

線形論理型言語を用いた人工化学のための推論の一手法

An Approach to Reasoning in an Artificial Chemistry
Using a Linear Logic Programming Language古野達也[†]
Tatsuya Furuno富永和人^{††}
Kazuto Tominaga

概要

人工化学系における分子の生成可能性を判定するために、線形論理に基づく論理型言語 LLP を用いて推論を行うことを試みた。線形論理とは命題を資源に見立てる論理体系である。この論理における命題を我々の人工化学における分子と見なすことで推論を行う。本研究は、この人工化学における推論を行うための LLP による記述法を与えた。そしてその記述法を用い、いくつかの人工化学系に対して推論のための記述を与えて実行し、正しく推論が行えることを確認した。

1. 背景と目的

人工生命と呼ばれる、計算機を用いて生命現象を研究する研究分野があり、その手法として人工化学がある [1]。人工化学は化学反応系を抽象的に表現する形式である。我々の研究室では文字列のパターンマッチと組み換えに基づく人工化学を提案し [2]、その人工化学系における分子の生成可能性を判定するための推論アルゴリズムを与えた [3]。

本研究はこのアルゴリズムによらず、線形論理 [4] に基づいて我々の人工化学に関する推論を行うことを試みる。線形論理とは命題を資源に見立てる論理体系である。我々の人工化学における分子をこの論理における命題と見なすことで、その個数を自然に扱った推論を行える。これにより、この種の人工化学における推論を線形論理を用いて行う手法を確立することを目指す。

処理系としては線形論理に基づいた論理型言語 LLP を用いる [5]。本研究では、この人工化学における推論を行うための LLP による記述法を検討する。

2. 人工化学

我々の研究室ではパターンマッチと組み換えに基づく独自の人工化学を提案している。この人工化学は原体と分体という概念を持つ。原体は自然界の原子に相当し、A や Abc のように大文字から始まる文字列で表す。分体は原体の列 (行と呼ぶ) の積み重なりであり、自然界の分子に相当する。分体は $O\#AB/1\#CD/$ のように表す。シャープ (#) 記号の前にある数字は「ずれ」といい、1 行目の先頭に対するその行の先頭の相対位置を表す。例えば $O\#AB/1\#CD/$ は、1 行目は原体 A と B で構成され、2 行目が原体 C と D で構成され、2 行目の先頭が 1 行目のそれに対して 1 つ右にずれているような分体を表す。

分体間の反応を組み換え規則という式で表す。組み換え規則は左辺の項に一致する分体群を右辺の項の分体群に組み換える。式 (1) は分体 $O\#A/$ と分体 $O\#B/$ を分体 $O\#AB/$ に組み換えることを表す。

$$O\#A/ + O\#B/ \rightarrow O\#AB/ \quad (1)$$

組み換え規則の各項はパターンという、パターン内では鬼札が使える。鬼札には原体鬼札と列鬼札の 2 種類がある。原体鬼札は任意の原体 1 つに一致し、 $\langle 1 \rangle$ などと表す。列鬼札は 0 個以上の原体からなる任意の原体列に一致し、 $\langle *1 \rangle$ や $\langle 2* \rangle$ など

と表す。 $\langle *1 \rangle$ は行の左端に、 $\langle 2* \rangle$ は右端のみに現れてよい。組み換え規則に鬼札を使った例が式 (2) である。これは、原体 A を先頭にもつ 1 行の分体と、任意の一つの原体からなる分体を組み換える。

$$O\#A\langle *1 \rangle/ + O\#\langle 2* \rangle/ \rightarrow O\#\langle 2* \rangle\langle *1 \rangle/ \quad (2)$$

この規則が $O\#AAAA/$ と $O\#B/$ に適用されると、それらは $O\#ABAAAA/$ に組み換わる。

人工化学系は初期分体の多重集合と組み換え規則の集合からなる。系は以下のように動作する。

1. 反応集合を初期化する。
2. 組み換え規則の集合から組み換え規則一つを選択する。
3. 選択した組み換え規則に一致する分体群があれば、その分体群にその規則を適用する。
4. 手順 2 に戻る。

