様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

令和　２ 年　 ６ 月　 ８ 日現在

機関番号：１３９０１
研究種目：挑戦的研究（萌芽）
研究期間：2017～2019
課題番号：１７Ｋ１９９６９

研究課題名（和文）連続時間領域における実時間プログラムの離散実行モデル

研究課題名（英文）A discrete execution model of dense-timed programs

研究代表者
結縁　祥治（YUEN, SHOJI）

名古屋大学・情報学研究科・教授

研究者番号：７０２３０６１２

交付決定額（研究期間全体）：（直接経費）　　4,800,000円

研究成果の概要（和文）：本研究では、連続時間で設計された振る舞いを離散時間環境で実行する際に生じる問題について研究を行った。CPSなどのソフトウェアでは、時間経過に応じて連続的に状態が変化するほかに、一定の条件が成立した場合には、離散的に遷移が発生する。このような振舞いはハイブリッドオートマトンでモデル化され、プログラムで実現する。しかし、プログラムは離散的なサンプリングに基づいて実行されるため、振舞いが異なる場合がある。プログラムとして関数型言語Haskellの領域特化言語であるYampaプログラムにおいて、時間オートマトンとの合成で離散動作意味をモデル化してその振舞いを検証する枠組みを提案した。

研究成果の学術的意義や社会的意義
連続的に動作するシステムに対して設計されたプログラムが実行環境が提供する離散的なサンプルのもとでどのように振舞うかを形式的に定義することで、プログラムと実行環境による複合的な問題点を形式的に定義することができるようになった。プログラムと実行環境とは個別には正しい振舞いをする場合であっても、組み合わせによって生じる不具合についての解析が可能になる。実時間性を持つプログラムが動作する条件を明確にすることによって、CPSなどにおいて厳密な安全性が要求される場面での信頼性を向上させることが可能になる。

研究成果の概要（英文）：We studied the formal treatment of the gap between continuous behaviour and discrete behaviour of software with dense-time behaviour, as seen in CPS (cyber-physical-Systems). Software for CPS implements the behaviour of hybrid automata, where discrete transitions occur along with continuous transitions. A discrete transition occurs if a specific condition holds over continuous variables. Since this hybrid behaviour cannot be directly implemented by software running on computers, a program runs discretely based on sampling. It is possible for the sampling to miss the critical moment to change the behaviour. We study Yampa programs, that is a DSL of Haskell describing hybrid automata. We modelled the discrete behaviour of a Yampa program that approximates the continuous behaviour by composing a hybrid automaton with a timed automaton, where the timed automata sample the behaviour of the hybrid automaton. We applied the model to Uppaal to check the property of a Yampa program.

研究分野：プログラミング言語

キーワード：実時間性　連続時間　離散時間環境　時間オートマトン　ハイブリッドオートマトン

1. 研究開始当初の背景

計算機が社会のあらゆる場面で用いられるようになり、実世界のシステム制御においてソフトウェアが安全性において重要な意味を持つ。CPS(サイバーフィジカルシステム) の制御ソフトウェアでは実行中における環境との入出力でシステムの振舞いが決定する。システムの安全性、信頼性を向上させるためには、実際に動作するメカニズムの信頼性とともに制御ソフトウェアの信頼性を向上させることが不可欠である。このようなソフトウェアにおける振舞いの正しさは、従来のソフトウェアシステムとは異なり、実行環境に依存して決定する。特に、実行環境との時間をはじめとする連続量の相互作用が重要である。デジタル計算機上で実行されるソフトウェアでは連続量を正確に扱うことは不可能であり、離散値による近似によって一定精度のもとに連続量に対する信頼性を確保している。

ソフトウェアが高度化するに従って、連続量と離散量のミスマッチによる影響は複雑となっている。特に時間はシステム全体で同期して変化する量であることから、本質的に連続的である時間経過を離散的に扱うことによって信頼性が低下することは、従来のソフトウェア検証の枠組みで直接的に扱われてこなかった。このため、時間経過として不完全な実行環境における実行意味を定め、その性質を検証することで、実世界との間で正しい相互作用を及ぼすための条件を明確にすることは、安全性が重視されるソフトウェアに対して必要な観点である。
全体のシステム設計のもととなるシステム制御においては、対象領域に対する連続性を仮定して微分方程式によって振舞いを定める。正しく設計されたシステムとその実現には振舞いのギャップが存在するにもかかわらず、その扱いは実現における暗黙的な技術であるため、解決のための基礎となるモデルや解析手法が十分研究されているとは言えない。この問題に対して、ソフトウェア検証手法のうちのモデル検査技法を用いて信頼性を向上させる手法の提案が必要である。

