高信頼自動運転システム実現にむけて

ソフトウェア工学・機械学習の視点からの技術俯瞰

国立情報学研究所　石川　冬樹

f-ishikawa@nii.ac.jp / @fyufyu

http://research.nii.ac.jp/~f-ishikawa/

ERATO MMSD

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

自己紹介

- ソフトウェア工学と先端自律・スマートシステム
  特に品質・ディペンダビリティに関する技術
  - 形式手法，テスティング，自己適応，
    最適化，要求分析，安全性論証など
  - サービス合成（クラウド・IoT），CPS/AI（特に車）
- 産業界向け教育・応用研究
  - トップエスイー，日科技連SQiP，電通大社会人博士
- 最近（1）：自動（運転）車ソフトのテスト
- 最近（2）：機械学習ベースAIのテスト・品質

機械学習工学研究会
MACHINE LEARNING SYSTEMS ENGINEERING

AIプロダクト品質保証コンソーシアム

形式手法

- 使っている人は使っている
  - クラウド内部の複製管理：網羅的な検査，1チームの成功から組織に展開（AWS）
  - モバイルアプリ：メモリリークに関する静的解析をリリースサイクル上で自動起動（Facebook）
  - 自動運転地下鉄やホームドアのコア部分：正しさの証明済みコードを生成，単体テスト不要に（ClearSy）
  - 組み込みチップ：複数組織に展開する外部仕様を厳密化，テスト，仕様に起因する不具合をゼロに（FeliCa）

[J. Abrial, Formal Methods in Industry: Achievements, Problems, Future, ICSE'06]
[C. Newcombe et al., How Amazon Web Services Uses Formal Methods, Comm. of the ACM'15]
[C. Calcagno et al., Moving Fast with Software Verification, NFM'15]
[T. Kurita et al., Practices for Formal Models as Documents: Evolution of VDM application to "Mobile FeliCa" IC Chip Firmware, FM'15]

とはいえ

- 開発対象全体の全問題に対処できるわけではない
- スケーラビリティに懸念
  - 特に，連続系・物理的挙動を扱う場合
- というかSimulinkモデル and/or コードしかない
  - （厳密に意味論が定まっている）モデルは，
    （メンテされて）ない
- 今のやり方を変えられない
  （上流にかける工数，エンジニアスキル，などなど）
  - 費用対効果には不確かさが多く踏み込めない
  - 移行コストを乗り越えられない

ということで,

私からは「力業」の話を

実行しまくって確認する！

「試行＆評価」サイクル！

キレイに言うと

「発見的・経験的アプローチ」

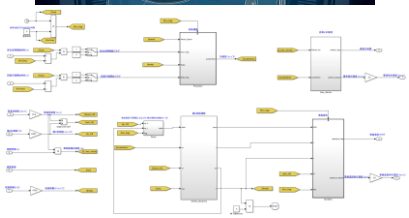
（ものすごく簡単な）問題イメージ

Simulinkモデル：
自動ブレーキ付の車の挙動

入力

ユーザの挙動
- アクセルペダル
- ブレーキペダル

環境
- 初期速度
- 歩行者の位置・動き
- 路面状況
- …



出力
- 衝突有無
- 衝突速度

- *実行可能なものがある（シミュレーションモデルやコード，実機）*
- *一つの入力を決めて，出力を得ることはできる*

➡ *「危ない」ケースを見つける*

・・・だけではなく何をする？

関連動向

ソフトウェアテスティング分野の一潮流

- サーチベースドテスティング（Search-based）
  - 最適化技術（特に進化計算・メタヒューリスティック）により「欲しいもの」を表すスコアを最大化するようなテストスイートやテストケースを生成
    - 例：「車が人に最も近づくような危ういテスト」ケースを生成
    - 例：「前バージョンと比べ出力が大きく変わる」入力を発見
    - 例：「高カバレッジで数が小さい」回帰テストスイートを生成

「試行＆評価」サイクル！

[S. Ali et al., Systematic Review of the Application and Empirical Investigation of Search-Based Test Case Generation, 2010]
[http://www.evosuite.org/]

