原始帰納関数を実現する項書換え系の遺伝的プログラミングによる学習

26-02

Using Genetic Programming to Learn Primitive Recursive Functions Implemented by Term Rewriting Systems

河口 信夫† 桑山 高史† 外山 勝彦‡ 稲垣 康善†
Nobuo KAWAGUCHI† Takashi KUWAYAMA† Katsuhiko TOYAMA‡ Yasuyoshi INAGAKI†

†名古屋大学大学院工学研究科計算理工学専攻 ‡中京大学情報科学部情報科学科
†Dept. of Comp. Sci. and Eng., Nagoya Univ. ‡Dept. of Comp. Sci., Chukyo Univ.

In this paper, we propose a learning method of primitive recursive functions implemented by term rewriting systems(TRS) using genetic programming(GP). Usually, in the process of learning recursive functions, it is difficult to guarantee the termination of the functions. So there is a requirement to limit the number of recursions. We adopt the structure of primitive recursive functions which is guaranteed to terminate, for the individual of GP as TRS. Additionally, using signature of the functions as a restriction of genetic operation, efficient learning for various functions are achieved. To obtain efficiency, we adopt the notion *penalty* for the recursive part. We confirm the effectiveness of it by the examination. Our method can learn "member" and "reverse" with additional support function.

1 はじめに

自動プログラミングの分野において，再帰関数を学習により獲得する研究がなされている．岸ら [岸 96] は，遺伝的アルゴリズムを用いてPrologの再帰述語を例から学習する手法を提案しており，member関数,reverse関数の学習に成功している．しかし，この手法は述語をビット列の遺伝子で表現することによる制約があり，応用性に欠ける．また，無限に再帰や単一化が起こる場合を避けることができないため，メタインタプリタによる処理を行っている．Kozaは木構造と遺伝子として扱う遺伝的プログラミング (Genetic Programming:GP)[Koza 92] により，フィボナッチ数等の再帰関数の学習に成功している．しかし，この手法も無限の再帰を排除するために特殊な関数を用いるため，一般的な再帰関数の学習手法とはいえない．

本稿では，原始帰納関数を実現する項書換え系 (Term Rewriting Systems:TRS)[二木 82] をGPにより学習する手法を提案する．一般に再帰関数の学習においては，生成された関数が停止性を持つかが不明であるため，従来の研究のようにメタインタプリタや特殊な関数を用いた再帰の制限が必要となる．我々は原始帰納関数のクラスでは停止性が保証されていることに注目し，原始帰納関数の形式を持つTRSを学習の対象として採用した．そのため，本手法では再帰の制限等の特殊な操作を必要としない．また，多くの有用な関数が原始帰納関数で表されることから，本手法の応用性の高さが期待される．遺伝的操作においては，TRSのシグニチャに関する制約を用いることにより，多様な関数に対する効率的な学習を可能にしている．また，本手法では目的とする関数が単一の関数で定義できない場合，複数の関数を同時に学習することも可能である．計算機実験によりmember関数,reverse関数等の学習を行い，再帰を行う部分を持たない個体に対してペナルティを与えることが効率的探索のために有効であることを確認した．

連絡先：河口 信夫　名古屋大学大学院工学研究科計算理工学専攻
〒464-01 名古屋市千種区不老町
Tel: 052-789-3630 Fax: 052-789-3800
Email: kawaguti@nuie.nagoya-u.ac.jp

2 原始帰納関数を実現するTRS

自然数上の原始帰納関数 F を実現するTRSは一般に次のように表現できる．

$$\begin{cases} F(\mathbf{x}, Z) \to h(\mathbf{x}) \\ F(\mathbf{x}, S(y)) \to g(F(\mathbf{x}, y), \mathbf{x}, y) \end{cases}$$

ここで，Z は零関数，S は後者関数である．以下では，この定義を一般の項集合上の原始帰納関数に自然に拡張したものとして議論を進める．便宜上 F を目的関数,h, g を構成する関数を基本関数と呼ぶ．また，シグネチャが定義されており，各関数は引数と返り値の型を持つ．この時，このTRSの合流性・停止性は保証される．

