自動運転車の安全性の数学的証明
――論理学の社会応用の一例として

蓮尾一郎

はすお　いちろう
国立情報学研究所，総合研究大学院大学（ソフトウェア科学・数理論理学）

自動運転が普及していくうえで，どのように安全性を保証し，社会に対して説明するかは大きな課題です。現在主流の統計的保証に対し，安全性を数学的定理として証明するという論理的アプローチが注目を集めています。本稿では，このアプローチを可能にするRSSの方法論と，その論理学的形式化の意義，さらに，続々と現れるICT技術の社会受容のために数学が果たすべき役割について論じます。

自動運転車の安全性保証

近年の情報技術の発展とともに，自動運転が急速に現実のものとして認識されつつあります。2017年頃の盛り上がりは大したもので，「あと数年で完全自動運転が実現する」といった言説が多く聞かれました。しかし2018年に起こった数件の不幸な事故のあと，自動運転への過度な期待はしぼんでしまったように見えます。

自動運転の本格普及までの障害としては，技術的なハードルはもちろんですが，**安全性保証**も大きな問題です。公道は文字通り「公」，すなわち社会のものですから，安全性が十分でなかったり，いざ事故を起こしたりした際に責任をとらない自動車を受け入れるわけにはいきません。何をもって「公道を走るために十分に安全」とするのか，すなわち**安全基準**についての技術的・社会的合意がいまだなされていないため，自動運転が普及しないというわけです。

安全基準の不在は，企業にとっても大きな**ビジネス的障害**です。明らかな安全基準がないと，企業が負うべき製造物責任の範囲を確定させることができません。自動運転車を売ることで将来的に莫大な賠償リスクを抱えることになってしまっては，研究開発と事業展開が及び腰になってしまいます。

このような安全性保証の課題は広く認識され，さまざまな取り組みが規制当局や規格化団体，業界団体などにより進められています。この取り組みのほとんどは**統計的な安全性保証**と呼ぶべきものです。

統計的安全性保証の代表的な1つが**事故統計**による保証です（走行距離100万マイルあたり事故何件など）。しかしこれはとても荒っぽい平均値にすぎず，たとえば「私の家の前の交差点では夕刻必ず事故が起きる（だがこの特定の状況は全体のごく一部）」という危険（再現可能エラーと呼ばれる）を排除することができません。

もう1つの代表的な統計的安全性保証はテストです。ここでは，テストシナリオ群をまず定めたうえで，これらのシナリオで事故を起こさないことを（多くの場合コストの安い計算機シミュレーションで）確認します。すると当然，重要な交通状況をできるだけ多くカバーするようなテストシナリオ群の選定が必要になります。交通状況には無限のバリエーションがあるため（道路形状，他車・歩行者の振る舞い，天候など），テストシナリオ群の選定は非常に困難な問題です。現在の安全基準策定の取り組みの多くは，この問題に関わっています。

しかし，統計的な安全性保証の取り組みには，いくつかの批判がつきまといます。批判の1つは**経験論的保証**であることであり（「これくらい試して安全だったから今後も安全だろう」），保証の度合いが十

分かどうかの判断は簡単な問いではありません。もう 1 つは**説明可能性の限界**であり，事故統計にしてもテストにしても「なぜ安全なのか」という問いを徹底的に突き詰めることは困難です。

自動運転安全性の数学的証明とその困難

そこで，安全性保証への全く異なるアプローチとして立ち現れるのが**数学的証明**です。自動運転車の安全性を定理として述べて，この定理を数学的に証明しようというこのアプローチは，帰納的 vs. 演繹的という意味で統計的取り組みと対比できます。論理的推論ステップを積み重ねて得られる数学的証明の**正しさは絶対**であり，経験論的保証とは全く違うレベルの強い保証を与えます。また，数学的証明の推論ステップの積み重ねは安全性の詳細な説明になっているため，**高い説明可能性**をもちます。

実際，情報通信技術(ICT)の多くの分野において，システムの「正しさ」の数学的証明は古くから試みられており，**形式検証 formal verification** と呼ばれています。ソフトウェア検証(プログラムにバグがないことの証明)やハードウェア検証(集積回路の設計の正しさの証明，1994 年の Pentium FDIV のバグ以降急速に普及した)など，多くの成功例があります。