おまけ：サーチベースドテスティングのレベル

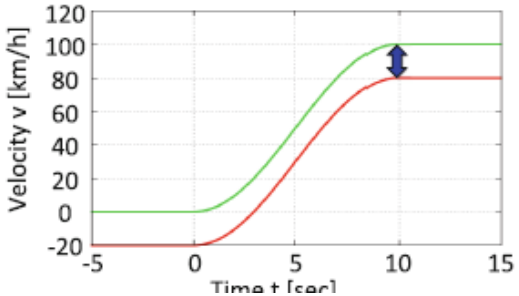
- 2018年のJava Unit Testing Competition
  - 比較用に複数ツールを組み合わせた「最強ツール」は，人が作ったものよりよいテストスイートを10秒で生成
  - ここでの評価基準は，コードカバレッジとミューテーションスコア（人工バグの検出率），テストケース数
- Facebookでの事例
  - モバイルアプリのContinuous Integrationに組み込み
  - テスト数が少なく，コマンド数が短く，多くのクラッシュを報告するようなテスト一式をまとめあげる
  - サーチベースドのバグ自動修正ツールも連動

[https://github.com/PROSRESEARCHCENTER/junitcontest/blob/master/README.md#6th-junit-contest]
[Molina et al, Java Unit Testing Tool Competition - Sixth Round]
[https://code.fb.com/developer-tools/finding-and-fixing-software-bugs-automatically-with-sapfix-and-sapienz/]

形式手法分野での一潮流

■検証式の定量化

シミュレーション結果例（２サンプル）



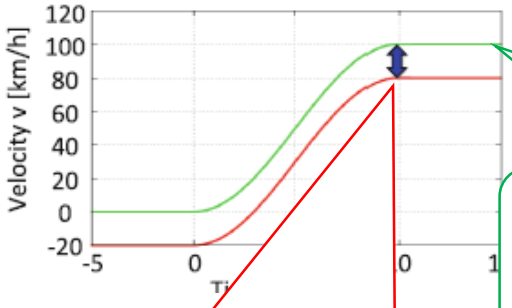
イベントBが起きてから**10秒以内**に
速度は **80km/h 以上**になる

[Fainekos et al., Robustness of temporal logic specifications for continuous-time signals, TheoCompSci'09]
Figure from [Akazaki et al, Time Robustness in MTL and Expressivity in Hybrid System Falsification, CAV'15]

形式手法分野での一潮流

■定量検証式の定量的な充足評価

シミュレーション結果例（２サンプル）



イベントBが起きてから10秒以内に
速度は 80km/h 以上になる

OK！10秒後，求められたよりも20km/h余裕を持って

ロバストな充足

OK！10秒後，ちょうどギリギリ求められた値クリア

危うい充足（違反に近い）

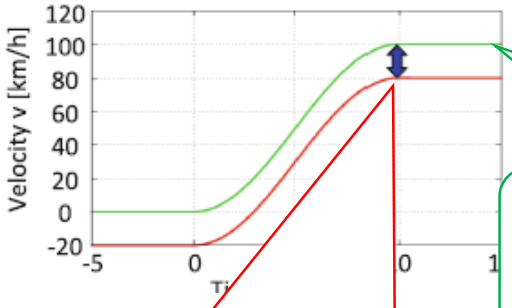
（時間についても同様な議論ができる）

[Fainekos et al., Robustness of temporal logic specifications for continuous-time signals, TheoCompSci'09]
Figure from [Akazaki et al, Time Robustness in MTL and Expressivity in Hybrid System Falsification, CAV'15]

形式手法分野での一潮流

■定量検証式に対する最適化を用いた反例探索

シミュレーション結果例（２サンプル）



イベントBが起きてから10秒以内に
速度は 80km/h 以上になる

OK！10秒後，求められたよりも20km/h余裕を持って

ロバストな充足

OK！10秒後，ちょうどギリギリ求められた値クリア

危うい充足（違反に近い）

「試行＆評価」サイクル！

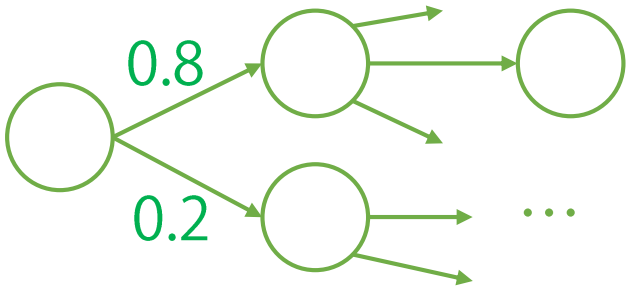
（時間についても同様な議論ができる）

➡「ロバスト度合い」の最適化問題に帰着

[Fainekos et al., Robustness of temporal logic specifications for continuous-time signals, TheoCompSci'09]
Figure from [Akazaki et al, Time Robustness in MTL and Expressivity in Hybrid System Falsification, CAV'15]