3 GPによるTRS学習システム

本システムの基本的なアルゴリズムはGP[Koza 92] に従う．目的関数の左辺の形式は基本的に変わらないのでユーザが与え，基本関数の合成により $h(\mathbf{x})$ 及び $g(r, \mathbf{x}, y)$ を獲得することによって，目的関数を実現するTRSを学習する (ここで r は再帰部 $F(\mathbf{x}, y)$ を表す変数)．組 (h, g) を解候補とし，GPの個体として扱う．GPの交叉等の遺伝的操作は，各関数のシグネチャの整合がとれたときのみ許される．

入力： 基本関数記号の集合，シグニチャ，事例 (目的関数の入出力対) の集合，GPのパラメータ
終了条件： 全ての事例を満たす個体の獲得または世代が最大世代数を超えた場合
アルゴリズム：

1. ランダムな個体の構成と母集団の形成
   個体はそれぞれ基本関数記号の集合及び h, g で用いる変数の集合から，シグネチャに従いランダムに構成する．
2. 適応度の計算，終了条件のチェック
   事例に対し，個体が定義するTRSを実行させ，適応度を求める．この時，個体 i の入力に対する誤答数を $m(i)$ とすると，適応度 $f(i)$ は以下の式で定義される．
   $$f(i) = \frac{1}{1 + m(i)}$$
3. 遺伝的操作（交叉および突然変異）
   適応度に比例した確率で個体のペアを抜き出し (ルーレット戦略)，h または g 同士で木構造間の交叉を行う．突然変異は個体をランダムに選んで行う．
4. 2にもどる

4 学習実験及び考察

本システムを用いてリストの要素の判定を行う member 関数，反転を行う reverse 関数等の学習実験を行った．獲得した member 関数の例を以下に示す．

$$\begin{cases} member(x, nil) \rightarrow false \\ member(x, cons(y, z)) \rightarrow if(eq(x, y), true, member(x, z)) \end{cases}$$

member 関数の実験では，基本関数として $if, eq, true, false, cons, nil$ を与えた．図 1 は事例数 30，母集団サイズ 50，最大世代数 50 で GP を 50 回実行した際の，各世代の最小誤答数，最大誤答数，平均誤答数の全体の実行における平均値のグラフである．なおアルゴリズムの 2 において，再帰部 r が g 上に出現しない個体に対してペナルティを設け適応度を下げた．この図から母集団全体が学習していく様子が確認できる．また図 2 に，再帰部に対するペナルティがある場合とない場合で実験を行なった際の，member 関数における各世代での目的関数獲得率の累積 (%) を示す．ペナルティがある場合は全試行の 84% で学習に成功したが，ない場合は 50% に留った．

