スライド
bit.ly/4u8Kvla

NII
MathTrust

自動運転の安全性説明へ
安全性評価のための数学・論理学的技術

SAKURAプロジェクト 最終成果報告会
2026/03/04

蓮尾 一郎

国立情報学研究所（NII） アーキテクチャ科学研究系 教授
同 数理的高信頼ソフトウェアシステム研究センター長

株式会社 イミロン CSO

自己紹介

NII MathTrust

蓮尾 一郎

- 数学 ＞ 数学基礎論・応用数学 ＞ 数理論理学
- 数学 ＞ 代数学 ＞ 圏論
- 情報学 ＞ ソフトウェア ＞ ソフトウェア科学・形式検証・数理論理学

- Radboud U. Nijmegen 計算機科学 （PhD）
  ＞ 京大数理研 （助教）
  ＞ 東大 コンピュータ科学専攻 （講師・准教授）
  ＞ 国立情報学研究所 （准教授・教授）

- 数学と情報学（ソフトウェア科学）の間で,
  数学（特に証明）の社会的使命を探求

共著書： 圏論の歩き方
（日本評論社，2015 & 2025）

自己紹介

NII
MathTrust

ERATO
蓮尾メタ数理システム
デザインプロジェクト
（2016-2025）

個人向けとしては
国内最大級の研究グラント
（研究員を~20名雇用）

➜ 成果の社会還元のため
自動車応用を追求

NII MathTrust

「安全性論証の形式化」＋「安全性保証の内容の形式化」

定義 完備距離空間とは任意のコーシー列が極限を持つ距離空間のことを言う．

定理 完備距離空間 X 上の収縮写像 $f: X \to X$ は不動点を持つ．さらに，この不動点は一意に定まる．

証明． $x \in X$ を任意に選び，点列 $x, f(x), f(f(x)), \ldots$ を考えると，f が収縮写像であることよりこれはコーシー列．よって完備性の定義より極限 x_0 を持つ．$x_0 = f(x_0)$ を示すには…．□

定義 自動運転車  とは…

定理 自動運転車  は安全である．

証明． …

数学では：

- **定理の言明 statement** を厳密に述べた上で，
- **厳密な証明**を与える

システム安全性保証への応用：

- **安全性保証の内容**を厳密に述べた上で，
- **厳密な安全性証明**を与える

NII MathTrust

「安全性論証の形式化」＋「安全性保証の内容の形式化」

定義 完備距離空間とは任意のコーシー列が極限を持つ距離空間のことを言う．

定理 完備距離空間 X 上の収縮写像 $f: X \to X$ は不動点を持つ．さらに，この不動点は一意に定まる．

証明. $x \in X$ を任意に選び，点列 $x, f(x), f(f(x)), \ldots$ を考えると，f が収縮写像であることよりこれはコーシー列．よって完備性の定義より極限 x_0 を持つ．$x_0 = f(x_0)$ を示すには…．□

定義 自動運転車  とは…

定理 自動運転車  は安全である．

証明. …

数学といえば証明

数学では：

- **定理の言明 statement** を厳密に述べた上で，
- **厳密な証明**を与える

システム安全性保証への応用：

- **安全性保証の内容**を厳密に述べた上で，
- **厳密な安全性証明**を与える

NII MathTrust

「安全性論証の形式化」＋「安全性保証の内容の形式化」

定義 完備距離空間とは任意のコーシー列が極限を持つ距離空間のことを言う．

定理 完備距離空間 X 上の収縮写像 $f: X \to X$ は不動点を持つ．さらに，この不動点は一意に定まる．

証明. $x \in X$ を任意に選び，点列 $x, f(x), f(f(x)), \ldots$ を考えると，f が収縮写像であることよりこれはコーシー列．よって完備性の定義より極限 x_0 を持つ．$x_0 = f(x_0)$ を示すには…．□

定義 自動運転車  とは…

定理 自動運転車  は安全である．

証明. …

確かに役に立ちそう！

自動運転の安全性証明ができればどれだけ楽か…

数学といえば証明

数学では：

- **定理の言明 statement** を厳密に述べた上で，
- **厳密な証明**を与える

システム安全性保証への応用：

- **安全性保証の内容**を厳密に述べた上で，
- **厳密な安全性証明**を与える

NII MathTrust

「安全性論証の形式化」＋「安全性保証の内容の形式化」

定義 完備距離空間とは任意のコーシー列が極限を持つ距離空間のことを言う.

定理 完備距離空間 X 上の収縮写像 $f: X \to X$ は不動点を持つ．さらに，この不動点は一意に定まる.

証明. $x \in X$ を任意に選び，点列 $x, f(x), f\big(f(x)\big), \ldots$ を考えると，f が収縮写像であ
て完備性
$f(x_0)$ を

そういえばこちらも必要

曖昧な言明には証明をつけられない（数学的な真偽が定まらない）

定義 自動運転車  とは…

定理 自動運転車  は安全である.

証明. …

数学では：

- **定理の言明 statement** を厳密に述べた上で，
- **厳密な証明**を与える

システム安全性保証への応用：

- **安全性保証の内容**を厳密に述べた上で，
- **厳密な安全性証明**を与える

NII MathTrust

「安全性論証の形式化」＋「安全性保証の内容の形式化」

定義 完備距離空間とは任意のコーシー列が極限を持つ距離空間のことを言う．

定理 完備距離空間 X 上の収縮写像 $f: X \to X$ は不動点を持つ．さらに，この不動点は一意に定まる．

証明. $x \in X$ を任意に選び，点列 $x, f(x), f(f(x)), \ldots$ を考えると，f が収縮写像であ
て完備性
$f(x_0)$ を

そういえばこちらも必要

曖昧な言明には証明をつけられない（数学的な真偽が定まらない）

定義 自動運転車  とは…

定理 自動運転車  は安全である．

証明. …

ん？ これもかなりうれしいぞ…

「『安全である』とは何か」がそもそも大問題

数学では：

- **定理の言明 statement** を厳密に述べた上で，
- **厳密な証明**を与える

システム安全性保証への応用

- **安全性保証の内容**を厳密に述べた上で，
- **厳密な安全性証明**を与える

NII MathTrust

「安全性論証の形式化」＋「安全性保証の内容の形式化」

定義 **完備距離空間**とは任意のコーシー列が極限を持つ距離空間のことを言う．

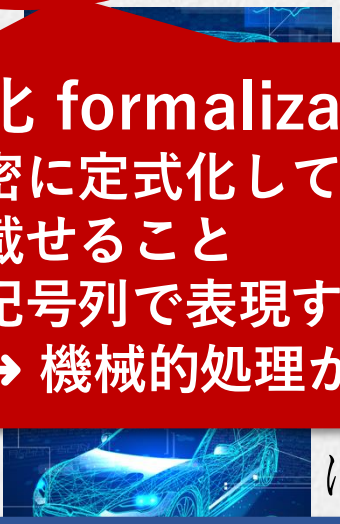
定理 完備距離空間 X 上の収縮写像 $f: X \to X$ は不動点を持つ．さらに，この不動点は一意に定まる．

証明. $x \in X$ を任意に選び，点列 $x, f(x), f\big(f(x)\big), \ldots$ を考えると，f が収縮写像であ
て完備性
$f(x_0)$ を

そういえばこちらも必要

曖昧な言明には証明をつけられない（数学的な真偽が定まらない）

定義 自動



形式化 formalization：
- **厳密に定式化して数学的議論の俎上に載せること**
- **（記号列で表現すること**
  ➜ 機械的処理が可能に）

定理 自動運転車 は安全である．

証明. …

ん？ これもかなりうれしいぞ…

「『安全である』とは何か」がそもそも大問題

数学では：
- **定理の言明 statement** を厳密に述べた上で，
- **厳密な証明**を与える

システム安全性保証への応用
- **安全性保証の内容**を厳密に述べた上で，
- **厳密な安全性証明**を与える

2つの技術をご紹介します

技術１：　各車責任の形式化に基づく安全性証明技術

RSS ルールの例

一車線同方向運転シナリオにおける追突回避のためのルール

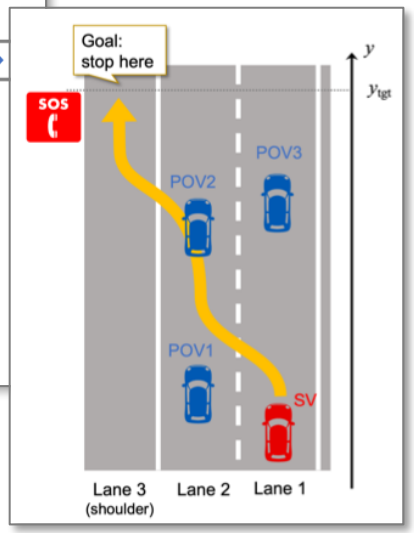
[Shalev-Shwartz et al., arXiv preprint, 2017]

NII MathTrust

- 自車（後方）に課すRSSルール：
  以下のペア (A, α)
  - RSS 条件 A：　（「A が成り立つうちは，まだ逃げられます」）
    車間距離を

$$d_{\min} = \left[v_r \rho + \frac{1}{2} a_{\max,\text{accel}} \rho^2 + \frac{(v_r + \rho a_{\max,\text{accel}})^2}{2a_{\min,\text{brake}}} - \frac{v_f^2}{2a_{\max,\text{brake}}} \right]_+$$

    以上確保すること
  - 適切反応 (proper response) α：　（「逃げ方はこれ」）
    反応時間 ρ 以内に 減速度 $a_{\min,\text{brake}}$ でブレーキすること
- 他車（前方）に課す RSS ルール：　最大限速度は $a_{\max,\text{brake}}$
- すると条件付き安全性補題が証明できる：
  RSS 条件 A が真である状態から適切反応 α を実行すれば，衝突は発生しない



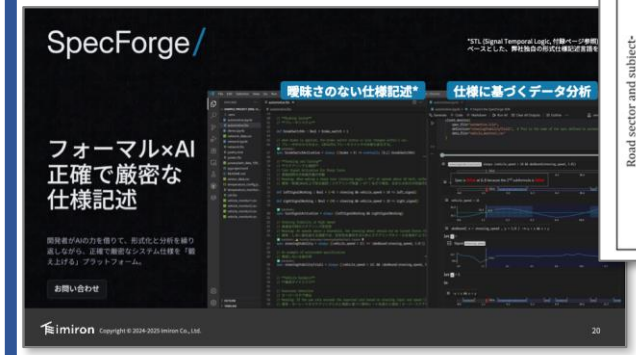
- 各車の責任内容を形式化し，
- 各車が責任を果たすという仮定のもとで，
  安全性を数学的に証明 ➔ **絶対の安全性保証**

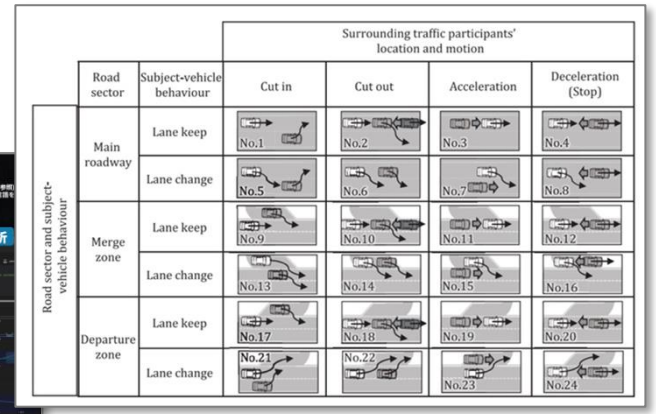
用途：

- E2E自動運転の走行時セーフガード
  （数学的安全性証明付き！）
- 安全性評価の定量的指標

技術２：　安全テストにおける高レベル機能シナリオの形式化技術

$$\mathbf{G}(p \rightarrow \mathbf{F}_{[0,T]} q)$$





- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  ➔ **高レベルシナリオの管理・活用の自動化**

用途：

- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成

2つの技術の比較

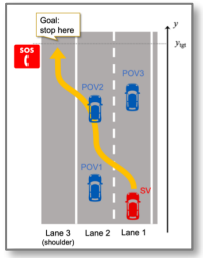
相補的な強みで，自動運転安全性の包括的保証＆説明

技術１：　各車責任の形式化に基づく安全性証明技術

- 各車の責任内容を形式化し，
- 各車が責任を果たすという仮定のもとで，
  安全性を数学的に証明 ➔ **絶対の安全性保証**

用途：

- E2E自動運転の走行時セーフガード
  （数学的安全性証明付き！）
- 安全性評価の定量的指標

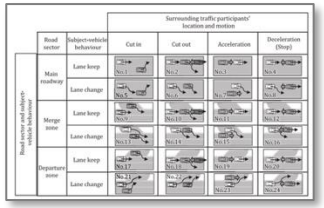


技術２：　安全テストにおける高レベル機能シナリオの形式化技術

- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  ➔ **高レベルシナリオの管理・活用の自動化**

用途：

- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成



数学的証明 「これがこうなるから安全．証明終」	安全性保証の カタチ	**経験論的保証** 「これだけのシナリオでテストして安全だったので大丈夫」 （というチェックの責任を果たしましたよ）
強い 論理的な絶対の保証	安全性保証の 強度・説明可能性	**比較的弱い** 経験論的な保証 ➔ 保証・説明の体系化が必要（本技術）
狭い 高速道，幹線道路の交差点（構造的）	主ターゲットたる 運転シーン	**広い** 一般道を含むあらゆる運転シーンに適用

アウトライン

- 自己紹介：
  数学・論理学・ソフトウェア科学における「形式化」とは？

- 技術１： 各車責任の形式化に基づく安全性証明技術
  - 追突防止自動ブレーキの例
  - 形式化による対象運転シナリオの拡大
  - 用途１： E2E自動運転の走行時セーフガード
  - 用途２： 安全性評価の定量的指標

- 技術２： 安全テストにおける高レベル機能シナリオの形式化技術

例： 追突防止自動ブレーキ

直感的で自然な考え方：

- 前車がブレーキをかけたら，自車もブレーキをかければよい
- 理想的状況では（反応遅れゼロ・同じ速度・同じ減速度），
  車間距離 0.01m でも追突防止できる
- しかし現実は理想的ではない → 適切な車間距離を維持

より精密な安全性論証へ：

- **予防条件**：
  適切な車間距離を常に維持する
- **安全動作**：
  予防条件の違反が予見される場合にはブレーキ

安全性論証：
安全動作実行によって，予防条件を常に真に保つことができる．
その帰結として特に，追突は起こらない

「適切な車間距離」は，次のパラメータ（仮定）に依存

- 最大反応遅れ時間 ρ
- 前車の速度 v_{front} ， 前車の最大減速度 b_{front}
- 後車の速度 v_{rear}， 後車の最大減速度 b_{rear}
- 後車の最大加速度 a_{rear}（反応遅れ時間中に加速）

例： 追突防止自動ブレーキ

直感的で自然な考え方：

- 前車がブレーキをかけたら，自車もブレーキをかければよい
- 理想的状況では（反応遅れゼロ・同じ速度・同じ減速度），
  車間距離 0.01m でも追突防止できる
- しかし現実は理想的ではない ➔ 適切な車間距離を維持