形式手法分野での別の一潮流

- 確率モデル検査
  - マルコフ過程などの確率モデルに対する検査



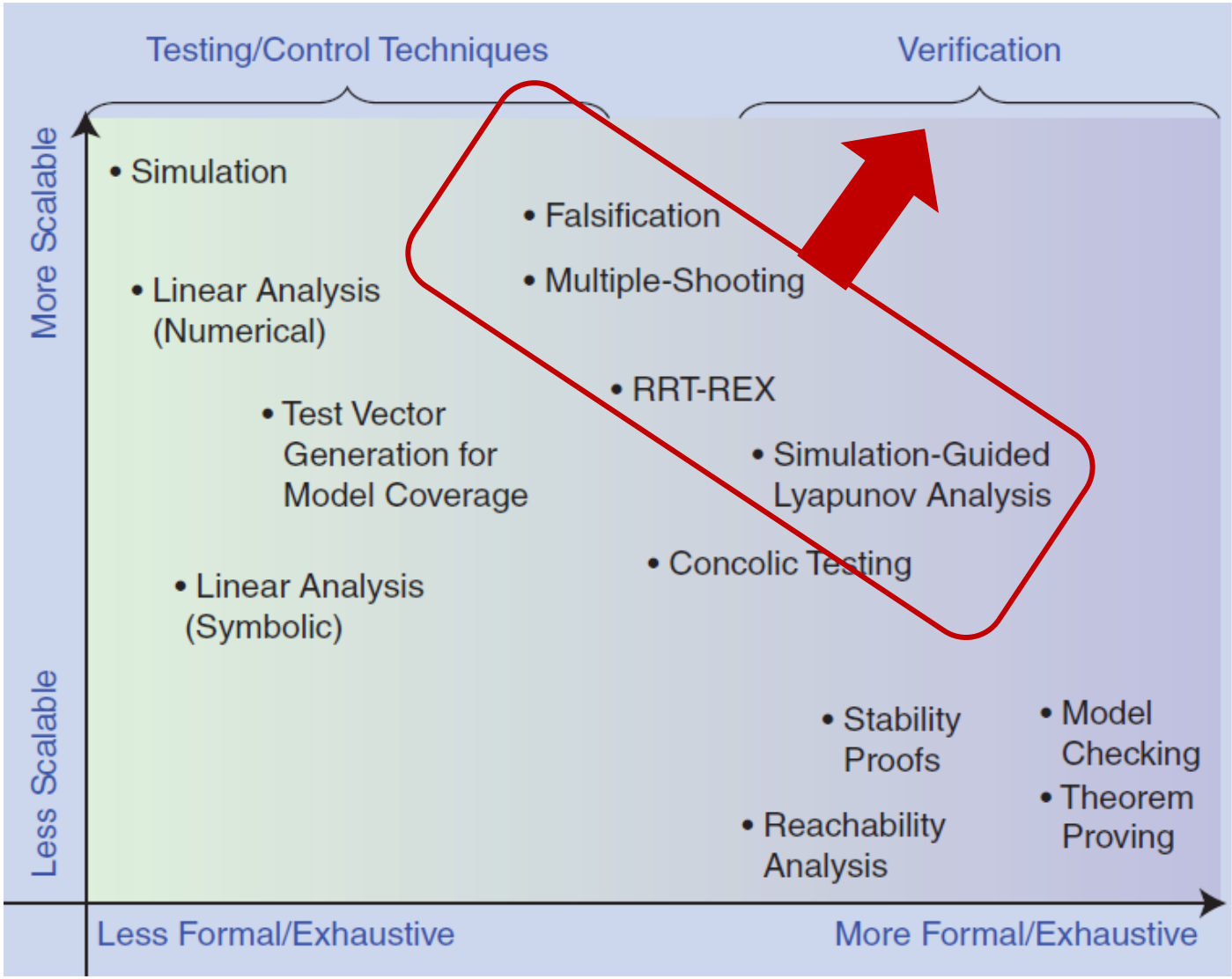
- 発生確率の計算は高コスト

- 検証項目に対する*「仮説検証」*をしてしまうことにする（統計的モデル検査）
  - とにかく大量に実行してしまえばよい
  - ドメイン知識・掘り出した知識で加速可能