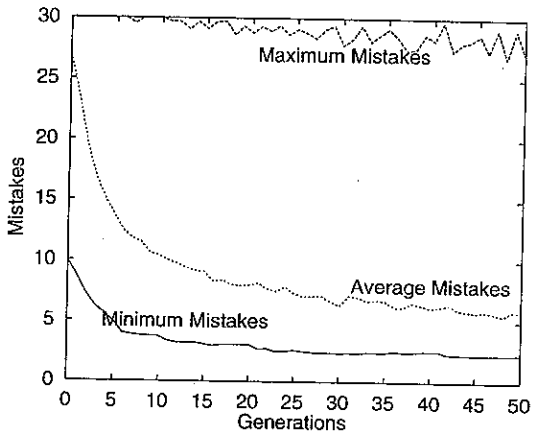

図 1: 50 試行における member 関数の平均誤答数等の推移

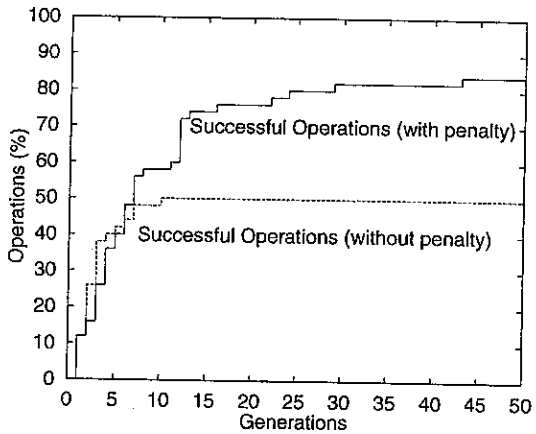

図 2: 50 試行における member 関数の累積獲得率

reverse 関数は単独の TRS の関数では定義できないため，補助関数を必要とする．そのため，補助関数も同時に学習を行なった．基本関数は $cons, nil$ のみである．実験は母集団サイズを 600 に増やし，世代 50，事例 40 で GP を 50 回実行した．獲得した reverse 関数の例を以下に示す．関数 A は同時に学習された補助関数を表している．

$$\begin{cases} reverse(nil) \rightarrow A(nil, nil) \\ reverse(cons(Y, Z)) \rightarrow A(cons(Y, A(nil, nl)), reverse(Z)) \\ A(X, nil) \rightarrow X \\ A(X, cons(Y, Z)) \rightarrow cons(Y, A(X, Z)) \end{cases}$$

50 試行における reverse 関数の累積獲得率を図 3 に示す．この実験では，reverse 関数に補助関数が含まれない場合にペナルティを与えている．ペナルティのある場合 66% の成功であるが，ペナルティのない場合 18% しか成功しない．このことから，再帰部や補助関数のペナルティが目的関数の効率的獲得に有効であることが確かめられた．また，本手法により再帰関数の獲得が可能であることが確かめられた．

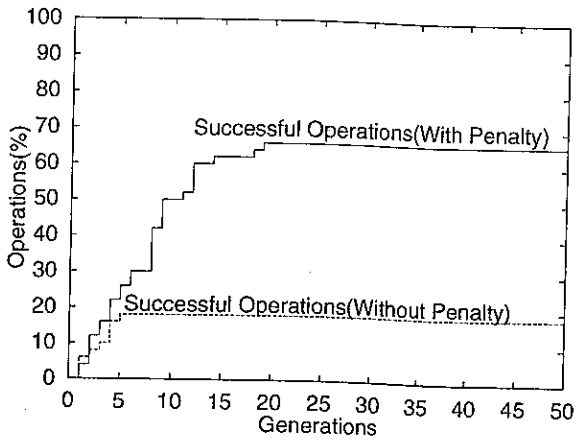

図 3:50 試行における reverse 関数の累積獲得率

5 まとめ

本稿では，遺伝的プログラミングを用いて原始帰納関数を実現する TRS を学習する手法を提案した．原始帰納関数の構造を学習の対象とすることにより，停止性に関する操作を必要とせず，十分な応用性を持つ再帰関数の学習が可能である．特に補助関数を必要とする関数においても補助関数を同時に学習することが可能であり，高い学習能力が認められた．

現在，学習成功率が低いのは，適応度の評価が計算結果の正誤のみにより行われるためと考えられる．Knuth-Bendix [Knuth 70] アルゴリズム等の TRS に関する性質を用いて，形式的に適応度評価を行うことにより，より高い学習成功率が得られることが期待される．また，現在の手法では再帰構造を持つ左辺項はユーザが与えることを仮定しているが，その学習による獲得も今後の課題である．

参考文献

[Koza 92] Koza, J.R.: Genetic programming : On the Programming of Computers by Means of Natural Selection, The MIT Press, 1992.

[Koza 94] Koza, J.R.: Genetic Programming II: Automatic Discovery of Reusable Programs , The MIT Press ,1994.

[岸 96] 岸, 小井土, 柴田 : GA を用いた番地型変数配置による述語の帰納的学習について 人工知能学会誌, Vol.11, No.4, pp. 112-122, 1996.

[Knuth 70] Knuth,D.E. and Bendix,P.B. : Simple Word Plobrems in Universal Algebras, *Computational Problems In Abstract Algebra*, Leech,L.(ed.), Pergamon Press, pp. 263-297, 1970.

[二木 82] 二木, 外山 : 項書換え型計算モデルとその応用. 情報処理, Vol. 24, No. 2, pp. 133-146, 1982.