研　究　報　告　書

「ハイブリッドシステムのための超準プログラミング言語理論を用いた形式手法」

研究タイプ：通常型
研究期間：　平成27年10月～平成31年3月
研　究　者：　末永幸平

1.　研究のねらい

情報通信技術の発展により，現代では多くのシステムが計算機を内蔵し，あるいは計算機に接続されている．このようなシステムは数学的には連続量の遷移と離散量の遷移とが組み合わさったハイブリッドシステムと呼ばれるクラスに属する．このようなヘテロ化したシステムについて，サイバーフィジカルシステムへの興味の高まりもあり，その挙動の正確な理解とシステムの正しさの検証が重要な社会的課題となっている．ハイブリッドシステムは連続量のみ，あるいは離散量のみからなるシステムよりも複雑であることが知られており，その挙動の正確な理解は学術的にも興味ある課題である．

本課題においては，ソフトウェア科学の分野で研究が進んでいる形式手法をハイブリッドシステムに適用可能にすることによって，ハイブリッドシステムの挙動の理解を可能とし，それらが意図通りに動作することを検証可能にすることをねらう．主要なアイデアの一つは，ハイブリッドシステムのモデリング言語として，提案者が 2011 年以降研究を進めている超準プログラミング言語を用いることである．これは無限小値を表す定数値で拡張されたプログラミング言語であり，その意味論は超準解析を用いて与えられている．無限小定数によってハイブリッドシステムに存在する連続的遷移を離散的遷移として表現することで，離散量のみからなるソフトウェアを対象としている既存の形式検証手法を，連続量を含むハイブリッドシステムにそのまま適用することが理論上は可能である．本課題においては，このアイデアを実用的に結実させるための道筋を立てる研究を行う．そのためにソフトウェア科学の分野で研究されている形式検証手法を，超準プログラミング言語でモデリングされたハイブリッドシステムに実際に適用し，実用的なハイブリッドシステムの検証までのギャップを埋めることを目標とする．

現在提案されている形式検証手法のほとんどが不変条件，すなわちプログラム中のある箇所において常に成り立っている条件の発見手法に帰着されることが知られているため，本提案では形式検証手法のうち，不変条件発見に基づく形式検証手法にフォーカスを絞る．不変条件の発見手法はソフトウェア科学の草創期から研究が続いており，近年特に目覚ましい発展が見られている．

2.　研究成果

（1）概要

IC3/PDR を用いたハイブリッドシステムの形式検証手法（[研究成果 D]）が本研究のメインの成果である．近年ソフトウェア検証において注目を集めている IC3/PDR と呼ばれる手法をハイブリッドシステムに適用可能なように拡張し，その正しさを証明し，実装・実験を行った．検証アルゴリズム中では，超準プログラムの検証手法に着想を得た手続きを用いている．本研究成果は現在査読付き国際会議に論文を投稿中であり，また特許出願を検討している．

その他に，既存のソフトウェア検証手法を超準プログラム検証に適用するために強化することを志向する過程で成果がいくつか得られた．多項式 p を用いて p=0 という形に書ける不変条件（代数的不変条件）の高速な生成手法を提案した（[研究成果 A]）．実験においては最大で 10 倍程度の高速化が観察された．本研究成果は査読付き国際会議プロシーディングスと査読付き国際ジャーナルで出版され [論文 1]，2 件の国内特許出願と 1 件の PCT 出願を行った．

また，Craig 補間と呼ばれる，論理式の充足不能性を証明する論理式を発見する手続きを改良する研究を行った（[研究成果 B]）．Craig 補間の発見はモデル検査において重要な技術であるが，従来手法では論理式のクラスが過度に制限されている手法か，制限されていない場合でも重要な入力について Craig 補間の発見に失敗する手法のみが提案されていた．本成果ではこれらの制限を緩和する改良を行った．本研究成果は査読付き国際会議プロシーディングス [論文 2] で出版され，1 件の国内特許出願を行った．