「試行＆評価」サイクル！

[Jha et al., A Bayesian Approach to Model Checking Biological Systems, CMSB'11]
[Zuliani et al, Bayesian statistical model checking with application to Stateflow/Simulink verification, FMSD'13]

さらに制御分野の潮流

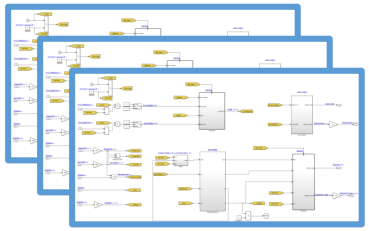


[Simulation-Based Approaches for Verification of Embedded Control Systems, IEEE Control Mag.'16]

プロジェクトでの取り組み

ERATO-MMSD Group3 の取り組み

「マルチプロダクトライン」エンジニアリング



- 多様な設計，環境，要求・ハザード

  を個別ではなくまとめて統合的に分析・テスト

我々の探索空間！

「インテリジェンス協働」によるテスト技術の深化

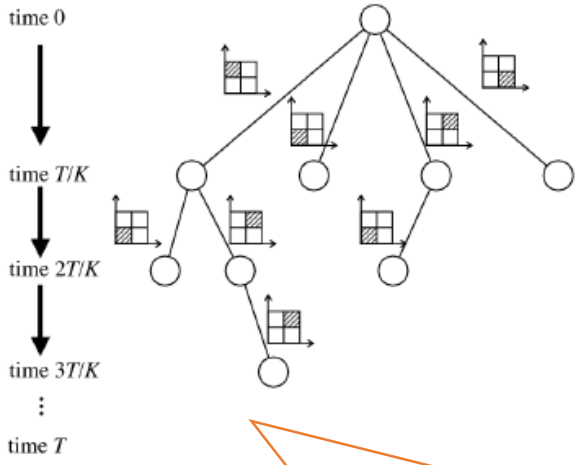
産業界からの問題投入により加速

- 検査式やモデルの構造に応じた探索
- 進化計算に限らないAI技術の活用
- 「悪いケース探し」に限らず知見発掘

[Ali et al., Towards a Framework for the Analysis of Multi-PLs in the Automotive Domain, 2019]

研究事例：より探索的な反例探索

- 系統的な探索手法を適用
  - たまたま見つけた悪い例に深入りしない
  - 探索空間全体に対する情報を随時管理・更新
- Monte-Carlo Tree Search
  - 囲碁プレーヤの構築などで有名なAI技術
  - 連続変化する入力シグナルを分割，「匂い」「怪しさ」を継続記録し「試行＆評価」サイクル

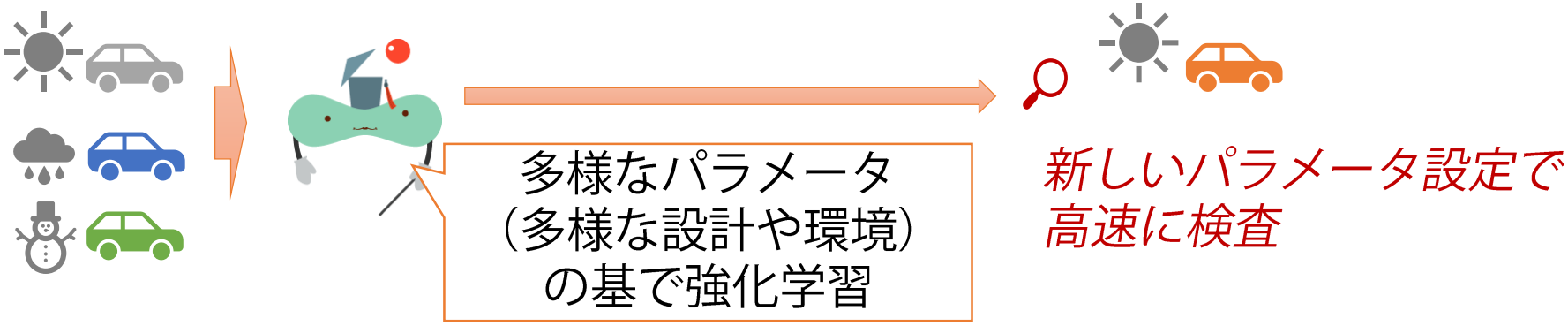


最初にアクセルを踏み込んだらどうだろうか？その次は・・・

[Zhang et al., Two-Layered Falsification of Hybrid Systems Guided by Monte Carlo Tree Search, EMSOFT/TCAD'18]

