

π 計算に対する動作視覚化システム

1 E - 4

塚崎 悟[†] 金指 文明[‡] 富樫 敦[¶]

静岡大学工学部知能情報工学科

1 はじめに

本論文では、モバイルプロセスの数学的モデルである π 計算の動作を視覚的にシミュレートするシステムを提案する。π 計算とは、Milner ら [1] によって提案されたプロセス計算であり、名前通信機能によりプログラミング言語での名前呼び出し (C 言語では Pointer) のような通信機能を有している。名前通信機能によりデータの通信のみでなく、ポートやプロセス自体の通信が可能となり、より複雑なプロセスの記述ができる。複雑なコンピュータプロセスを表現できるが、数式による表現であるため、直観的にプロセスの動作を理解することが難しい。本システムは、プロセス式の意味をアニメーションを利用することで表現し、プロセス式の動作を理解するための補助ツールとなる。

一方、ネットワークの普及により、分散・並列処理システムの開発に関する基礎理論および開発手法が重要となっている。π 計算は分散・並列処理プロセスの数学的モデルであるため、システムの仕様記述言語として利用することができる。π 計算で仕様記述を行い、この仕様記述を基に実際に動作するシステムを構築できれば有用な開発手段となる。本論文では、シミュレータ設計時に π 計算のプロセス式からプログラミング言語へ変換するための簡単なフレームワークを提案する。実際に Java 言語によりフレームワークを実装し、これを利用して本論文のシステムを設計・開発を行う。

2 シミュレータの概要

π 計算には、通信ポートから1つのデータの入出力を扱う Monadic π 計算 [1] と複数のデータを同時に扱える Polyadic π 計算 [2] がある。本論文では前者の Modadic π 計算に対するシミュレータを構築する。図 1 に Monadic π 計算の構文を示す。なお、ここで示す π 計算の構文は完全な定義ではなく、本研究で考慮する構文のみを示した。

| | | |
|---------|--------------|---------------------------------------|
| $P ::=$ | 0 | 無動作プロセス (<i>inaction</i>) |
| | $\tau.P$ | 静アクション (<i>silent prefix</i>) |
| | $x(y).P$ | 入力アクション (<i>bound input prefix</i>) |
| | $\bar{x}y.P$ | 出力アクション (<i>output prefix</i>) |
| | $P + P$ | 選択 (<i>summation</i>) |
| | $P P$ | 合成 (<i>composition</i>) |
| | $(\nu x)P$ | 制限 (<i>restriction</i>) |
| | $!P$ | 繰り返し (<i>replication</i>) |

図 1: π 計算の構文

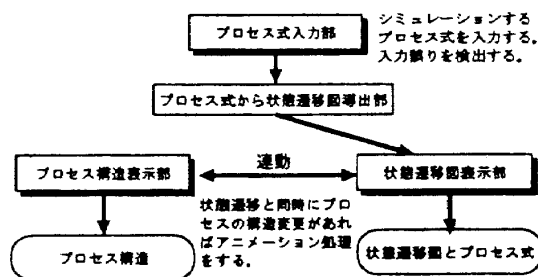


図 2: 構成図

P, Q, R, \dots はプロセス, x, y, z, \dots は名前を表す。 0 は、デッドロックを示す。 τ は観測できない内部アクションを表す。 α をアクションとすると $\alpha.P$ は、 α の実行後 P のように振舞うプロセスを表す。 $x(y).P$ や $(\nu y)P$ に出現する y のスコープは P であり、 y によって束縛されていることを意味している。 繰り返し演算子 $!P$ は、 P の無限個の複製の並行合成を示している。 動作は基本的に 2 個以上のプロセスの並行合成により全体としての動作が規定される。

本論文のシミュレータは、プロセス式入力部、プロセス構造表示部、状態遷移図表示部の 3 つの構成要素から成り立っている (図 2 参照)。プロセス式入力部は、シミュレートしたいプロセス式を図 1 の π 計算の構文にしたがって入力するためのインターフェイスである。入力されたプロセス式の構文誤りの検出や名前の一覧を表示することで入力時ミスを防ぐ機能を有する。実際のシミュレーションの前に、プロセス式から状態遷移システムを構成する。構成された状態遷移図は、状態遷移図表示部 (図 3) により表示される。シミュレーションは、導

A Visual system for the π-calculus.

[†] Satoru Tsukasaki, Dept. of Computer Science, Shizuoka University.

[‡] Fumiaki Kanezashi, Graduate school of Science and Engineering, Shizuoka University.