さらに，検証に必要な不変条件を多くの場合に簡略化することのできるプログラム変換を提案した（[研究成果 C]）．このプログラム変換はコンパイラにおける最適化でよく研究されているループ融合変換に着想を得たものであるが，従来のループ融合変換では扱えるプログラムに制限が多かった．本成果ではこの制限を緩和したプログラム変換を提案し，実験でその効果を確認した．本研究成果は査読付き国際会議プロシーディングス [論文 4] で出版され，1 件の国内特許出願と 1 件の PCT 出願を行った．

さらに研究に付随する成果（[研究成果 E]）として，超準プログラミング言語を新たに関数型言語に適用する研究 [論文 3]，深層学習を用いて定理証明を行う研究 [論文 5] をそれぞれ国際会議プロシーディングスで出版した．

（2）詳細

[研究成果 A] 代数的不変条件の生成手法: 代数的不変条件，すなわち多項式 p を用いて p = 0 と書ける不変条件を生成する手法を研究した．代数的不変条件を計算するための既存手法として，テンプレート法と呼ばれる手法が提案されている．テンプレート法は，未定係数を含む多項式テンプレートを生成し，この多項式テンプレートが不変条件になるために未定係数がどのような制約を満たすべきかを計算した上で，制約をソルバで解消する手法である．制約の生成においては，代数的不変条件を構成する多項式全体の集合が多項式環におけるイデアルをなすことに着目して，その有限個の生成元を Buchberger アルゴリズムによって計算する手法があった．しかし，この手法はシステムが大規模になるとスケールしにくいという問題が知られていた．

この問題点を解決するために，プログラムに軽量な事前解析を行うことでテンプレートのサイズを縮減する手法を提案した．研究の核となるアイデアは，本研究によって明らかとなった「代数的不変条件が存在するならば，次数付き斉次多項式によって構成される不変条件が存在する」という性質である．この性質は，物理学の分野で用いられる「量の次元」の概念をプログラム解析に導入することに対応しており，この性質を用いて代数的不変条件の探索範囲を絞ることに成功した．

具体的にはこのアイデアを数学的に厳密な形で定式化し，その正しさの証明を行った．また，論文執筆時点で最も効率的に不変条件を求めることのできた Cachera et al. [Cachera et al. SAS 2012] の FastInd とよばれる手法に対し，「量の次元」の解析を事前に行う拡張を行い，解析で得られた情報を用いて不変条件の探索範囲を狭めるアルゴリズムを定式化した．「量の次元」の解析には，Kennedy の次元型システムを用いた自動解析を用いた．定式化したアルゴリズムを実装し Cachera et al. の用いたベンチマークで実験したところ，最大で 10 倍程度の性能改善が見られた（図 1）．

実験結果

FastInd が用いたベンチマークで実行時間 (ms) を比較 (抜粋, 青: FastInd, 赤: 本技術)

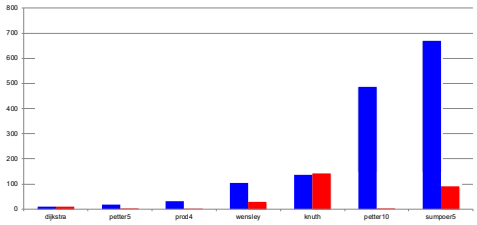


図 1: 提案手法の性能

この成果を論文としてまとめてプログラム解析に関する国際会議 SAS 2014 およびジャーナル Theoretical Computer Science にて発表した [論文 1]．また，京都大学を通じて 2 件の国内特許出願を行い，JST の支援を得て PCT 出願を行った．