研究事例： 事前訓練による反例探索

- 探索方法を学習
  - 毎回ランダムから探索するのを避ける
  - 囲碁プレーヤが様々な相手に対し事前訓練するように
- モデルファミリー式に対する反例生成器の構築
  - 試行錯誤の度に行う検査を高速に
  - 設計や環境の多少の変化があっても高速に検査



[Kato et al., Falsification of Cyber-Physical Systems with Reinforcement Learning, MT-CPS'18]

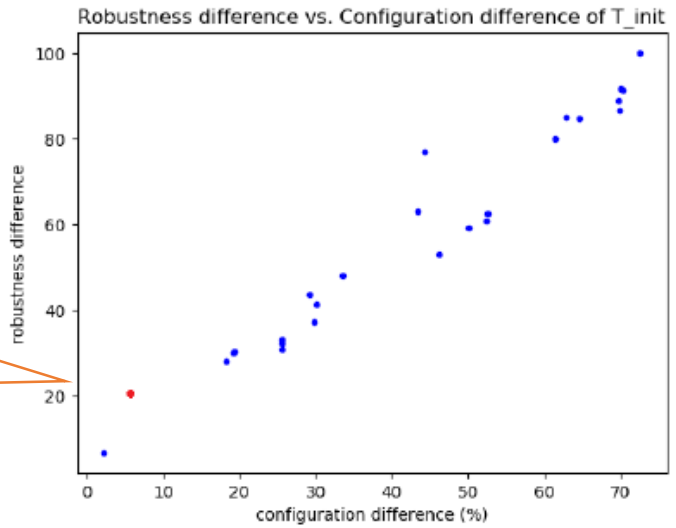
研究事例： 安定性（敏感性）分析

- 不安定なパラメータ領域を探索

  *パラメータ値の少しの変更が安全や品質に大きく影響する*ような状況を報告

  - 設計パラメータ（避けるか重点的にテスト）も，
    環境パラメータ（重点的にテスト）も

動き出し時間設定の
0.12 秒の差により，
衝突無しの状況が
20km/h衝突に変わる



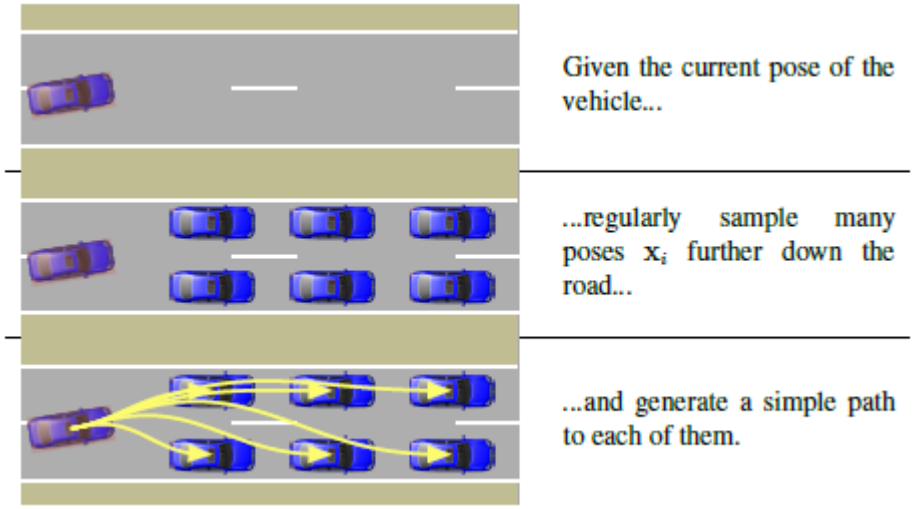
[Lee et al., Stability Analysis for Safety of Automotive Multi-Product Lines: A Search-Based Approach, GECCO'19]

システムテストレベルへ

システムレベルの例： 経路計画

- 自車の経路を定める部品

[Figure from McNaughton, Parallel Algorithms for Real-time Motion Planning]



- 短期・長期のゴール・制約を踏まえて直近の制御操作を決める，というサイクルの反復
- 障害物との距離，レーン遵守，車両の物理的限界，なめらかさ・快適さ，法令遵守，運転慣習遵守，・・・