より精密な安全性論証へ：

- **予防条件**：
  適切な車間距離を常に維持する
- **安全動作**：
  予防条件の違反が予見される場合にはブレーキ

安全性論証：
安全動作実行によって，予防条件を常に真に保つことができる．
その帰結として特に，追突は起こらない

- **「適切な車間距離」はどれくらい？**
- **安全性論証を証明できるか？**

➔ 数学的形式化！

「適切な車間距離」は，次のパラメータ（仮定）に依存

- 最大反応遅れ時間 ρ
- 前車の速度 v_{front} ， 前車の最大減速度 b_{front}
- 後車の速度 v_{rear}， 後車の最大減速度 b_{rear}
- 後車の最大加速度 a_{rear}（反応遅れ時間中に加速）

例： 追突防止自動ブレーキ

- **予防条件**：
  適切な車間距離を常に維持する
- **安全動作**：
  予防条件の違反が予見される場合にはブレーキ

安全性論証：
安全動作実行によって，予防条件を常に真に保つことができる．
その帰結として特に，追突は起こらない

数学的「定式化」 ➔ RSS
[Shalev-Shwartz et al., arXiv preprint, 2017]

- **予防条件（“RSS Condition”）**：
  次の値以上の車間距離を常に維持する

$$d_{\min} = \left[v_r\,\rho + \frac{1}{2} a_{\max,\mathrm{accel}}\,\rho^2 + \frac{(v_r + \rho\, a_{\max,\mathrm{accel}})^2}{2a_{\min,\mathrm{brake}}} - \frac{v_f^2}{2a_{\max,\mathrm{brake}}} \right]_+$$

- **安全動作（”proper response”）**：
  反応時間 ρ 以内に，減速度 $a_{\min,\mathrm{brake}}$ でブレーキ

条件付き安全性定理：
予防条件が成立する状況において安全動作を実行すれば，
追突は起こらない

証明： むずかしくない．
高校物理の等加速度直線運動
＋ （ちょっと面倒な）場合分け）

例： 追突防止自動ブレーキ

数学的「定式化」
[Shalev-Shwartz et al., arXiv preprint, 2017]

- **予防条件（“RSS Condition”）**：
  次の値以上の車間距離を常に維持する

$$d_{\min} = \left[v_r\,\rho + \frac{1}{2} a_{\max,\mathrm{accel}}\,\rho^2 + \frac{(v_r + \rho\, a_{\max,\mathrm{accel}})^2}{2a_{\min,\mathrm{brake}}} - \frac{v_f^2}{2a_{\max,\mathrm{brake}}} \right]_+$$

- **安全動作（”proper response”）**：
  反応時間 ρ 以内に，減速度 $a_{\min,\mathrm{brake}}$ でブレーキ

条件付き安全性定理：
予防条件が成立する状況において
安全動作を実行すれば，
追突は起こらない

証明： むずかしくない．
高校物理の等加速度直線運動
＋ （ちょっと面倒な）場合分け）

考察：

- 安全性の厳密な証明ができた！
- 各車の内部動作原理に踏み込まない
  「センサー統合はどうなってる？」「ルールベースでなくE2Eだけど？」とか言わない
- 代わりに，**外から見た振る舞いルール**（＝「責任」）に立脚
  - 自車： 必要により安全動作を行いながら，予防条件を保持
  - 他車： 最大減速度など
- 事故は起こる．が，条件付き安全性定理により，誰かが責任を守らなかったことがわかる

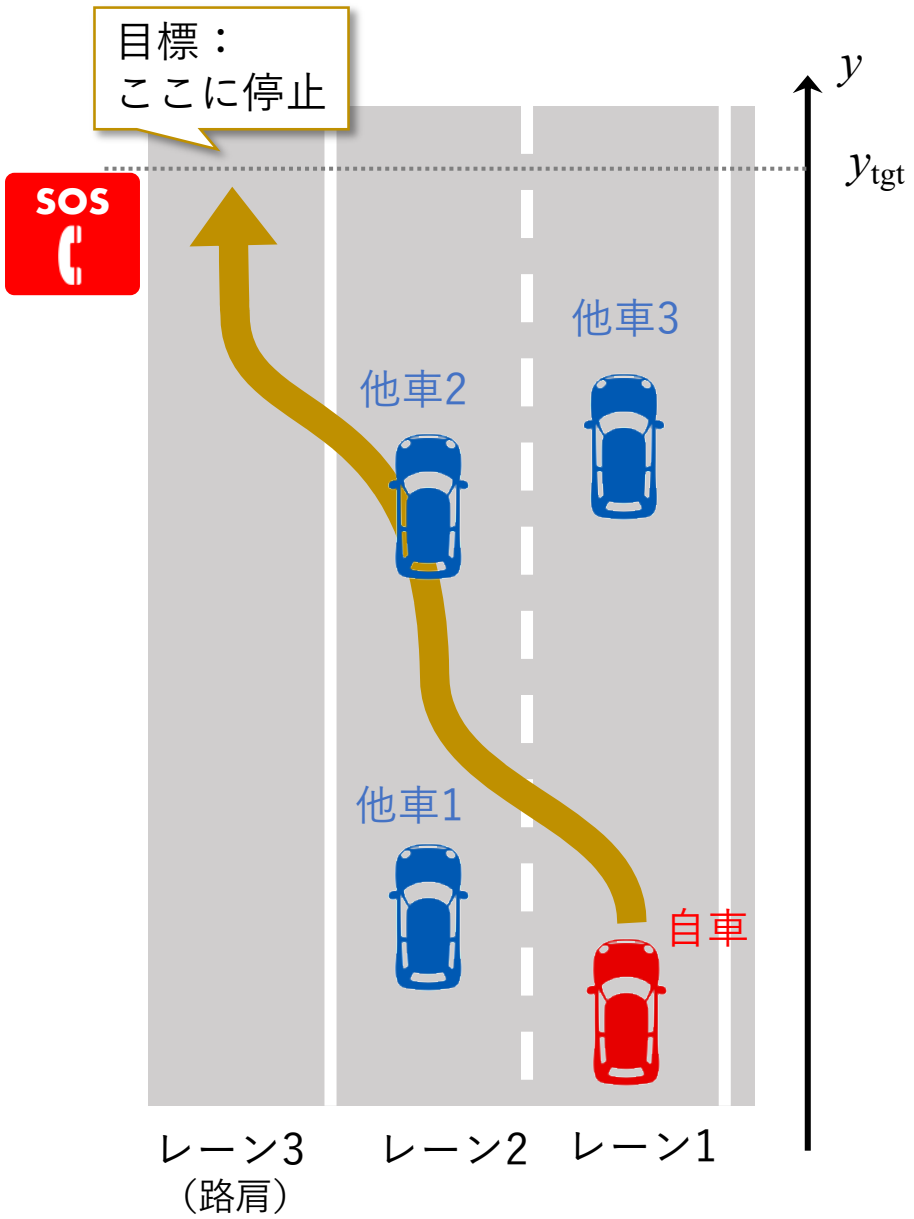
そもそも，現在の
公道システムは
「事故ゼロ」でなく
「責任追求」が基本

アウトライン

- 自己紹介：
  数学・論理学・ソフトウェア科学における「形式化」とは？

- 技術１： 各車責任の形式化に基づく安全性証明技術
  - 追突防止自動ブレーキの例
  - 形式化による対象運転シナリオの拡大
  - 用途１： E2E自動運転の走行時セーフガード
  - 用途２： 安全性評価の定量的指標

- 技術２： 安全テストにおける高レベル機能シナリオの形式化技術



RSS から GA-RSS へ

[Shalev-Shwartz et al., arXiv preprint, 2017]

[Hasuo et al., IEEE T-IV, 2023]

- このような運転シナリオ（路肩停止）ではどうか？
- 他車１の前で合流？ 後で？
  追い越すために加速したら
  停止位置をオーバーランするかも？
  …
  → 複雑さが段違い

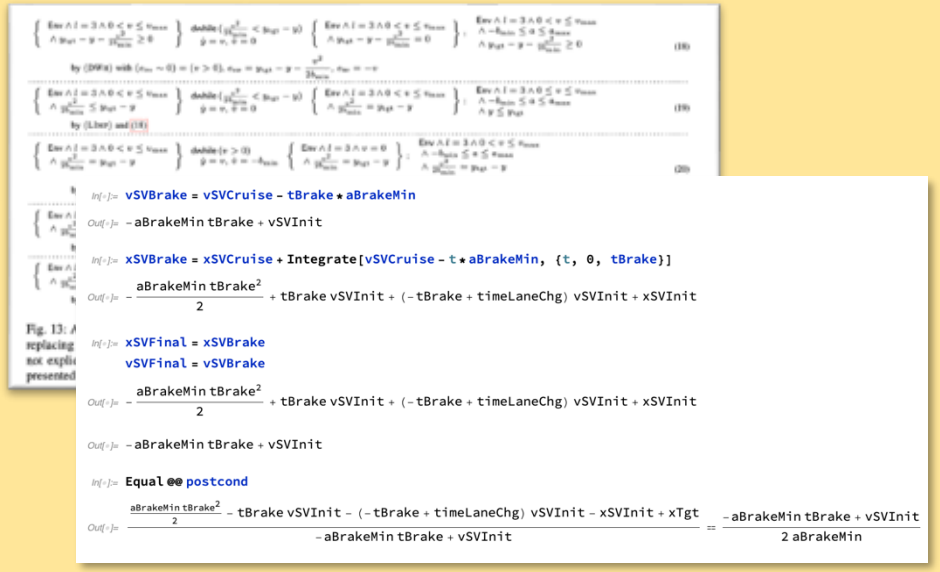
- （マツダさんからの宿題でした）

数学的「定式化」 vs. 数学的「形式化」

非形式的証明
（紙とペンで書く）

- 大量（数千ページとか），間違えやすい
- 後から or 他人による チェック・解析が困難

形式的証明
（コンピュータ上で書く）

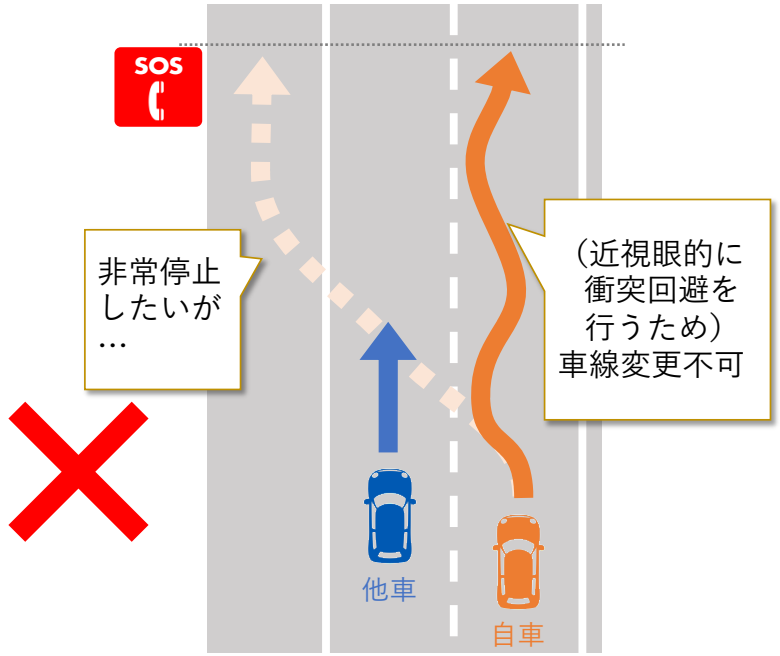


```
In[•]:= vSVBrake = vSVCruise - tBrake * aBrakeMin
Out[•]= -aBrakeMin tBrake + vSVInit

In[•]:= xSVBrake = xSVCruise + Integrate[vSVCruise - t * aBrakeMin, {t, 0, tBrake}]
Out[•]= -(aBrakeMin tBrake^2)/2 + tBrake vSVInit + (-tBrake + timeLaneChg) vSVInit + xSVInit

In[•]:= xSVFinal = xSVBrake
        vSVFinal = vSVBrake
Out[•]= -(aBrakeMin tBrake^2)/2 + tBrake vSVInit + (-tBrake + timeLaneChg) vSVInit + xSVInit

Out[•]= -aBrakeMin tBrake + vSVInit

In[•]:= Equal @@ postcond
Out[•]= (aBrakeMin tBrake^2/2 - tBrake vSVInit - (-tBrake + timeLaneChg) vSVInit - xSVInit + xTgt)/(-aBrakeMin tBrake + vSVInit) == (-aBrakeMin tBrake + vSVInit)/(2 aBrakeMin)
```

- あらかじめ決まった証明言語と 書き換え規則で 証明の各ステップを記述（「記号推論」）
- 証明の正しさのチェックが ソフトウェア（proof checker）により可能

我々の成果 [Hasuo et al., IEEE T-IV, 2023]：
RSS 証明の**形式化**による RSS ルールの本格展開

RSS
Responsibility-Sensitive Safety
（責任感知型安全論）
[Shalev-Shwartz et al., arXiv, 2017]
- 安全ルールの基本的方法論
- 国際規格化の動き
  (IEEE 2846)
- 複雑なシナリオに対するルール策定・証明手法は未整備
- **特に，衝突回避以外の目的への対応事例がない**

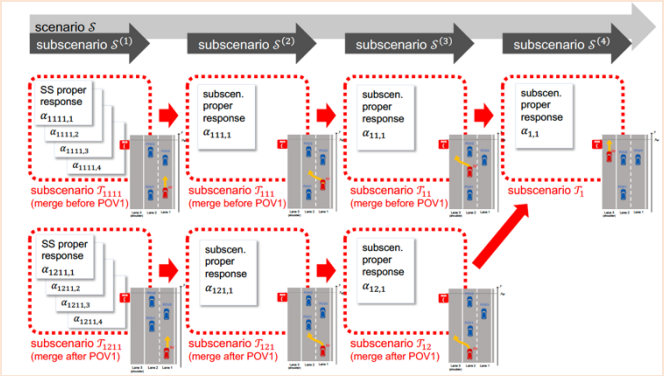


微分プログラム論理 dFHL（今回の成果）

$$\frac{\begin{array}{ll} \text{inv}: & A \Rightarrow e_{\text{inv}} \sim 0 \quad e_{\text{var}} \geq 0 \wedge e_{\text{inv}} \sim 0 \Rightarrow \mathcal{L}_{\dot{\mathbf{x}}=\mathbf{f}}\, e_{\text{inv}} \simeq 0 \\ \text{var}: & A \Rightarrow e_{\text{var}} \geq 0 \quad e_{\text{var}} \geq 0 \wedge e_{\text{inv}} \sim 0 \Rightarrow \mathcal{L}_{\dot{\mathbf{x}}=\mathbf{f}}\, e_{\text{var}} \leq e_{\text{ter}} \\ \text{ter}: & A \Rightarrow e_{\text{ter}} < 0 \quad e_{\text{var}} \geq 0 \wedge e_{\text{inv}} \sim 0 \Rightarrow \mathcal{L}_{\dot{\mathbf{x}}=\mathbf{f}}\, e_{\text{ter}} \leq 0 \end{array}}{\{A\}\ \text{dwhile}\,(e_{\text{var}} > 0)\,\dot{\mathbf{x}} = \mathbf{f}\ \{e_{\text{var}} = 0 \wedge e_{\text{inv}} \sim 0\} : e_{\text{inv}} \sim 0 \wedge e_{\text{var}} \geq 0} \text{(DW)}$$

