数理解析研究所講究録
第 1568 巻 2007 年 1-6

項置換による限定子除去アルゴリズムにおける論理式簡略化について

屋並 仁史
YANAMI HITOSHI*
(株) 富士通研究所/(独) 科学技術振興機構
FUJITSU LABORATORIES LTD./CREST, JST†

穴井 宏和
ANAI HIROKAZU
(株) 富士通研究所/(独) 科学技術振興機構
FUJITSU LABORATORIES LTD./CREST, JST‡

1 はじめに

「ものづくり」における設計問題を考える際、個々の制約をパラメタを文字変数とする多項式を用いた方程式、不等式で、また、全体の制約条件はそれらの方程式、不等式を原子式としてもつような一階述語論理式で表現できることが多い。このようにして与えられた制約式から限定子 $(\forall, \exists)$ の付いた記号を除去し、パラメタのみの同値な関係式を導くアルゴリズムを限定子除去法 (quantifier elimination, QE) という。近年、産業界で生じる諸問題に QE を応用する試みが盛んになってきている [9, 10]。

我々は数式処理ソフト Maple 上で、実代数制約式 (first-order formula over the reals) を解くためのツールボックス SyNRAC(Symbolic-Numeric toolbox for Real Algebraic Constraints) を開発している [2]。本稿では項置換による限定子除去アルゴリズムの効率に大きな影響を及ぼす、式の簡略化について SyNRAC を用いて実験・考察する。

2 背景

ある言語上の理論 T において、任意の一階述語論理式に対して、それと等価な、限定子のない論理式 (quantifier-free formula) が存在するとき、T で限定子除去可能である、と言う。前節で述べたように工学上の問題を論理式として表現し、限定子除去法 (QE) によって解こうとする場合、T として実閉体の基本理論 (elementary theory of real closed fields) を考えると、問題の定式化、計算機による解法ともに都合がよい。この時、入力論理式の quantifier-free 部は通常整数係数多項式による等式または不等式を論理演算子 $(\vee, \wedge, \neg, \Rightarrow, \Leftrightarrow$ など) で結合したものとして表される。論理式の中に出てくる等式や不等式は原子式

*This work has been partially supported by CREST, Japan Science and Technology Agency.
†yanami@labs.fujitsu.com
‡anai@jp.fujitsu.com

(atomic formula) と呼ばれる。このような設定の下で、与えられた入力論理式に対して、それと等価な限定子無しの論理式を出力することを実代数制約式を解く、と言う。

ここで QE の歴史をごく簡単に述べる。最初にこの問題に大きな貢献をしたのは A. Tarski である。Tarski は 1930 年代に、実閉体の基本理論で quantifier elimination 可能であることを示した。Tarski は QE アルゴリズムも初めて提示した[1)]。1975 年、G. E. Collins が Cylindrical Algebraic Decomposition を利用した QE アルゴリズム (QECAD) を提案した [11]。このアルゴリズムは Tarski のものを計算量的に大きく改良したが、当時の計算機環境では実装して問題を解くにはまだまだ厳しかった。その後 80 年代に入り、Collins のアルゴリズムの改良が次々に発表された。CPU 処理速度の向上、メモリ容量の増加も後押しとなり、1990 年前後から計算機への実装が試みられるようになった。

QECAD アルゴリズムは、任意の論理式を入力として受け付けるため、general QE と呼ばれる。これに対して、特別な形の入力論理式のみを受け付け、そういう式のみに特化した効率的な QE アルゴリズムを考える、という研究もある。それらは special QE と呼ばれる。次節では special QE の一つを取り上げる。

3 QE by Virtual Substitution

本節では special QE アルゴリズムの一つである、項置換による限定子除去法 (quantifier elimination by virutual substitution, QEVS) を概説する。1980 年代後半に、次元の低い入力制約式を解くアルゴリズムとして項置換による限定子除去法が V. Weispfenning [3] により提案された。その後、[4, 8, 5] などによりアルゴリズムは改良された。このアルゴリズムでは

$$\mathrm{Q}x\ \varphi(p_1,\ldots,p_m,x), \tag{1}$$

ただし $\mathrm{Q} \in \{\forall, \exists\}$、という形の論理式を解くことが基本となる。限定子一つを除去できれば、一般の入力

$$\mathrm{Q}_1x_1\cdots\mathrm{Q}_nx_n\varphi(p_1,\ldots,p_m,x_1,\ldots,x_n) \tag{2}$$