2. 研究の目的

本研究では、連続的な時間経過に対して正しく動作するプログラムの離散実行条件について研究する。実時間プログラムの振舞いに対して離散的な実行意味を与えることを目的とする。離散実行では、連続実行をサンプリングに基づいて近似する。サンプリング意味を形式的に定義することで、離散実行意味に基づいたプログラム検証によってプログラムが保証する実時間性を明確に表現することが可能になる。実時間プログラムの動作のうち、システム全体に大きな影響を与える動作が実行されずにスキップされると致命的な不具合を起こす可能性がある。これは実時間プログラムが持つ連続時間上の意味と実際の離散的実行意の間の相違に起因する。具体的なプログラミング言語として、関数型プログラミング言語 Haskell のハイブリッドオートマトンの領域特化言語である Yampa を対象とし、Yampa プログラムの記述の意味と実行環境の振る舞いについて定式化を行うことで実際のプログラムの離散的振る舞いの性質を検査するための枠組みを定める。

Yampa 言語では、連続した時間の型として Time を持ち、プログラムの振る舞いは時間から値への信号関数として記述される。概念的には連続したなめらかな信号関数を値の変化に応じて切り替えることによってシステムの振る舞いを記述する。Yampa プログラム全体としては、振舞い意味としてはハイブリッドオートマトンの振舞いを信号関数で記述する他に、信号関数をどのようにサンプルして実行するかということも記述する。本研究では、ハイブリッドオートマトンの振舞いが Yampa プログラムの仕様であるとみなす。

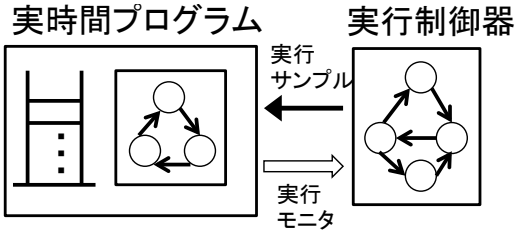


図 1: 離散実行意味

Yampa プログラムの仕様をハイブリッドオートマトンに変換し、サンプルのための時間オートマトンを合成することによって、ハイブリッドオートマトンの振舞いを離散的な振舞いに変換する。この振舞いを解析することによって、離散的実行がプログラムの仕様を満たすかどうか検査できるようになる。実際にモデルを構成し、時相

論理式によって性質を検査する。本研究では、モデル検査によって反例が導出された場合、プログラムの仕様が正しいことを前提とすれば、環境に問題があり、プログラム実行の前提が成立してないことが確かめられる。離散実行のサンプル時間間隔はアプリケーションレベルで実現した場合には、環境に依存して変動することがあり、このための障害を解析するための手法を与えることができるようになる。

モデル検査器 Uppaal を用いて実際に検査を行って実際に提案手法が可能であることを示す。

3. 研究の方法

(1) Yampa 言語のプログラムから信号関数とその切替え関数である switch 関数を取り出すことによってプログラムから仕様としてのハイブリッドオートマトンを抽出する。Yampa は Haskell 言語を含むので記述力が高すぎるので、ハイブリッドオートマトン記述に必要な言語要素を simple-Yampa として定義する。

(2) 構成したモデルに対して Uppaal を用いてモデル検査を実施する。時間オートマトンのサンプル制御において、サンプルが正確でない場合、どのような振舞いをするかについて網羅的にモデル検査を実施する。

(3) より広いクラスの Yampa プログラムへの適用を目標として、DTPDA モデルの拡張を行い、サンプル意味論を適用する。

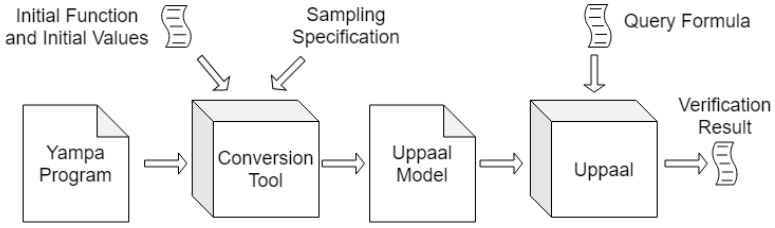
(4) より規模の大きな Yampa プログラム（Haskanoid などのゲームプログラム）に対するサンプルモデル検査を実施する。また他の実時間プログラムに対するテスト手法についても検討する。