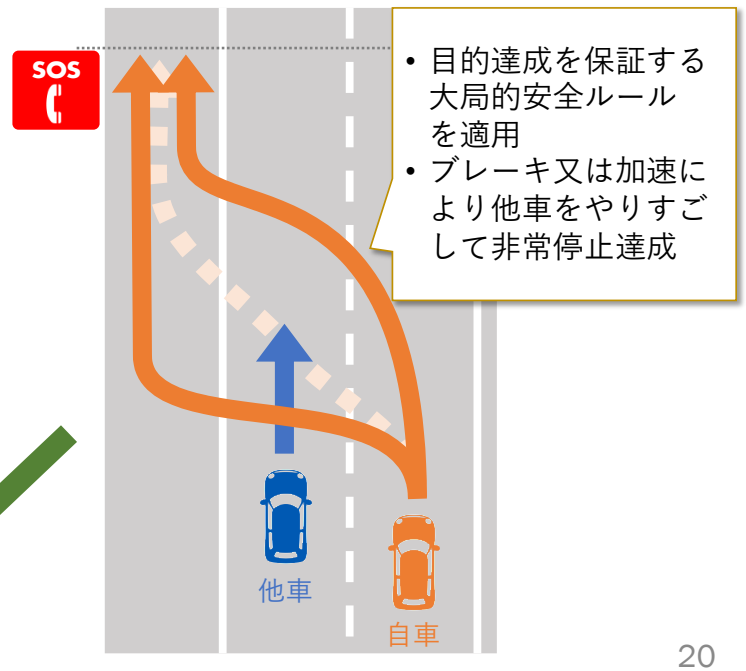
- 安全ルール導出・証明のための論理体系

dFHL による逐次的推論・ルール導出ワークフロー（今回の成果）



- 複雑な行動計画を分割，それぞれ論理的解析し，結果を結合
- 自動推論によるツールサポート

GA-RSS（我々の成果）
Goal-Aware
Responsibility-Sensitive Safety
[Hasuo+, IEEE T-IV, 2023]
- 衝突回避に加え，緊急停止等の**目的達成**もサポート
- 複数の行動を組み合わせた大局的安全ルール
- 現実の複雑な交通シナリオへの適用において必須



追突防止自動ブレーキで言い換えると…

NII MathTrust

非数学的「論証」

- **予防条件**：
  適切な車間距離を常に維持する
- **安全動作**：
  予防条件の違反が予見される場合にはブレーキ

安全性論証：
安全動作実行によって，予防条件を常に真に保つことができる．
その帰結として特に，追突は起こらない

数学的証明（非形式的）

RSS [Shalev-Shwartz et al., arXiv preprint, 2017]

- **予防条件（“RSS Condition”）**：
  次の値以上の車間距離を常に維持する

$$d_{\min} = \left[v_r\,\rho + \frac{1}{2} a_{\max,\text{accel}}\,\rho^2 + \frac{(v_r + \rho\, a_{\max,\text{accel}})^2}{2a_{\min,\text{brake}}} - \frac{v_f^2}{2a_{\max,\text{brake}}} \right]_+$$

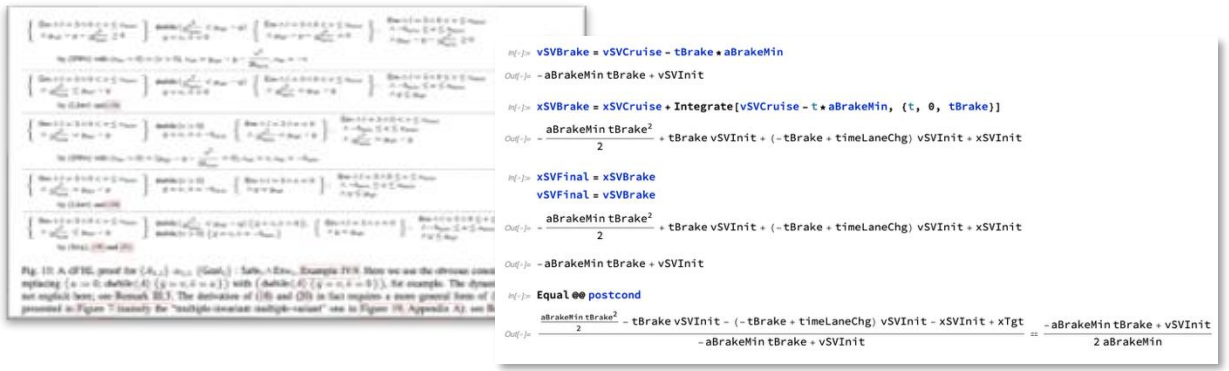
- **安全動作（”proper response”）**：
  反応時間 ρ 以内に，減速度 $a_{\min,\text{brake}}$ でブレーキ

条件付き安全性定理：
予防条件が成立する状況において安全動作を実行すれば，
追突は起こらない

証明：むずかしくない．
高校物理の等加速度直線運動
＋（ちょっと面倒な）場合分け）

数学的証明（形式的）

GA-RSS [Hasuo+, IEEE T-IV, 2023]



```
In[•]:= vSVBrake = vSVCruise - tBrake * aBrakeMin
Out[•]= -aBrakeMin tBrake + vSVInit

In[•]:= xSVBrake = xSVCruise + Integrate[vSVCruise - t * aBrakeMin, {t, 0, tBrake}]
Out[•]= -aBrakeMin tBrake^2/2 + tBrake vSVInit + (-tBrake + timeLaneChg) vSVInit + xSVInit

In[•]:= xSVFinal = xSVBrake
        vSVFinal = vSVBrake
Out[•]= -aBrakeMin tBrake^2/2 + tBrake vSVInit + (-tBrake + timeLaneChg) vSVInit + xSVInit

Out[•]= -aBrakeMin tBrake + vSVInit

In[•]:= Equal @@ postcond
Out[•]= (aBrakeMin tBrake^2/2 - tBrake vSVInit - (-tBrake + timeLaneChg) vSVInit - xSVInit + xTgt)/(-aBrakeMin tBrake + vSVInit) == (-aBrakeMin tBrake + vSVInit)/(2 aBrakeMin)
```

形式化のご利益：

- 計算機による証明チェック・メンテナンス
  「場合分けのヌケモレはない？」等
- 計算機実装による走行時運用
- 複雑な運転シナリオへの応用

アウトライン

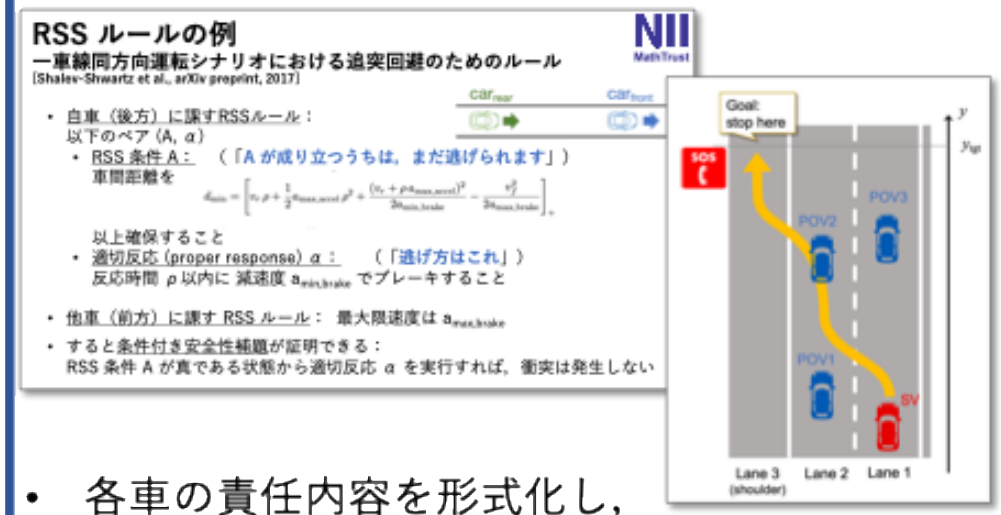
- 自己紹介：
  数学・論理学・ソフトウェア科学における「形式化」とは？

- 技術１： 各車責任の形式化に基づく安全性証明技術
  - 追突防止自動ブレーキの例
  - 形式化による対象運転シナリオの拡大
  - 用途１： E2E自動運転の走行時セーフガード
  - 用途２： 安全性評価の定量的指標

- 技術２： 安全テストにおける高レベル機能シナリオの形式化技術

説明可能・証明可能な安全性保証
走行時・設計時の双方で活用

技術１： 各車責任の形式化に基づく安全性証明技術



- 各車の責任内容を形式化し，
- 各車が責任を果たすという仮定のもとで，
  安全性を数学的に証明 → **絶対の安全性保証**

用途：
- E2E自動運転の走行時セーフガード
  （数学的安全性証明付き！）
- 安全性評価の定量的指標

用途１：
E2E 自動運転の走行時セーフガード

「まだ逃げられるための条件」

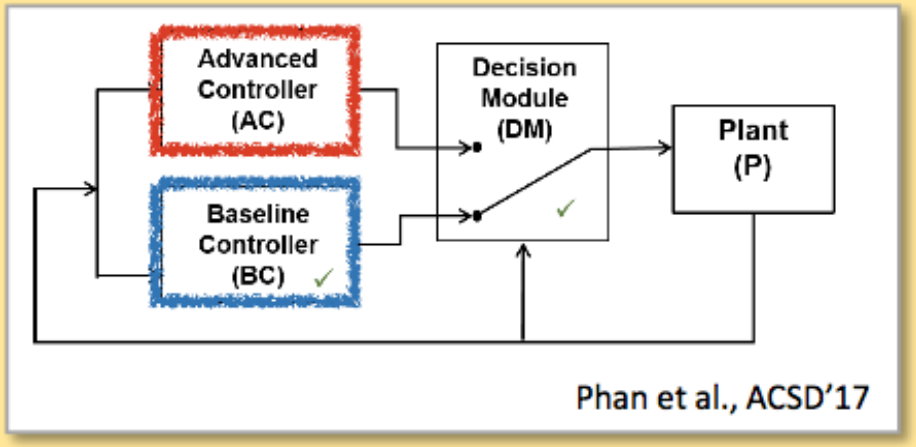
- **予防条件**：
  適切な車間距離を常に維持する
- **安全動作**：
  予防条件の違反が予見される場合
  にはブレーキ

「やばくなったらこうやって逃げろ」

→ **走行時セーフガードとして実装** （cf. 自動ブレーキ）

- 予防条件を監視
- 余裕がなくなったら介入して，安全動作を実行
  （セーフガード対象は，E2E自動運転でも，人間でも）

数学的安全性証明を走行時実装
「追突防止自動ブレーキ」をあらゆる運転シナリオへ



安全性確保のための simplex architecture

- **AC（通常系）**が性能を追求
  （E2E自動運転，人間，…）
- **BC（安全系）**が安全動作を実行
- **DC（切換器）**が予防条件を監視
  - まだ余裕あり➔
    「AC，やってみなはれ」
  - 余裕なし ➔ BC に切替

