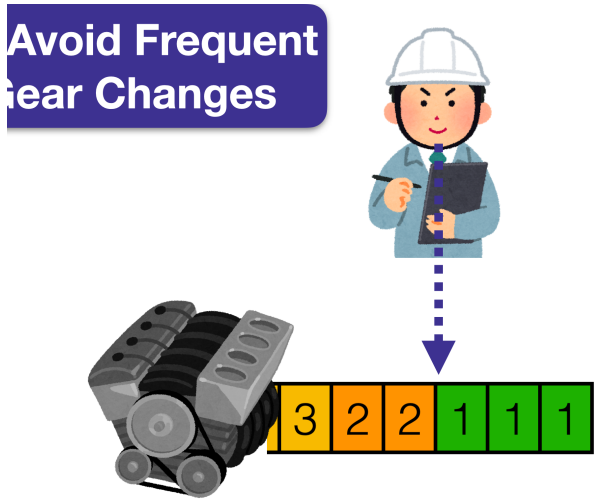
- →
  確率的システム，timed automaton の研究の勃興（特に欧州）
- 物理情報システム，工業製品の品質保証にも有効
- ただし，
  （連続量の有無に関わらず）
  モデル検査の物理情報システムや工業製品への実適用は，モデリングのコストがバカ高い…
  航空宇宙くらいしか割に合わない？

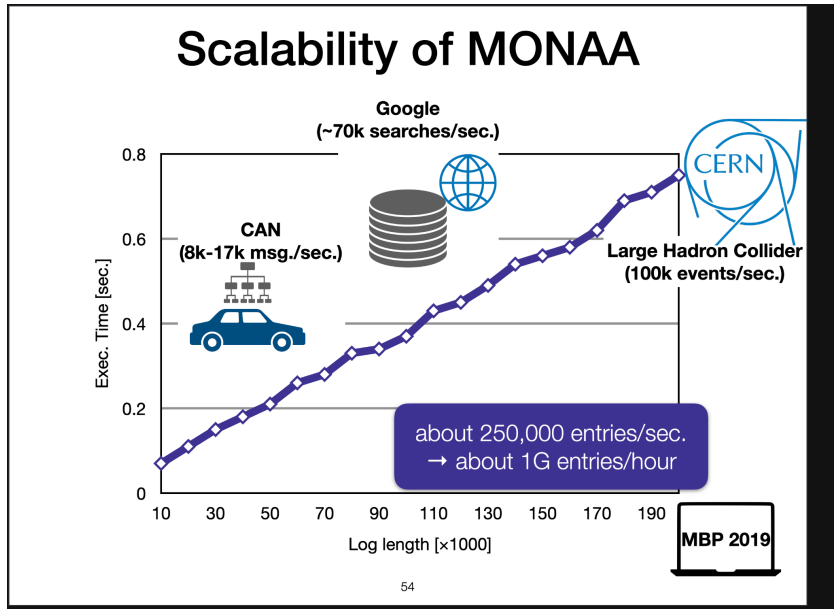
実行時検証，モニタリング

- （参考）モデル検査
  - 入力： システムモデル M，仕様 ϕ
  - 出力： M の任意の実行トレース w が ϕ を満たすか否か
    $(\forall w \in L(M).w \vDash \phi)$
- 実行時検証，モニタリング
  - 入力： システムの実行トレース w，仕様 ϕ
  - 出力： w が ϕ を満たすか否か $(w \vDash \phi)$
- 簡単すぎてつまらない？ ➔ no!
  - モデルが不要 ➔ 産業界の実プロセスですぐに利用可能
  - 現場では「目の子モニタリング」「ad-hoc 自動モニタリング」が横行
    ➔ ご利益がわかりやすい
  - ログはしばしば巨大（右図，MONAA は [Waga+, MT-CPS'18]）
    ➔ パフォーマンスへの要求が大きい．
    線形時間計算量・定数空間計算量が望ましい
  - Timed pattern matching は理論的にも challenging.
    実行トレース： 連続時間スタンプ付きイベント列（右）
    仕様： 連続時間制約付き論理式／オートマトン（下）

$$G_{[0,\infty)}((\text{gear} = 1 \wedge \text{RPM} > 3500) \Rightarrow \text{gear} = 1 \,\mathcal{U}_{[0,1.5]}\, \text{gear} = 2)$$

```
C 2928.56
B 2928.6
F 2928.8
E 2928.84
C 2929.04
B 2929.12
F 2929.28
E 2929.32
C 2929.56
```





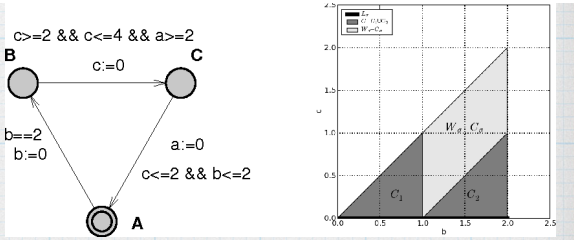
実行時検証，モニタリング ERATO MMSD の成果群

[Waga+, EMSOFT'18] [Andre+, ICECCS'18]
[Waga+, NFM'19] [Waga+, CAV'19]
[Waga, FORMATS'19] [Waga+, ICCPS'21] …

- Timed automaton 理論の応用として取り組む
  - オートマトン
    ＋連続量

* 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]

  - 産業界のニーズ： モニタリングの枠組みの flexibility
    - 仕様にパラメータを含む
    - 文字列等のデータを含む
    - Boolean に限らない「充足度合い」の定量的意味論
    - …

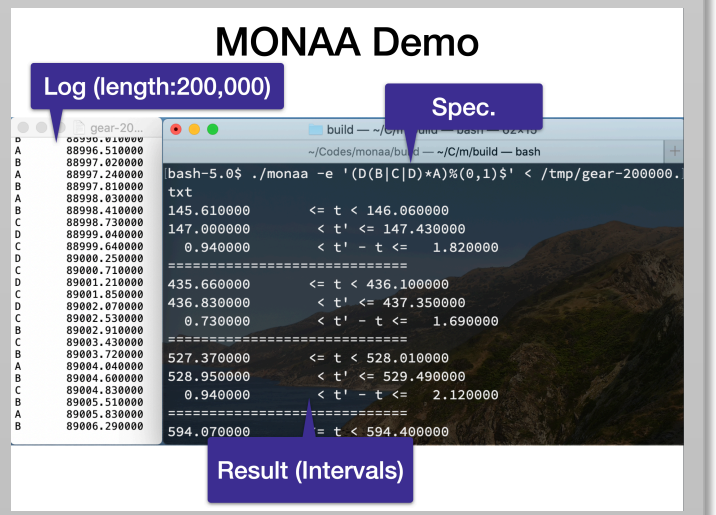
- 方針： polyhedra による空間抽象化の限界を追求
  - Zone 抽象化，線形計画法で効率的操作
  - その範囲の中で枠組みの表現可能性・flexibility を上げていく
  - モデル検査が非決定的になる setting でも，モニタリングであれば実効的に動作

