氏名（本籍）	木(き)村(むら)成(しげ)伴(とも)　（宮城県）
学位の種類	博士（情報科学）
学位記番号	情博第4号
学位授与年月日	平成7年3月24日
学位授与の要件	学位規則第4条第1項該当
研究科，専攻	東北大学大学院情報科学研究科（博士課程）情報基礎科学専攻
学位論文題目	Inductive Inference of Algebraic Processes (代数的プロセスの帰納推論)
論文審査委員	(主　査) 東北大学教授　白鳥　則郎　　東北大学教授　丸岡　章 東北大学教授　佐藤　雅彦　　東北大学教授　富樫　敦

論文内容要旨

Chapter 1. Introduction

The study of process algebra started from the latter half of 1970's, to provide mathematical semantics for concurrent systems. Algebraic formalization techniques are used as the descriptive languages for communicating processes and concurrent programs. They are also applied to the verification problem due to their mathematical for mality. The processes, however, have the features such as non-determinacy and concurrency. So their operational semantics are completely different from those of the traditional automata and formal languages.

Considering the above aspects, we propose an inductive synthesis algorithm for a process defined by a subset of CCS. The algorithm synthesize a process form facts, which must be satisfied by the target process. In general, such facts are infinitely many, and therefore are given to the algorithm one by one. When n facts are input to the algorithm, it generates the process which satisfies the n facts already fed to it. This generation process is repeated infinitely many times. The correctness of our algorithm is stated as follows. The sequence of processes generated by the algorithm converges in the limit to a process, which cannot be distinguished from the intended one (if it could be known) by a given enumeration of facts.

Chapter 2. Preliminaries

In this chapter, we briefly review the preliminary notions of algebraic processes of CCS and facts of processes. In this research, facts of processes are logical formulae of HM-logic and μ-calculus. A sub-calculus, called μ^{-}-calculus, is defined and its properties discussed.

Chapter 3. Synthesis Algorithm Based on Enumeration of Process

This chapter describes an algorithm that synthesizes a process which is strong eguivalent to the intended one. For systhesis, logical formulae of HM-logic are considered to describe spefic properties of the process. This method is an incremental technique, similar to the one used by Shapiro in the model inference.

First, a refinement operator ρ_o is defined. It constructs any guarded and sequential processes in finite time. By using ρ_o, a linear ordered set of all guarded and sequential processes P_h is constructed.

The proposed algorithm in this chapter is inductive one. It generates a process which satisfies given formulae of HM-logic, which are the properties of the intended target process. In general, There are infinitely many formulae. So each formula is input to the synthesis algorithm one by one. When n-th formula is input, the output from the algorithm is a process satisfying n formulae which are already input. This synthesis muchanism of the algorithm is described as follows. Suppose a sequence of input formulae of the algorithm is $f_1, \ldots, f_n, \ldots$, and a corresponding sequence of output processes is $p_1, \ldots, p_n, \ldots$. When the formula f_n is input, the algorithm has already output the process p_{n-1} satisfying $f_1, \ldots, f_{n-1}$. If this p_{n-1} satisfies f_n, the algorithm output p_{n-1} is p_n, since p_{n-1} satisfies $f_1, \ldots, f_n$. Otherwise, the algorithm searches a process satisfying this n-th formula too, from P_h.

The algorithm is a non-terminating procedure. Therefore, its correctness is shown by using the concept of convergence in the limit, which has been a key idea in inductive learning paradagm. Assume an algorithm that reads in an enumeraion of facts, and returns processes sequentially. After some time, if the output process is always the same p, then the inferred sequence by this algorithm converges in the limit to p over the enumeration of facts.

The validity of the synthesis algorithm is also shown by the following theorem. Under the assumption of this synthesis algorithm, if there exists a process p_o satisfying all enumueration of facts, the inference by this algorithm converges in the limit to a process, which is strong equivalent to p_o.

Chapter 4 . Synthesis Algorithm Based on Model Construction

In the previous chapter, the synthesis algorithm for algebraic processes is based on enumeration of processes. When some formulae are input, this algorithm searches for a process, which satisfies given formulae, from an ordered set of all processes. The method adapted in the algorithms is based on inductive inference. However, it is rather promitive regarding efficiency.

This chapter presents a new algorithm which synthesize secursive processes gradually from the information of the input formulae. In the previous chapter, formulae of HM-logic is adopted. But the expressive power of HM-logic is not strong enough to efficiently synthesize recursive processes by the way of model construction. Therefore, the algorithm in this chapter adopts formulae in μ-calculs.