用途１：
E2E 自動運転の走行時セーフガード

「まだ逃げられるための条件」

- **予防条件**：
  適切な車間距離を常に維持する
- **安全動作**：
  予防条件の違反が予見される場合にはブレーキ

「やばくなったらこうやって逃げろ」

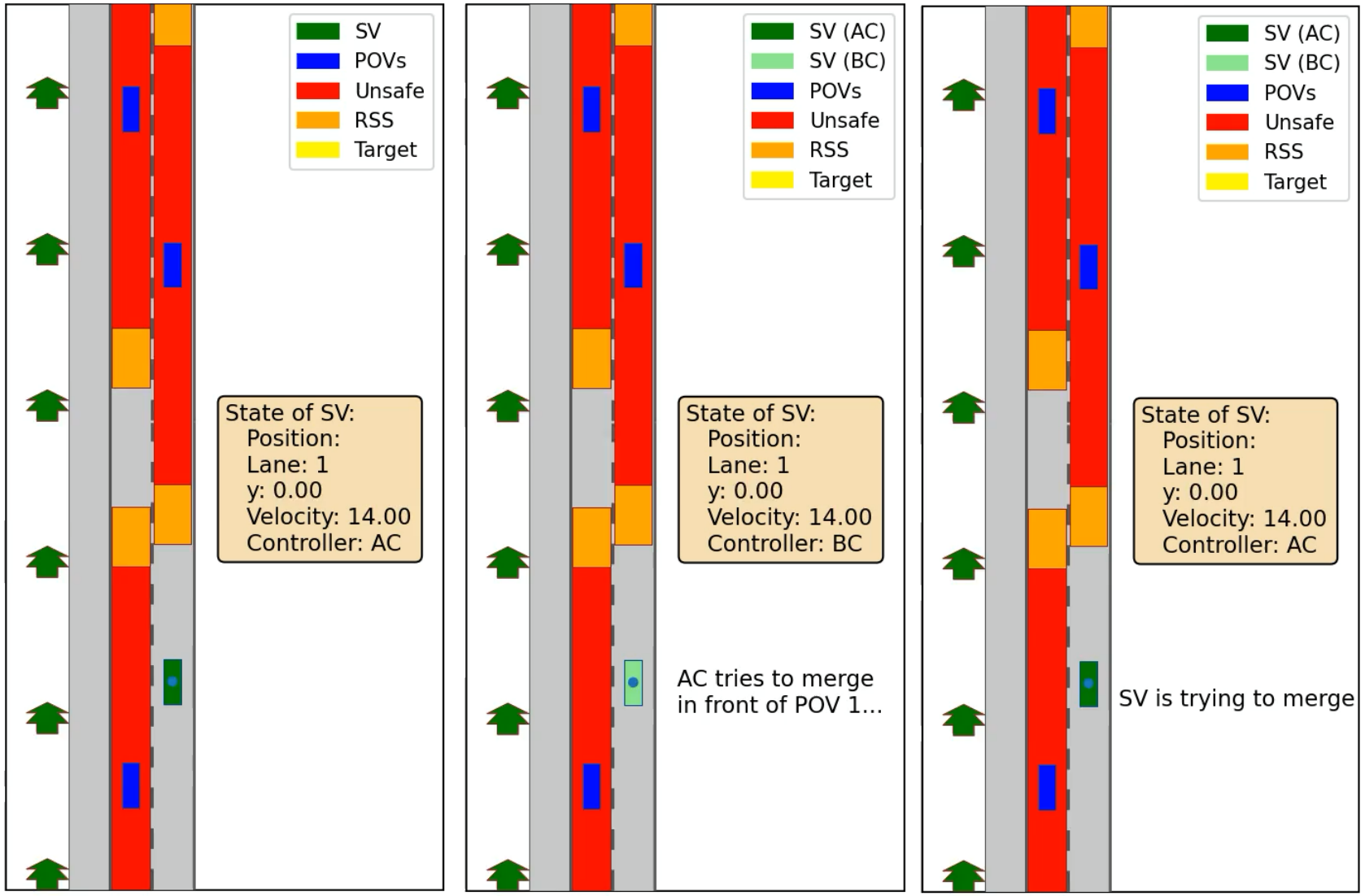
➔ 走行時セーフガードとして実装（cf. 自動ブレーキ）

- 予防条件を監視
- 余裕がなくなったら介入して，安全動作を実行
  （セーフガード対象は，E2E自動運転でも，人間でも）

安全エンベロープ実行例１

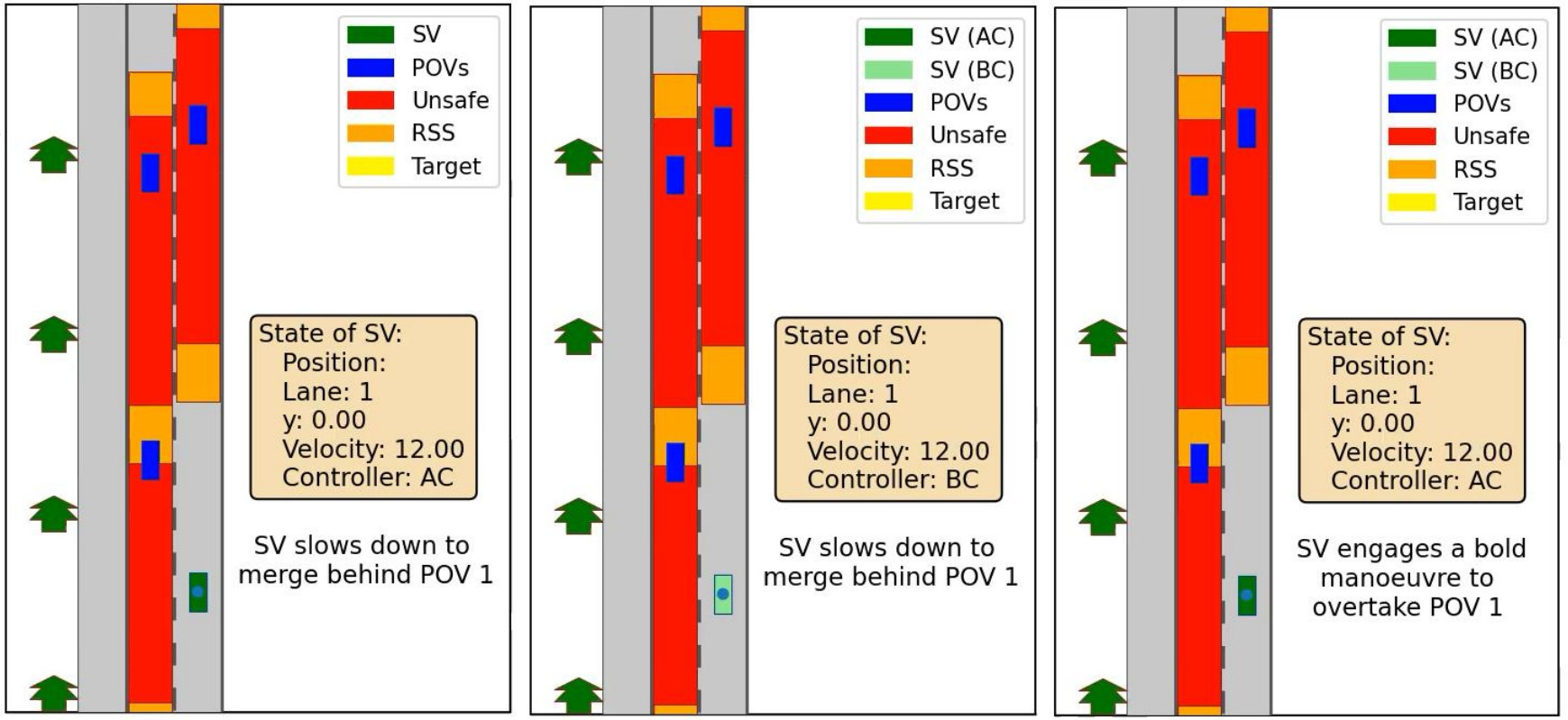
- AC: 安全系なし
- AC+RSS：
  既存研究 [Shalev-Shwartz et al., arXiv, 2017] の RSS ルールを安全エンベロープとして実装
  （近視眼的に衝突回避）
- AC+RSS^{GA}：
  我々 [Hasuo+, IEEE T-IV] の RSS ルールを安全エンベロープとして実装
  （長期的視野で目標達成も保証）

- AC は安全でない（危険な割り込み）
- AC+RSS は路肩に到達できず
- AC+RSS^{GA} は長期的視野で減速 → 他車の後ろで合流，安全性と目標達成（路肩停止）両方を実現



安全エンベロープ実行例２

- AC: 安全系なし
- AC+RSS：
  既存研究 [Shalev-Shwartz et al., arXiv, 2017] の RSS ルールを安全エンベロープとして実装
  （近視眼的に衝突回避）
- AC+RSS^{GA}：
  我々 [Hasuo+, IEEE T-IV] の RSS ルールを安全エンベロープとして実装
  （長期的視野で目標達成も保証）

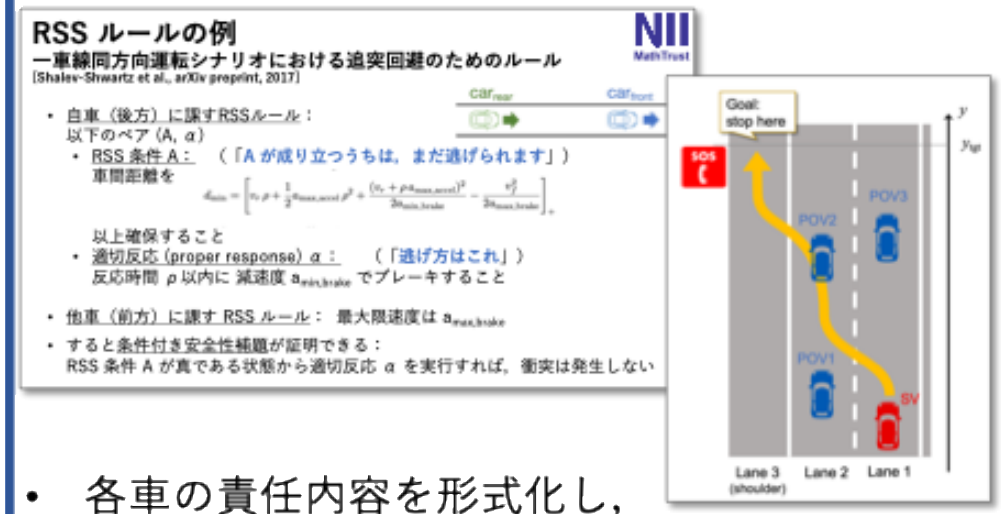


AC　　AC+*RSS*　　AC+*RSS*^{GA}

- AC, AC+RSS は安全な路肩停止に成功，しかし遅い
- AC+RSS^{GA} は（安全性保証のもと）加速して追い越し，
  他車の前で合流して路肩停止
- ➔「自動運転車は安全性ばかり気にして前に進まない」へのアンチテーゼ！

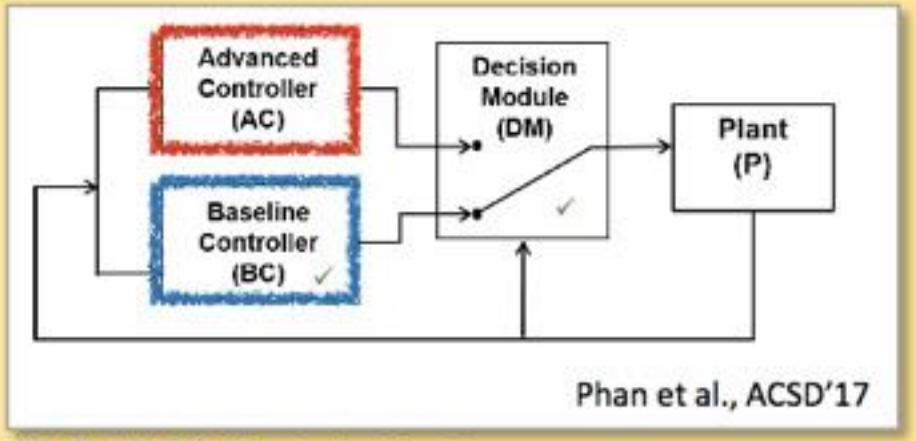
数学的安全性証明を走行時実装「追突防止自動ブレーキ」をあらゆる運転シナリオへ

技術１： 各車責任の形式化に基づく安全性証明技術



- 各車の責任内容を形式化し，
- 各車が責任を果たすという仮定のもとで，安全性を数学的に証明 → **絶対の安全性保証**

用途：
- E2E自動運転の走行時セーフガード（数学的安全性証明付き！）
- 安全性評価の定量的指標



安全性確保のための simplex architecture

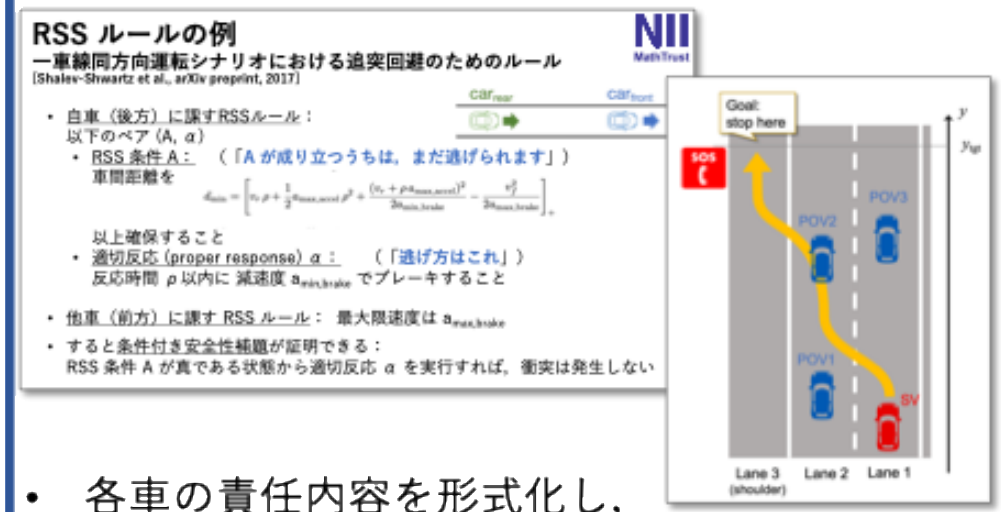
- **AC（通常系）**が性能を追求（E2E自動運転，人間，…）
- **BC（安全系）**が安全動作を実行
- **DC（切換器）**が予防条件を監視
  - まだ余裕あり → 「AC，やってみなはれ」
  - 余裕なし → BC に切替

用途１：
E2E 自動運転の走行時セーフガード

- アイデア自体は追突防止自動ブレーキと同じ
- しかし，車線変更・合流・交差点など，**様々な運転シナリオに展開可**

「逃げ代（にげしろ）」の数学的定式化による 内包的 intensional ・シナリオ特化の安全性指標

技術１： 各車責任の形式化に基づく安全性証明技術

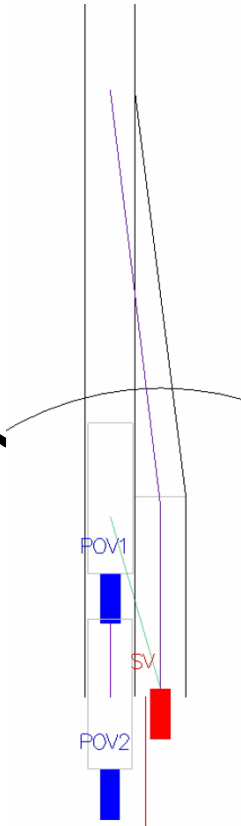


- 各車の責任内容を形式化し，
- 各車が責任を果たすという仮定のもとで，
  安全性を数学的に証明 → **絶対の安全性保証**

用途：
- E2E自動運転の走行時セーフガード
  （数学的安全性証明付き！）
- 安全性評価の定量的指標

用途２： 安全性評価の定量的指標

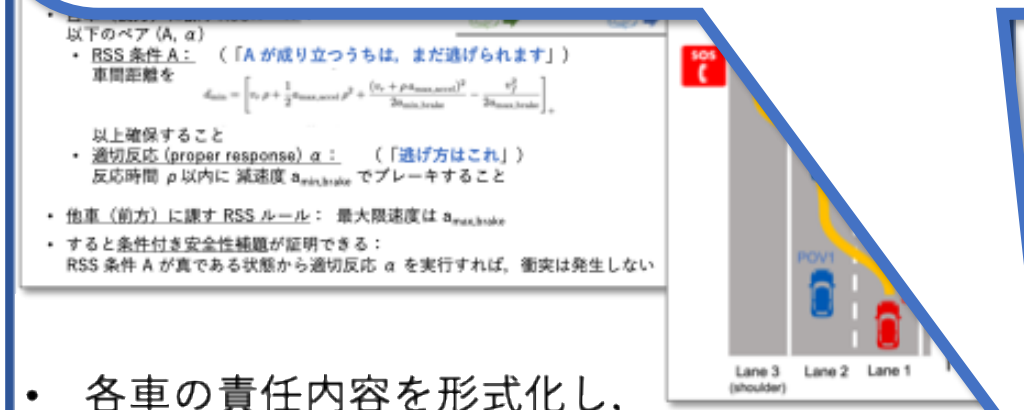
- 既存の安全性指標（車間距離，TTC，…）
  → **外延的 extensional**，各車行動への言及なし
- RSS距離
  [Shalev-Shwartz et al., arXiv, 2017] の追突安全距離
  → **内包的 intensional**，行動ベース解析
- GA-RSS 安全性指標
  [Hasuo et al., IEEE T-IV, 2023]に基づく
  → **内包的指標をあらゆる運転シナリオへ**
  - 今合流したのは，どれくらい
    『逃げ代』があった？
    （安全行動ができなくなるまでの余裕）



「まだ逃げられるための条件」

- **予防条件**：
  適切な車間距離を常に維持する
- **安全動作**：
  予防条件の違反が予見される場合にはブレーキ

「やばくなったらこうやって逃げろ」



- 各車の責任内容を形式化し，
- 各車が責任を果たすという仮定のもとで，
  安全性を数学的に証明 → **絶対の安全性保証**

用途：

- E2E自動運転の走行時セーフガード
  （数学的安全性証明付き！）
- 安全性評価の定量的指標

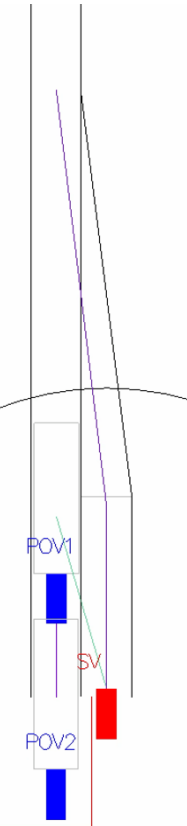
…行：用途２

…」の数学的定式化による…シナリオ特化の安全性指標

…２：安全性評価の定量的指標

…存の安全性指標（車間距離，TTC，…）
→ **外延的 extensional**，各車行動への言及なし

- RSS距離
  [Shalev-Shwartz et al., arXiv, 2017] の追突安全距離
  → **内包的 intensional**，行動ベース解析
- GA-RSS 安全性指標
  [Hasuo et al., IEEE T-IV, 2023]に基づく
  → **内包的指標をあらゆる運転シナリオへ**
  - 今合流したのは，どれくらい
    『逃げ代』があった？
    （安全行動ができなくなるまでの余裕）



NII MathTrust

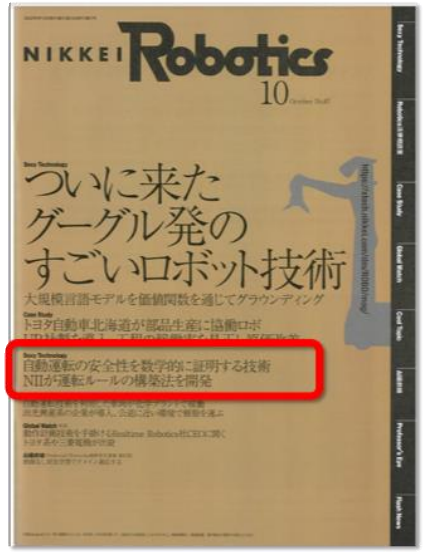
IEEE AV Decision Making WG → IEEE 2846
自動運転安全性の論理的検証の規格化を追求