[研究成果 B] 半代数的な Craig 補間の生成手法: 多くのプログラム検証手法，特にモデル検査と呼ばれる手法においては，検証対象のプログラム c をある有限な述語集合 S の真偽値に着目して抽象化し，抽象化されたプログラムを全数探索することで不変条件を求める述語抽象化と呼ばれる手法がよく用いられる．この述語集合 S を自動的に求める手法として反例駆動抽象精緻化（CEGAR）と呼ばれる手法が多く用いられる．この手法においては，検証の開始時に述語集合 S を空集合に初期化し，現在の述語集合から得られる c の抽象化において仕様が成り立たないようなプログラムの実行 s が存在するかをチェックする．s が存在し，かつ s が真の反例ではなく抽象化が粗いために発見された偽反例である場合には，S に追加すべき新たな述語を s を解析して発見し，拡大された S で再び c の抽象化を求める．この手続きを検証が成功するか真のバグが見つかるまで繰り返す．

偽反例 s から追加する述語を発見するためのヒューリスティクスとしてよく用いられるのが Craig 補間と呼ばれる手法である．一般に論理式 A と B の Craig 補間とは，論理式 A∧B が充足不能であるときに定義される以下を満たす論理式 ϕ である: (1) A⇒ ϕ が恒真，(2) ϕ ∧ B が充足不能，(3)ϕに現れる変数は A と B の両方に現れる変数のみ．モデル検査の文脈においては，実行 s が偽反例である場合に「s がプログラム c の実行であり，かつ仕様を満たさない」という意味の A∧B という形をした充足不能な論理式を構成することができる．また，A と B の Craig 補間を S に追加することが検証に有用であることが経験的に知られている．

これまでに提案された Craig 補間を求めるための手法のほとんどは A と B が線形制約の場合しか扱えず，かつ求まる Craig 補間も線形制約として表現可能なものに限られていた．ハイブリッドシステムにおいては，A，B，求めるべき Craig 補間がそれぞれ線形制約では表現できない場合があり，これらの制限は望ましくない．これらの制限のない手法としては Dai et al. [Dai et al. CAV 2013] による手法があるが，この手法では A と B の形によっては Craig

補間を求められない場合があった.

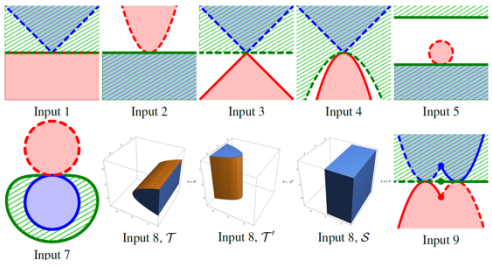
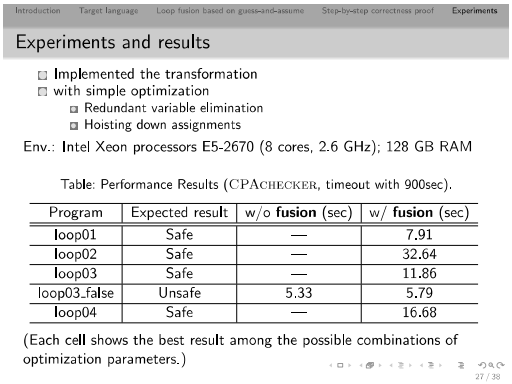


図 2: 提案手法で求められる Craig 補間. A と B は平面上の青と赤で示された領域. 緑で示された領域が求められた Craig 補間.

この問題を解決するために, 本研究項目では Dai et al. の手法の改良を行った. Dai et al. の手法は A と B が半代数的制約, すなわち多項式の不等式からなる制約として与えられたときに, $A \wedge B$ が充足不能であることの certificate と呼ばれる多項式を求める問題を半正定値計画問題に還元し, この問題をソルバによって解き, 得られた certificate から Craig 補間を求める. この certificate の構成法を工夫することで Dai et al. の手法では求められない Craig 補間が求められることを確認した (図 2).

研究内容はプログラミング言語に関する国際会議 APLAS 2017 において発表し [論文 2], また東京大学と京都大学が権利者となって特許出願を行った.