4. 研究成果

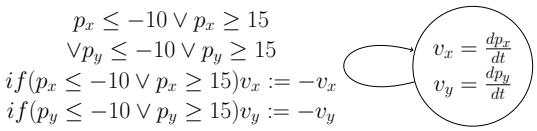
(1) Yampa プログラムからのモデル抽出と Uppaal によるモデル検査

本項目が本研究で行った研究の主な結果である。

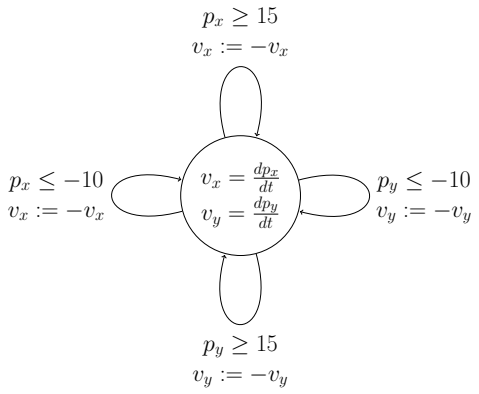
構文を制限した simple-Yampa プログラムから Uppaal のモデルを抽出して、モデル検査を行う手法を提案した。Simple-Yampa は、信号関数の定義と信号関数を切り替える以下のよう switch 文のみに制限したプログラム仕様記述とする。Simple-Yampa で記述されたプログラムを Uppaal 入力とする変換プログラムを作成した。全体を図のような枠組みで検証を行う。



この枠組において、枠内でボールがバウンドする様子をシミュレートする２つの SimpleYampa プログラム (Ha1Program.hs, Ha4Program.hs) を記述し、２つのハイブリッドオートマトンが抽出された。



Ha1Program.hs



Ha4Program.hs

2 つのプログラムに対して、サンプルトリガを生成する時間オートマトンを合成し、以下の Uppaal における Query である時相論理式 ψ_1 を与えたところ、Ha1Program.hs の振舞いはサンプル時間によっては、ψ_1 は満たさず、Ha4Program.hs の振舞いはサンプル時間に関わらず ψ_1 を満たすことが示された。ψ_1 は、ボールが枠の外の一定範囲内に留まることを示す。したがって、ψ_1 を満たさないことは、ボールが枠を非常に大きくはずれ、最悪の場合は戻ってこない場合があることを示している。

$$\psi_1 \equiv \texttt{A[]}(O_x\texttt{>} - 100 \texttt{ and } O_x\texttt{<}100 \texttt{ and } O_y\texttt{>} - 100 \texttt{ and } O_y\texttt{<}100)$$

この際に Uppaal は値として整数型のみをサポートするため、この実験においては、整数の対を固定小数点数として計算するモデルを構成してモデル検査を行った。またサンプルトリガを生成する際に jitter を発生するようにしてモデル検査を行った。その実行結果を以下の表に示す。

表 1: Checking ψ_1 by sampling with jitter

	Ha1program.hs		Ha4Program.hs	
Sampling Time Range and Step	Time(sec)	Memory (KB)	Time(sec)	Memory(KB)
Range [1.0,1.0] (Constant)	0.024	51,712	0.133	51,684
Range [0.9,1.1]/Step 0.1	3.891	92,584	231.357	2,893,852
Range [0.8,1.2]/Step 0.1	6.969	97,524	505.442	5,266,224
Range [0.7,1.3]/Step 0.1	10.537	135,016	833.631	7,782,628
Range [0.6,1.4]/Step 0.1	14.368	175,704	1,157.481	10,466,364
Range [0.95,1.05]/Step 0.05	14.294	296,468	1,066.576	17,366,568
Range [0.9,1.1]/Step 0.05	25.11	409,348	3,063.726	38,701,728
Range [0.85,1.15]/Step 0.05	35.954	487,160	5,214.78	57,265,228
Range [0.8,1.2]/Step 0.05	47.872	609,384	7,730.198	75,440,968
Range [0.8,1.2]/Step 0.2	1.148	53,556	39.123	403,816
Range [0.6,1.4]/Step 0.2	2.086	55,392	82.459	753,052
Range [0.4,1.6]/Step 0.2	3.401	57,512	137.783	1,186,116
Range [0.2,1.8]/Step 0.2	4.685	92,884	209.389	1,733,452

(Uppaal 4.1.22(64bit) on Ubuntu 18.04LTS server(kernel 4.15.0) with Intel Xeon E5-2690 v2@3.00GHz and 128GB memory.)