- 国立情報学研究所（NII）はフルメンバーとして参加
- 月1回ミーティング

実用化に向けて

複数社とパイロットプロジェクト
用途１・２において feasibility を確認

- （そもそも）マツダさんとの共同研究
- T2 さんと共同プロジェクト
  （研究成果活用企業 （株）イミロンを通じて）
- 安全性指標の活用事例も
- 一般からの関心



日経
ロボティクス
2022年12月号

岩波「科学」
2023年3月号

自動運転の安全性を“証明”する：T2とイミロンが形式手法の活用による安全論証で共同プロジェクト開始

次世代モビリティの社会実装に向け、 数学的アプローチによる安全性評価・論理的説明性を強化

イミロン　2025年8月5日 11時30分

3

株式会社T2（東京都千代田区、代表取締役CEO：熊部 雅友、以下「T2」）は株式会社イミロン（東京都千代田区、代表取締役：足立 正和、以下「イミロン」）と、自動運転レベル4認可取得に向けた形式的安全論証の共同検証プロジェクトを、2025年8月より開始しました。本取組は、自動運転車の安全性を形式手法と数学的証明によって厳密に立証する、世界的にも先進的な挑戦であ り、実運用への適用は国内初の試みとなります。

アウトライン

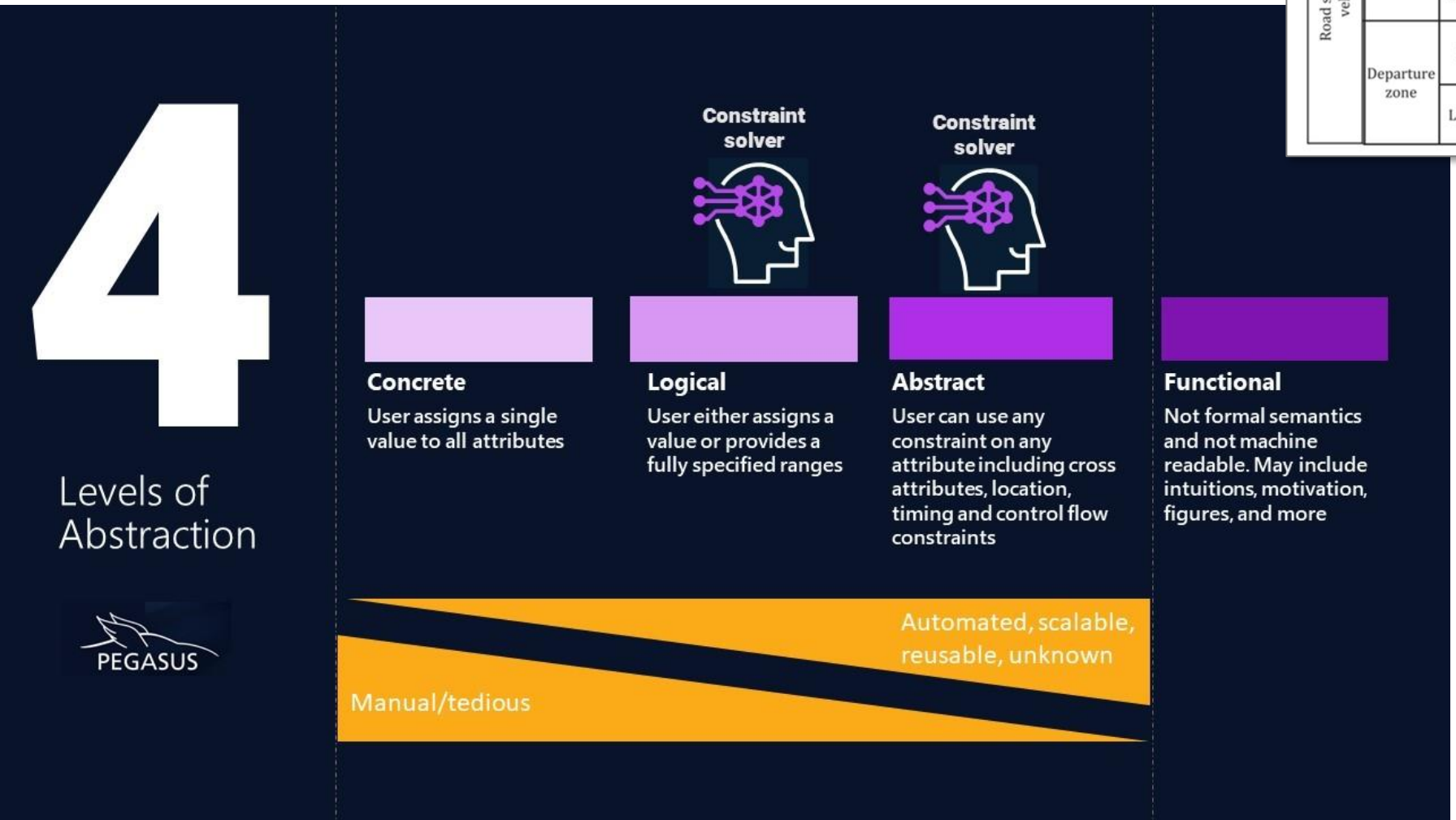
- 自己紹介：
  数学・論理学・ソフトウェア科学における「形式化」とは？

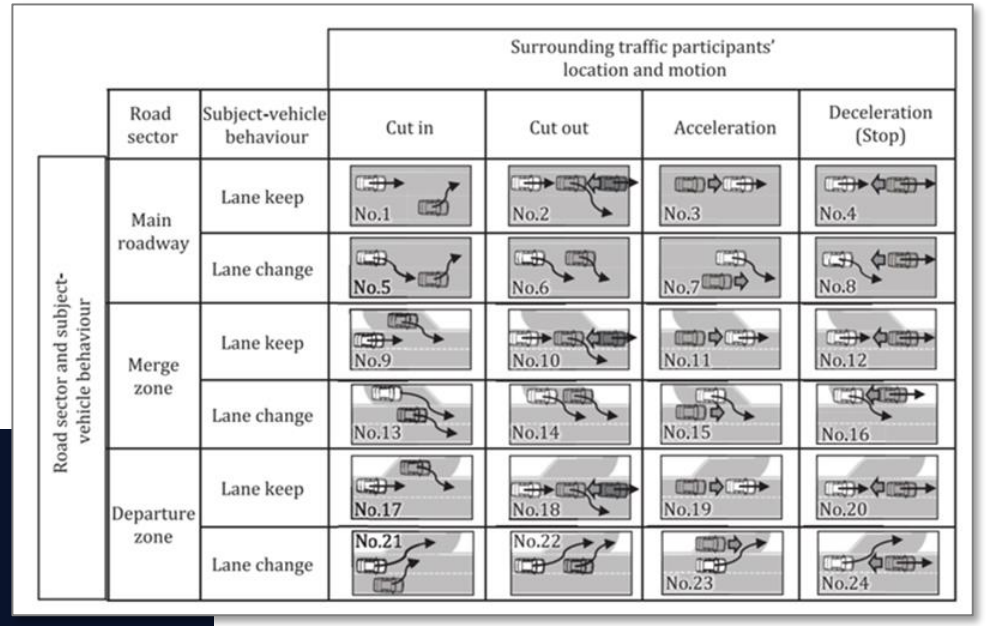
- 技術１： 各車責任の形式化に基づく安全性証明技術
  - 追突防止自動ブレーキの例
  - 形式化による対象運転シナリオの拡大
  - 用途１： E2E自動運転の走行時セーフガード
  - 用途２： 安全性評価の定量的指標

- 技術２： 安全テストにおける高レベル機能シナリオの形式化技術

ここで質問です

ASAM/Pegasus の
シナリオ抽象度レベル foretellix.com より





自工会・SAKURA・
ISO 34502の
自動運転外乱シナリオ

Q.
自動運転外乱
シナリオの
抽象度レベルは？

生成（制約を解く）

生成（値を選ぶ）

具体シナリオ

x = 1.3, y = 2,
b = true, …　変数値 fixed

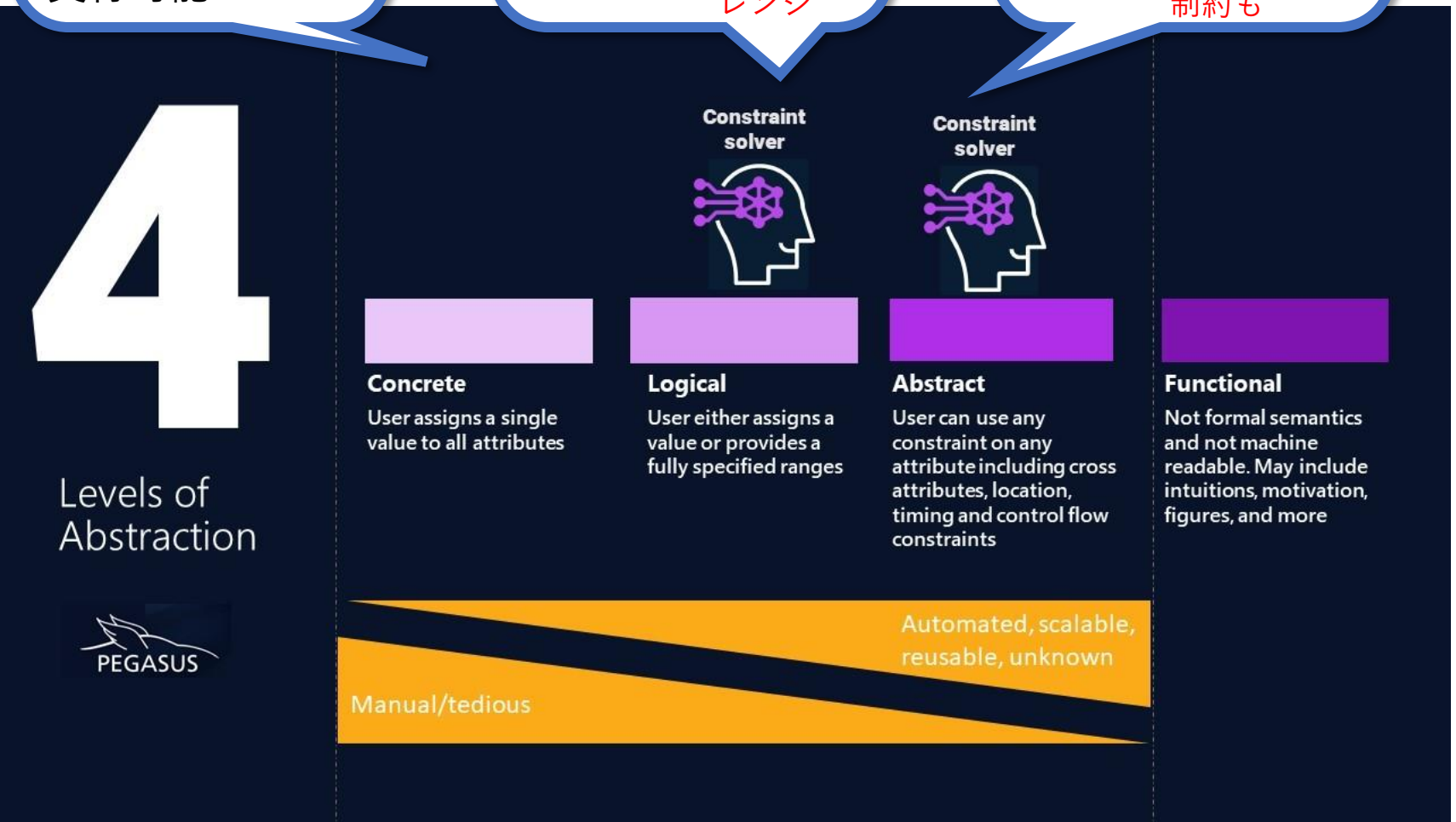
➜ シミュレータ実行可能

「論理」シナリオ

0.9 ≦ x ≦ 1.3,
0 ≦ y ≦ 4,
b ∈ {true, false},
…　各変数のレンジ

「抽象」シナリオ

x = 0.9 + 0.1 y,
0 ≦ y ≦ 4,
b ∈ {true, false},
…　変数間の関係制約も



ASAM/Pegasus の
シナリオ抽象度レベル foretellix.com より

生成（制約を解く）

生成（値を選ぶ）

具体シナリオ

x = 1.3, y = 2,
b = true, …

➜ シミュレータ実行可能

「論理」シナリオ

0.9 ≦ x ≦ 1.3,
0 ≦ y ≦ 4,
b ∈ {true, false},
…

各変数のレンジ

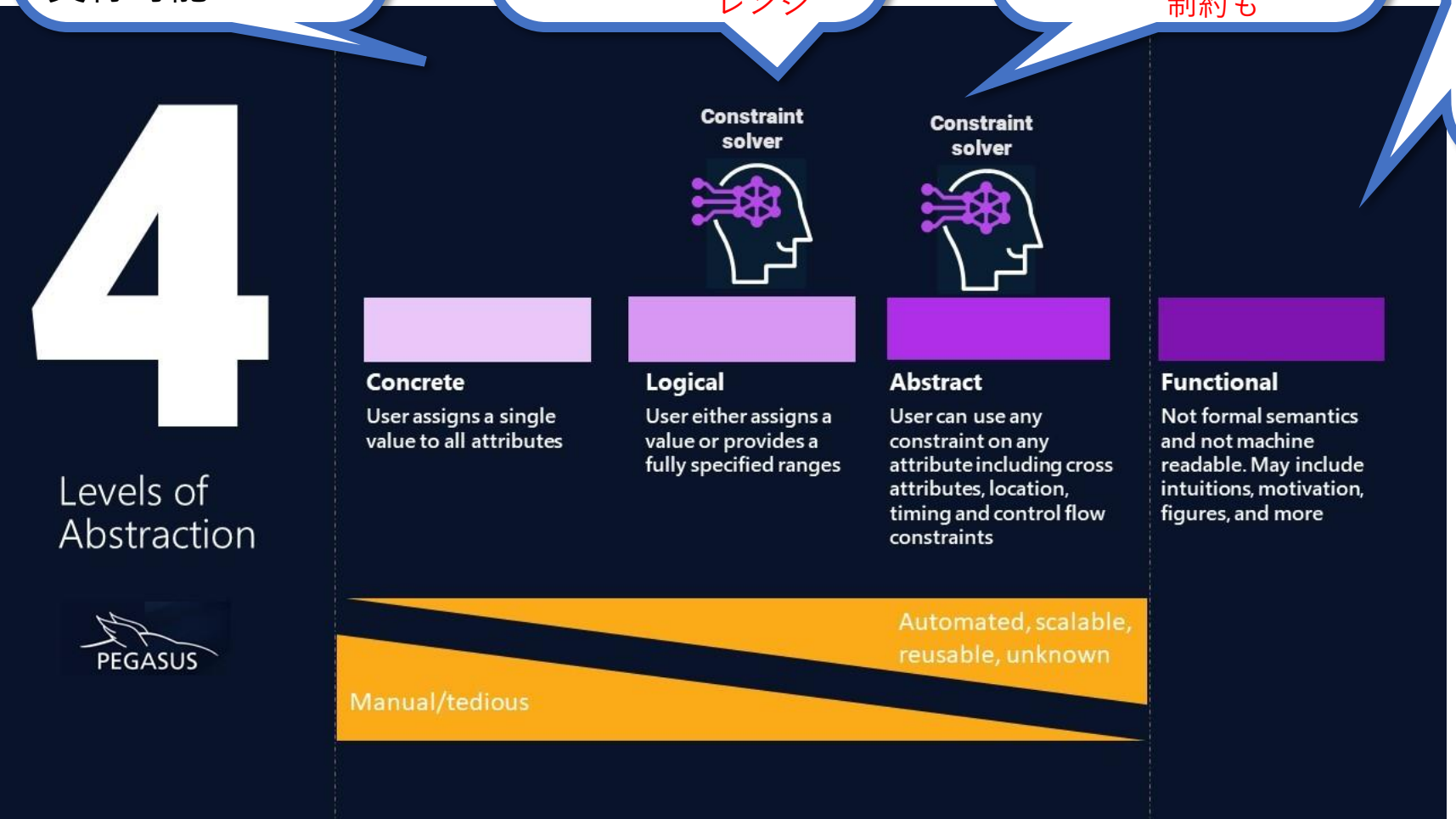
「抽象」シナリオ

x = 0.9 + 0.1 y,
0 ≦ y ≦ 4,
b ∈ {true, false},
…

変数間の関係制約も

「機能」シナリオ

- “Not formal semantics and not machine readable”
- “May include intuitions, motivations, figures, and more”