パラメータ空間を探索？

- パラメータ空間
  - 直接的にはシミュレータで設定可能な全可能性？
    （車両の重さなど，歩行者や他の車の位置・動き，
    信号の変化，道路の形状や状況，・・・）

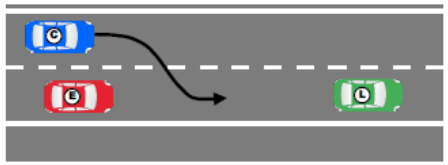
➡ **安全論証にほぼ貢献しない設定の方が大多数**

  - 安全過ぎる： 歩行者が遠くの方で道路を横切っている
  - 危険過ぎる： 自車が全方向から他の車に迫られて
    どうやっても避けられない
  - 現実的でない： 道路を横切って向かいの店へと
    移る（これはあり得る），そういう車が何十台も続く
  - ・・・

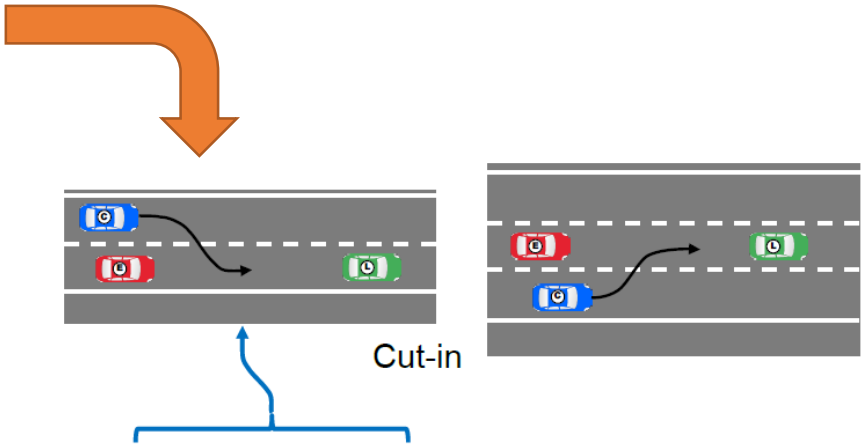
シナリオでの区切り（Pegasus Projectより）

Cut-in

- Start situation
  - Ego car (**E**) drives on highway lane
  - Other vehicle (**C**) on adjacent lane
  - Potentially further vehicles involved
- Evolution
  - **C** moves into E-lane in front of **E**
- Criticalities
  - **C** cuts in with little distance to **E**
  - **C** brakes after cutting in
  - Low TTC(**E**,**C**)



TTC: Time to collision



Cut-in (left, from behind) (regular traffic situation)

- Step 1:
  - Velocity [m/sec]: E , L: [22-36]; E-L: [-4,4]; C: [23-67]; C-E: [1,45];
  - Position [m]: L-E: [33,100]; E-C: [0,30];
  - Distributions: may be multivariate binomial (nontrivial correlations), or multivariate gamma-distributions
  - …
- Step 2: Cut-in starts (C crosses lane marking) Δt: [2,20]
  - Velocity [Δ m/sec]: L: [-7,+7]; C: [-50,+5]; C-E: [-5,40]; C-L:[-12,50]
  - Position [m]: L-E: [25,110]; C-E: [1,60]; L-E: [5,100]
  - …
- Step 3: Cut-in completed (C has crossed lane marking halfway) Δt: [0.5,4]
  - Velocity [Δ m/sec]: …
  - …

[Hunger et. al., Test Specifications for Highly Automated Driving Functions: Highway Pilot, 2017
https://www.pegasusprojekt.de/en/information-material?file=files/tmpl/pdf/AVT%20Symposium%202017%20Test%20Specifications%20for%20HAD_Folien%20.pdf]

テスト仕様・設計のマイニングを含めた探索

- 概念的にシナリオを絞っても，実装上のパラメータ（範囲）の決定問題が残る

➡ **「意味のある」テスト仕様・設計も探索**

  - 「試す意義がない」パラメータ（の範囲）
  - 「シナリオの意図」に合うパラメータ（の範囲）
  - 「難しい，しかし衝突を避けることが可能」である状況を表すパラメータ
  - 更新前後・設計の選択肢の差異を浮かび上がらせるパラメータ
  - ・・・