は Q_nx_n から順に一つずつ除去すればよい。この時、限定子を 1 個除去する過程を 1 ラウンドと呼ぶことがある。また、(1) で $\mathrm{Q}=\forall$ の時は $\forall x\varphi(x)\Longleftrightarrow\neg(\exists x\neg\varphi(x))$ により全称記号を存在記号に書き換え、さらに quantifier-free 部 $\neg\varphi(x)$ はド・モルガンの法則と原子式の書き換えを使って否定を含まない論理式に書き直すことができる。したがって、

$$\exists x\varphi(p_1,\ldots,p_m,x) \tag{3}$$

の形の入力論理式が解ければ一般の入力 (2) を解くことができる。

存在命題 (3) は限定子付き変数を実軸上を動かしながら、quantifier-free 部が真となる点があるか否かが問題となる。Weispfenning は x を含まない項の有限集合 S を具体的に構成して

$$\exists x\varphi(x)\Longleftrightarrow\bigvee_{t\in S}\varphi(x//t) \tag{4}$$

と書き換えられることを示した[2)]。この集合 S を $\exists x\varphi(x)$ の elimination set と言う。このアルゴリズムは elimination set をいかに小さく取れるかが効率化のカギになる。具体的な S の取り方、$|S|$ を小さくする工夫については [3, 4, 8, 5] を参照されたい。

[1)] Tarski のアルゴリズムは elementary recursive ではなく、計算効率という点では難があった。

[2)] $\varphi(x//t)$ は $\varphi(x)$ の中の文字 x を全て項 t で書き換えて（必要ならば式の変形を経て）得られる論理式である。

4 論理式の簡略化

前節で紹介した QEVS は (4) 式からも分かるように限定子を除去することにより入力論理式の quantifier-free 部に現れる限定記号付変数を他の項で置換したものがいくつか[3]生じ、それらの ∨ 結合が次のラウンドの quantifier-free 部となるため、計算過程で quantifier-free 部が急速に長くなる。そのまま計算を進めるとメモリ不足で計算が止まってしまうこともしばしばある。また、最後まで計算が進んだとしても、出力が長い論理式になった場合に人間が見て理解しにくいこともある。したがって QEVS では適宜 quantifier-free 部を簡潔な形に書き換えることが重要である。これを論理式の簡略化と呼ぶ。しかし、論理式の簡略化もコストのかかる計算であり、QE アルゴリズム内のどの位置で、どの程度の簡略化アルゴリズムを適用するかにより、全体の計算時間に大きな影響が出る。

本稿では実験的に 3 種類の簡略化アルゴリズムを用意し、簡略化する場所も 3 箇所用意していくつかの組み合わせで例題を解いて計算時間を比較した。簡略化する場所としては次の 3 ヶ所を考えた：

t) quantifier-free 部を elimination set に属する元で置換した後、

r) 限定子を 1 個除去した後、

l) 解を出力する直前。

また、簡略化アルゴリズムは次の 3 つを用意した：

0) 何もしない、

1) quantifier-free 部のうち flat-part[4] だけを簡略化する、

2) quantifier-free 部全体を簡略化する。

どの場所でどの程度の簡略化をするかを t, r, l と 0, 1, 2 を組にして並べることで表す。例えば、$t0r0l2$ は出力直前で論理式全体を簡略化し、それ以外の場所では簡略化は全く行わないことを意味する[5]。

計算機実験の例題としては

$$F_n = \exists x_1 \cdots \exists x_n \; (m < a_1x_1 + a_2x_2 + \cdots + a_nx_n < M) \tag{5}$$

を用いた。その理由として、いくつかの条件の ∧ 結合の存在命題は工学上の問題でよく見られる形であること[6]、また、簡略化がうまく働けば QE の結果 F_n は

$$\begin{aligned} F_n \iff & (a_1 \neq 0 \;\wedge\; m < M) \\ & \vee (a_1 = 0 \;\wedge\; a_2 \neq 0 \;\wedge\; m < M) \\ & \vee \cdots \\ & \vee (a_1 = 0 \;\wedge\; \cdots \;\wedge\; a_{n-1} = 0 \;\wedge\; a_n \neq 0 \;\wedge\; m < M) \\ & \vee (a_1 = 0 \;\wedge\; \cdots \;\wedge\; a_{n-1} = 0 \;\wedge\; a_n = 0 \;\wedge\; m < 0 \;\wedge\; 0 < M) \end{aligned}$$

