

生物シミュレーション構築のための 確率的構文を導入した論理型言語の提案

平田 裕和* 西山 裕之† 大和田 勇人†

東京理科大学大学院 理工学研究科 経営工学専攻* 東京理科大学 理工学部 経営工学科†

1 序論

近年、計算生物学の分野では生物システムをモデル化し、シミュレーションをすることによって、様々な生命現象を理解しようとする試みが行われている。この生物システムのモデル化する際、論理型言語を用いたプログラミングが行われている。論理型言語では、データ間の関係を命題としてプログラムし、パターンマッチングを行いながら命題が成立するかどうかを再帰的に調べることでプログラムが実行される。論理型言語を利用することで、生物システムにおける要素や要素間における相互作用を宣言的に記述することができる。

また、この論理型言語におけるプログラミングにおいて、論理型言語に確率的な概念を導入するという試みが行われている。しかし、従来の確率的論理型言語においては、項の確率的な記述を直接的に行うことができない。このことは、シミュレーション等の特に確率的な記述が多く含まれる事象を従来の言語を用いて記述する際には、プログラムが煩雑になるという問題が考えられた。

そこで本論文では従来の論理型言語に確率的な構文を導入することで拡張し、生物シミュレーションを構築する際に、プログラムをより容易に記述できるような言語の提案、実装を行う。また、生物システムの構築に利用できるライブラリを提供し、例として酵素反応のシミュレーションを示す。

2 設計方針

本研究で提案する言語の設計方針を以下のように定める。

- 論理型言語への確率的な構文の導入

提案言語では論理型言語の構文をベースとし、確率表現に関する構文を組み込む。確率的な記述は簡潔に記述できるようにし、記述性、可読性の高い言語を目指す。

- 制約論理型言語の利用

制約論理プログラミング [1] の概念を利用することにより、プログラム実行の効率化を図る。制約論理プログラミングでは制約により探索する対象領域の絞り込みを行う。これにより制約を満たさない解を事前にカットし、効率的な探索が可能となる。本研究では従来の制約論理型言語を利用することを想定する。

- 汎用的に利用できるライブラリの提供

様々な確率的記述に利用できるような汎用的なライブラリを提供する。このライブラリを用いることで、より効率的なプログラミングを支援する。

3 提案言語

本論文で提案する言語は制約論理型言語に、確率的な構文を導入することで拡張する。論理型言語に関する基本的な構文は Edinburgh Prolog[2] に準拠している。

3.1 構文規則

提案言語ではプログラムは節の集合として論理型言語の形式で記述する。節は以下の二つの形式のどちらかをとる。

$$H. \quad (1)$$

$$H : -\{B_1\}, \{B_2\}, \dots, \{B_n\}. \quad (2)$$

ここで H はヘッド、 B_1, \dots, B_n はボディである。ここで (2) のボディの項 $\{B_1\}, \dots, \{B_n\}$ について、本言語では選択確率に関する記述を加えることにより、項の実行確率を制御している。ある項に対して確率的な記述を加える場合、項に $\{ \}$ で囲み、さらに選択確率を項の先頭に付加する。つまりボディの項 T_i が実行される選択確率を λ_i とすると、項 $\{B_i\}$ は以下のように記述することができる。

$$\{\lambda_i.T_i\} \quad (3)$$

選択確率 λ は 0 から 1 までの実数の値をとる。 $\lambda = 1$ の場合は $\{ \}$ を省略し、単に項 T_i と記述することもできる。

また、項 T_1, \dots, T_n の選択確率をそれぞれ $\lambda_1, \dots, \lambda_n$ とし、この項 T_1, \dots, T_n の中から一つの項を選択し実行したい場合は以下のように記述する。

$$\{\lambda_1.T_1; \dots; \lambda_n.T_n\} \quad (4)$$

ここで選択確率は $0 \leq \lambda_1 + \dots + \lambda_n \leq 1$ でなければならない。また、 $\sum_{i=1}^n \lambda_i \leq 1$ であった場合は、 $1 - \sum_{i=1}^n \lambda_i$ の確率でいずれの項も実行されない。さらに、ネスト構造を利用して確率的なボディの項の記述をすることもできる。

3.2 実行規則

提案言語で書かれたプログラムにおいて、各節は以下のようない手順で実行される。

Step1 ヘッドとマッチする論理式が与えられる。節が規則を表し、ボディ部が記述されている場合は Step2 へ。

Step2 ボディを実行する。ボディの中に $\{ \}$ がある場合は Step3 へ。

Step3 対応する $\{ \}$ で囲まれた範囲内の選択確率 $\lambda_1, \dots, \lambda_n$ の和を計算する。 $\sum_{i=1}^n \lambda_i$ が $[0,1]$ にない場合はエラーを出力する。

