効率的で正しいプログラムの自動生成

「機能と構成」領域　小川　瑞史

要　　旨

人間が作成するプログラムを効率の面で凌駕するプログラムを自動生成することをめざす．具体的には，簡単な関数型プログラムによる仕様記述を基づき，組合せ理論とプログラム変換手法を適用し理論的・実用的に効率の良いプログラムを生成する．現在，関係データベース検索における未解決問題の解の生成，および効率の良いコントロールフロー解析の生成を提案した．今後は大規模なプログラムに対する実用性の検証をめざす．

1　研究のねらい

効率的で正しいプログラムを得るために，計算機が可能なサポートには

- 人間の書いたコードの解析に基づく最適化とエラー検出，および
- 仕様に基づく自動生成

の二つがある．後者では，プログラムの正しさは，仕様と自動生成系が正しければ自動的に保証される．

本研究では，後者の立場をとり,「効率的で正しいプログラムの自動生成」というテーマを設定した．しかし，一般的なプログラムを対象に自動生成を試みれば，構成的証明からのプログラム抽出の研究でも広く知られるように簡単に決定不能性に陥る．したがって，有用な応用領域への適切な制限が必要になる．また，理論計算機科学では論理の枠組みを用いることが多いが，その中にもるべき内容を数学（特に組み合わせ理論）から借りてくることで，対象とする領域では人間のコーディングを凌駕する自動生成が可能と考えた．

具体的には，応用領域の制限として，

1. 関係データベースやデータマイニング,
2. コントロールフロー解析

を，また組み合わせ理論としてそれぞれ

1. 順序の理論（Well-Quasi-Order や Kruskal 型定理の構成的証明）
2. グラフの代数的構成法（グラフの木分割や木幅）

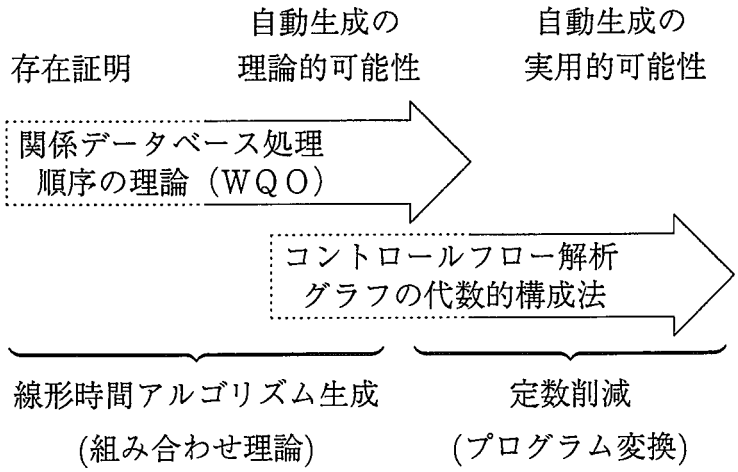

図１：研究計画の二つの流れ

を用いた(図 1)．このような効率的なプログラムの自動生成には，存在証明，自動生成の理論的可能性，自動生成の実用的可能性，の３つの段階がある．効率的アルゴリズムの存在証明はできても実際に生成できない状況は，特に Kruskal 型の定理を用いた場合にしばしば生じる．これを構成的証明を借りてきて乗り切きろう，というのが第一のアイデアである．

また自動生成の理論的可能性が示されたとしても，それは実用的可能性をそのまま示している訳ではない．典型的には，自動生成されたプログラムの計算量評価は良くても，実際には定数が爆発して現実的でない状況である．ここでは，実際のコントロールフローグラフは比較的良い構造をもっているという観察を出発点として，グラフの代数的構成法，およびその上の関数型プログラミング＋プログラム変換を用いよう，というのが第二のアイデアである．これらのアイデアはそれぞれ応用領域・組み合わせ理論における選択の 1., 2. に対応している．

2　研究方法と成果

順序の理論に基づく不定データベースの質問処理

当初は 1. について進めた．具体的には，関係データベース処理において van der Meyden (現ニューサウスウェールズ大) が 1993 年に提案した未解決問題，時間概念をもつ不定データベース上の選言（∨ を含む）単項質問処理の線形時間アルゴリズム，を対象とした．この問題は線形時間アルゴリズムの存在は示されていたが，その具体的構成法が知られていたなかったものである．