のような形になるので答えの検証が容易であること、が挙げられる。

---

[3] S は最大で論理式に含まれる原始式の数程度になる。

[4] quantifier-free 部の中で原子式が ∨ または ∧ のみで結合されている部分を指す。

[5] これらの記号は互いに排反ではないことに注意。例えば $r2$ (各ラウンド後に quantifier-free 部全体を簡略化) する場合は $l2$ (最終ラウンド後の論理式簡略化) も含んでいる。

[6] $m < f < M$ は $m < f \;\wedge\; f < M$ を略記したものである。

5 実験結果

前節 (5) の形の命題 F_1 から F_{13} について SyNRAC に実装されている QE コマンドを適用した。計算環境は Pentium IV 3.80 GHz CPU, 1.99 GB RAM である。簡略化の場所と種類の組合せとして $t0r0l2$, $t0r1l2$, $t1r0l2$, $t0r2l0$, $t2r0l2$ の 5 種類を選び、計算途中での quantifier-free 部に含まれる原子式の数とプログラムの実行時間を計測した。実験結果をグラフで示したものが Figure 1, Figure 2 である。論理式の大きさは式に含まれる原子式の数で判定した。計算途中で全く簡略化を行わなかった場合 ($t0r0l$␣)、論理式の膨張により $n = 6$ 程度で最終ラウンドに達する前にメモリ不足でプログラムが止まってしまった。

計算実験の結果、quantifier-free 部を項置換した後では何もせず、ラウンドごとに flat-part だけを簡略化し、出力直前に論理式全体に簡略化アルゴリズムを適用する ($t0r1l2$) 組合せが最も計算時間が短かった。ラウンドごとに quantifier-free 部全体に簡略化アルゴリズムを適用する方法は原子式の数を常に小さく保ち続けられるが、簡略化アルゴリズム自体の計算時間が全体の実行時間を遅くしていることが分かる。

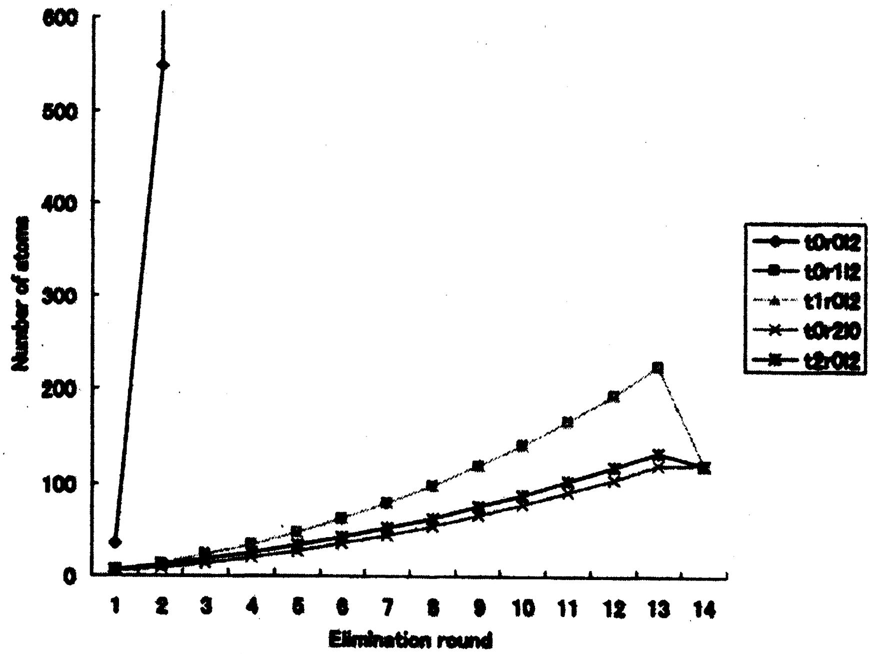

Figure 1: Growth of formulas when solving F_{13}

6 まとめと今後の課題

本稿では項置換による限定子除去アルゴリズム (QEVS) を取り上げ、式の簡略化戦略について実験・考察した。QEVS では計算過程で論理式が指数的に長くなるため、途中で簡略化アルゴリズムを入れないとすぐに手に負えなくなってしまうことが分かった。しかし、ラウンド毎に論理式全体に対して簡略化を行うのは度が過ぎていることも見てとることができた。

今後は他のタイプの問題でも今回と同様のことが言えるのか、それとも入力論理式の違いにより、簡略化戦略を変える必要があるのかを調べたい。また、本稿で取り上げた例題のように出力がきれいな形には