実行時検証，モニタリング
成果例： パラメータ付き timed pattern matching

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

（パラメータなし）
timed pattern matching

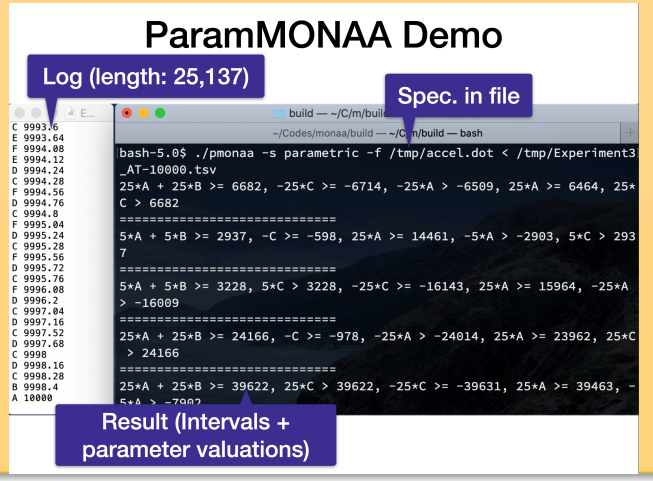
- 入力：
  - timed word　$w = (a, 0.12)\ (b, 1.28)\ \cdots$
  - specification ϕ
    "no occurrence of c for 6 sec. after b"
- 出力：
  - All intervals [t, t'] such that w's restriction $w|[t, t']$ to it satisfies the spec ϕ
- By a laptop, ~ 1M events/sec



パラメータ付き
timed pattern matching

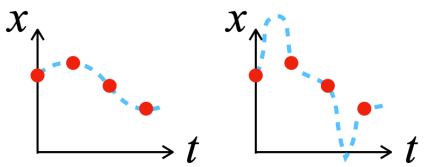
- 入力：
  - timed word　$w = (a, 0.12)\ (b, 1.28)\ \cdots$
  - パラメータ付き specification $\phi(p)$
    "no occurrence of c for p sec. after b"
- 出力：
  - All triples {p0, t, t'} such that w's restriction $w|[t, t']$ to the interval [t, t'] satisfies the spec instance $\phi(p0)$
- By a laptop, ~ 10K events/sec

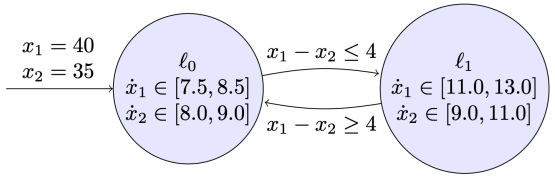
こういうしきい値を決めるのが現場では大変…



実行時検証，モニタリング 研究の展開と展望

- オートマトン理論の応用先として豊穣
  - 実用ニーズ大（企業との共同研究）
  - モデル不要 ➔ すぐにご利益が出る
- モデル検査が undeciable でも，モニタリングはしばしば実効的に動作
- 展開１： 部分モデルの活用 [Waga, Andre, Hasuo, ICCPS'21]
  - 離散サンプルの補間は根源的課題（下左）
  - ”bounding model”（下右） を用いてあり得る補間を制限 ➔ false positive を削減





(a) A bounding model $\mathcal{M}$ for the platooning example, expressed as an LHA

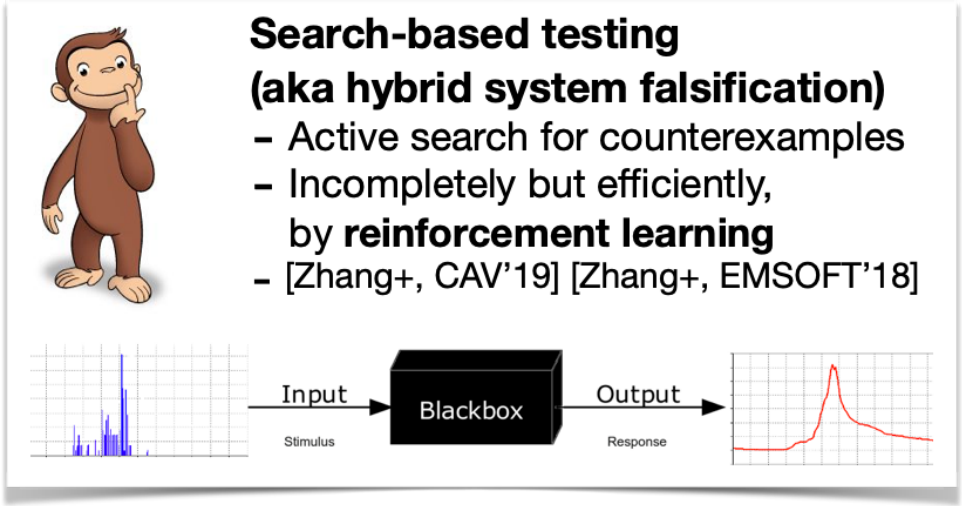
- 展開２： 形式仕様記述支援
  - システムモデルは不要．しかし，形式仕様記述はそれでもむずかしい （ユーザーは論理・オートマトンに習熟しているとは限らない）
  - 形式仕様記述の対話的支援ツールを作成中

Outline

Messages:
- オートマトン理論の応用先としてのモニタリング
- 産業上のニーズも大
- Polyhedra 抽象（←線形計画法）で連続時間・データもさばける
- 形式仕様記述がやはり実用の壁

	モデル（➔ 実応用可能性）	不確かさの種類	解析手法
確率的モデル検査，確率的検証 [Kura+, TACAS'19] [Phalakarn et al., CAV'20] を例に	モデル要 (whitebox)	known unknowns	formal
実行時検証，モニタリング [Waga+, CAV'19] を例に	モデル不要	unknown unknowns	formal
サーチベーステスト＋論理・離散的構造 [Zhang+, CAV'19] を例に	blackbox モデルのみ要	unknown unknowns	testing
自動運転システムの安全性検証 [Kobayashi+, NFM'21] を例に	部分的 モデル要	unknown unknowns	formal