線形時間アルゴリズムの存在証明は，Higman の補題を用いて,「より充足可能な（∨ を含

$\{P(a),Q(b),R(c),a<b<c\}$, $\{Q(a),R(b),P(c),a<b<c\}$, $\{R(a),P(b),Q(c),a<b<c\}$,
$\{P(a),Q(b),a<b,Q(c),R(d),c<d,R(e),P(f),e<f\}$,
$\{P(a),Q(b),P(c),a<b<c,Q(d),R(e),d<e\}$, $\{Q(a),R(b),Q(c),a<b<c,R(d),P(e),d<e\}$,
$\{R(a),P(b),R(c),a<b<c,P(d),Q(e),d<e\}$, $\{P(a),R(b),P(c),a<b<c,Q(d),R(e),d<e\}$,
$\{Q(a),P(b),Q(c),a<b<c,R(d),P(e),d<e\}$, $\{R(a),Q(b),R(c),a<b<c,P(d),Q(e),d<e\}$,
$\{P(a),Q(b),P(c),Q(d),a<b<c<d,R(e)\}$, $\{Q(a),R(b),Q(c),R(d),a<b<c<d,P(e)\}$,
$\{R(a),P(b),R(c),P(d),a<b<c<d,Q(e)\}$, $\{P(a),R(b),P(c),R(d),a<b<c<d,Q(e)\}$,
$\{Q(a),P(b),Q(c),P(d),a<b<c<d,R(e)\}$, $\{R(a),Q(b),R(c),Q(d),a<b<c<d,P(e)\}$.

図2：単項質問 φ に対する極小不定データベース

まない）単項質問が多い」という不定データベース間の関係 $\preceq$ が well-quasi-order (WQO) であることから得られる．つまり，$\preceq$ が WQO であることは，任意のデータベースの集合に対し，($\preceq$ に関する）極小元が有限個しかないことを意味している．したがって，選言単項質問を満たす極小不定データベース $\{D_1,\cdots,D_k\}$ を先に計算しておけば，各 D_i との比較は線形時間でできることがわかっているので，全体として線形時間で質問処理がなされる．

しかし極小不定データベースが有限個であることがわかっていても，具体的に計算することは容易ではない．たとえば，単項質問 $\varphi=\psi_1\vee\psi_2\vee\psi_3$, ただし

$$\begin{cases}\psi_1 &= \exists xyz[P(x)\wedge Q(y)\wedge R(z)\wedge\ x<y<z],\\ \psi_2 &= \exists xyz[Q(x)\wedge R(y)\wedge P(z)\wedge\ x<y<z],\\ \psi_3 &= \exists xyz[R(x)\wedge P(y)\wedge Q(z)\wedge\ x<y<z].\end{cases}$$

において，その極小不定データベースの集合は図2のようになる．一番上の行が含まれるのは明らかだが，それ以外についてはパズルを解くような考察を要する．

本研究では，Higman の補題の構成的証明から極小元の計算法を抽出することで，自動的に極小不定データベースの計算，すなわち線形時間質問処理プログラムの自動生成が可能となった．この結果は2001年の秋，国際会議 TACS01 で発表し（国際会議2），続いて幸いにも特集号に招待され採録となった（海外論文誌3）．

しかし，1. については定数の削減の良いアイデアがみつからなかったこと (fold/unfold 変換は一つの可能性である)，また問題として自然なものがあまり設定できなかったため，中断して2. に力をそそぐことにした．

グラフの代数的構成法に基づくコントロールフロー解析

2. については，Thorup（現 ATT Labs）が1998年に示した実際のプログラムのコントロールフローグラフは比較的良い構造を持っているという観察（具体的には，GOTO-free C プログラムならば木幅は6以下，Java プログラムではほとんど木幅が3以下などが知られる）に基づき，コントロールフロー解析に問題領域のターゲットをしぼった．

論理式による記述

$$Conn(V) \;=\; \forall V_1\, \forall V_2\, (Part(V, V_1, V_2)$$
$$\Rightarrow (V_1 = \phi \;\vee\; V_2 = \phi \;\vee\; \exists v_1 \in V_1\, \exists v_2 \in V_2\, (Next(v_1, v_2) \vee Next(v_2, v_1))))$$
$$Part(V, V_1, V_2) \;=\; (V_1 \cup V_2 = V) \wedge (V_1 \cap V_2 = \phi)$$

関数型プログラムによる記述

```
conn [x] = True
conn (x:xs) = if marked x then nm xs || (marked (head xs) && conn xs)
                          else conn xs
nm [x] = not (marked x)
nm (x:xs) = not (marked x) && nm xs
```

図３：最大連続部分列和問題の仕様記述

最適化定理

もともとグラフの木幅が有界なとき，さまざまな性質の判定が線形時間でできることが知られていた（80年代からの Courcelle, Arnborg や Borie などの研究）．たとえば，ある制約のもとで，ある尺度において最適（たとえば重み和が最大または最小）となる部分集合を決定する最適化問題などはその例である．