近年，ICT システムがあらゆる分野に進出し，物理システムと融合して**物理情報システム cyber-physical system**(CPS)と呼ばれるパラダイムを生み出しました。家電，自動車，航空宇宙など現代の工業製品の大多数が CPS の例になっており，より最近のバズワードである IoT(Internet of Things)とも CPS は強く関連します。もちろん自動運転車は CPS の代表例です。上記の成功(ソフトウェア・ハードウェアの形式検証)を受けた自然な流れとして，形式検証を CPS に適用しようとする試みが 15 年程前から盛んに行われてきました。

しかし，形式検証を CPS に応用するには大きな困難が 1 つあります。この困難ゆえ，自動運転車の安全性証明はなかなか実現しませんでした。

その困難とはすなわち**モデリングの難しさ**です。

定義 **完備距離空間**とは任意のコーシー列が極限をもつ距離空間のことをいう．

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

証明． 点 $x \in X$ 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．$x_0 = f(x_0)$ であることは容易に示される．□

定義 **自動運転車**とは …??

定理 自動運転車は安全である．

証明． ???

図 1―数学における定理と定義(上)，形式検証における定義，すなわちモデリング(下)

まず大前提となる数学の話をしましょう。数学で何かを証明するためには，登場するすべての概念の正確な**定義**が必要です。たとえば図 1 上では，Banach の不動点定理を証明するため，まず完備性の定義を行っています(ここで証明を詳細に追う必要はありません)。ひるがえって自動運転車の形式検証において，安全性定理を述べてこれを証明するためには，登場する「自動運転車」という概念を定義する必要があります(図 1 下)。この定義は，自動運転車の振る舞いの正確な数学的記述でなければならず，また同時に，単純で本質を捉えていることが望ましいです(単純でないとその後の証明が大変になる)。この定義を行う営みは，たとえば自然現象を微分方程式で記述するように，数学の外にある実体や現象を数学の俎上に載せてあげるための**数理モデリング**の営みにほかなりません。

そして，**自動運転車**(および多くの CPS)の**モデリングは非常に難しい**のです。難しさの 1 つは自動運転車それ自身の複雑さです。デジタル制御と，物理システムとしての自動車，さらに物体認識用のニューラルネットなど，自動運転車は多種多様なコンポーネントを組み合わせた巨大なシステムです。この総体の動作原理を単純な数学的定義に落とし込むのは大変な作業です。また，他所から買ってきた部品は内部の動作原理が不明なブラッ

クボックスであり，そもそもモデリングが不可能です。

モデリングが難しいもう 1 つの理由は，交通システム全体が複雑なマルチエージェントシステムであることです。自動運転車の安全性証明のためには自車だけの議論だけでは不十分であり，他車や歩行者などの振る舞いも定義して，自車との関係を論じる必要があります。しかし，他車や歩行者の振る舞いのモデリングはそもそも非常に難しく，さらにこれらエージェントの相互作用の複雑さは，エージェントの数に対して指数的に増加します。

そもそも，自車が敵対的な他車に追い立てられる状況では衝突回避は不可能です。よって安全性証明では「自車が責任をもって衝突を回避すべき状況」と「衝突が起きるかもしれないが自車には責任がない状況」を峻別する必要があります。

RSS の方法論：ちょうどよい「割り切り方」

上記のモデリングの困難を乗り越えて自動運転車の安全性の数学的証明を可能にするために提案されたのが，RSS(responsibility-sensitive safety，責任感知型安全論)と呼ばれる方法論です。イスラエルの Mobileye 社の 3 人の研究者(機械学習が専門)による論文[1]に記されたこの方法論は，実は「困難に負けずモデリングをやりぬく」というものではなく，むしろ「困難なモデリングをうまく回避しながら，それでも実効的で役に立つ数学的証明を書く」という，「割り切り方」の方法論です。