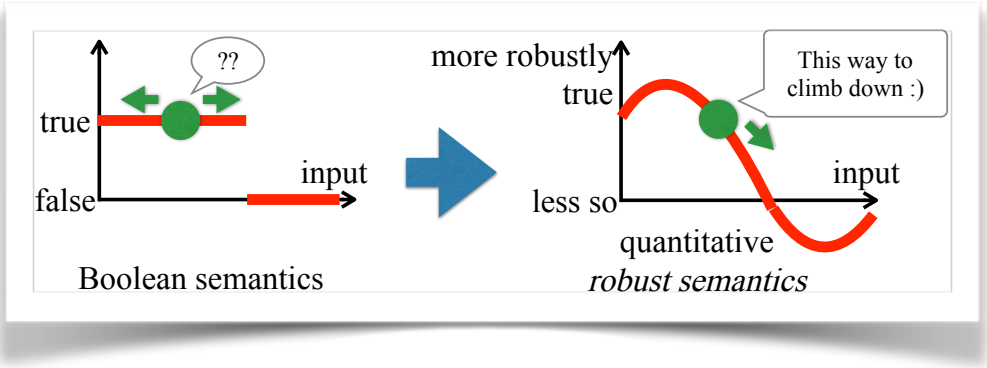
サーチベーステスト＋論理・離散的構造

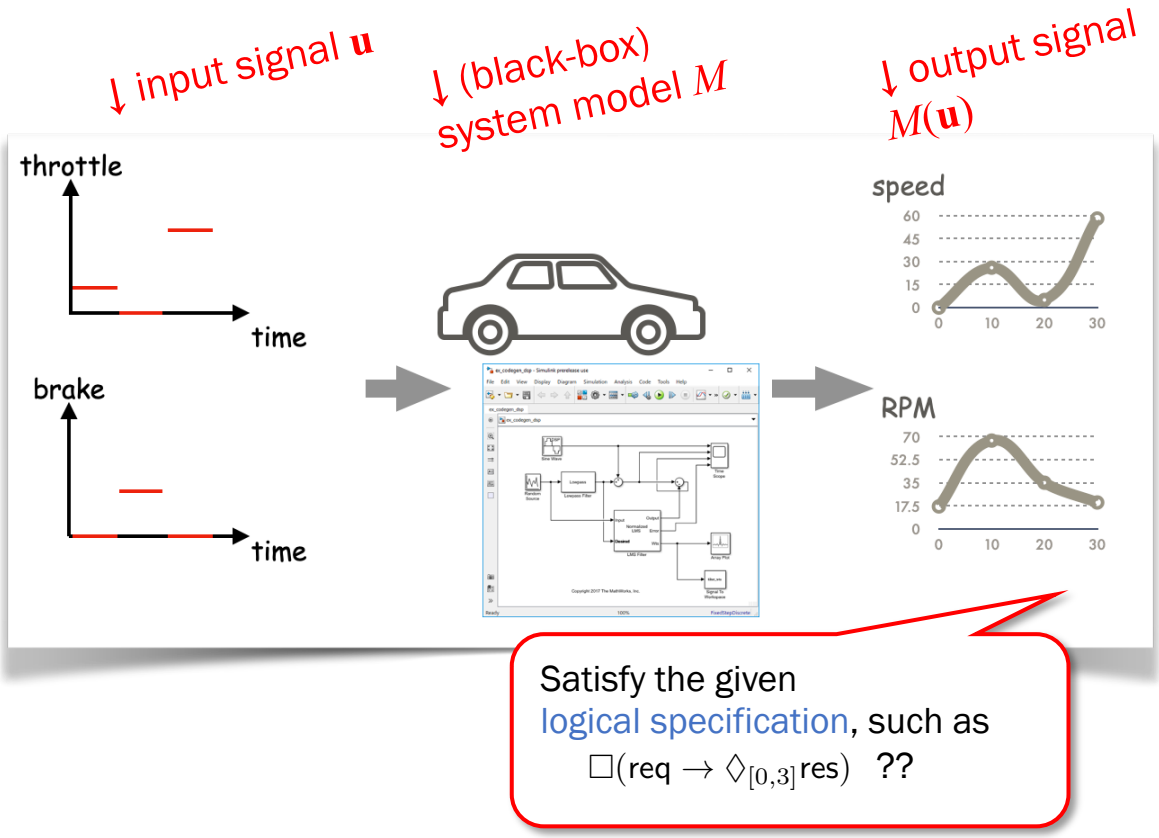


- サーチベーステスト：
  「危ない入力を探す」
- システムはブラックボックス
  - 危ない出力から論理的に逆算，は不可
  - 入力を色々試して探すしかない
    ➔ メタヒューリスティクス（進化計算，…），強化学習
- Hybrid system falsification （反例生成） [Fainekos & Pappas, TCS'09]
  - 「危なさ」を論理式の形式仕様で定義
  - 定量的意味論で，論理式を最適化の目的関数に変換

Hybrid System Falsification

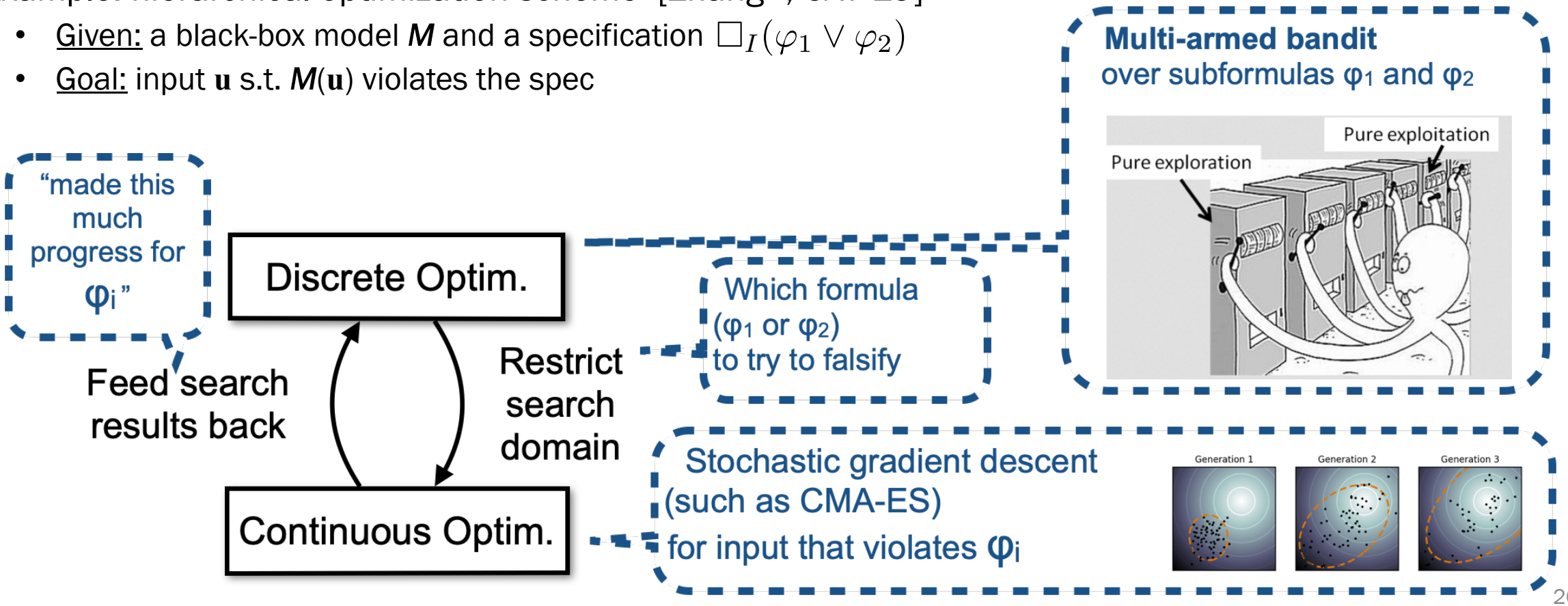
- Hybrid-system falsification: logic-powered search-based testing
- Goal:
  find $\mathbf{u}$ such that the output $M(\mathbf{u})$
  violates the given logical specification
- Enabling technology [Fainekos & Pappas, TCS'09] :
  quantitative robust semantics
  ➔ falsification by optimization