3. 線形論理型言語を用いた人工化学に関する推論

線形論理は、命題を資源として扱う論理体系である [4]。この論理においては、命題を使用するとそれを消費し、二度と使えなくなる。線形論理に基づいた論理型言語 LLP [5] は Prolog の上位機能を持つ。この言語を用いることで、人工化学に関する推論を、分体の数を考慮して行える。

分体と命題を以下のように対応付ける。左側は我々の人工化学における分体の記述、右側が LLP における命題である。

$$O\#A/ \iff m([0], [[a]])$$

$$O\#AB/ \iff m([0], [[a, b]])$$

LLP において分体を表す述語 m は 2 つ引数を持つ。ひとつ目はずれのリスト、ふたつ目は原体列のリストである。原体列は原体のリストである。

組み換え規則は述語 $rule$ を使って以下のように記述する。二重矢印の左側に示した組み換え規則を二重矢印の右側のように LLP で記述する。述語 $rule$ の第一引数は組み換え規則の左辺のパターンであり、第二引数は右辺のパターンを表す。

$$O\#A/ + O\#B/ \rightarrow O\#AB/ \iff$$

$$rule(m([0], [[a]]), m([0], [[b]]),$$

$$m([0], [[a, b]])).$$

パターンに鬼札を使うには以下のように変数を用いる。原体鬼札 $\langle 1 \rangle$ を変数 $X1$ で、列鬼札 $\langle *1 \rangle$ を変数 $Xs1$ で表している。

$$O\#\langle 1 \rangle B/ \iff m([0], [[X1, b]])$$

$$O\#\langle *1 \rangle/ \iff m([0], [[a, b|Xs1]])$$

左端の列鬼札 ($\langle *1 \rangle$ など) がある場合には、以下のように原体列の連結とずれの調整を行なう。列鬼札 $\langle *1 \rangle$ をもつ 1 行全体を変数 $X1$ および $Y1$ とし、ひとつ目の $append$ で Y と a, b を合わせたものが $Y1$ に等しいことを、ふたつ目の $append$ で Y と a, b を合わせたものが $X1$ に等しいことを表している。

この組み換えによって Y の長さだけ 2 行目のずれが大きくなるので、パターンにおけるずれ ($1\#\langle 2* \rangle/$ の 1) に Y の長さを加えたものを新しいずれ ($Yd2$) としている。

[†] 東京工科大学大学院バイオ・情報メディア研究科

^{††} 東京工科大学コンピュータサイエンス学部

```

0#<*1>AB/ + 0#<2>/ -> 0#<*1>AB/1#<2>/ <=>
rule( (m([0],[X1]), m([0],[[X2]])),
      m([0,Yd2],[Y1,[X2]]) ) :-
  append(Y, [a,b], Y1), append(Y, [a,b], X1).
length(Y, L), Yd2 := L + 1.

```

以上のように定めた分体群と組み換え規則の記述を用いて、ある系におけるある分体の生成可能性を判定する。初期分体群を S、生成されるかどうかを調べる分体を O とし、(S) -<> create(O) で生成可能性を判定する。述語 hantei は、目的的分体を生成するのに必要な分体がすべて初期分体でまかなえたなら生成可能であることを表示する。

ひとつ目の create 節は、与えられた引数が初期分体の一つであれば、その分体を消費して真となる。

与えられた引数が初期分体でない場合は、ふたつ目の create 節によって、その分体を生成する組み換え規則の左辺に当たる分体が生成できるかを確認する。

```

% 0#X/ + 0#Y/ -> 0#Z/
hantei(S, O) :-
  ( (S) -<> create(O) ) ->
  (write('OK!'),nl);
  (write('Fail...'),nl) ).

create(m(Dd0,Da0)) :- m(Dd0,Da0), erase.

create(m(Dd0,Da0)) :-
  rule((m(Xd, Xa),m(Yd, Ya)),m(Dd0,Da0)),
  ((create(m(Xd, Xa)),create(m(Yd, Ya)));
  create( (m(Xd, Xa),m(Yd, Ya)) )).

```

4. 実行例

いくつかの簡単な人工化学系に対して推論のための記述を与えて実行した。その一例を以下に示す。

% 初期分体

```
% 0#AB/:3
```

```
% 0#CD/:1
```

```

% 規則 1 0#<*1>AB/ + 0#CD/ -> 0#<*1>AB/1#CD/
rule( (m([0],[Xa1]),m([0],[[c,d]])),
      m([0,Zd2],[Za1,[c,d]]) ) :-
  append(Z, [a,b], Za1), append(Z, [a,b], Xa1),
  length(Z, L), Zd2 := L + 1.