RSS の要点は，図 2 のように

- 「各車が RSS ルールを遵守すれば安全」という条件付き安全性補題と，
- 「各車が RSS ルールを遵守する」というルール遵守仮定に，

安全性定理を分割することにあります。そして，数学的証明の対象を条件付き安全性に限るという「割り切り」を行います。

この(論理的に自明な)定理の分割がなぜそこまで重要なアイデアなのか，理由が 2 つあります。

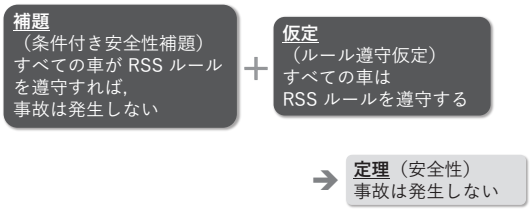

図 2―RSS における安全性定理の分割

まず 1 つは，この分割によって数学的証明が実現可能になることです。RSS ルールと呼ばれる規則の具体例をすぐ下で述べますが，これは数学的に厳密でありながら十分に単純な規則であり，その仮定のもとで安全性を数学的に証明することは不可能ではありません(論文[1]ではほぼ高校物理の範疇)。

しかしこれでは「複雑なところは全部仮定に押し込めてしまおう」という安易な逃げに聞こえるかもしれません。RSS の定理分割(図 2)のもう 1 つの重要性は，RSS ルールの遵守という仮定が社会的契約としてうまく機能するような，ちょうどよい粒度のものになっているということです。自動車をめぐる現在のエコシステムにおいて(安全基準，規制，事故責任，保険など)，「RSS ルールを責任をもって遵守せよ」という要請は非常に受け入れやすいものになっています。例を見てみましょう。

RSS ルール，最初の例

論文[1]に記されている RSS ルールの例を述べます。この RSS ルールは図 3 の運転シナリオに対するもので，前の車 $\mathrm{car}_{\mathrm{front}}$ がさまざまな振る舞いをしうる状況で，衝突回避の責任がある後ろの車 $\mathrm{car}_{\mathrm{rear}}$ が従うべき条件を述べたものです。

具体的に，論文[1]に記されたこの RSS ルールは，次のことを $\mathrm{car}_{\mathrm{rear}}$ に要請します。

- **RSS 条件**：車間距離が図 4 の式で表される安全距離を下回らないこと。
- **適切反応** proper response：上記 RSS 条件の違反が予見される場合，反応時間 ρ 以内に減速度 b_{min} でブレーキすること。

そして，この RSS ルールに対する条件付き安

図 3―一車線同方向運転シナリオ

$$\max\left(0,\ v_r\rho+\frac{1}{2}a_{max}\rho^2+\frac{(v_r+a_{max}\rho)^2}{2b_{min}}-\frac{v_f^2}{2b_{max}}\right)$$

図 4―一車線同方向運転シナリオの RSS ルールにおける安全距離

ここで v_f, v_r は car_{front}, car_{rear} の現在の速度，a_{max} は car_{rear} の最大加速度，b_{max} は car_{front} の最大減速度，b_{min} は car_{rear} の最大減速度，ρ は car_{rear} の反応時間。詳細は文献 2 を参照。

全性補題(「RSS ルールを守れば安全」，図 2 参照)は，正確には次のように述べられます。(詳細を追う必要はありません。文献 2 にもっと詳しい説明があります。)

補題(図 3 のシナリオの条件付き安全性補題)

car_{front} の加速度が常に $-b_{max}$ 以上であり，また，car_{rear} の加速度が常に a_{max} 以下であるとする。この仮定のもとで次が成り立つ：上記 RSS 条件が成り立つ任意の時点から上記適切反応を実行すれば，car_{rear} は停止するまでに car_{front} に衝突しない。

少し持って回った言い方になっていますが，本質はあくまで「(RSS 条件と適切反応からなる)RSS ルールを守れば安全である」というものです。

図 4 の RSS 条件で重要なのは，この条件が(1)数学的に定式化された厳密なものであり，かつ(2)速度などの現在の値のみに言及し，未来の値には言及していないことです。よって，この RSS 条件が成り立つかどうかは現在の時点で判定可能です。すなわち上記の条件付き安全性補題は，**未来の安全性を現在の RSS 条件に帰着**していると理解できます。

この RSS ルールが社会的契約としてちょうどよい粒度のものになっていることも，確かに見て取れます。図 4 の条件は位置・速度といった外部から測定可能な物理量と，最大加速度・減速度という基本的な性能指標を使って述べたものですから，RSS 条件の成立・不成立は客観的に判定可能です。また，RSS ルールが一般的なものであり，メーカーや車種に依存しないことも重要です。各車の内部構造などには一切立ち入ることなく適用可能であり，他の車種への適用の際には a_{max}, b_{max}, b_{min} の 3 つのパラメータの値を変えるだけです。条件付き安全性補題の証明がそう難しくないことも見て取れると思います。