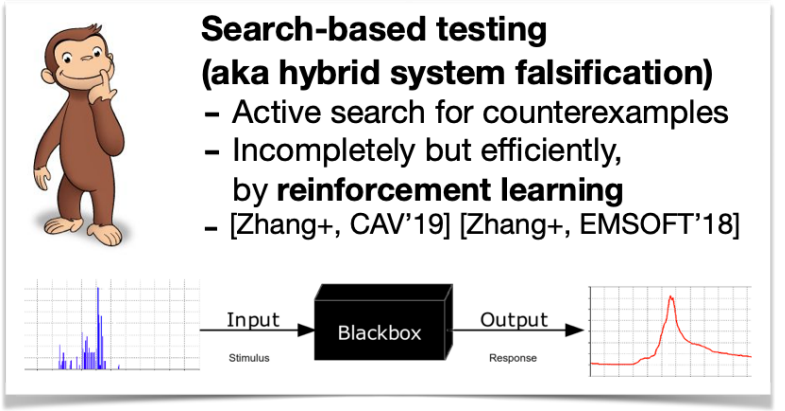
Exploiting Logical Structures

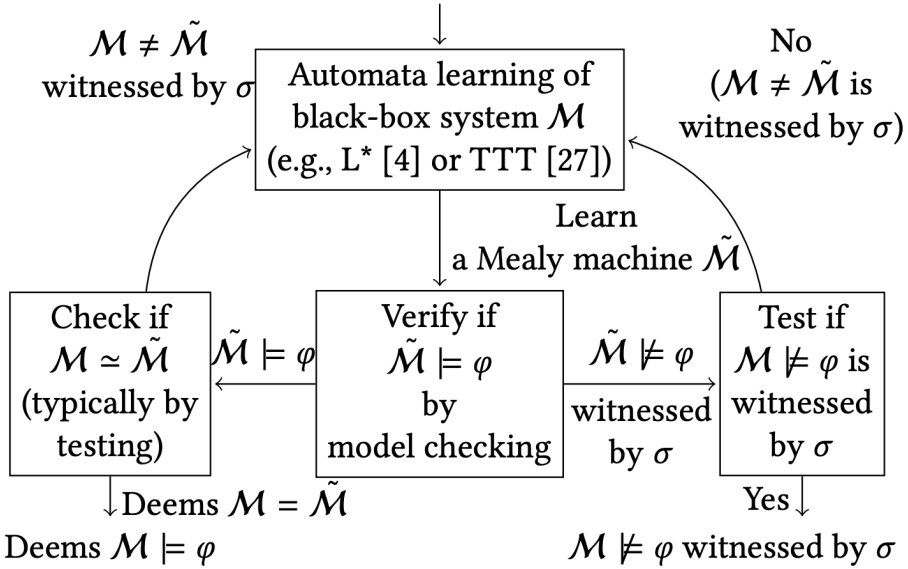
- In hybrid system falsification (and statistical ML in general), efficiency comes mainly from **gradient descent** (which is continuous)
- **Question:** can we also exploit discrete structures for efficiency and explainability?
- Example: hierarchical optimization scheme [Zhang+, CAV'19]
  - Given: a black-box model ***M*** and a specification $\square_I(\varphi_1 \vee \varphi_2)$
  - Goal: input $\mathbf{u}$ s.t. ***M***($\mathbf{u}$) violates the spec



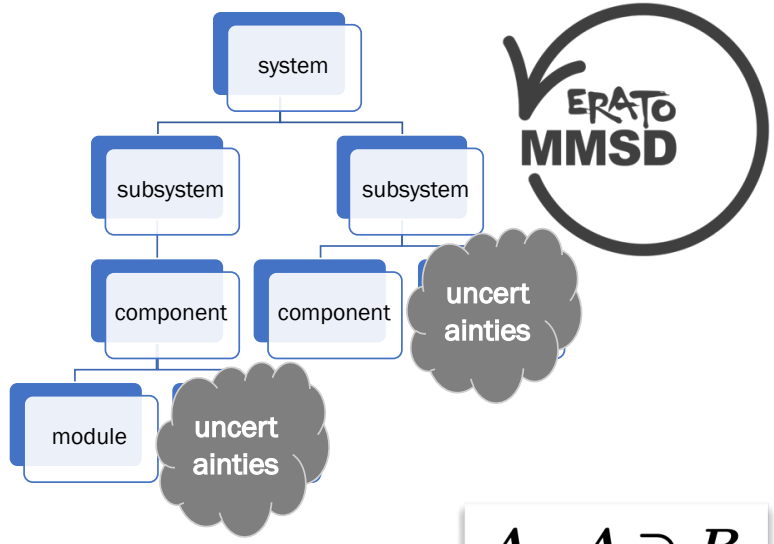
サーチベーステスト＋論理・離散的構造 研究の展開と展望

- 「論理構造
  ＋数値最適化・メタヒューリスティクス」
  はまだまだ掘れるテーマ
  - [Sato+, ADHS'21]:
    制約付き最適化で論理積を解釈,
    危険発見確率を大きく向上
  - 「統計AI＋論理AI」にも近い
- モデル推定との組み合わせ
  [Waga, HSCC'20] [Shijubo+, PPL'21] [Okudono+, AAAI'20] [Gutierrez+, GECCO'20]
  - 入力を試しながら，モデルをだんだん
    ホワイトボックス化
  - 近似モデルでモデル検査することで
    反例入力を推定
- 産業応用が容易
  （トヨタ，Volvo，Airbus などから適用報告）

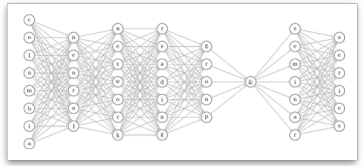




少し脱線：
Logical Confinement of Uncertainties



- Need to cope with uncertainties
  - Physical environment (rich interface)
  - Numeric optimization
  - Data-driven decision making, statistical reasoning
- A related big challenge: reconsiling logical and statistical reasoning
  - Obj.-level stat. reasoning: NNs in CPS
  - Meta-level stat. reasoning: testing
  - Meta-level logical reasoning: safety arguments and proofs



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

Statistical reasoning		Logical reasoning
Allow noisy data	Errors in input	Axioms are absolute
No guarantee	Correctness of concl.	Logical guarantee (mathematical proofs)
High Automatic pattern discovery from data	Scalability	Low Manual preparation of axioms
Low Decision making by “weights”	Explainability	High Explicit deduction processes as proofs
Statistical ML Neural networks, regression, …	Used in	Logic programming (Most) human communication

Logical Confinement of Uncertainties

ML Component as
“Super-Sharp but Not-Totally-Reliable Colleague”

LSD - Genius ft. Sia, Diplo, Labrinth

* Often super sharp
  “Wonderful idea! How did you get it?”
* … for no clear reason
  “It just came down to me…”
* … and not always so
  “How come you made this stupid mistake?”

