5F−1

等式理論の完備化手続き自動化の検討 —RMS利用による効率化—†

近藤 久　栗原 正仁　大内 東
北海道大学工学部

1 はじめに

項書き換えシステム (*term rewriting system*:TRS) の重要な性質として停止性 (*termination*) と合流性 (*confluence*) がある．これらの性質の検査の自動化は，プログラムの合成，変換，検証などにおいて重要な役割を果たすが，この問題は一般に決定不能であることが知られ，いくつかの十分条件が知られているだけである．その中でもKnuthとBendixによって提案された完備化手続き[1]は停止性と合流性をみたす完備 (*complete*) なTRSを生成する手続きであり，完備なTRS:Rは，

- 唯一の計算結果 (正規形 (*normal form*)) が有限ステップで求まる．
- Rを向き付けられていない等式の集合とみたとき，任意の等式 $s=t$ が Rの論理的帰結かどうかを効率的に決定することができる．

という性質をもつため，この手続きは非常に有効である．ただし，現在の完備手続きは停止性を保証する項の集合上の半順序をあらかじめ利用者に要求したり，実行中に問い合わせを行ない，完全な自動化はなされていない．

本稿では，完備化続きの自動化に対する問題点とその解決法及び効率化のためのRMS利用について述べる．

2 完備化手続き

完備化手続きは，等式の集合 Eと簡約順序 (*reduction ordering*) が与えられたとき，可能ならば完備なTRS:Rを生成する．これは，"すべての危険対 (*critical pair*)$\langle s,t\rangle$ が $s \to_R^* u \leftarrow_R^* t$ となるなら，停止するTRSは合流性をみたす．" さらに "停止性は簡約順序によって等式を向き付けることによって保証される．" ということに基づいている．抽象的には，次の推論規則と見なすことができる[2]．

Delete:	$(E\cup\{s\leftrightarrow s\};R)\vdash(E;R)$	
Compose:	$(E;R\cup\{s\to t\})\vdash(E;R\cup\{s\to u\})$	if $t\to_R u$
Simplify:	$(E\cup\{s\leftrightarrow t\};R)\vdash(E\cup\{s\leftrightarrow u\};R)$	if $t\to_R u$
Orient:	$(E\cup\{s\leftrightarrow t\};R)\vdash(E;R\cup\{s\to t\})$	if $s\succ t$
Collapse:	$(E;R\cup\{s\to t\})\vdash(E\cup\{u\leftrightarrow t\};R)$ if $s\to_R u$ by $l\to r\in R$ with $s\rhd l$	
Deduce:	$(E;R)\vdash(E\cup\{s\leftrightarrow t\};R)$ if $s\leftarrow_R u\to_R t$	

これらの推論規則を適用した結果，等式の集合 Eが空になるならば，完備な Rが得られる．

3 完備化手続きの自動化の問題点

完備化手続きの自動化を行なうためには，推論規則Orientで用いる簡約順序を実行中に求めなければならない．本稿では，簡約順序として辞書式経路順序 (*lexicographic path ordering*:LPO)[3]を用いる．LPOを用いるためには項を構成する演算子の集合上に選好順序 (*precedence*) を決定しなければならない．この選好順序の決定によっては完備化に失敗してしまう．例えば，等式の集合 Eを

$$\begin{cases}(x+y)+z\leftrightarrow x+(y+z) & 1)\\ f(x)+f(y)\leftrightarrow f(x+y) & 2)\end{cases}$$

とし，選好順序を$+\succ f$とするとOrientによって2)は書き換え規則 $r_2: f(x)+f(y)\to f(x+y)$ となり，1)は書き換え規則 $r_1:(x+y)+z\to x+(y+z)$ となる．r_1，r_2からDeduceを用いて新たな等式を求め，Orientを適用すると，$r_3: f(x+y)+z\to f(x)+(f(y)+z)$ が得られる．さらに，r_2，r_3から同様の推論によって新たな書き換え規則 $f^2(x+y)+z\to f^2(x)+(f^2(y)+z)$ が得られる．この推論は無限に続き，書き換え規則の無限列を生じる．これは，完備化を通常のバックトラック法を用いて行なう場合に生じる問題点であり，通常のバックトラック法を用いて選好順序を決定することはできない．