ASAM/Pegasus の
シナリオ抽象度レベル foretellix.com より

生成（値を選ぶ）

生成（制約を解く）

具体シナリオ

x = 1.3, y = 2,
b = true, …

➜ シミュレータ実行可能

「論理」シナリオ

0.9 ≦ x ≦ 1.3,
0 ≦ y ≦ 4,
b ∈ {true, false},
…

各変数のレンジ

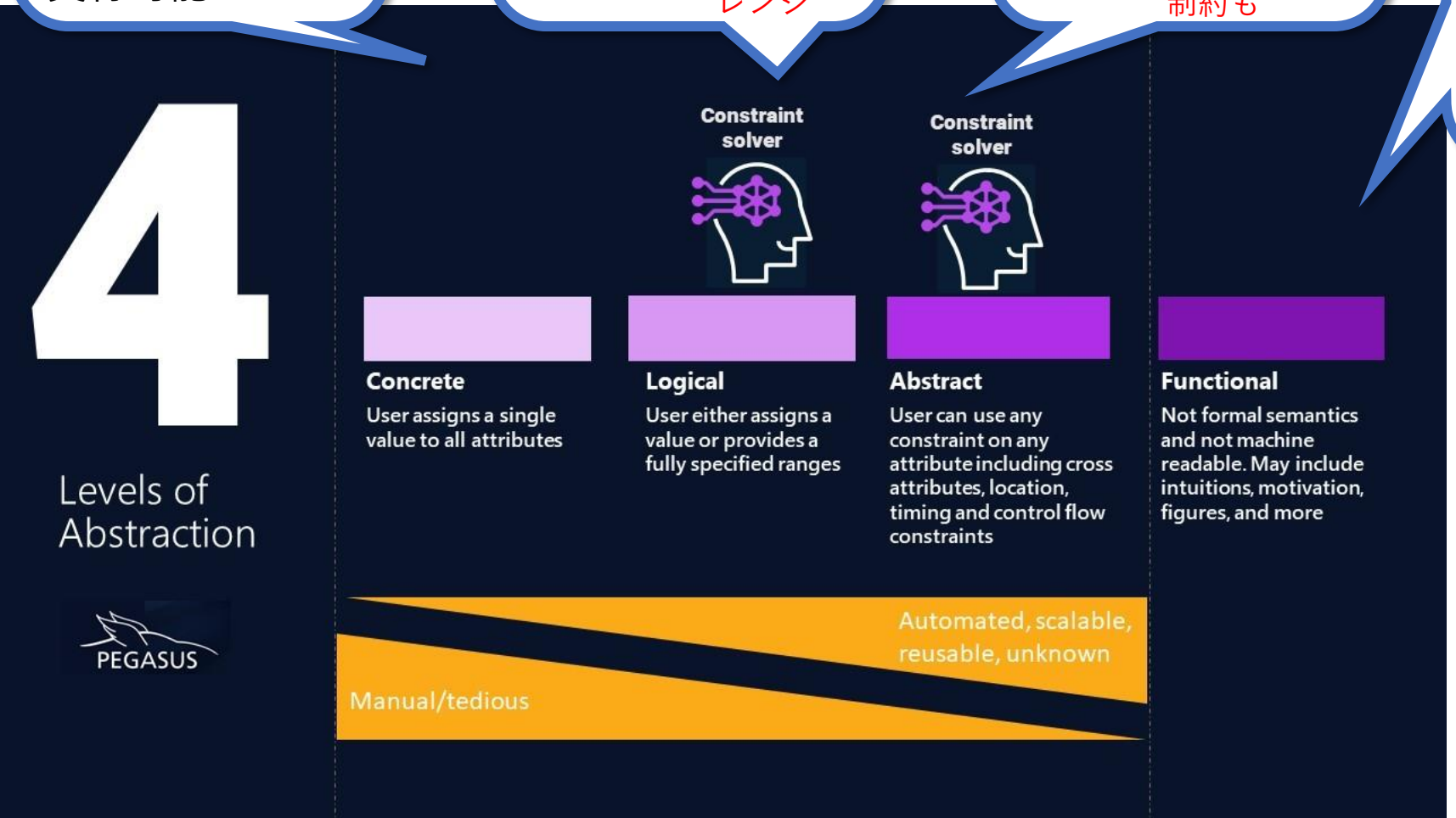
「抽象」シナリオ

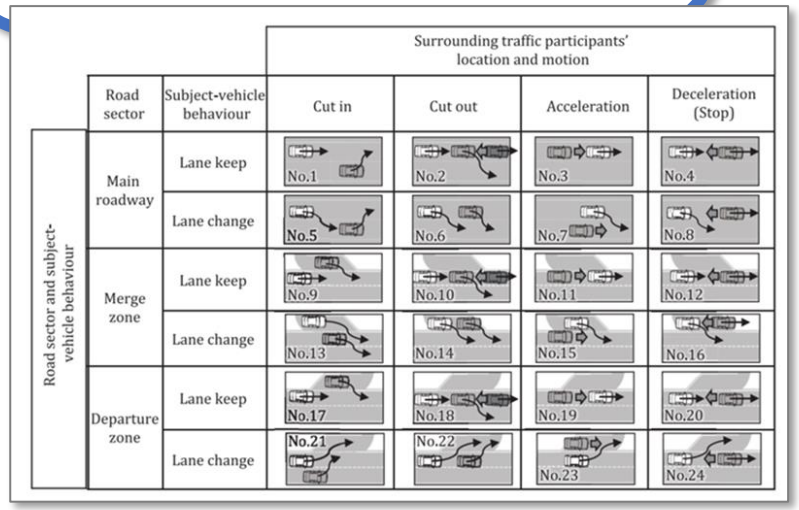
x = 0.9 + 0.1 y,
0 ≦ y ≦ 4,
b ∈ {true, false},
…

変数間の関係制約も

「機能」シナリオ

- “Not formal semantics and not machine readable”
- “May include intuitions, motivations, figures, and more”





自工会外乱シナリオは機能シナリオ？

ASAM/Pegasus のシナリオ抽象度レベル foretellix.com より

生成（値を選ぶ）

生成（制約を解く）

具体シナリオ

x = 1.3, y = 2,
b = true, …

➜ シミュレータ実行可能

「論理」シナリオ

0.9 ≦ x ≦ 1.3,
0 ≦ y ≦ 4,
b ∈ {true, false},
…

各変数のレンジ

「抽象」シナリオ

x = 0.9 + 0.1 y,
0 ≦ y ≦ 4,
b ∈ {true, false},
…

変数間の関係制約も

「機能」シナリオ

- “Not formal semantics and not machine readable”
- “May include intuitions, motivations, figures, and more”

形式化したい！

4

Levels of Abstraction

PEGASUS

Constraint solver

Constraint solver

Concrete
User assigns a single value to all attributes

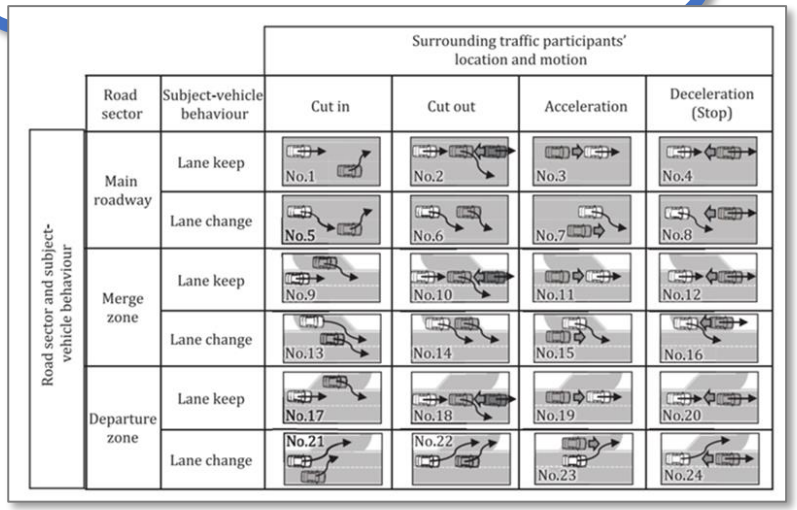
Logical
User either assigns a value or provides a fully specified ranges

Abstract
User can use any constraint on any attribute including cross attributes, location, timing and control flow constraints

Functional
Not formal semantics and not machine readable. May include intuitions, motivation, figures, and more

Automated, scalable, reusable, unknown

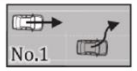
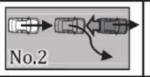
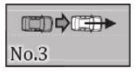
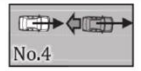
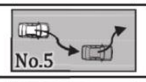
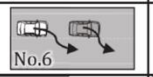
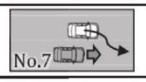
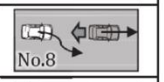
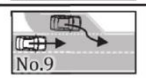
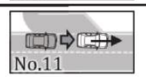
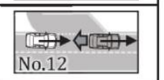
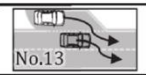
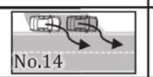
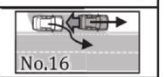
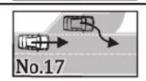
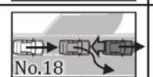
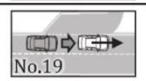
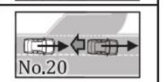
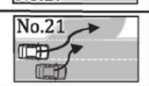
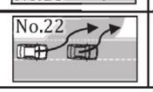
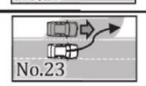
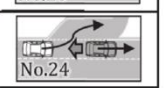
Manual/tedious



自工会外乱シナリオは機能シナリオ？

ASAM/Pegasus のシナリオ抽象度レベル foretellix.com より

… 形式化しました.

Road sector	Subject-vehicle behaviour	Cut in	Cut out	Acceleration	Deceleration (Stop)
Main roadway	Lane keep	 No.1	 No.2	 No.3	 No.4
	Lane change	 No.5	 No.6	 No.7	 No.8
Merge zone	Lane keep	 No.9	 No.10	 No.11	 No.12
	Lane change	 No.13	 No.14	 No.15	 No.16
Departure zone	Lane keep	 No.17	 No.18	 No.19	 No.20
	Lane change	 No.21	 No.22	 No.23	 No.24

Jesse Reimann, Nico Mansion, James Haydon, Benjamin Bray, Agnishom Chattopadhyay, Sota Sato, Masaki Waga, Étienne André, Ichiro Hasuo, Naoki Ueda, and Yosuke Yokoyama. 2024. Temporal Logic Formalisation of ISO 34502 Critical Scenarios: Modular Construction with the RSS Safety Distance. In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24). Association for Computing Machinery, New York, NY, USA, 186–195. https://doi.org/10.1145/3605098.3636014

（三菱電機さんとの共同研究）

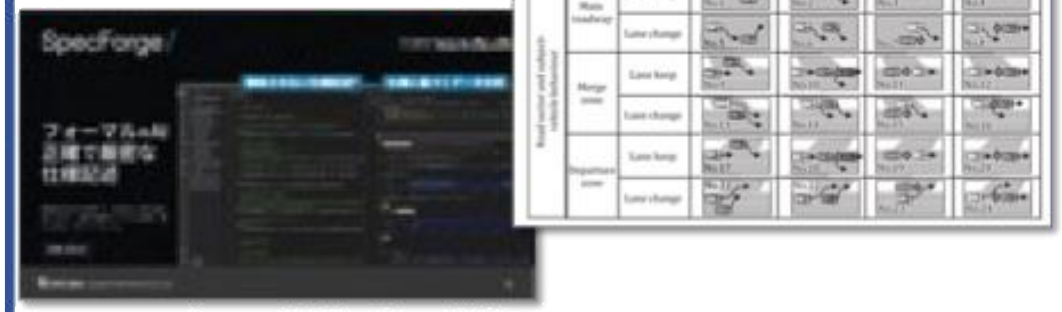
$scenario_i(SV, POV, L) := initSafe(SV, POV) \wedge roadSector_i(SV, POV) \wedge disturb_i(SV, POV, L)$, $i = 1, \ldots, 24$ (cf. this is (1). *initSafe* is from §4.3)
$disturb_i(SV, POV, L) := initialCondition_i(SV, POV, L) \wedge behaviourSV_i(SV, L) \wedge behaviourPOV_i(POV, SV, L)$, $i = 1, \ldots, 24$ (cf. (2) in §3)

i	$roadSector_i$ (cf. §4.1)	i	$initialCondition_i$ (cf.§4.2)	$behaviourSV_i$ (cf.§4.4)	$behaviourPOV_i$ (cf.§4.5)
1 – 8	$mainRoad(SV, POV)$	1	$\top$	$laneKeep(SV, L)$ $\mathcal{U}\, danger(SV, POV)$	$cutIn(POV, SV)$
		2	$sameLane_3(SV, POV_1, POV_2, L)$ $\wedge aheadOf(SV, POV_1)$ $\wedge aheadOf(POV_1, POV_2)$	$laneKeep(SV, L)$ $\mathcal{U}(\neg sameLane(SV, POV_1, L))$	$leavingLane(POV_1, L)$ $\wedge(laneKeep(POV_2, L)$ $\mathcal{U}(\neg sameLane(POV_2, POV_1, L)$ $\wedge danger(SV, POV_2)))$
		3	$aheadOf(POV, SV)$ $\wedge(sameLane(SV, POV, L)$ $\vee inAdjLanes(SV, POV, L))$	$laneKeep(SV, L)$ $\mathcal{U}\, danger(SV, POV)$	$accel(POV, SV, L)\ \mathcal{U}\ danger(SV, POV)$
		4	$aheadOf(SV, POV)$ $\wedge(sameLane(SV, POV, L)$ $\vee inAdjLanes(SV, POV, L))$	$laneKeep(SV, L)$ $\mathcal{U}\, danger(SV, POV)$	$decel(POV, SV, L)\ \mathcal{U}\ danger(SV, POV)$
		5	$\top$	$leavingLane(SV, L)$	$cutIn(POV, SV)$
		6	$\top$	$leavingLane(SV, L)$	$cutOut(POV, SV, L)$
		7	$aheadOf(POV, SV)$	$enteringLane(SV, L)$	$accel(POV, SV, L)\ \mathcal{U}\ danger(SV, POV)$
		8	$sameLane(SV, POV, L)$ $\wedge aheadOf(SV, POV)$	$leavingLane(SV, L)$	$decel(POV, SV, L)\ \mathcal{U}\ danger(SV, POV)$
9–16	$mergeZone(SV, POV)$	9–16	$initialCondition_{i-8}$	$behaviourSV_{i-8}$	$behaviourPOV_{i-8}$
17–24	$departZone(SV, POV)$	17–24	$initialCondition_{i-16}$	$behaviourSV_{i-16}$	$behaviourPOV_{i-16}$