* Great to have one in a team;
  you don’t want all to be like that

Logical Confinement of Uncertainties

ML Component in Safety-Critical Cyber-Physical Systems

* Take its opinion with a pinch of salt
* **Suggestion**, instead of **decision**

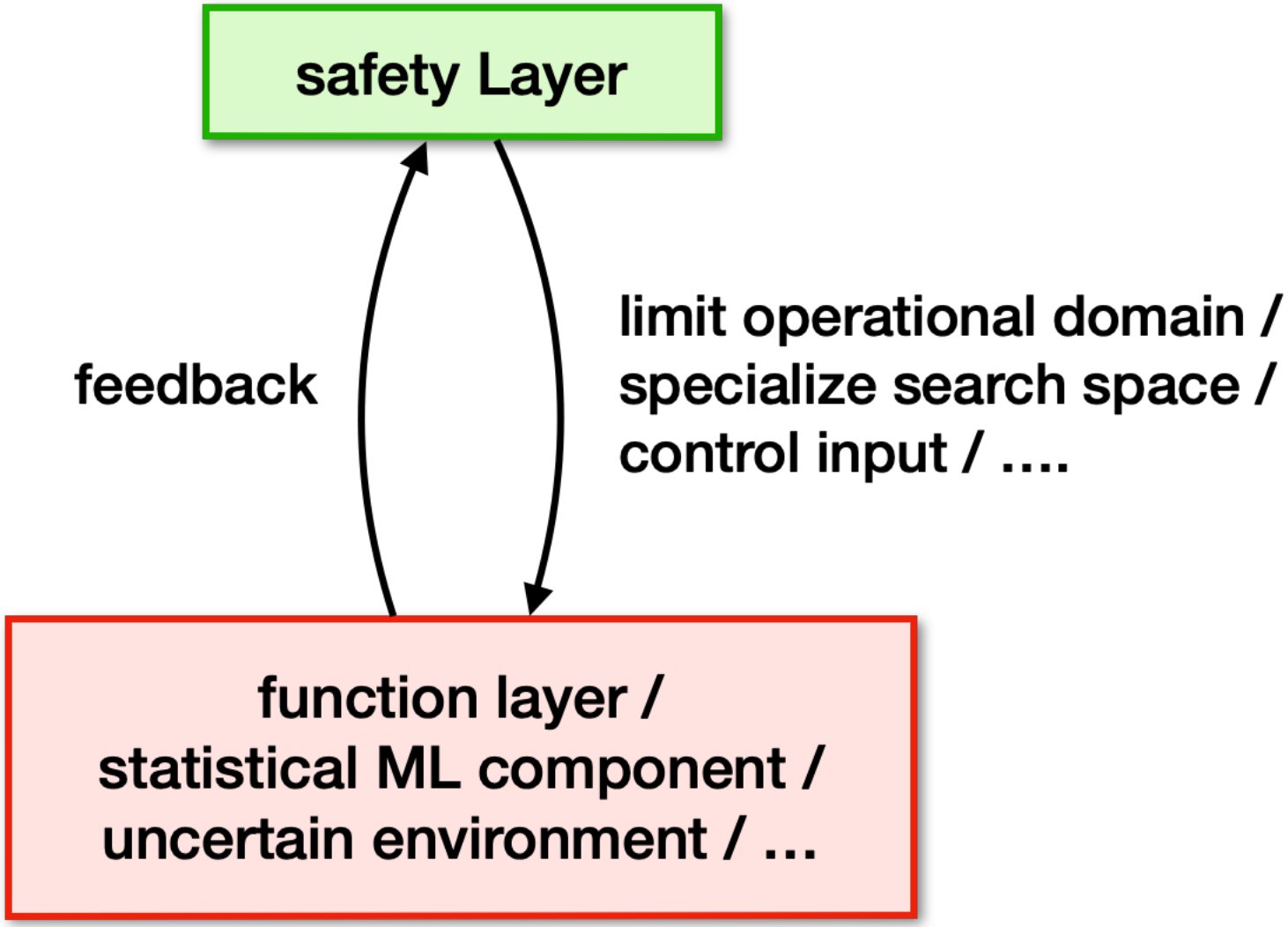
Extreme example: proof check

* System-level assurance

Towards safe AI and XAI...

- A common approach: examine the internal working of AI
- Here: safety envelope, logical confinement

Logical Confinement of Uncertainties



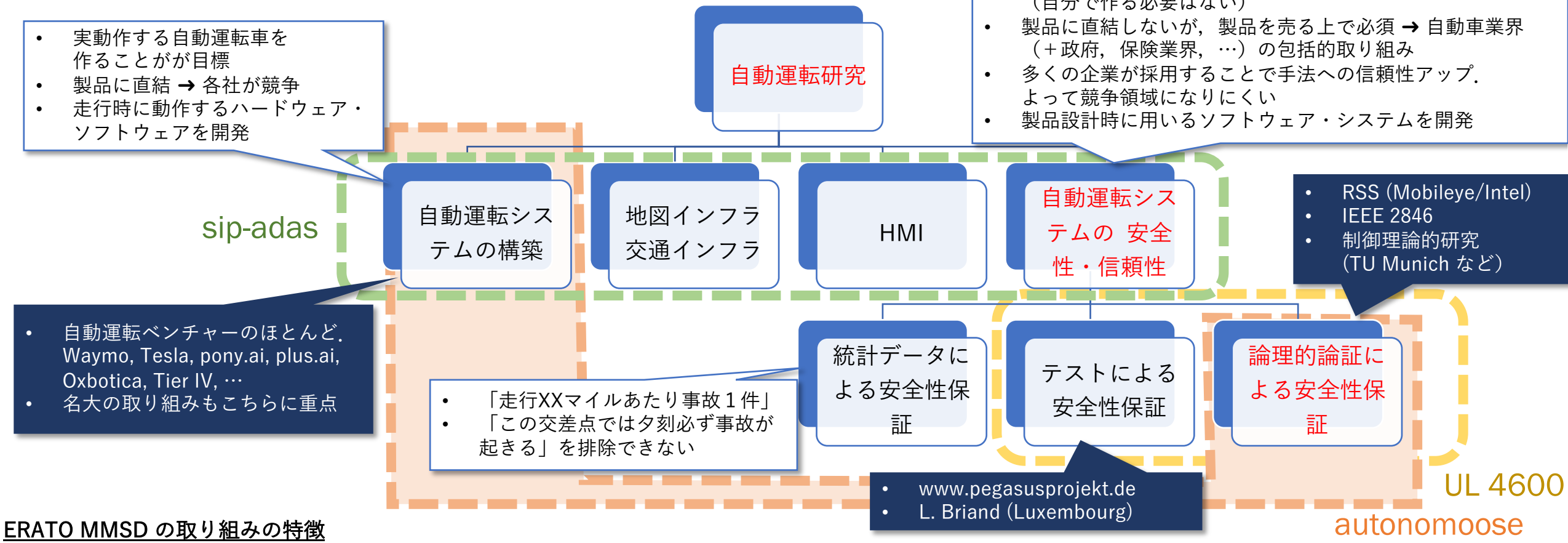
Outline

Messages:
- 論理を使った賢いバグ探し
- ニーズ大，実適用容易．わかりやすいご利益
- 論理と数値最適化の組み合わせ．階層的最適化フレームワーク
- 「不確かさの論理的封じ込め」とも関連

	モデル（→ 実応用可能性）	不確かさの種類	解析手法
確率的モデル検査，確率的検証 [Kura+, TACAS'19] [Phalakarn et al., CAV'20] を	モデル要 (whitebox)	known unknowns	formal
実行時検証，モニタリング [Waga+, CAV'19] を例に	モデル不要	unknown unknowns	formal
サーチベーステスト＋論理・離散的構造 [Zhang+, CAV'19] を例に	blackbox モデルのみ要	unknown unknowns	testing
自動運転システムの安全性検証 [Kobayashi+, NFM'21] を例に	部分的モデル要	unknown unknowns	formal