しかし，線形時間といっても定数が爆発するため，理論的な計算量評価にとどまっていた．たとえば，グラフとしてもっとも単純なリストにおいて，プログラム変換の例題としてしばしばとりあげられる最大連続部分列和問題の線形時間プログラムは，連続部分列の論理式による記述から動的プログラミングとして自動生成される．しかしその定数（動的プログラミングの保持する状態数）は $2^{2^{2^{2^5}}}$ に達する．

本研究では，最適化定理を多項式データ型の場合に制限して，仕様を論理式のかわりに直接関数型プログラミングで記述し(必要に応じて)fusion/tuppling 変換を施すことで劇的な定数の改善（最大連続部分列和問題では 2^3) が可能となることを示した（国内論文誌２）．図３に最大連続部分列和問題の仕様記述を，それぞれ論理式および関数型プログラムで示した．論理式では V が選ばれたリストの要素の集合を，関数型プログラムでは `marked x` によって選ばれたリストの要素を現している．

さらに，いくつかの問題への応用（国内論文誌１，３）．および拡張（投稿中１）を試みている．

コントロールフローグラフの代数的構成法

上記の多項式データ型の最適化定理が出発点であったが，コントロールフローグラフに応用するには，グラフの適切な代数的表現が必要になる．試行錯誤の末，最終的にSP項という概念に到達し，国際会議ICFP03にて発表した（国際会議4）．簡単なプログラムを例としたコントロールフローグラフのSP項による表現（木幅が2の場合）を図4に示す．

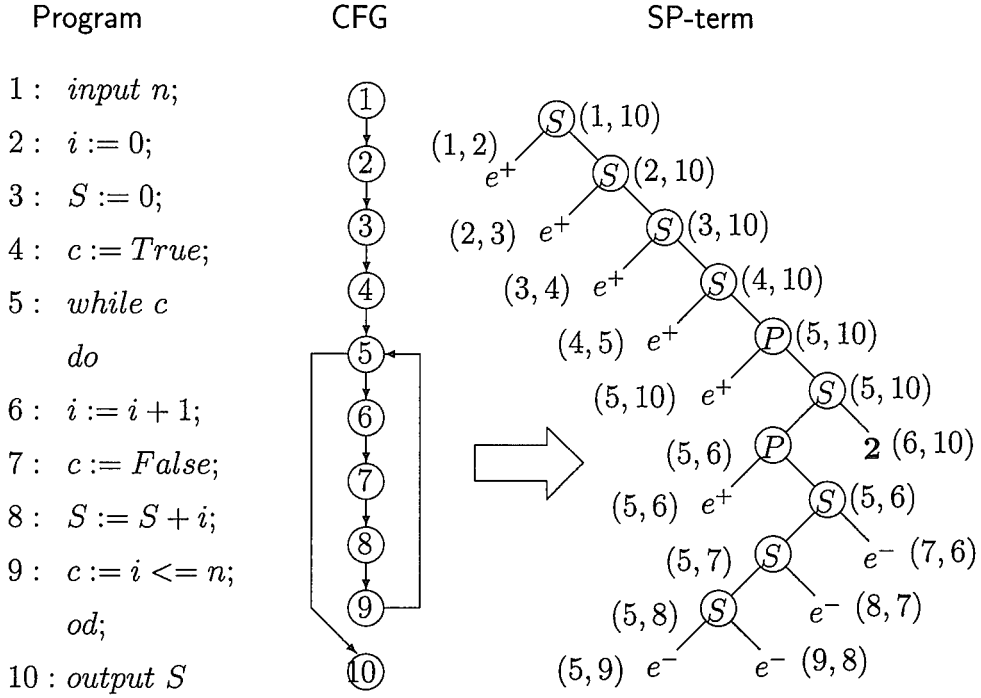

図4：SP項によるコントロールフローグラフの表現の例

木幅が有界なグラフにおいて，いくつかの代数的表現は既に知られていたが，木幅の増大につれ再帰構成子の個数が爆発するため，実際のプログラミングはほとんど不可能であった．それに対し，SP項は再帰構成子の数を木幅によらず2に抑えたため，実際に(仕様記述として)プログラミングが可能になった．代償として，定数構成子の数は増えるが，それらについてはほとんど等質な扱いが可能で，プログラマの負担にはならない．

SP項上の実際のプログラミング，および自動生成の実例として，不要変数解析や最適レジスター割り当ての仕様を記述した（国際会議4）．通常のコントロールフロー解析がプログラムの状態遷移の追跡を解析結果が収束するまで繰り返すのに対し，SP項を用いた不要変数解析の記述ではそれぞれ1回の上昇型構造再帰ならびに下降型構造再帰により実現される．