RSSの課題：多様な運転シナリオへ

前節では，RSS の「割り切り方」(図 2)の具体例を見ました。ここで同時に明らかなのは，(1)RSS ルールは運転シナリオごとに策定して，その安全性を証明すべきものであること，そして(2)上記の例の運転シナリオ(図 3)は非常に単純なものにとどまっていることです。さて，**RSS の方法論は，多様な運転シナリオに適用できるのでしょうか。**

我々の最近の研究成果[3]はまさにこの問いから始まりました。マツダ株式会社と共同で行ったこの研究において，我々は特に図 5 の運転シナリオに注目しました。

この運転シナリオは自動運転車にとって，以下に述べる **ODD 逸脱対応のため特に重要**です。人間の運転者の迅速な補助を想定しない，いわゆるレベル 4 以上の自動運転では，交通状況が自動運転車の設計時想定(ODD, operational design domain)を外れて自動運転車が「もう責任もてません」となった際に，とにかく自車を安全なところに停止させる必要があります。そのために図 3 の単純なシナリオでは同一車線停止を行うわけですが，自動運転の初期の有力な ODD と考えられている高速道路では，同一車線でなく路肩停止が必須です。

同時に，図 5 が図 3 と比較して**遥かに複雑なシナリオ**であることも明らかだと思います。レーン変更は人間にとっても難しいタスクですが，このシナリオではさらに停止位置 y_{tgt} も指定されているため，「他車 1 の前と後，どちらに合流するのか？」「他車 1 の前に合流するために加速するとして，加速しすぎて停止位置 y_{tgt} をオーバーラ

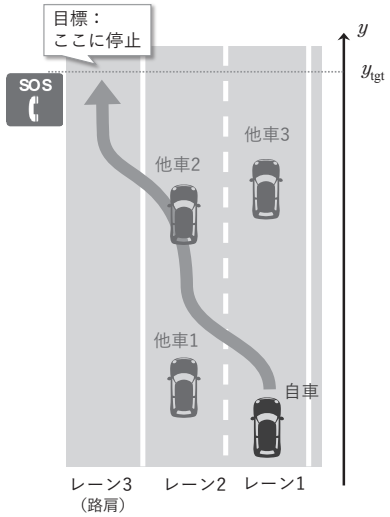

図 5―非常路肩停止シナリオ

ンしてしまわないだろうか？」といったような心配が次々に現れます。

RSS の方法論は論文[1]で示されましたが，具体的な運転シナリオへの応用は図 3 の単純なもの（およびそのバリエーション）にとどまっていました。特に図 5 のような複雑な運転シナリオへの適用はなされていませんでした。このギャップを埋め，RSS の方法論を多様な運転シナリオに適用して，自動運転車の安全性保証と社会受容・本格展開を進めようとしたのが我々の最近の成果[3]です。そこで用いたのが，証明を扱う数学である**論理学**と，論理学をソフトウェアの形式検証に使ってきた**ソフトウェア科学**，両者の学術的蓄積です。

以下，論文[3]の技術的内容の説明のため，関連する論理学とソフトウェア科学の基礎的内容を少し紹介します。抽象度が高く読みにくい場合は遠慮なく読み飛ばしてもらって，最後の自動運転の将来展望の話で合流してください。

論理学における証明の形式化

RSS の方法論は数学的証明に関するものですから，その実運用のためには，証明を扱う数学である論理学の活用が必要です。しかし我々の論文[3]以前は RSS の論理学的解析を行う研究は非常に少なく，この不在を埋める必要がありました。具体的には，RSS における証明の形式化を行いました。

証明の形式化 formalization とは何でしょうか？　計算機（コンピュータ）の出現より古くから存在する由緒正しい概念ですが，ここでは，

証明の形式化

＝証明を計算機で扱えるように記号化すること

と説明することにします。形式化された証明は計算機の中に書かれているもので，特に，証明中の各推論ステップが論理的に正しいかどうかを確認する**証明チェック** proof check を計算機が行うことができます。逆に，形式化されていない証明とは，人間が（日本語などの自然言語も使って）紙にペンで書いた証明です。