自動運転研究の地平

ERATO MMSD は学際研究の牽引役と，わかりやすいキーアプリのため，自動運転研究に戦略的集中



- 実動作する自動運転車を作ることがが目標
- 製品に直結 ➜ 各社が競争
- 走行時に動作するハードウェア・ソフトウェアを開発

- 与えられた自動運転車の安全性保証が目標（自分で作る必要はない）
- 製品に直結しないが，製品を売る上で必須 ➜ 自動車業界（＋政府，保険業界，…）の包括的取り組み
- 多くの企業が採用することで手法への信頼性アップ．よって競争領域になりにくい
- 製品設計時に用いるソフトウェア・システムを開発

- RSS (Mobileye/Intel)
- IEEE 2846
- 制御理論的研究 (TU Munich など)

- 自動運転ベンチャーのほとんど．Waymo, Tesla, pony.ai, plus.ai, Oxbotica, Tier IV, …
- 名大の取り組みもこちらに重点

- 「走行XXマイルあたり事故１件」
- 「この交差点では夕刻必ず事故が起きる」を排除できない

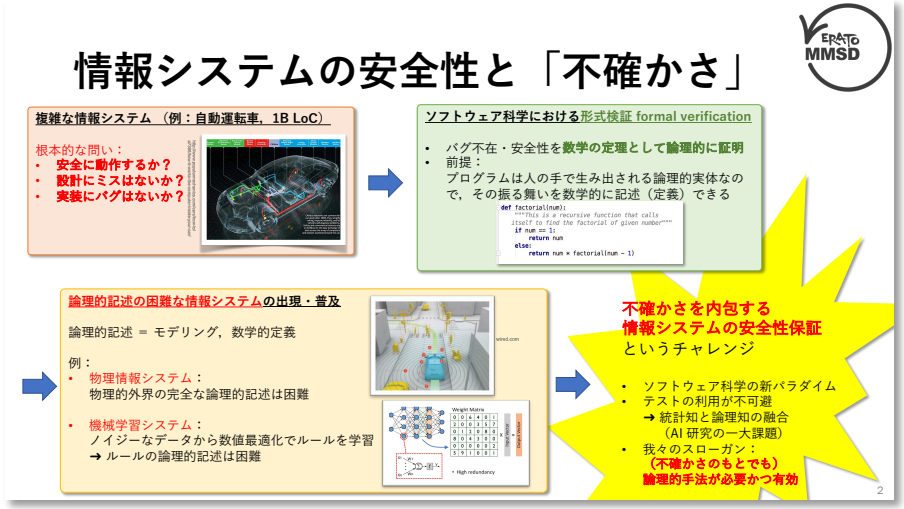
- www.pegasusprojekt.de
- L. Briand (Luxembourg)

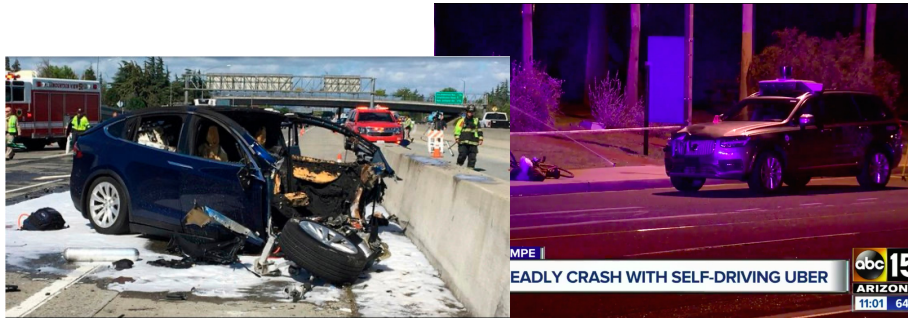
ERATO MMSD の取り組みの特徴

- 安全性・信頼性を確かめる研究（⇔ 自動運転車を作る研究）
- 自動運転安全性の**論理的論証**（最近の関心の高まり ➜ RSS, 国際規格，IEEE 2846 など）　（⇔ 統計データ，テストによる保証）
- ソフトウェア科学・数学・論理学に基づく，基礎的・長射程の研究
- テストや実システム構築とも密接に協働（グループ２@ Waterloo，産業界協働）➜ 応用への即応性も確保

自動運転安全性の論理的論証

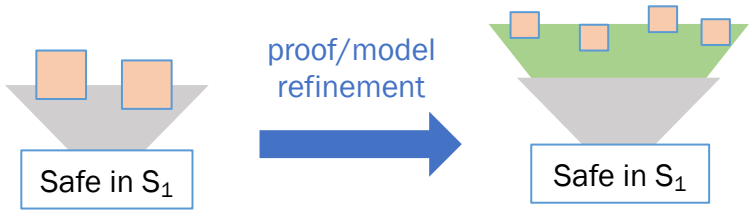
- ?? 「モデル化不可能なので検証不能」と言ったはずでは…？
- → 検証（＝証明）でなく論証.
  ”safety case” “safety argument”
- 安全であることの論理的議論の積み上げ
  - 設計者が duty of care を果たしていることの証拠
  - 事故の際には，論証を辿ることで原因の特定が容易に
  - 「テスト結果の解釈」とも考えられる
- 自動運転安全性の基準・規格は五里霧中状態
  - ISO 26262/21448, UL 4600, IEEE 2846, …
  - （2018年の事故以降 hype がしぼんだ大きな理由はこれ）
  - 論理的論証・議論に注目が集まっている．ルールベースの安全性解析
    例： UL 4600, IEEE 2846, RSS (responsibility-sensitive safety)