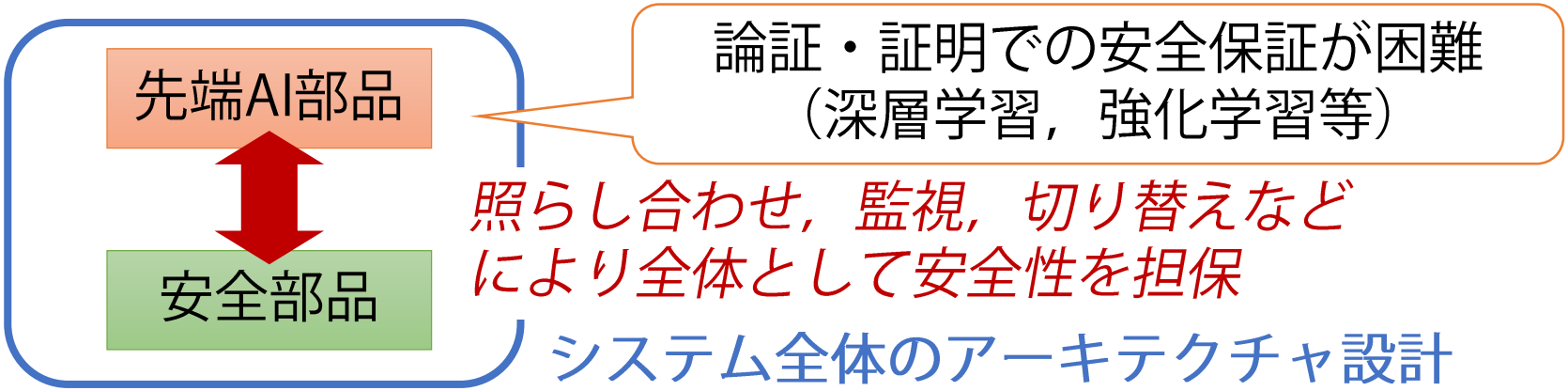
現在進行中のメインテーマ！

おわりに

本当は「テスト」
だけの話ではない！

演繹と帰納の融合

- システム全体をブラックボックステスト，以上？
  - 問題が大きすぎる（例：不具合の原因追及が困難）
- 自動運転や，機械学習工学一般での典型課題



➡ ***形式手法（演繹的保証）による問題分割の力・強い保証の力をテスト（帰納的保証）と組み合わせ***

- 手持ち技術：「証明リファクタリング・再利用」

[T. Kobayashi et al., Consistency-Preserving Refactoring of Refinement Structures in Event-B Models, 2019]

機械学習ベースAIシステムのための深化

- 自動運転・物理系に限らず最近の重要課題
  - 例： 機械学習工学研究会キックオフシンポ（18/5/17）
  - 例： Open QA4AI Conference（19/5/17）

- 発見的・経験的テストの応用は世界的にも注目
  - 例： 誤認識が起きる方向にノイズの足し方を寄せていく
- 本プロジェクトでも，理論を踏まえて
  将来を見据えた取り組みが進行中
  - 例： 「全体平均」に過ぎない精度に変わる評価指標や，
    その指標に応じたテストやモデル構築・訓練

グループ3の全体像（ボトムアップ）

お話しできなかった
トピックも

演繹的な分析・論証

形式仕様，段階的詳細化，定理証明，保証ケース

証明リファクタリングに
よる証明の再利用部品化

連携・融合技術

制御工学アプローチ
による保証を扱う形式手法

演繹と帰納の連携・融合による
AI部品入りシステムの安全論証

実世界・AIの不確かさ

不確かなシステムに対する
継続的な保証ケース進化

組込み連携での
高速な実行時問題検出

MCTSによる
効率的反例探索

安定性分析

設計比較・修正

AI技術の活用

強化学習によるモデルファミリ
向け反例探索器生成

「注目すべき」
シナリオの発見

知見を掘り出す広義のテスト

帰納的な分析・テスト

サーチベースド工学，反例探索，統計的分析・検査

まとめ

- 発見的・経験的なアプローチ

  「実行可能なもの」があれば適用できる，

  （一つの）重要で現実的なアプローチ

  - 実世界の難しさとの戦いも主軸に

    （不確実性・オープン性）

  - 理論・演繹と経験・帰納の協働へ

  - 皆様からの問題投げ込みを受けて活発化・技術深化

  ➡ トップダウンの取り組みとツール化へ