最適レジスタ割り当ては，プログラムの実行時に各コントロールポイントにおけるレジスタにどの変数を割り当てるか，最適な割り当て（レジスタからメモリへの格納およびメモリからレジスタへのロードの回数の和の最小化）を求める問題である．これはコントロー

ルフローグラフがSP項により多項式データ型として現されるため，前記の最適化定理を応用することで実現される．

その他の話題

研究を進めていると，副作用としてテーマとはやや異なる成果もでてくる．たとえば，

- 書換え系における基本的な性質（海外論文誌1，2，国際会議1，3など．このうち前者二つについては，さきがけ開始前に主たる研究成果は得ていたが，論文化における改訂作業をさきがけ開始後に行った），
- ω言語の正規性とWQOとの関係（海外論文誌4），
- コントロールフロー解析についてより伝統なモデル検査に基づく自動生成（国内論文誌4）
- 解析の定式化（口頭発表6）

などの結果も得た．

3　今後の展開

昨年4月のみずほ銀行統合時のオンラインシステム，本年3月の航空管制システムのダウンにみられるように社会的波及効果の大きい大規模システムの信頼性を高めることは急務である．しかし，従来から知られる論理的な正しさを示す検証法は，人間のガイド抜きには応用することが難しく，高い検証コストに見合うシステムは数が限られる．

今後，本研究では，前記のSP項のアイデアを用いて，大規模なシステムに適用することを想定したプログラム解析の自動生成系の実装をめざす．すなわち，大規模システムの論理的正しさの保証のかわりに，バグの生じやすい様々な状況（変則性）を検出するプログラム解析を自動生成することにより漸近的にプログラムの信頼性を高めていこうという発想である．その際に，SP項は実用的な効率をもつプログラム解析の実現においてキーとなる可能性を秘めていると期待している．つまり，木幅が3程度でもかなりの比率で実際のプログラムがカバーできる可能性がある．しかも，SP項への変換が失敗するかどうか（つまり木幅が3以下であるかどうか）の判定はかなり速く実行することができるので，失敗した場合は通常の方法を用いればよい．実際に，JAVAなどでは90%～95%のプログラムのコントロールフローグラフが木幅が3以下であるという測定結果も知られている．

SP項の概念は，まだ得たばかりであり，理論面でも整備しなくてはいけない問題が残っている．たとえば，ＳＰ項は始代数ではないので，現時点では自動生成のための仕様（関数

型プログラムで記述される）の正しさはユーザーの責任となっている．SP 項の完全な公理化を得ることができれば，それに対する自動的なサポートも可能になる．現在では，まだ部分的結果（口頭発表１，２，７，投稿中２）しか得ていないので，それを明確にしたい．

また，木幅が大きくなるにつれて，やはり仕様記述（プログラミング）が複雑になるが，「木幅が２の場合の仕様記述＋到達性の記述」により包括的なプログラミングが可能になる可能性があり，これも明確にしていきたい．

4 成果リスト

論文誌 (国際)

1. Ken Mano, Mizuhito Ogawa, Unique Normal Form Property of Compatible Term Rewriting Systems - A New Proof of Chew's Theorem -, Theoretical Computer Science, 258 (1-2), pp.169-208, 2001.
2. Zurab Khasidashvili, Mizuhito Ogawa, Vincent van Oostrom. Perpetuality and Uniform Normalization in Orthogonal Rewrite Systems. Information and Computation, 164 (1), pp.118-151, 2001.
3. Mizuhito Ogawa. A Linear Time Algorithm for Monadic Querying of Indefinite Data over Linearly Ordered Domains. Information and Computation, 186(2), pp.236-259, 2003, Fourth International Symposium on Theoretical Aspects of Computer Science special issue.
4. Mizuhito Ogawa. Well-Quasi-Orders and Regular ω-languages. Theoretical Computer Science 掲載予定，Third International Colloquium on Words, Languages and Combinatorics special issue.

論文誌 (国内)

1. 篠埜 功, 胡 振江, 武市 正人, 小川 瑞史. ナップサック問題およびその発展問題の統一的解法, コンピュータソフトウェア 18 (2), pp.59-63, 2001.
2. 篠埜 功, 胡 振江, 武市 正人, 小川 瑞史. 最大重み和問題の線形時間アルゴリズムの導出. コンピュータソフトウェア 18 (5), pp.1-17, 2001.
3. Isao Sasano, Zhenjiang Hu, Masato Takeichi, Mizuhito Ogawa. Derivation of Linear Algorithm for Mining Optimized Gain Association Rules. コンピュータソフトウェア 19 (4), pp.39-44, 2002.
4. 山岡裕司, 胡振江, 武市正人, 小川瑞史．モデル検査技術を利用したプログラム解析器

の生成ツール．情報処理学会論文誌:プログラミング 掲載予定.