```

```

% 規則 2 0#<*1>AB/1#CD/ + 0#AB/
%
% -> 0#<*1>ABAB/1#CD/
rule( (m([0,Zd2],[Xa1,[c,d]]),m([0],[[a,b]])),
      m([0,Zd2],[Za1,[c,d]]) ) :-
  append(Z, [a,b,a,b], Za1),
  append(Z, [a,b], Xa1),
  length(Z, L), Zd2 := L + 1.

```

```

% 規則 3 0#<*1>ABAB/1#CD/ -> 0#<*1>ABAB/ + 0#CD/
rule( (m([0,Xd2],[Xa1,[c,d]]),
      (m([0],[Za1]),m([0],[[c,d]]))) ) :-
  append(Z, [a,b,a,b], Za1),
  append(Z, [a,b,a,b], Xa1),
  length(Z, D2), Xd2 is D2 + 1.

```

この系はいくつかの 0#AB/と一つの 0#CD/を使っ

て、0#ABAB.../を生成する。規則 1 で 0#AB/と 0#CD/が 0#AB/1#CD/になり、規則 2 で 0#AB/1#CD/と 0#AB/が 0#ABAB/1#CD/になり、規則 3 で 0#ABAB/と 0#CD/が離れる。そして規則 1 に戻り、同様の動作を繰り返す。最終的に初期分体から 0#ABABAB/5#CD/が生成できる。以下は初期分体からこの分体が生成可能であるかどうかを判定する記述である。生成可能であることが示される。

```

| ?- hantei( (m([0],[[a,b]]), m([0],[[a,b]])),
             m([0],[[a,b]]), m([0],[[c,d]])),
          m([0,5],[[a,b,a,b,a,b],[c,d]]) ).

```

OK!

初期分体を変更して、分体 0#ABABAB/5#CD/を生成するのに必要な分体 0#AB/がひとつ足りない状態にし、推論を行ったところ、意図した通り、生成不能という結果が得られた。

```

| ?- hantei( (m([0],[[a,b]]), m([0],[[a,b]])),
             m([0],[[e,f]]), m([0],[[c,d]])),
          m([0,5],[[a,b,a,b,a,b],[c,d]]) ).

```

Fail...

5. まとめと今後

本研究では、線形論理型言語を用いて、我々の人工化学における系について推論するための記述法を考案した。組み換え規則に鬼札を用いるこの人工化学に推論のための記述を与えることができた。また実際に簡単な系について記述を行い推論を行ったところ、意図した通りの結果が得られた。

今後は、より複雑な系の推論ができるように記述法を拡張する。現在は生成可能かどうか判定する目的的分体としてひとつしか指定できない。また、右辺にパターンを3つ以上持つ組み換え規則を扱えない。これらが可能となるように記述法を拡張する。さらに、人工化学系の定義を LLP の記述に自動変換するソフトウェアを開発する。

参考文献

- [1] Peter Dittrich, Jens Ziegler, and Wolfgang Banzhaf. Artificial chemistries — a review. *Artif. Life*, Vol. 7, pp. 225–275, June 2001.
- [2] Kazuto Tominaga, Yoshikazu Suzuki, Keiji Kobayashi, Tooru Watanabe, Kazumasa Koizumi, and Koji Kishi. Modeling biochemical pathways using an artificial chemistry. *Artif. Life*, Vol. 15, pp. 115–129, January 2009.
- [3] Kazuto Tominaga. An approach to reasoning in an artificial chemistry. In editor Hamid R. Arabnia, editor, *Proceedings of the 2006 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'06)*, Vol. II, pp. 839–845. CSREA Press, 2006.
- [4] Jean-Yves Girard. Linear logic: its syntax and semantics. In *Proceedings of the workshop on Advances in linear logic*, pp. 1–42, New York, NY, USA, 1995. Cambridge University Press.
- [5] Banbara Mutsunori, Kang Kyoung-Sun, and Tamura Naoyuki. An abstract machine for a compiler system of a linear logic programming language. *コンピュータソフトウェア*, Vol. 18, No. 1, pp. 195–216, 2001-01-15.