動作意図を含む「機能シナリオ」を数学的形式化 計算機による管理・活用が可能に

技術２： 安全テストにおける高レベル機能シナリオの形式化技術

$$\mathbf{G}(p \rightarrow \mathbf{F}_{[0,T]} q)$$



- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  → **高レベルシナリオの管理・活用の自動化**

用途：
- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成

時相論理（temporal logic）を活用

- 命題論理 （∧, ∨, ⇒, ¬ ）
  ＋ 時相演算子
  （**G** これからずっと，**F** 未来のどこかで，…）
- 例： **G** (request ⇒ $\mathbf{F}_{[0,T]}$ response)
  「request が発生したら，T秒以内に response が返ることが，今後ずっと保証される」
- 単純な言語，習得はそう難しくない
  （プログラミング言語の if 分岐条件 ＋ 時相，
  大学の情報系講義であれば1~2回）
- 表現能力と複雑さのバランス
  （一般に，いろいろ書ける言語は処理が重い）
- 数学・ソフトウェア科学における理論研究の積み重ね
  Turing 賞 (Pnueli, Clarke, …)

NII MathTrust

動作意図を含む「機能シナリオ」を数学的形式化 計算機による管理・活用が可能に

技術２： 安全テストにおける高レベル機能シナリオの形式化技術

$$\mathbf{G}(p \rightarrow \mathbf{F}_{[0,T]}q)$$

- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  → **高レベルシナリオの管理・活用の自動化**

用途：
- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成

用途： 走行データに対する自動マッチング

実走行データの中から，当該シナリオにマッチする部分を**高速・全自動で抽出**.

（時相論理モニタリング・実行時検証の研究成果を活用）

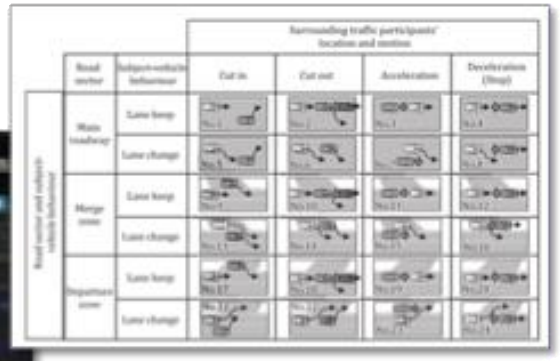
BEFORE：
シナリオごとに抽出用の C コードを個別に開発
- 工数？ （数百行）
- 記述の属人性？ 透明性，説明可能性？
- パフォーマンス？

AFTER：
シナリオを時相論理式で記述し，汎用マッチングツールへ
- 工数 よし（数行～数十行）
- 記述の説明可能性 （論理式は読みやすい）
- パフォーマンス （SOTA の汎用アルゴリズム）

動作意図を含む「機能シナリオ」を数学的形式化
計算機による管理・活用が可能に

技術２： 安全テストにおける高レベル機能シナリオの形式化技術

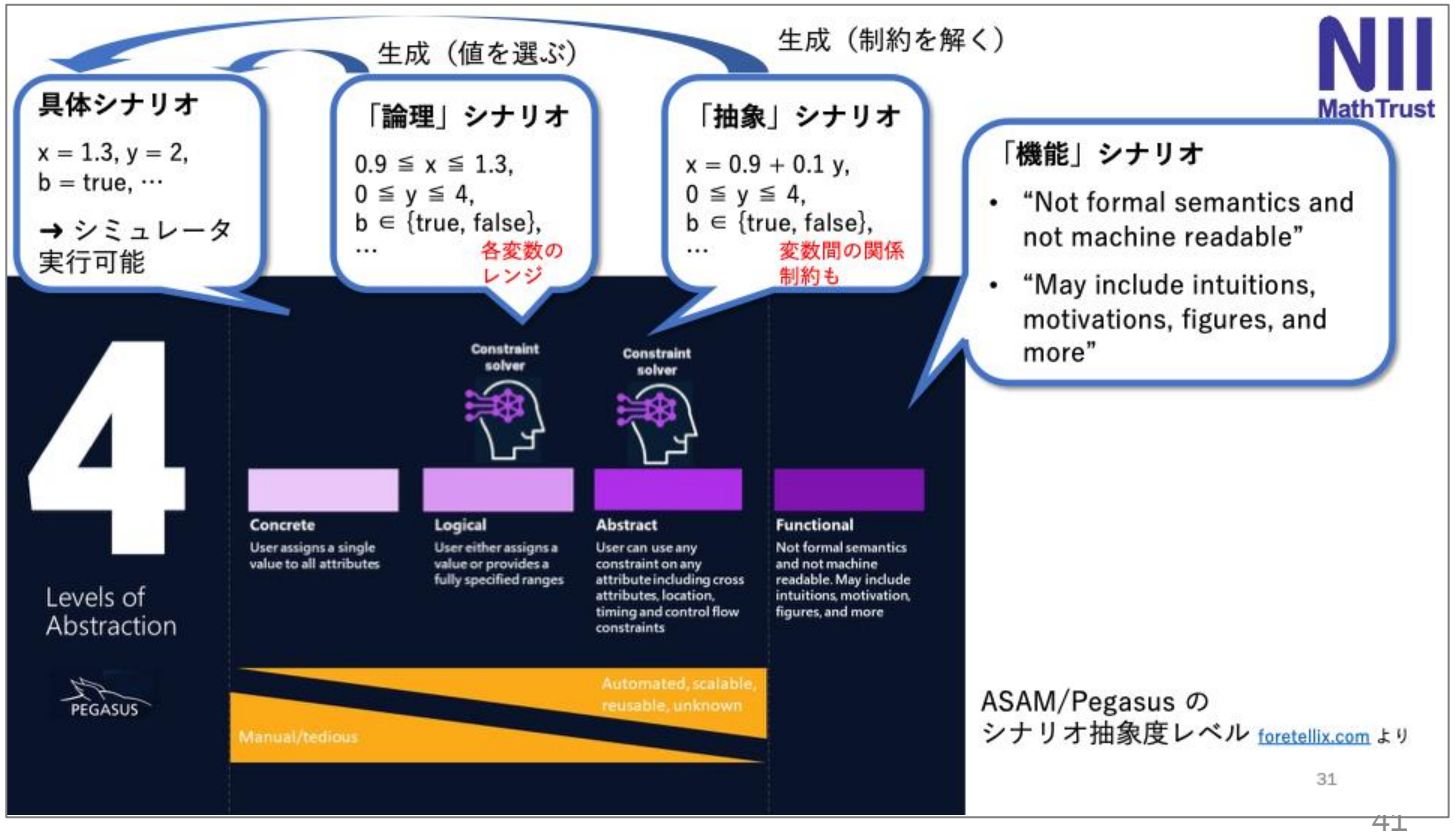
$$\mathbf{G}(p \rightarrow \mathbf{F}_{[0,T]}q)$$



- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  → **高レベルシナリオの管理・活用の自動化**

用途：
- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成

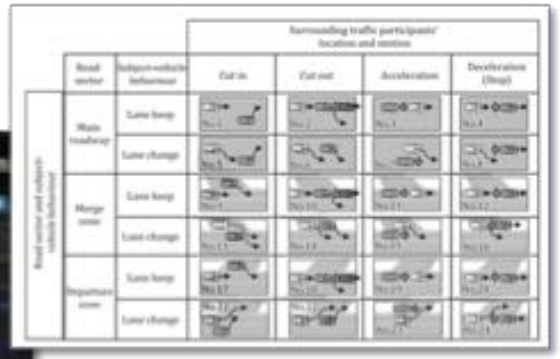
用途： 低レベル具体シナリオの自動生成



動作意図を含む「機能シナリオ」を数学的形式化
計算機による管理・活用が可能に

技術２： 安全テストにおける高レベル機能シナリオの形式化技術

$$\mathbf{G}(p \rightarrow \mathbf{F}_{[0,T]} q)$$

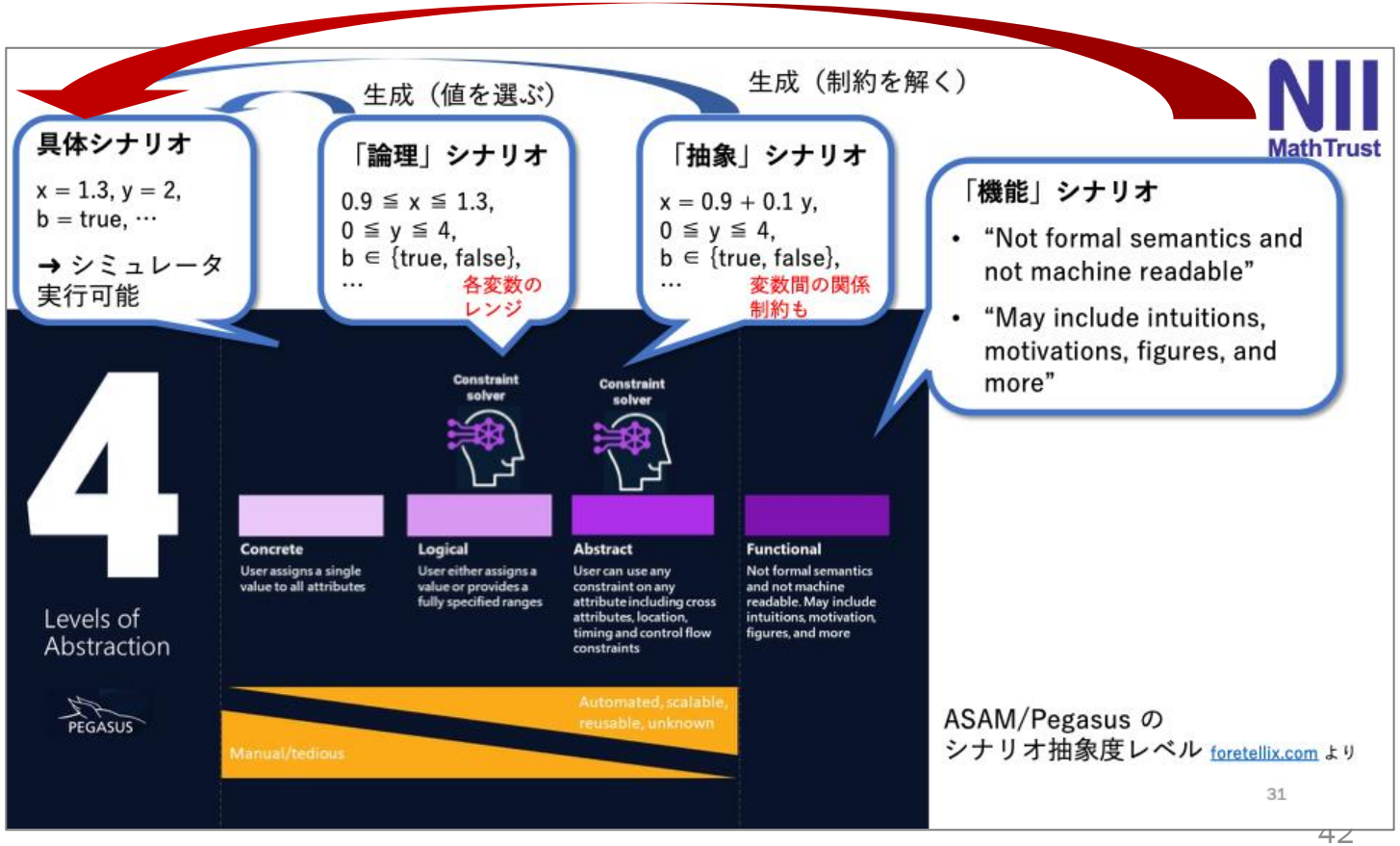


- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  → **高レベルシナリオの管理・活用の自動化**

用途：
- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成

用途： 低レベル具体シナリオの自動生成

直接の自動生成が可能
[Sato+, CAV'24]，MILP 最適化により，多くは <1秒，最悪ケースでも10秒で

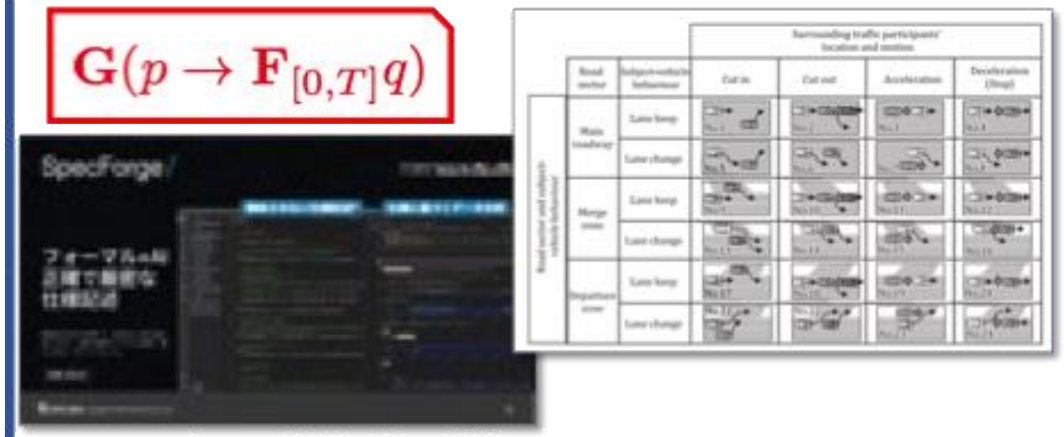


NII MathTrust

動作意図を含む「機能シナリオ」を数学的形式化
計算機による管理・活用が可能に

技術２： 安全テストにおける高レベル機能シナリオの形式化技術

$$\mathbf{G}(p \rightarrow \mathbf{F}_{[0,T]} q)$$



- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  → **高レベルシナリオの管理・活用の自動化**

用途：
- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成

用途： 機能シナリオの意味の厳密化

論文 [Reimann, Mansion, et al., SAV'24] で，
形式化シナリオの highD データセット評価により
次を発見：

- ISO 34502 で「加速 accelerate」と記述されているのは，
- 「加速 ないし 後車の方が前車より速い」としたほうが，よりハイレベルの意図に叶う

（具体的には，このように意味を拡張しないと，
別の方法で定義した危険シーンの取りこぼしが発生）

NII MathTrust

動作意図を含む「機能シナリオ」を数学的形式化
計算機による管理・活用が可能に

技術２： 安全テストにおける高レベル機能シナリオの形式化技術

$$\mathbf{G}(p \rightarrow \mathbf{F}_{[0,T]}q)$$

- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  → **高レベルシナリオの管理・活用の自動化**

用途：
- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成

用途： 機能シナリオの管理タスクの自動化

高速道路では 6 x 4 シナリオだが,
一般道では？ 数百～数千シナリオ？
管理は？ Excel？

→ 数学的形式化により**データベース管理が可能に**

- タグ付け
- 条件による絞り込み
- 具体シナリオ・テスト結果との紐づけ
- 統計的解析と可視化
- 矛盾・重複の検索
  （時相論理式の充足可能性チェック satisfiability）
- バージョン管理 → トレーサビリティ
- …