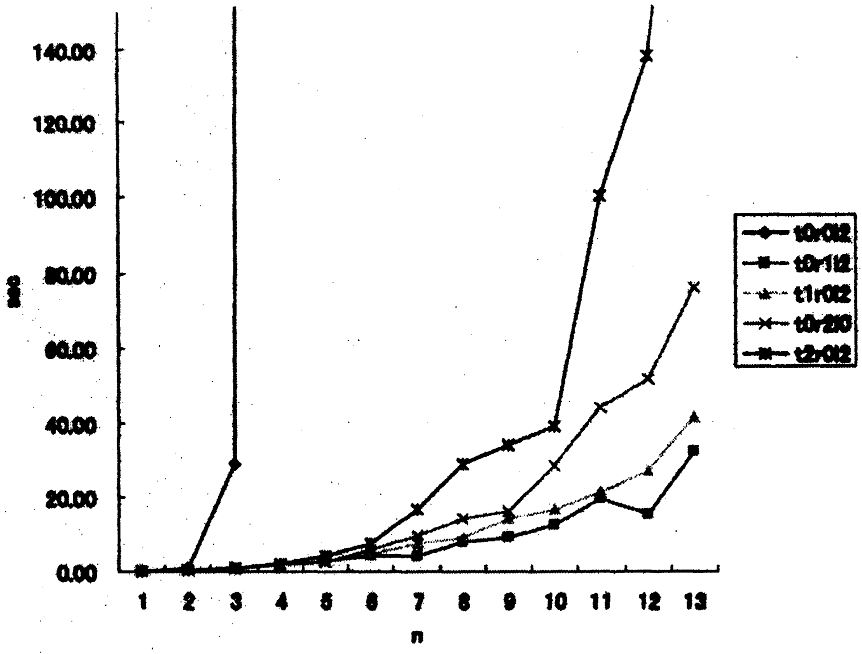

Figure 2: Solving time, F_1 through F_{13}

ならない場合に、論理式に含まれる原子式の数が“式が簡略であること”を測るのに適当な指標と言えるのか、他に良い指標は考えられないか、という問題も残されている。

References

[1] Anai, H., Yanami, H.: SyNRAC: A maple-package for solving real algebraic constraints. In: Proceedings of International Workshop on Computer Algebra Systems and their Applications (CASA) 2003 (Saint Petersburg, Russian Federation), P.M.A. Sloot et al. (Eds.): ICCS 2003, LNCS 2657, Springer (2003) 828–837

[2] Yanami, H., Anai, H.: Development of SyNRAC – formula description and new functions. In: Proceedings of International Workshop on Computer Algebra Systems and their Applications (CASA) 2004 : ICCS 2004, LNCS 3039, Springer (2004) 286–294

[3] Weispfenning, V.: The complexity of linear problems in fields. Journal of Symbolic Computation **5** (1988) 3–27

[4] Loos, R., Weispfenning, V.: Applying linear quantifier elimination. The Computer Journal **36** (1993) 450–462 Special issue on computational quantifier elimination.

[5] Weispfenning, V.: Quantifier elimination for real algebra—the quadratic case and beyond. Applicable Algebra in Engineering Communication and Computing **8** (1997) 85–101

[6] Hong, H.: Quantifier elimination for formulas constrained by quadratic equations. In Bronstein,

M., ed.: Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC 93), Kiev, Ukraine, ACM, ACM Press (1993) 264–274

[7] González-Vega, L.: A combinatorial algorithm solving some quantifier elimination problems. In Caviness, B., Johnson, J., eds.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation. Springer, Wien, New York (1998) 365–375

[8] Weispfenning, V.: Applying quantifier elimination to problems in simulation and optimization. Technical Report MIP-9607, FMI, Universität Passau, D-94030 Passau, Germany (1996) To appear in the Journal of Symbolic Computation.

[9] Anai, H., Hara, S.: Fixed-structure robust controller synthesis based on sign definite condition by a special quantifier elimination. In: Proceedings of American Control Conference 2000. (2000) 1312–1316

[10] Anai, H., Yanami, H., Sakabe, K., Hara, S.: Fixed-structure robust controller synthesis based on symbolic-numeric computation: design algorithms with a cacsd toolbox (invited paper). In: Proceedings of CCA/ISIC/CACSD 2004 (Taipei, Taiwan). (2004) 1540–1545

[11] Collins, G.E.: Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In Caviness, B., Johnson, J., eds.: Quantifier Elimination and Cylindrical Algebraic Decomposition. Texts and Monographs in Symbolic Computation. Springer, Wien, New York (1998) 85–121