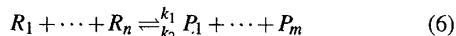
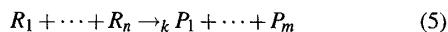
Step4 各項の選択確率に従って、任意の項を選び実行する。
選択した項の中に { } がある場合、Step3 から再帰的に実行する。

4 生物システムの記述

本章では、生物システムを記述するために利用できる汎用的なライブラリのひとつである、化学反応式 [3] の記述法を示す。また、このライブラリを用いて酵素反応の例を示す。

4.1 化学反応式

ここでは以下のような二つの基本的な化学反応式を提案言語を用いて記述する。



式(5)は最も一般的な化学反応式であり、式(6)は化学反応のうち可逆反応を表した化学反応式である。 R_1, \dots, R_n は反応物質であり、これらは P_1, \dots, P_m の生成物質へと変換される。 k は反応速度に関する速度定数である。式(5)の化学反応式は以下のように記述される。

```
reaction(..., R_i, ...), [..., P_j, ...], [..., k, ...] :-
  (R_i > 0, (R'_i = R_i - 1, P'_j = P_j + 1);
   true).
```

このプログラムではまず、化学変化に必要な反応物質が存在しているかをチェックする。存在していれば反応物質、生成物質の量を更新している。そしてこの処理を再帰的に呼び出す。もし反応物質が不足している場合は処理を停止する。また、この *reaction* を用いるときは選択確率により実行頻度を規定することにより、化学反応の反応速度を表現する。

また式(6)の可逆反応式は以下のように記述される。

```
rev_reaction(..., R_i, ...), [..., P_j, ...], k_1, k_2) :-
  reaction(..., R_i, ...), [..., P_j, ...], k_1),
  reaction(..., P_i, ...), [..., R_j, ...], k_2).
```

このプログラムでは反応物 R_i から生成物 P_i への化学反応と、生成物 P_i から反応物 R_i への逆反応が並列に動作するようになっている。

4.2 酵素反応

酵素反応とは、酵素が触媒する生化学反応である [3]。ここでは前小節で示した化学反応式に関する記述の適用例として、酵素反応の例を示す。

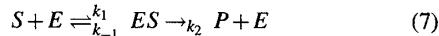
酵素反応は図1のような関係で表わされる。S は基質、E は



図1 酵素反応

酵素であり、この二つが結合する事により複合体 ES を形成する。この複合体は E と S に解離でき、また生成物 P と基質 E

に変換することもできる。酵素反応を式で表すと以下のようになる。



この酵素反応は前小節で示した *reaction* を用いて記述すると、以下のようになる。

```
enz_reaction(S, E, SE, P, k_1, k_{-1}, k_2) :-
  calc_kinetic(V1, V2, S, E, SE, P, k_1, k_{-1}, k_2),
  V1.rev_reaction([S, E], SE, [k_1, k_{-1}]),
  V2.reaction(SE, [P, E], k_2).
```

ここで用いられる *calc_kinetic* は反応速度を算出するための述語であり、反応速度は反応物質の濃度と速度定数 k より求められる。*calc_kinetic* によって算出された反応速度 $V1, V2$ をそれぞれ *rev_reaction*, *reaction* の選択確率として利用する。

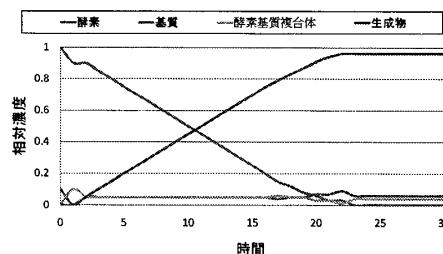


図2 酵素反応シミュレーション結果

図2はこのプログラムに基づいて行った酵素反応シミュレーションの一例である。このシミュレーションでは初期値として、酵素、基質、酵素基質複合体、生成物の数をそれぞれ10, 100, 0, 0 とし、速度定数として $k_1 = 0.1$, $k_{-1} = 0.0001$, $k_2 = 0.5$ とした。また、縦軸を基質の初期値を1.0とした相対濃度としている。この結果から基質が消費されるにつれて、生成物が生成されていく様子が示されている。この結果は[3]に示される結果ともほぼ一致している。

5 結論

本論文では、生物システムのシミュレーションを構築のための論理型言語の提案を行った。この言語では論理型言語を拡張し、確率的な実行の概念を組み込んだ。確率的な節の選択を可能にしたことにより、生物システムでのシミュレーションにおいて反応速度を表現する事を可能にした。また、化学反応に関するライブラリを提示し、これを用いることで酵素反応を容易に記述できることを示した。

参考文献

- [1] 沟口文雄, 古川康一, J.L.Lassez : 制約論理プログラミング. 共立出版, 1989.
- [2] D.L. Bowen. DEC-10 Prolog User's Manual. D.A.I. occasional paper 27, University of Edinburgh, December 1981.
- [3] P.W. Atkins. et al. Physical Chemistry. W H Freeman & Co, 2006.