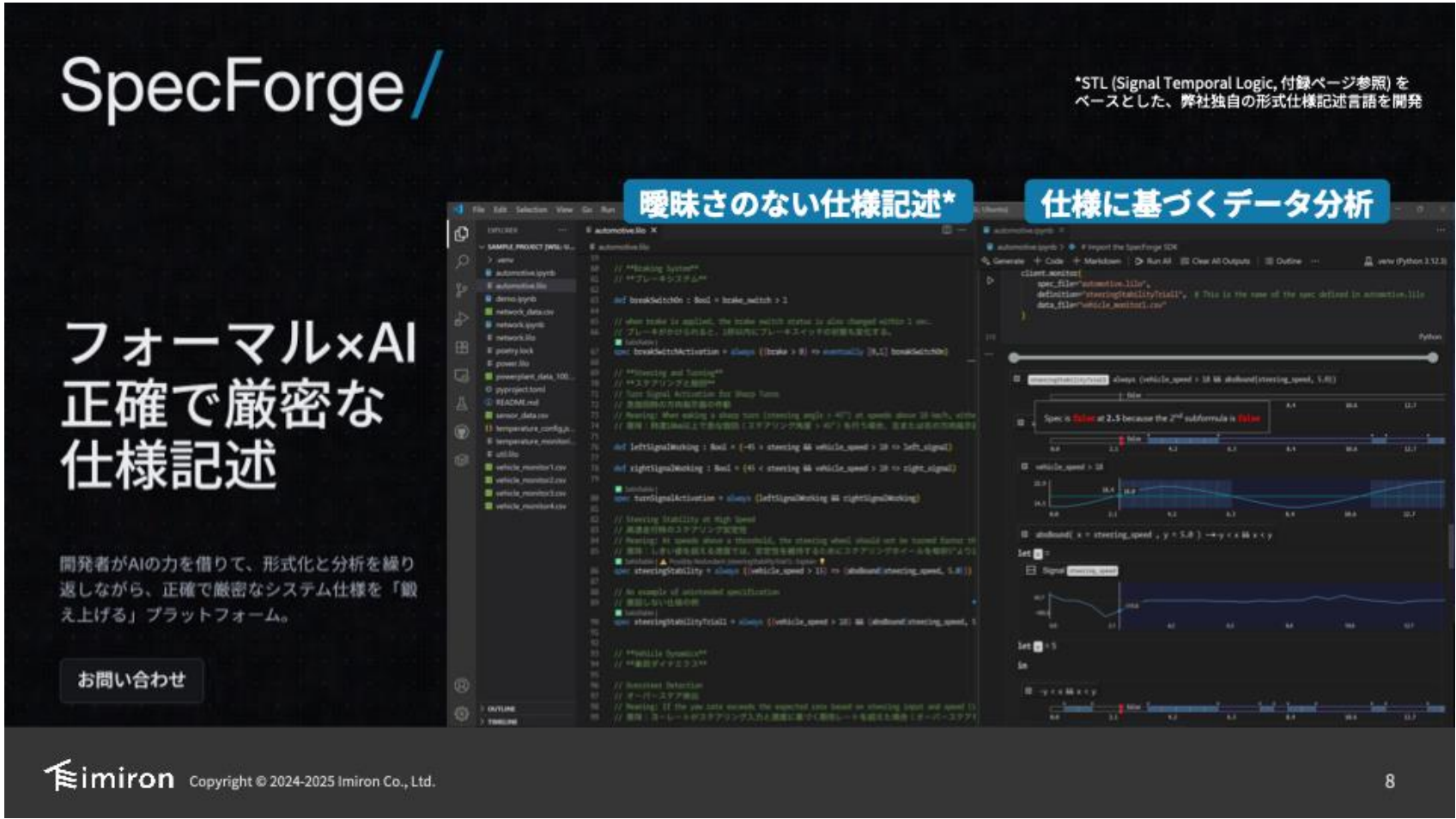
（「**テストシナリオの github**」というイメージ）

NII MathTrust

時相論理仕様オーサリング＆管理ツールを研究成果活用企業からリリース

国立情報学研究所からの
知財ライセンシングを受けた
株式会社イミロンが
ツール開発・販売

- 上記用途のいくつかを実装済み
  （特にオーサリング・マッチング・テスト生成）
- テストシナリオ管理ツール・チームウェアとしての機能を開発中



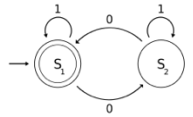
株式会社イミロン： 国立情報学研究所の研究成果活用企業

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。16年10月-25年3月まで
JST START **イミロン設立** imiron	**ソフトウェア品質の論理的説明技術による、自動運転の本格普及の実現** 22年11月-25年3月。事業プロモーター： インキュベイトファンド株式会社

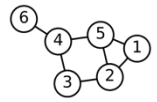
技術力

理論計算機科学 - Wikipedia

$P \to Q$ 数理論理学

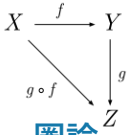


オートマトン

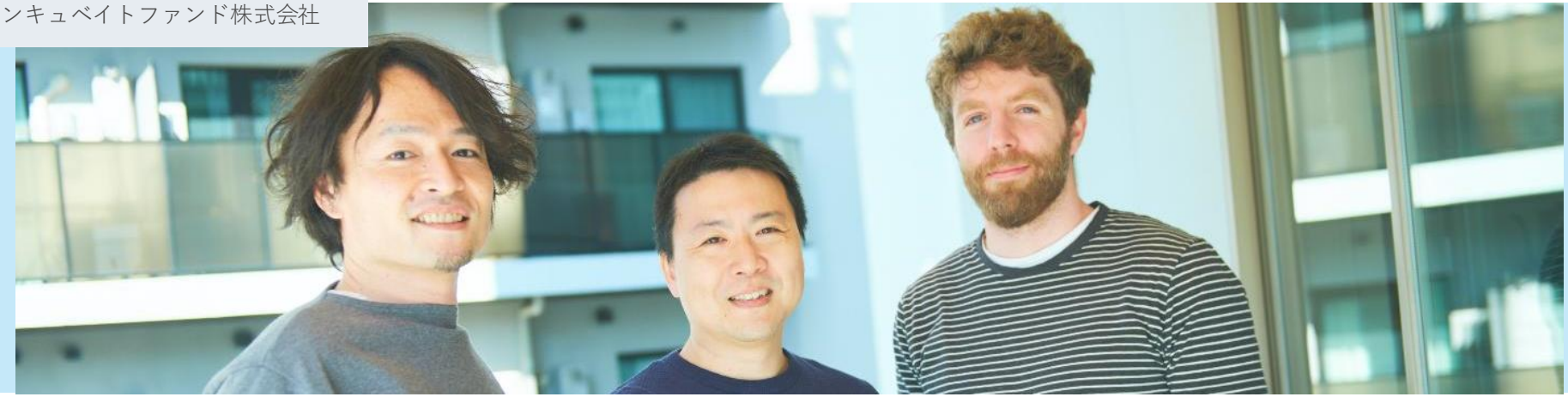


グラフ理論

$\Gamma \vdash x : Int$ 型理論



圏論



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

取締役 (CSO)・創業者

蓮尾 一郎 教授

東京大学/ラドバウド大学 (蘭)
➡ 京大、東大等で教職を歴任

研究業績 論文 221 編

主な受賞歴 '24年 文部科学大臣表彰 科学技術賞

代表取締役 (CEO)・創業者

足立 正和 博士 (工学)

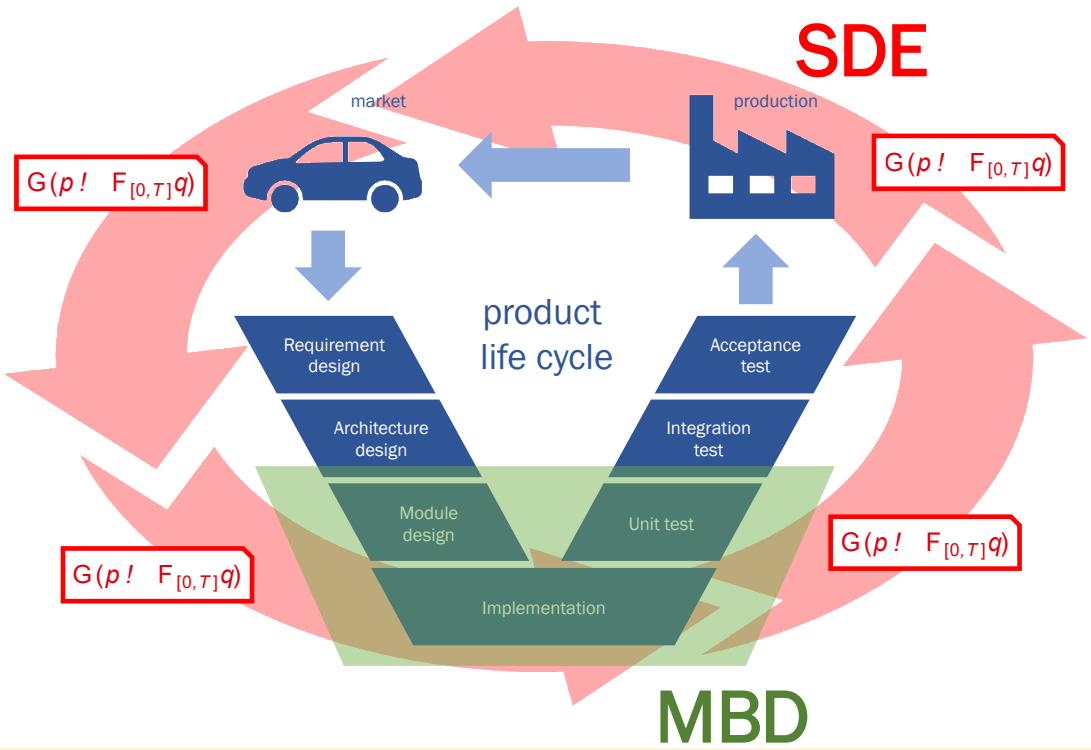
大阪大学
➡ カーネギーメロン大学 (ポスドク)
➡ 株式会社 豊田中央研究所
➡ 株式会社 デンソー
➡ デンソードイツ Corp. R&D ヘッド

取締役 (CTO)・創業者

ジェームズヘイドン 博士 (学術)

インペリアル・カレッジ・ロンドン
➡ オックスフォード大学
➡ 欧州スタートアップ テックリード
(Artificial, Tweag etc.)
➡ 蓮尾研究室 テックリード

時相論理は工業製品の要求仕様一般に適用可
➜ Specification-Driven Engineering へ



	モデルベース開発 MBD	仕様駆動エンジニアリング SDE
ターゲット	設計・実装・V&V	認証・保守を含むライフサイクル全般
最初に必要なもの	システムモデル（工数大）	形式仕様（数行〜数十行）
利点	V&V 工程の left-shift	• 要求仕様の厳密化，ソフトウェアサポート • V&V 工程の自動化 • トレーサビリティ

V&V asks

digitalized in MBD ⇓

digitalized in SDE ⇓

if a system (model) M satisfies a spec φ

➜ 相補的，両端からの設計工程DX

47

アウトライン

- 自己紹介：
  数学・論理学・ソフトウェア科学における「形式化」とは？

- 技術１： 各車責任の形式化に基づく安全性証明技術
  - 追突防止自動ブレーキの例
  - 形式化による対象運転シナリオの拡大
  - 用途１： E2E自動運転の走行時セーフガード
  - 用途２： 安全性評価の定量的指標

- 技術２： 安全テストにおける高レベル機能シナリオの形式化技術

NII MathTrust

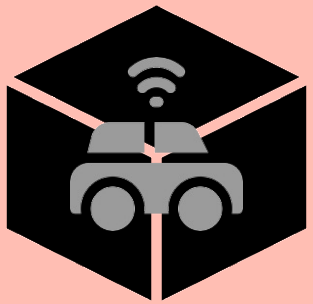
来たるべき「侵襲的」情報技術と人間社会の間の説明可能インターフェイス

- （統計的）AIと人間の関わり方，２つの未来像

- AIは「侵襲的」．社会受容のやり方を注意深く考える必要がある

- ぜひ右の未来像，「人間中心の情報化社会」をめざしていきたい

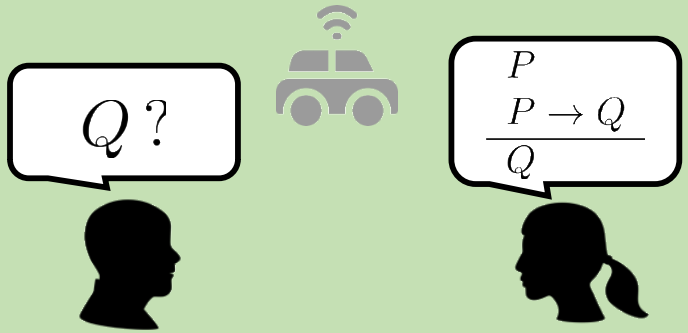
ブラックボックス安全性保証



- 中身の見えない「全体主義的」安全性論証
- 議論，精査，批判的検討，改善が困難

vs

説明可能な安全性保証



- 説明責任，トレーサビリティを満たす安全性論証を論理的に構成
- ICT システムの安全性保証は終わりがない課題
  → 社会全体での取り組みをサポート
- 我々の目指す未来像

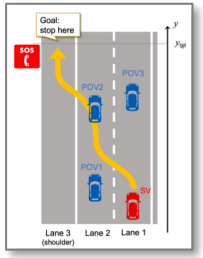
自動運転安全性の保証・説明のための数学・論理学

NII MathTrust

技術１：　各車責任の形式化に基づく安全性証明技術

- 各車の責任内容を形式化し，
- 各車が責任を果たすという仮定のもとで，
  安全性を数学的に証明 ➔ **絶対の安全性保証**

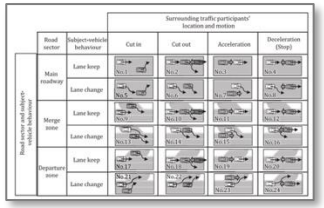
用途：
- E2E自動運転の走行時セーフガード
  （数学的安全性証明付き！）
- 安全性評価の定量的指標



技術２：　安全テストにおける高レベル機能シナリオの形式化技術

- テストシナリオの意図を記述する高レベル「機能シナリオ」の内容を形式化
  ➔ **高レベルシナリオの管理・活用の自動化**

用途：
- 機能シナリオの意味の厳密化
- 走行データにおける自動マッチング
- 条件による絞り込み等，管理タスクの自動化
- 低レベル具体シナリオの自動生成



数学的証明 「これがこうなるから安全．証明終」	安全性保証のカタチ	**経験論的保証** 「これだけのシナリオでテストして安全だったので大丈夫」（というチェックの責任を果たしましたよ）
強い 論理的な絶対の保証	安全性保証の強度・説明可能性	**比較的弱い** 経験論的な保証 ➔ 保証・説明の体系化が必要（本技術）
狭い 高速道，幹線道路の交差点（構造的）	主ターゲットたる運転シーン	**広い** 一般道を含むあらゆる運転シーンに適用