自動運転車のみならずソフトウェアやハードウェアなど，**形式検証一般において証明の形式化は必須**です（なので「形式」検証）。形式検証で書く証明は，煩雑な場合分けが多く，ダラダラ長くて盛り上がりに欠けるものです（「必殺のアイデアが決まって気持ちいい！」というカタルシスがない）。人間がこのような証明を書くと多くの誤りを犯します。また，社会に対する説明可能性・アカウンタビリティの面でも，安全性の証拠が 1000 ページの手書きのノートという状況は望ましくありません。逆に，形式化した証明は計算機上のデータであるため，証明チェックだけでなく翻訳や可視化，改訂など，さまざまな計算機的処理のために将来にわたって使い回せる，いわば人類の財産です。

証明の形式化のために，証明を行う「環境」としての**形式論理体系**をまず定義します。形式論理体系は次の 2 つの構成要素からなります。

- **証明言語**：どのような記号列を証明および定理・補題の言明において用いてよいのかを定める形式言語
- **導出規則**：証明中に用いることのできる論理的推論ステップを，記号列の変換として定める書き換え規則

プログラミング言語のようなものとイメージしてもらってもよいかもしれません（証明言語＝プログ

ラミング言語，導出規則＝プログラムの実行系）。重要なのは，人間が証明を書く際と違って（ノートに何かを書きながら数学的意味を想像している），形式論理体系は意味を考えない純粋な記号操作の体系であることです。記号操作であるため計算機実装が可能であり，機械的な証明チェックが可能です。

形式論理体系の例：ホーア論理

形式論理体系の例として，プログラムの形式検証のためのホーア論理を紹介します。我々が論文[3]で用いた形式論理体系の原型であり，ソフトウェア科学の基礎となっている由緒正しい論理体系です。詳細は教科書[4]を参照してください。

ホーア論理の証明の例を図6に示します。以下，この例を説明していきます。

ホーア論理は，よくある命令形プログラムの振る舞いについて，その性質を証明するための形式論理体系です。ホーア論理においては，

$$\{A\}\, c\, \{B\}$$

の形の3つ組（ホーア3つ組）を順次導いていきます。$\{A\}\, c\, \{B\}$ は(1)事前条件 A のもとで(2)プログラム c を実行すると(3)プログラム実行の終了後事後条件 B が成り立つ，と主張するものです。

たとえば図6の最後の結論では，(1)事前条件 $x-1\leq 2$（つまり $x\leq 3$）のもとで(2)プログラム $y:=x;\ y:=y-1$ を実行すると（x の値を y に代入し，さらに1減らす）(3)事後条件 $y\leq 2$ が成り立つ，と主張しています。確かに成り立ちそうです。

ホーア論理の証明言語を正確に述べると次のようになります。事前条件・事後条件は，変数と整数が現れる不等式，およびその論理結合（$\wedge$，$\vee$，$\neg$ など）で得られる論理式です。プログラムは，代入命令（$y:=x$ など）およびそれらの逐次合成（セミコロン ; でつなげる），if分岐，whileループです。ここではとりあえず図6の例で考えれば大丈夫です。

形式論理体系のもう1つの構成要素である導出規則についてはどうでしょうか。今回の例に関連するもののみを抜粋して図7に示します。

最初の規則（:=）は横線の上が空なので（仮定がない），下のホーア3つ組をいきなり導出してよいと言っています。ここで $A[a/x]$ は，論理式 A において変数 x が現れているところをすべて式 a に置換した論理式を表します。実際，図6の証明では(1)，(2)のホーア3つ組がこの規則で導かれていますね。

$$\begin{array}{lll}
(1) & \{x-1\leq 2\} \quad y:=x & \{y-1\leq 2\} \\
 & & \text{by Rule } (:=) \\
(2) & \{y-1\leq 2\} \quad y:=y-1 & \{y\leq 2\} \\
 & & \text{by Rule } (:=) \\
(3) & \{x-1\leq 2\} \quad y:=x;\ y:=y-1 & \{y\leq 2\} \\
 & & \text{by (1), (2), and Rule } (;)
\end{array}$$

図6―ホーア論理の証明の例

$$\frac{}{\{A[a/x]\}\, x:=a\, \{A\}}\ (:=) \qquad \frac{\{A\}\, c\, \{B\} \quad \{B\}\, d\, \{C\}}{\{A\}\, c; d\, \{C\}}\ (;)$$