Our Approach: Decompositional Proofs with Open Assumptions

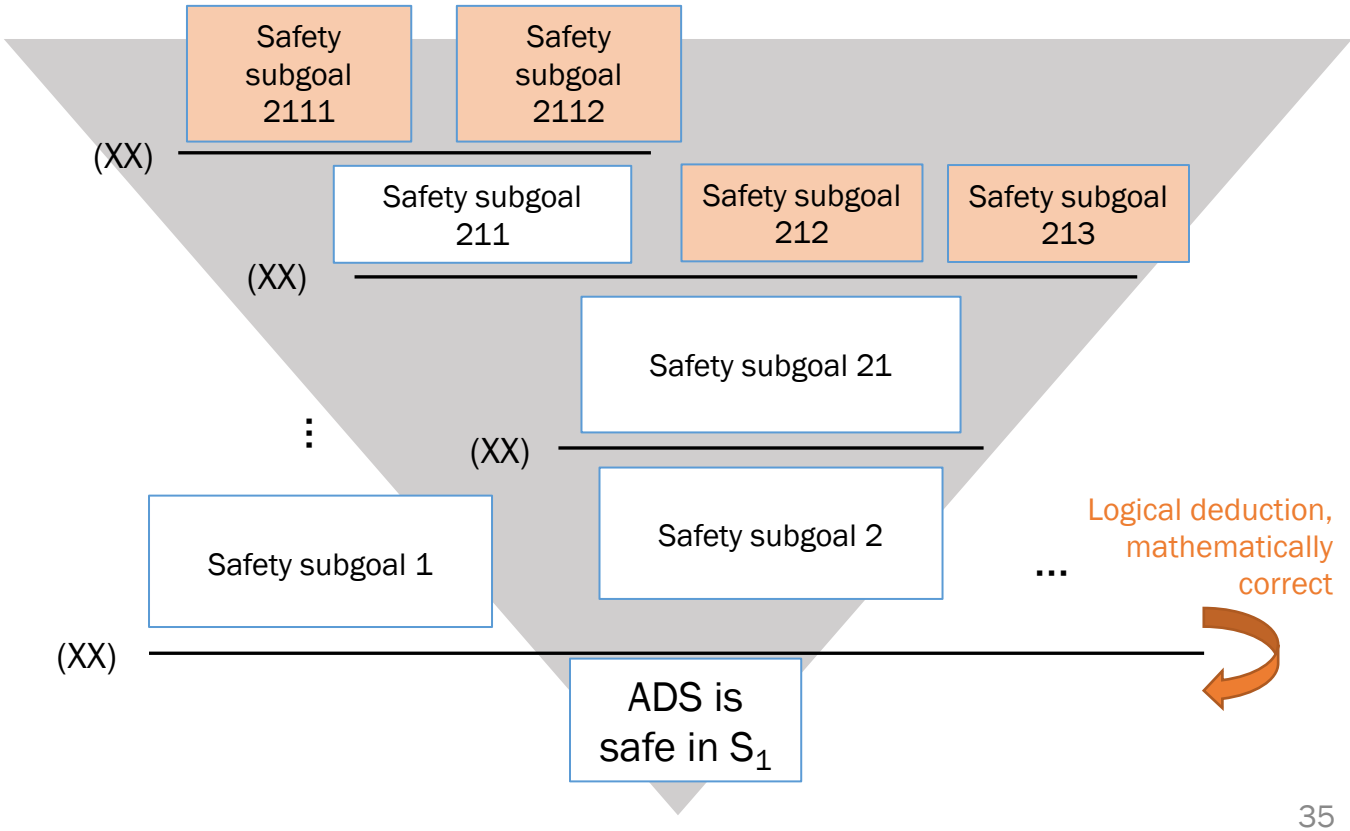
- Goal:
  a decompositional formal proof for ADS safety
  - Rigorous. Mathematical correctness
  - Uncertainties confined into open assertions,
    which can be tested or monitored
  - Can be refined by further efforts



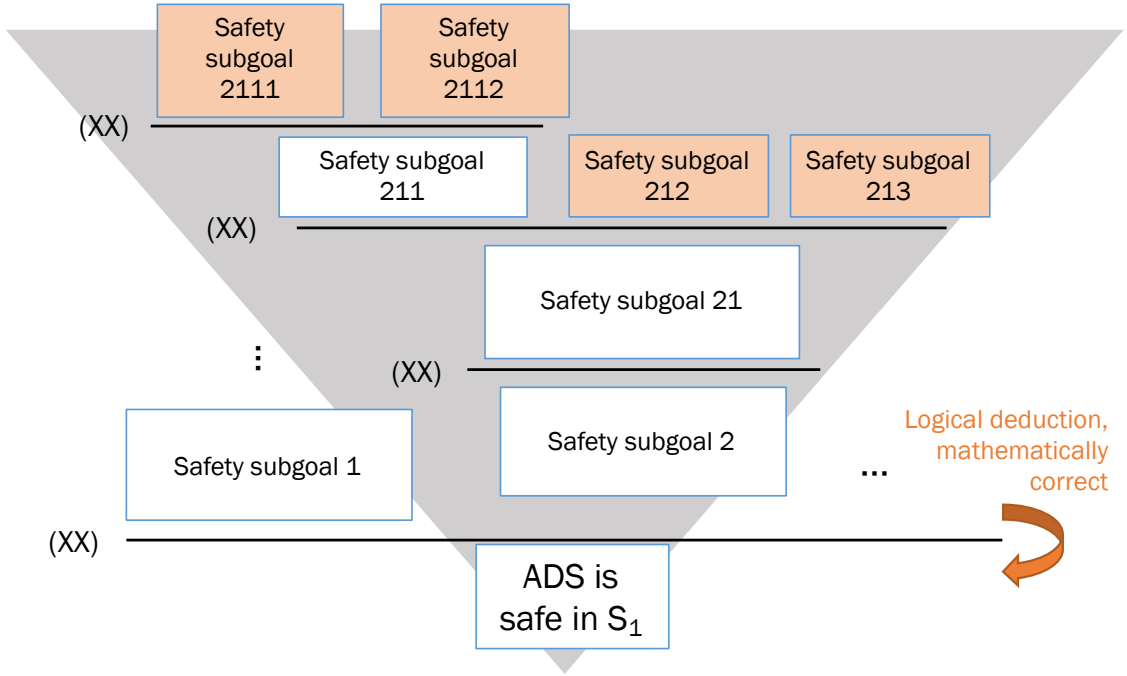
  - Tree-style arguments are not rare…
    but this one is formal
    with mathematical correctness
- Ongoing collaboration with a car company & Waterloo

Technical element:
Event-B and Rodin
- Formal modeling and verification framework
- Allows iterative refinement
  → suited for accommodating partial proofs
- A technical result for accommodating uncertainties: [Kobayashi+, NFM'21]

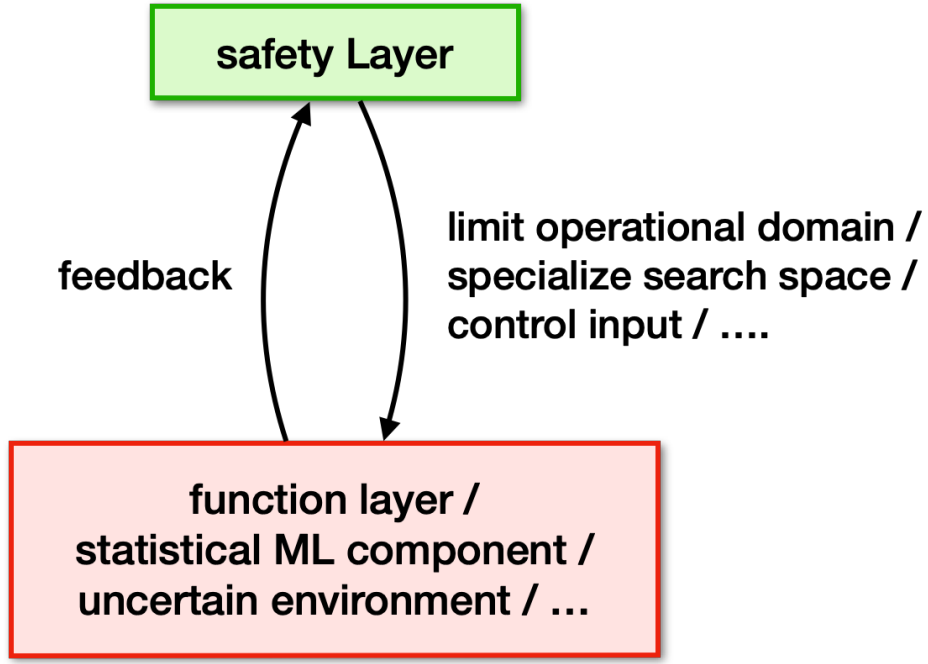
Open assertions (unproved subgoals)
→ tested offline, or runtime-monotored