The input and output sequences of the algorithm in this chapter are same as the one of the previous chapter. When f_n is input to the algorithm, it synthesizes p_n from p_{n-1} and f_n and relevant information which is obtained until p_{n-1} is constructed. This information is a set of formulae which p_{n-1} must satisfy. The set of formulae of p_i is written by C_i. $p_i{:}C_i$ describes a process p_i with information C_i. In fact, any subprocess p_i has this set of formulae. The algorithm is brifly described as follows. Suppose the algorithm output p_{n-1} is C_{n-1}, where $p_0 = 0$ and $C_0 = \emptyset$. When f_n is input to the algorithm, then :

- If $f_n = \mathbf{tt}$, output p_{n-1} as p_n.
- If $f_n = \mathbf{ff}$, the input sequence of formulae is inconsistent.
- If $f_n = f \wedge f'$, apply f and f' to p_{n-1} in succession.
- If $f_n = f \vee f'$, first apply f. If inconsistency occurs by this application, backtrack to the point before f is applied, and then apply f'.
- *If* $f_n = \langle a \rangle f$, apply f to a certain process q, which is a subprocess after performing a from p_{n-1}. If there is no such subprocess, then add a new a-branch to p_{n-1} and apply f to the process q after this a-branch. And also apply any formula g to q, where $[a]g \in C_{N-1}$.
- If $f_n = [a]f$, add f_n to C_{n-1}, and apply f to any subprocess after performing a from p_{n-1}.
- If $f_n = \mu x.f(x)$ or $\nu x.f(x)$, apply $f(x)$ to p_{n-1}.
- If $f_n = x$ and x is a variable of $\mu x.f(x)$, then apply $f(x)$ to p_{n-1}.

- If $f_n=x$ and x is a variable of $\nu x.f(x)$, then unify the current node, i.e. the subprocess of p_{n-1} where f_n is applied, to the node where its original formula $\nu x.f(x)$ is applied.

Of course, some techniques are needed during application of variables, so that the above algorithm terminates at each step. Finally we have the following result. The sequence of the algorithm converges in the limit to a process, which is strong equivalent to the target process.

Chapter 5. A Prototype for the Process Synthesis System

This chapter introduces a prototype system SORP (Synthesizer of Recursive Processes) based on the synthesis algorithm presented in the previous chapter. This system is a graphical user interface to synthesize a process. It displays a process graph when the algorithm synthesizs a process by each input formula. The system is implemented by using SICStus Prolog and X-window system on Sun Sparc Station 2.

Chapter 6. Concluding Remarks and Related Works

This chapter concludes and briefly discusses future problems. Our method is compared to related works and we show the reason why inductive inference is more suitable to synthesize a process.

審査結果の要旨

近年，並列性や非決定性を有する分散システムが注目されている。代数的プロセスは，並列的かつ非決定的に振舞うシステムのモデル化に適するが，その記述が数学的に厳密であるため，記述されたシステムは読解性に乏しく，初心者にとって記述が容易ではないなどの難点がある。一方，具体的性質からプロセスを獲得する手法は，信頼性の高い通信プロトコルなどを自動生成するための基礎を与え，読解性や記述性に関する問題点を解決するための有効な手法として期待されており，その手法の確立が重要な課題となっている。著者は，これらの問題を解決するための方法として，計算機による機械学習の一手法である帰納推論を用い，具体的な性質の枚挙からそれらを満たすプロセスを帰納的に推論するための方法に関する研究を行なった。本論文はこれらの成果を取りまとめたものであり，全編6章からなる。

第1章は序論である。

第2章では，代数的プロセスの定義とプロセスの性質を表現するための論理体系の定義を与え，それらの諸性質を導出している。

第3章では，Shapiroのモデル推論の手法を基礎に，意図したプロセスの満たす論理式の枚挙から，プロセスを合成するアルゴリズムを提案している。このアルゴリズムによる出力系列は，意図したプロセスと強等価なプロセスに極限において収束することを示している。この結果より，全ての遷移先有限なプロセスを生成することが可能となり，この章での結果はプロセスが帰納推論によって導出可能であることを理論的に示した有用な成果である。

第4章では，前章のアルゴリズムの効率化について議論し，より現実的な2種類のアルゴリズムを提案している。具体的には，ある制限された再帰的プロセスのクラスを対象とし，入力された具体例の情報を用いて最終的なプロセスを構成的に合成するアルゴリズムとその正当性を理論的に示している。一般に，プロセスは並列的で非決定的に動作するため，形式言語やオートマトンなどとは本質的に異なり帰納推論が困難とされていた。この章での結果はこれらの問題点を克服すると同時に，この分野の発展に有用な知見を与えるものであり，柔軟な分散システムを開発する上での基礎となる重要な成果である。

第5章では，プロセスの合成支援システムの試作を行ない，通信プロトコルなどの具体例に適用することによって，本論文で与えた手法の有効性を実証している。これは，興味深い結果である。

第6章は結論である。

以上要するに本論文は，具体的な事実から非決定性や並列性を含んだシステムを高信頼的に獲得するための理論的な基礎を与えたものであり，情報基礎科学に寄与するところが少なくない。

よって，本論文は博士（情報科学）の学位論文として合格と認める。