上記の問題点を解決するには，パラレルに完備化を行なわなければならない．まず，等式 $s=t$ を $s\to t$ と $t\to s$ にすることのできるすべての選好順序を求める．上記の例で，1)式は選好順序{ }のもとで，$(x+y)+z\to x+(y+z)$ に向き付けることができ，逆に向き付けることは任意の選好順序のもとで不可能である．2)式は選好順序$+\succ f$のもとで $f(x)+f(y)\to f(x+y)$，$f\succ+$のもとで $f(x+y)\to f(x)+f(y)$ に向き付けることができる．したがって，2つの完備化列 C_1，C_2が生じる．つまり，

C_1 $(E_0;R_0)\vdash(E_1;R_0\cup\{(x+y)+z\to x+(y+z)\})\vdash(E_2;R_1\cup\{f(x)+f(y)\to f(x+y)\})$

C_2 $(E_0;R_0)\vdash(E_1;R_0\cup\{(x+y)+z\to x+(y+z)\})\vdash(E_2;R_1\cup\{f(x+y)\to f(x)+f(y)\})$

である．さらに C_1，C_2をパラレルに処理することによって，最終的に C_2が完備化に成功し，完備なTRS:$R=\{(x+y)+z\to x+(y+z), f(x+y)\to f(x)+f(y)\}$ を得る．

この例では比較的簡単に完備化を行なうことができるが，実際には探索空間が非常に大きなものとなり，組み合わせ的爆発が生じる．例えば，ある文脈で行なった推論が他の文脈で生じる場合がある．これを避けることがこの方法を実用的なものとする．

4 RMS利用による効率化

4.1 RMS

RMS(*Reason Maintenance System*) は3章で述べた推論(探索)の問題点を避け，効率を改善するために人工知能の分野で提案された一手法である．

RMSを利用することによって全体的な問題解決器は "問題領域に関する推論を行う部分(PS)" と "PSの推論を記録する部分(RMS)" に分けられる．つまり，RMSはPSの行った推論を記録し，PSの推論を効率よく行わせるための援助を行う機構である．

† On automatization of completion procedure of equational theory –Efficient completion based on RMS–
Hisashi KONDOH, Masahito KURIHARA, Azuma OHUCHI
Faculty of Engineering, Hokkaido University

PS は RMS に対して問い合わせを行うことによって，これまで記録された推論から得られるいくつかの情報を受け取り，PS は受け取った情報を用いて効率よく推論を行うことができる．

RMS として de Kleer の ATMS(*Assumption-based TMS*)[4] がよく知られている．ATMS は PS の推論を ATMS ノードと呼ばれるデータを用いて記録する．ATMS ノードは概念的に次のように表される．

$$\gamma_{datum} : (datum, label, justification)$$

datum は PS が推論して得た事実，label(ラベル) はその事実を成り立たせる環境 (environment) の集合である．環境は PS が推論を行うために用いた仮定の集合である．justification(正当化) はこの ATMS ノードが他の ATMS ノードからどのように導かれたかを表す依存関係である．ATMS ではラベルを持たせることによって多重文脈を扱うことができる．

4.2 完備化手続きにおける RMS

本 RMS は ATMS に基づき，RMS ノードは概念的に ATMS と同様である．仮説は選好順序対と書き換え規則とした．

$$(s = t \, or \, s \to t, \{\{(f, g), s' \to t' \cdots\}, \cdots\}, \{(\gamma_1, \gamma_2, \cdots), \cdots\})$$

ここで $s = t$ あるいは $s \to t$ は 2 章の推論規則を適用して得られた等式，書き換え規則である．$\{(f, g), s' \to t' \cdots\}$ は等式あるいは書き換え規則を成り立たせる環境であり，選好順序対，書き換え規則の集合である．$(\gamma_1, \gamma_2, \cdots)$ は正当化である．datum スロットが等式であるノードを等式ノード，書き換え規則であるノードをルールノードと呼ぶ．本完備化手続きで扱う各完備化列 (E_i, R_i) はそれぞれ，$E_i \subseteq$ 現在の全等式ノードの集合，$R_i \subseteq$ 現在の全ルールノードの集合を持つ．