[研究成果 C] 軽量なプログラム解析を可能とするプログラム変換: 「研究のねらい」の節で述べた通り, プログラム検証においては不変条件を求めることが重要であるが, プログラムの形によっては不変条件が複雑となるために, 軽量なプログラム解析が困難となる場合が多い. 特に複数のループ文が逐次に実行されるプログラムにおいては, 比較的シンプルなプログラムであっても, 不変条件が線形制約で表現できないために最先端の検証器が検証に失敗することが見られる. 逐次処理はハイブリッドシステムにおいても多く見られるため, ハイブリッドシステムの検証手法においてもこの問題を扱う必要がある.


Introduction Target language Loop fusion based on guess-and-assume Step-by-step correctness proof Experiments

Experiments and results

- Implemented the transformation
- with simple optimization
  - Redundant variable elimination
  - Hoisting down assignments

Env.: Intel Xeon processors E5-2670 (8 cores, 2.6 GHz); 128 GB RAM

Table: Performance Results (CPAchecker, timeout with 900sec).

Program	Expected result	w/o **fusion** (sec)	w/ **fusion** (sec)
loop01	Safe	—	7.91
loop02	Safe	—	32.64
loop03	Safe	—	11.86
loop03_false	Unsafe	5.33	5.79
loop04	Safe	—	16.68

(Each cell shows the best result among the possible combinations of optimization parameters.)

27 / 38

図 3: 実験結果. w/o fusion の列は本手法を適用しなかった場合, w/ fusion の列は本手法を適用した場合の, 検証にかかった時間を示す. 時間が示されていないセルは 900 秒経っても検証ができなかったことを表す.

本研究項目においては, 逐次に実行される複数のループ文を含むプログラムを, 単一のループ文を含むプログラムに変換する手法を提案した. 本変換はコンパイラにおける最適化で研究されているループ融合変換に着想を得ている. しかしながら, コンパイラの文脈におけるループ融合変換では, 融合するループ文が共通の変数を有する場合には, 融合前のプログラムと融合後のプログラムの意味を同一にするという要請から, 一般に容易には融合を行うことができない. 本研究においては, プログラム検証においては融合後のプログラムの意味を同一にする必要は必ずしもなく, 安全性が成り立つかどうかのみが保存されればよいという点に着目し, guess and assume という軽量なループ融合変換手法を提案した. この手法を実装し, 最先端のモデル検査器である CPAChecker と SeaHorn でいくつかの例を用いて実験したところ, 融合前

はすべてのプログラムの検証に失敗したのに対し，融合後は現実的な時間で検証に成功することに確認した．

研究内容はプログラミング言語に関する国際会議 PEPM 2018 で発表し［論文 4］，京都大学が権利者となって国内特許出願を行い，JST の支援を得て PCT 出願を行った．

[研究成果 D] ハイブリッドシステムのためのモデル検査アルゴリズム: 本 研究項目においては，ソフトウエア検証において近年大きな成果を上げている仕様駆動到達性検査（IC3/PDR）と呼ばれるモデル検査手法をハイブリッドシステムに拡張するための理論基盤を確立した．

本研究成果を論文にまとめ，現在投稿中である．特許出願を検討しているため，本手法の詳細な内容については非公開とする．

[研究成果 E] 付随するその他の成果: その他の付随して得られた成果を記す．これまでの超準プログラミング言語理論は命令形言語とデータフロー言語と呼ばれるプログラミング言語についてのみ提案されていた．プログラミング言語に関する国際会議 APLAS 2017 にて発表した［論文 3］においては，超準プログラミング言語の関数型言語への拡張を提案した．

プログラム検証においては，自動定理証明器を用いることが多い．自動定理証明は一般には決定不能な問題であり，既存の証明器はヒューリスティクスを用いて証明を行っている．このヒューリスティクスに機械学習を用いる手法を研究した．最小論理と呼ばれる論理のための深層学習を用いた自動証明器を教師付き学習を用いて訓練しテストしたところ，テストに用いた命題の 93%について正しい証明を生成できることを確認した．成果をまとめてプログラミング言語に関する国際会議 APLAS 2018 にて発表した［論文 5］．