図7―ホーア論理の導出規則の抜粋

図6の(2)の導出をもう少し説明します。ここでは「プログラム $y:=y-1$（y の値を1減）を実行したあと $y\leq 2$ になるためには，もともと y はどのような条件を満たさなければならないだろう？」というのが問題です。人間には「事前条件は $y\leq 3$」と簡単にわかるのですが，規則（:=）では，事後条件 $y\leq 2$ における y の現れを $y-1$ に置換することで，同値な事前条件 $y-1\leq 2$ を得ています。これが形式論理体系における記号操作による推論の例です。

図7の2つ目の規則（;）では，横線の上の2つの仮定（導出済みのホーア3つ組）を組み合わせて，プログラムの逐次合成 $c; d$ についてのホーア3つ組を導出しています。図6の例では(3)の導出にこの規則を用いています。ここでもやはり規則の適用は機械的な記号操作であり，「条件 B が共有されている」という記号列の一致を確認しただけです。

以上，証明の形式化，すなわち記号操作による機械的推論の例として，ホーア論理を紹介しました。計算機の中で証明を書いて，その正当性（図7のような導出規則に従っているかどうか）を計算機でチェックすることのイメージができたかと思います。（ここまで，形式論理体系の意味の話を割愛しています。興味が

あれば文献 4 を御覧ください。）

ホーア論理の拡張によるRSSの形式化と適用範囲の拡大

論文[3]では，RSS の形式化のためにホーア論理を拡張して，新たな形式論理体系 dFHL（differential Floyd-Hoare Logic）を提案しました。dFHL はホーア論理を次の 2 つの機能で拡張したものです：（1）物理ダイナミクスのための**微分方程式**，（2）事前条件・事後条件に加えて，プログラム実行中常に保証されるべき**安全性条件の追加**（よってホーア 3 つ組でなく，安全性条件 S を加えたホーア 4 つ組 $\{A\}\,c\,\{B\}:S$ を導出していく）。

dFHL による RSS の形式化のデモンストレーションとして，我々は論文[3]で図 5 の非常路肩停止シナリオに取り組み，RSS ルールを導出しました。この成功により，RSS の適用範囲を図 3 のような単純な運転シナリオから，非常路肩停止のような重要かつ複雑な運転シナリオへ拡大しました。

この成果の要点は 2 つあります。まず 1 つは，RSS ルールによって衝突回避だけでなく**目標達成も保証**できるようになったことです。ホーア 4 つ組 $\{A\}\,c\,\{B\}:S$ は RSS の文脈で次のように解釈できます：（1）事前条件 A が RSS 条件，（2）プログラム c が適切反応，（3）安全性条件 S が衝突回避の要請，そして（4）事後条件 B が運転シナリオの目標。たとえば非常路肩停止シナリオでは，事後条件 B を「路肩の指定位置に停止していること」とすれば，シナリオ目標の達成を保証できます。

もう 1 つの要点は，dFHL による **RSS ルールの逐次導出**です。ホーア論理の大きな特徴に，プログラムの各部分に対する証明を行ったあと結果を組み合わせるという逐次導出があります（図 6 参照，代入命令に対する証明 2 つを推論規則（;）で組み合わせている）。我々の非常路肩停止シナリオの解析においても同様に，運転シナリオ全体を「レーン変更の準備」「レーン変更」「路肩に移動」「路肩で停止」の 4 つのサブシナリオに分割し，サブシナリオそれぞれについて証明を行ったあと，その結果を dFHL の推論規則（;）で組み合わせるという逐次導出を行っています。この逐次導出によって，**複雑な運転シナリオの取り扱いを可能**にしています。

RSSの社会応用と将来展望

しばらく論理学の技術的内容をお話ししてきました。この先は自動運転という応用の大局的視点に戻って，RSS の社会応用について論じたいと思います（技術的内容を読み飛ばした方，戻ってきてください！）。

RSS の方法論は安全性定理を補題と仮定に分割し，補題を数学的に厳密に証明する一方，仮定を社会的契約として各車に要請する，というものでした（図 2）。その応用は単なる安全性保証にとどまらず，次のようなものが考えられています。