[¶] Atsushi Togashi, Department of Computer Science, Shizuoka University.

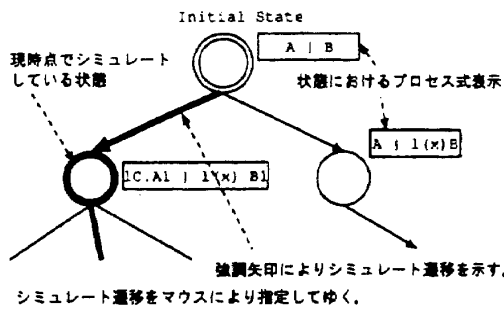


図 3: 状態遷移図表示部

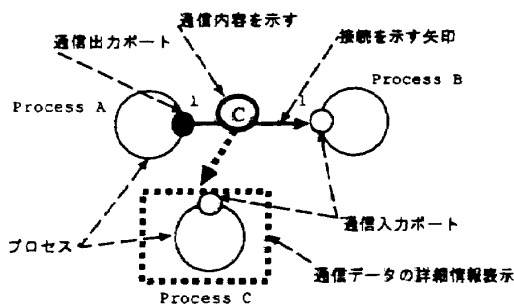


図 4: アニメーション表示の例

出された状態遷移図に従って実行される。様々な状態が考えられる場合は、ユーザが遷移を指定し、その遷移によるシミュレーションを実行する。プロセス構造表示部(図4)は、プロセス式を図的に表現して時々刻々と変化形状を直観的にわかるように簡単なアニメーション表示をする。π計算における変化は、主にプロセス間同士での通信による。よって、重視する表示内容は、プロセス間での通信経路の表現、データの送受信内容、ポートの変化である。これらを動的に図表現する。データ通信状況の表現は、ポートから出力されるデータの形式など現状でのプロセスの内部状況を表示する。これを表示することで、動的なプロセスの表示の他、内部のデータ状況も把握できるため、理解を深めることができる。

3 π計算プロセス実装フレームワーク

本シミュレータを実装するにあたり、一部でπ計算的な意味モデルを利用して設計を行った。その為、π計算による仕様をプログラミングする簡単なフレームワークを構築しこれを利用して実装を行っている。図5はフレームワークの構成を表している。π計算によるプロセスを実装する場合に、全ての実装されるプロセスはクラスProcessを継承することになる。クラスProcessは、通信のための入出力ポート操作や後述する管理エージェントとのやりとりを可能とする機能を持つ。またプロセス固有の処理を含めることができる。このプロセスは管理エージェントと呼ばれる通信の仲介を行う部分に登録される。実際に通信はプロセス間同士が行うが、どのプ

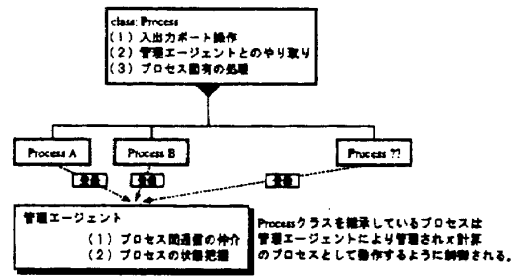


図 5: π計算フレームワーク概要

ロセス同士が通信するのか、どのポートと接続するかといった処理はこの管理エージェントが司る。実装時にフレームワーク利用者はこの管理エージェントの存在は特に気にしなくてもプログラムを構成することができる。

4 おわりに

本論文では、π計算の視覚シミュレータをπ計算からJavaプログラムを構成するフレームワークを利用することで設計・実装を行った。視覚シミュレータは、Monadic π計算に対するシミュレータを実装したが、この計算系よりも複雑なPolyadic π計算をシミュレートできるシステムに拡張したいと思っている。また実装に利用したフレームワークは、本論文の目的であるシミュレータを開発するためにならかなり特化した仕様である。このフレームワークでは他のアプリケーションに適用することは難しい。今後の課題として、今回のフレームワークを拡張することで一般的なアプリケーションの開発に耐えうるフレームワークを提案したいと思っている。

謝辞本論文は、一部文部省科学研究費(基盤研究(C)08680343, 重点領域研究09245214), 電気通信普及財団, 東海産業技術新興財団, 栢森情報科学新興財団の援助による。

参考文献

- [1] Robin MILNER, Joachim PARROW, David WALKER, A calculus of mobile processes, Reports ECS-LFCS-89-85 and -86, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University, 1989.
- [2] Robin MILNER, The Polyadic π-Calculus: a Tutorial, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University, 1991.