Logical Confinement of Uncertainties



as an instance of



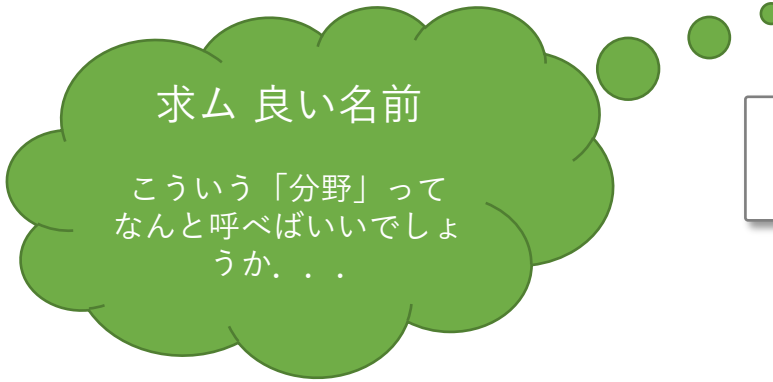
Outline

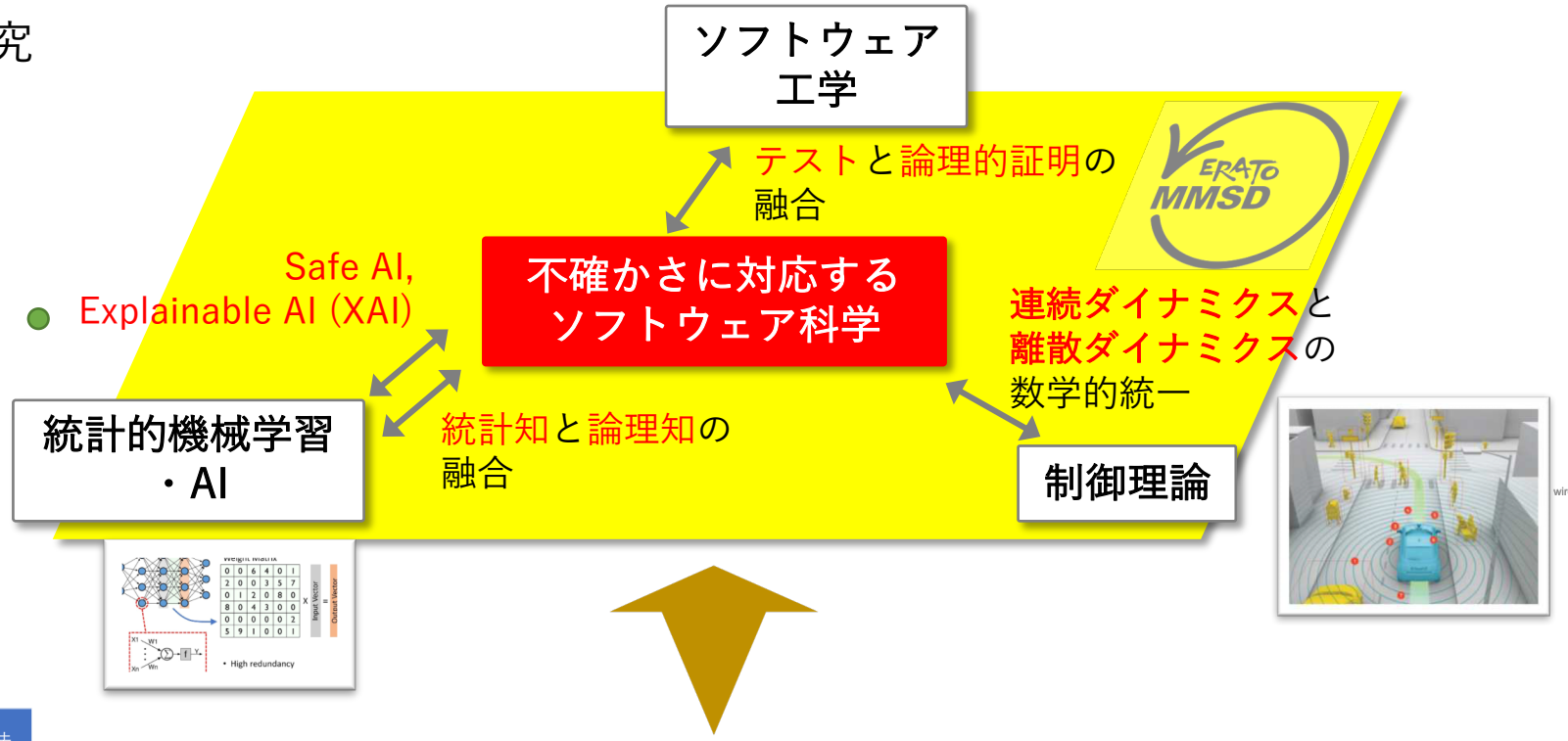
Messages:
- 自動運転の安全性保証は未開の領域．社会受容には必須
- 論理的論証，ルールベースの安全性保証への関心の高まり
- Decompositional proofs with open assertions が論理的論証に対応
- Event-B & Rodin によるモデルの逐次的詳細化が有効

	モデル（→ 実応用可能性）	不確かさの種類	解析手法
確率的モデル検査，確率的検証 [Kura+, TACAS'19] [Phalakarn et al., CA	モデル要 (whitebox)	known unknowns	formal
実行時検証，モニタリング [Waga+, CAV'19] を例に	モデル不要	unknown unknowns	formal
サーチベーステスト＋論理・離散 構造 [Zhang+, CAV'19] を例に	blackbox モデルのみ要	unknown unknowns	testing
自動運転システムの安全性検証 [Kobayashi+, NFM'21] を例に	部分的 モデル要	unknown unknowns	formal

まとめ：
不確かさに対応するソフトウェア科学

- 不確かさのもとでのソフトウェア研究
  ➔ 研究スコープの拡大が必要
- 論理的理論・手法はそれでも
  有効かつ必要





	モデル（➔ 実応用可能性）	不確かさの種類	解析手法
確率的モデル検査，確率的検証 [Kura+, TACAS'19] [Phalakarn et al., CAV'20] を例に	モデル要（whitebox）	known unknowns	formal
実行時検証，モニタリング [Waga+, CAV'19] を例に	モデル不要	unknown unknowns	formal
サーチベーステスト＋論理・離散的構造 [Zhang+, CAV'19] を例に	blackbox モデルのみ要	unknown unknowns	testing
自動運転システムの安全性検証 [Kobayashi+, NFM'21] を例に	部分的モデル要	unknown unknowns	formal

ご清聴ありがとうございました．蓮尾 一郎（国立情報学研究所，twitter @IchiroHasuo）
- 関連する興味をお持ちの方，お声をおかけください．一緒にやりましょう！
- 大学院生（総研大）募集中．トピック・環境・経済的待遇などご相談ください．