3． 今後の展開

学術面においては，本課題のメインの成果である［研究成果 D］をさらに実用的な検証手法につなげるための研究を行いたいと考えている．現在の［研究成果 D］の検証アルゴリズムは，その過程である論理式の充足不能性を証明するための Craig 補間条件や，ある論理式が与えられた微分方程式に沿った連続時間ダイナミクスで不変であることの証明を要求する．現状のプロトタイプ実装においては，これらの作業を非常に限られた場合のみ自動的に行うように実装されており，その他の場合においては人手で情報を与えることが必要となっている．これらの作業のうち自動的に行える範囲を，［研究成果 A,B］を適用することで広げていくことを考えている．また，検証対象のシステム自体を[研究成果 C］の結果を応用して検証が容易なシステムに変換する手法を研究する．

今回得られた研究成果においては，可能なものについては特許出願を行っている．今後これらの知財を用いて研究を社会実装することを目指したい．

4． 自己評価

研究目的の達成状況について: IC3/PDR を用いたハイブリッドシステム検証手法である［研究成果 D］が本研究のメインの成果である．この検証手法は，当初の構想で想定していたような，ハイブリッドシステム中の連続時間ダイナミクスをすべて無限小定数によって離散化した上で検証するものではなく，連続時間ダイナミクスの微分方程式によるモデリングを用いつつ検証を行う．当

初想定していた手法は，理論上は安全性の検証が可能であるものの，現実的には離散化の結果得られる超準プログラムにおける「連続性」の性質を利用することが難しくなり，現状のソフトウェア検証技術を超準プログラムに適用して有用な検証器を得ることがやや困難であった．そのため，[研究成果 D] では微分方程式でモデル化された連続時間ダイナミクスの情報を利用するように既存のモデル検査手法である IC3/PDR を拡張しつつ，その過程で用いる手続きにおいて超準プログラムの検証に着想を得た手法を用いている．当初構想とは異なる形の成果となったものの「ハイブリッドシステム検証に超準プログラミング言語理論を適用する」という当初の目的の成果としては一定の結果が出たと考えている．

[研究成果 A,B,C] は，当初の構想通りに既存のソフトウェア検証手法を超準プログラムに適用するために強化することを志向する過程で得られた成果である．上述の通り [研究成果 D] においては超準プログラムを直接的に扱うことは行わなかったものの，[研究成果 A,B,C] は，それぞれの成果において査読付き論文が出版されていること，京都大学と東京大学が本研究成果の特許を受ける権利を承継して特許出願が行われたこと，PCT 出願においてヒアリングの結果 JST からの支援が得られたことから分かる通り，ソフトウェア検証のための要素技術としては重要な成果をあげることができたと考えている．また，「今後の展開」の項にも記した通り，これらの成果を [研究成果 D] に統合することで，ハイブリッドシステムのためのよりよい形式検証手法を構築することが可能であると考えている．

以上をまとめると，当初の目論見とは違う形に研究が進んだが，最終的にはある程度満足の行く研究成果が得られたと考えている．

研究の進め方について: 研究体制を組むにあたっては学生 RA を雇用して研究補助業務に従事させることにより効率的な推進を心がけた．また，プロジェクト中で研究的要素のない作業を業者に委託することにより，研究を効率的に推進した．研究費の執行にあたっては，当初計画通りの執行を重視しつつ，研究の進展状況に応じて JST から認められた範囲内での費目間の転用等を活用することにより，柔軟かつ効率的な執行を心がけた．

研究成果の科学技術及び社会・経済への波及効果: IC3/PDR のハイブリッドシステムへの適用 ([研究成果 D]) は，この手法のソフトウェア検証での成功にも関わらず，これまでに提案されていなかった．今後よりよいハイブリッドシステム検証手法につながるポテンシャルは大きいと考えている．また，[研究成果 A,B,C] は，これまでにソフトウェア検証手法の研究ではあまり重視されていなかった，線形でない不変条件を要求するプログラムの検証を前進させる成果であると評価している．