国際会議 (査読付)

1. Zurab Khasidashvili, Mizuhito Ogawa, Vincent van Oostrom. Uniform Normalization beyond Orthogonality. Proceedings of the 12th International Conference on Rewriting Techniques and Applications (RTA01), Lecture Notes in Computer Science 2051, pp.122-136, May 2001, Springer-Verlag.
2. Mizuhito Ogawa. Generation of a Linear Time Query Processing Algorithm Based on Well-Quasi-Orders. Proceedings of the Fourth International Symposium on Theoretical Aspects of Computer Software (TACS2001), Lecture Notes in Computer Science 2215, pp.283-297, October 2001, Springer-Verlag.
3. Mizuhito Ogawa. Call-by-Need Reductions for Membership Conditional Term Rewriting Systems. The 3rd International Workshop on Rewriting Strategies in Rewriting and Programming (WRS03), June 2003, Electronic Notes in Theoretical Computer Science 86(4) 掲載予定, Elsevier.
4. Mizuhito Ogawa, Zhenjiang Hu, Isao Sasano. Iterative-Free Program Analysis, Proceedings of the 8th ACM SIGPLAN International Conference on Functional Programming (ICFP03), pp.111-123, August 2003, ACM Press.

投稿中

1. Isao Sasano, Mizuhito Ogawa, Zhenjiang Hu. Derivation for Maximum Marking Problems with Accumulation, October 2003.
2. Mizuhito Ogawa. Complete Axiomatization for an Algebraic Construction of Graphs. October 2003.

口頭発表

1. Mizuhito Ogawa. Complete Axiomatization for an Algebraic Construction of Graphs. Proceedings of the First Asian Workshop on Programming Languages and Systems pp.199-207, Simgapore, December 2000.
2. Mizuhito Ogawa. Complete Axiomatization of an Algebraic Construction of Graphs. 5th Program Transformation Workshop, Yokohama Techno Tower Hotel Famiel, Yokohama, Japan, March 2001.

3. 小川瑞史. 非線型なメンバーシップ条件付項書換系のNVNF-逐次性. 日本ソフトウェア科学会第18回大会, September 2001.
4. Isao Sasano, Zhenjiang Hu, Masato Takeichi, Mizuhito Ogawa. Derivation of Linear Time Algorithm for Mining Optimized Gain Association Rules. 日本ソフトウェア科学会第18回大会, September 2001.
5. Mizuhito Ogawa. Complete Axiomatization for an Algebraic Construction of Graphs. Proceedings of The RIEC International Symposium on Rewriting in Proof and Computation (RPC'01), pp.199-207. Itsutsubashi Kaikan, Sendai, October 2001. RIEC report.
6. Mizuhito Ogawa. Abstract Interpretation over Infinite Abstract Domains. Proceedings of the 2nd Asian Workshop on Programming Languages and Systems, pp.183-191, KAIST, Daejeon, Korea, December 2001, ROPAS Techenical Memorandum 2001-16.
7. Mizuhito Ogawa, Complete Axiomatization of an Algebraic Construction of Graphs. Workshop on Proving, Solving and Computing, KKR Hotel Tokyo, Tokyo, Japan, Feburary 2002.
8. Mizuhito Ogawa, Efficient Algebraic Data Structure for Graphs: Tentative Survey. 2002 IEEE International Conference on Systems, Man and Cybernetics (SMC02), Hammamet, Tunisia, October 2002.
9. Mizuhito Ogawa. Algebraic Construction of Graphs with Bounded Tree Width and Its Applications, - Catamorphic Appraoch to Program Analyses -. Proceedings of the 3rd Asian Workshop on Programming Languages and Systems, pp.58-73, Shanghai Jiao Tong University, Shanghai, China, November 2002.
10. 山岡裕司, 胡振江, 武市正人, 小川瑞史. モデル検査技術を利用したプログラム解析器の生成ツール. 情報処理学会 第42回プログラミング研究会, January 2003.

学位論文

小川瑞史. 関数型プログラムの自動解析・検証・生成. 東京大学大学理学系研究科（情報科学）, April 8, 2002授与.

謝辞

さきがけの恵まれた研究環境を与えて頂いたことに対し, 片山卓也研究総括, アドバイザの皆様, 科学技術振興機構の皆様に感謝いたします.