- **事故の責任所在の特定**：条件付き安全性補題により，事故が起こった際には RSS ルールに準拠していない車がいたことになります。その車に責任があると特定できます。
- **自動運転の安全性規格**：RSS ルールは規格として使いやすく，さらに数学的に厳密で安全性証明済みというお墨付きまであります。IEEE などですでに試みがあります。
- **保険料率計算**：損害保険会社は自動車業界における大きなステークホルダーです。RSS はリスク軽減の明確な証拠になります。

各車による RSS ルールの遵守（ルール遵守仮定，図 2）について，どう保証するか不思議に思われるかもしれません。しかしこれにも明快な答えがあります。

図 8 を見てください。この**安全アーキテクチャ**は自動車や航空機など人命に関わるシステムで広く用いられている多重性確保の仕組みです。複雑な制御器 AC と，安全性のみを追求する単純な制御器 BC とを，DM が適切に切り替えることで，「安全性に余裕があれば AC で性能を追求，切迫

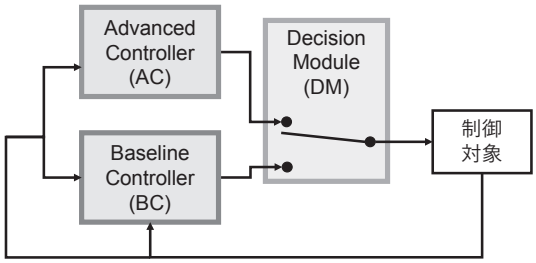

図 8―安全アーキテクチャ

してきたら BC に切り替えて安全性をとにかく保証」という制御を行います。ポイントは，AC がブラックボックスであっても，BC と DM が「安全ガード」となり，システム全体の安全性を保証できることです。

RSS ルールは安全アーキテクチャとしてそのまま実装可能です。すなわち，各メーカーがこだわって作った AC に対し，適切反応を BC，RSS 条件を DM として追加することで，「RSS 条件の違反が予見されたら制御を BC に切り替える」という制御が実現し，ルール遵守仮定を充足できます。

結び：論理学の社会応用の新たな形

以上，自動運転車の安全性の数学的証明について，RSS という方法論と論理学の役割について説明しました。RSS の割り切り方は論理学の研究をしてきた私にとって非常に新鮮なものです。数学的証明は確かな事実を積み上げるものだと思ってきましたが，RSS の証明はそれとは違って，数学的に巨大な仮定(ルール遵守仮定)のうえに積み上げるものです。さらにこの強い仮定を「現在の自動車業界的に要請しやすい」という社会的・応用駆動的尺度で正当化する，というのも新鮮でした。このような実際的な仮定のおきかたを他の応用分野でも探索すれば，論理学の社会応用は大きく広がるのではと夢想します。

また，数学的証明の社会的役割についても考えさせられます。RSS の証明は，絶対不変の真理を樹立するものというよりは，安全性のためにどの仮定をどのように用いたかという議論の精密なドキュメントと思ったほうがしっくりきます。自動運転車の安全性への終わりなき社会的取り組みにおける，説明・議論のための重要なメディア，というわけです。

より一般に，次々現れる AI などの ICT 技術について，技術をブラックボックスとせず人間の管理下におくための社会的取り組みにおいて，議論の枠組みとしての数学的証明の重要性は今後ますます高まると予想します。論理学者としては強い責任を感じる一方，この新応用から生まれる理論研究の飛躍の可能性に大きな興奮を感じるところです。

文献

1―S. Shalev-Shwartz et al.: arXiv preprint, arxiv.org/abs/1708.06374(2017)

2―I. Hasuo: arXiv preprint, arxiv.org/abs/2206.03418(2022)

3―I. Hasuo et al.: IEEE Trans. Intelligent Vehicles, to appear (2022). プレプリント版が arXiv から入手可能 arxiv.org/abs/2207.02387

4―G. Winskel: *The Formal Semantics of Programming Languages*. MIT Press (1993). 和訳(末永幸平監訳): プログラミング言語の形式的意味論入門. 丸善出版(2023)

蓮尾一郎 はすお いちろう

国立情報学研究所教授，総合研究大学院大学教授。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。専門はソフトウェア科学・数理論理学。情報学における数学的構造，特に圏論的な構造に興味があります。数学的抽象論ならではの情報学的ブレイクスルーを目指して研究しています。

https://group-mmm.org/~ichiro/