結果として、このモデルにおいてサンプルによる振舞いの相違を定式化することができた。(研究会発表 2 件)

(2) DTPDA モデルの拡張

DTPDA モデルを Yampa プログラムのモデリングに応用するために、変数の Update および DTPDA の遷移状態に不変式 (invariant) を導入する拡張を行った。いずれも到達可能性について決定不能であることが得られ、このモデルを適用するためには、新たな手法が必要であることがわかった。この拡張については本研究期間では十分な結果を得られなかったので、今後の研究課題とした。(国際会議発表および予稿週 2 件)

(3) 実時間プログラムに対するテスト手法

実用的な実時間に対するモデル検査は効率的なツールを用いたとしてもコストが大きいことが示されたため、新たな手法として、強化学習を用いたテスト手法について研究を行い、初歩的な結果を得た。ここでは、Simulink のモデル記述に対して、テストのパラメータを学習アルゴリズムを用いてより効率的に不具合を発見する。従来結果よりも精度の高い学習パラメータを設定し、より少ない試行で不具合発見が可能となったが、学習パラメータのコスト計算が上昇したため、時間評価については大きな改善は得られなかった。ここでの新たな知見は、検

査する性質の真偽だけでなく、どの程度性質を満たすかという数値評価をおこなったことである。この手法は連続動作の離散実行における精度に応用できると期待できる。（研究会発表１件）

５．主な発表論文等

〔雑誌論文〕　計5件（うち査読付論文　5件／うち国際共著　5件／うちオープンアクセス　1件）

１．著者名 Imai Keigo、Yoshida Nobuko、Yuen Shoji	４．巻 172
２．論文標題 Session-ocaml: A session-based library with polarities and lenses	５．発行年 2019年
３．雑誌名 Science of Computer Programming	６．最初と最後の頁 135～159
掲載論文のDOI（デジタルオブジェクト識別子） 10.1016/j.scico.2018.08.005	査読の有無 有
オープンアクセス オープンアクセスとしている（また、その予定である）	国際共著 該当する

１．著者名 Li Guoqiang、Wen Yunqing、Yuen Shoji	４．巻 61
２．論文標題 Updatable timed automata with one updatable clock	５．発行年 2018年
３．雑誌名 Science China Information Sciences	６．最初と最後の頁 1-14
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/s11432-016-9027-y	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Imai Keigo、Yoshida Nobuko、Yuen Shoji	４．巻 10319
２．論文標題 Session-ocaml: A Session-Based Library with Polarities and Lenses	５．発行年 2017年
３．雑誌名 Lecture Notes in Computer Science	６．最初と最後の頁 99～118
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-319-59746-1_6	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Wang Yuwei、Wen Yunqing、Li Guoqiang、Yuen Shoji	４．巻 10610
２．論文標題 Nested Timed Automata with Diagonal Constraints	５．発行年 2017年
３．雑誌名 Lecture Notes in Computer Science	６．最初と最後の頁 396～412
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-319-68690-5_24	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 Wang Yuwei、Li Guoqiang、Yuen Shoji	４．巻 10606
２．論文標題 Nested Timed Automata with Invariants	５．発行年 2017年
３．雑誌名 Lecture Notes in Computer Science	６．最初と最後の頁 77～93
掲載論文のDOI（デジタルオブジェクト識別子） 10.1007/978-3-319-69483-2_5	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

〔学会発表〕　計4件（うち招待講演　0件／うち国際学会　0件）

１．発表者名 中根里空、結縁祥治
２．発表標題 離散時間実行環境におけるYampaプログラムに対するUppaalを用いた振舞い検証
３．学会等名 電子情報通信学会ソフトウェアサイエンス研究会SS2018-52
４．発表年 2019年

１．発表者名 市橋友樹、結縁祥治
２．発表標題 離散時間実行環境におけるYampaプログラムの振舞いモデル
３．学会等名 電子情報通信学会ソフトウェアサイエンス研究会 SS2017-24 pp.19-24
４．発表年 2017年

１．発表者名 結縁祥治、平岡祥
２．発表標題 凍結クロックを持つ稠密時間プッシュダウンオートマトンの到達可能性のための記号ゾーン解析手法について
３．学会等名 電子情報通信学会ソフトウェアサイエンス研究会 SS2018-65 pp.7-12
４．発表年 2018年

１．発表者名 大脇亮太,結縁祥治
２．発表標題 Simulinkモデルに対するChainerRLを用いたハイブリッド頑健性に基づく時相理論仕様の不具合導出
３．学会等名 電子情報通信学会システム数理と応用研究会MSS2019-67 pp.53-58
４．発表年 2020年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考