研究期間内に産業的なインパクトのあるハイブリッドシステム検証手法につなげることを目指していたが，残念ながら現状では直ちに産業利用が可能な成果には至っていない．しかしながら，今回の研究で得られた成果，特に IC3/PDR とハイブリッドシステムを橋渡しするという [研究成果 D] の成果を今後の産業利用につなげられるように，研究期間後も研究を継続したいと考えている．

5. 主な研究成果リスト

(1) 論文(原著論文)発表

1. Kensuke Kojima, Minoru Kinoshita, Kohei Suenaga: Generalized homogeneous polynomials

for efficient template-based nonlinear invariant synthesis. Theor. Comput. Sci. 747: 33-47 (2018)
2. Takamasa Okudono, Yuki Nishida, Kensuke Kojima, Kohei Suenaga, Kengo Kido, Ichiro Hasuo: Sharper and Simpler Nonlinear Interpolants for Program Verification. Proc. of APLAS 2017: 514-533 (2017)
3. Hirofumi Nakamura, Kensuke Kojima, Kohei Suenaga, Atsushi Igarashi: A Nonstandard Functional Programming Language. Proc. of APLAS 2017: 514-533 (2017)
4. Akifumi Imanishi, Kohei Suenaga, Atsushi Igarashi: A Guess-and-Assume Approach to Loop Fusion for Program Verification. Proc. of PEPM 2018: 2-14 (2018)
5. Taro Sekiyama, Kohei Suenaga: Automated Proof Synthesis for the Minimal Propositional Logic with Deep Neural Networks. Proc. of APLAS 2018: 309-328 (2018)

（2）特許出願

研究期間累積件数：6 件（公開前の出願件名については件数のみ記載）

1.
発　明　者：末永 幸平, 樹下 稔, 小島 健介
発明の名称： 不変条件生成装置、コンピュータプログラム、不変条件生成方法、プログラムコード製造方法
出　願　人：京都大学
出　願　日：2016/02/01
出 願 番 号： 特願 2016-017441

2.
発　明　者：末永 幸平, 樹下 稔, 小島 健介
発明の名称： 不変条件生成装置、コンピュータプログラム、不変条件生成方法、プログラムコード製造方法
出　願　人：京都大学
出　願　日：2016/02/01
出 願 番 号： 特願 2016-017419

（3）その他の成果（主要な学会発表、受賞、著作物、プレスリリース等）

学会等での発表:

- A Guess-and-Assume Approach to Loop Fusion for Program Verification. Akifumi Imanishi, Kohei Suenaga, and Atsushi Igarashi. PEPM 2018　2018 年 1 月 7 日（口頭・査読付き）
- Sharper and Simpler Nonlinear Interpolants for Program Verification. Takamasa Okudono, Yuki Nishida, Kensuke Kojima, Kohei Suenaga, Kengo Kido and Ichiro Hasuo. APLAS 2017　2017 年 11 月 27 日（口頭・査読付き）
- A Nonstandard Functional Programming Language. Hirofumi Nakamura, Kensuke Kojima, Kohei Suenaga and Atsushi Igarashi. APLAS 2017　2017 年 11 月 27 日（口頭・査読付き）
- Generalized Homogeneous Polynomials for Efficient Template-Based Nonlinear Invariant

Synthesis. 末永 幸平. SAS 2016　2016 年 9 月 8 日（口頭・査読付き）

その他の講演・アウトリーチ活動等:

- 研究取材記事:「映像と記事で描く AI 時代と科学研究の今：研究への情熱」JST Web ページにて取材記事と動画の掲載. 2018 年 2 月 9 日.
  http://www.jst.go.jp/kisoken/jyonetsu/interview/h29/suenaga.html