以下，各推論規則の適用による RMS ノードについて具体的に述べる．

(1)Delete

等式ノードの等式 $s = t$ が $s \equiv t$ であるノードを RMS データベースより削除する．このとき，$s \equiv t$ を含むすべての完備化列 $(E_i \cup \{s \equiv t\}, R_i)$ を (E_i, R_i) に変更する．

(2)Compose

Compose した結果，新たなルールとなるならば正当化にしたがって新ルールノードを作成する．

例えば，$(l \to r, \{\ldots\}, \{\ldots\})$ のノードに対して $l' \to r'$によって Compose 可能であるならば，$(l \to r', \{Env1\}, \{(\gamma_{l \to r}, \gamma_{l' \to r'})\})$ を新たなルールノードとする．ここで $Env1$ は正当化から得られる環境である．

(3)Simplify

等式ノードの正当化にしたがって Simplify し，Compose と同様の処理を行なう．

(4)Orient

等式ノードの等式 $s = t$ を $s \to t$ と $t \to s$ のルールとするため LPO によって向き付ける．例えば，$\gamma_{s=t} : (s = t, \{Env1, Env2\}, \{\ldots\})$ の等式ノードを向き付けたとする．このとき $s \to t$ と $t \to s$ を仮説とし，$s \succ_{lpo} t$，$t \succ_{lpo} s$ とすることのできる選好順序対をすべて求める．$s \succ_{lpo} t$ には $\{f \succ g\}$ で $t \succ_{lpo} s$ には $\{h \succ g\}$ で向き付けれたとすると，PS は 2 つの正当化，$\gamma_{s=t}, \Gamma_{s \to t}, \gamma_{f \succ g} \Rightarrow \gamma_{s \to t}$と$\gamma_{s=t}, \Gamma_{t \to s}, \gamma_{h \succ g} \Rightarrow \gamma_{t \to s}$を RMS に与える．この結果

$$(s \to t, \{Env3, Env4\}, \{(\gamma_{s=t}, \Gamma_{s \to t}, \gamma_{f \succ g})\})$$

$$(t \to s, \{Env5, Env6\}, \{(\gamma_{s=t}, \Gamma_{t \to s}, \gamma_{h \succ g})\})$$

ができる．ここで $Env3$ から $Env6$ は正当化から計算された環境である．PS は $s = t$ を含むすべての完備化列 $(E_i \cup \{s = t\}, R_i)$ を $(E_i, R_i \cup \{$新ルールノード$\})$ に変更する．この操作によって完備化列は 2 倍になる．

(5)Collapse

Compose，Simplify と同様に，ルールノードの正当化にしたがって Collapse し，新たな等式ノードとする．

(6)Deduce

2 つのルールノードから Deduce によって新たな等式ノードを得る．例えば，$(l \to r, \{Env1, Env2\}, \{\ldots\})$ と $(l' \to r', \{Env3\}, \{\ldots\})$ から危険対を求めるとする．このとき求まった危険対を $s = t$ とすると，PS は正当化$\gamma_{l \to r}, \gamma_{l' \to r'} \Rightarrow \gamma_{s=t}$を RMS に与え，新たな等式ノードとして $(s = t, \{Env1 \cup Env3, Env2 \cup Env3\}, \{(\gamma_{l \to r}, \gamma_{l' \to r'})\})$ を得る．このとき $l \to r$と $l' \to r'$を含むすべての完備化列 (E_i, R_i) を $(E_i \cup \{$新等式ノード$\}, R_i)$ に変更する．

4.3 ラベル更新

本 RMS におけるラベル更新を例を用いて述べる．例えば，2本のルール $l \to r$と $l' \to r'$から Deduce によって等式 $s = t$ を得たとし，$s = t$ は等式ノード$\gamma_{s=t} : (s = t, \{Env1\}, \{\ldots\})$ として RMS データベースに登録されているとする．このとき，ルールノードをそれぞれ，

$\gamma_{l \to r}$: $(l \to r, \{Env2\}, \{\ldots\})$

$\gamma_{l' \to r'}$: $(l' \to r' \{Env3, Env4\}, \{\ldots\})$

とする．PS は正当化として$\gamma_{l \to r}, \gamma_{l' \to r'} \Rightarrow \gamma_{s=t}$ を与える．このとき RMS は$\gamma_{l \to r}$と$\gamma_{l' \to r'}$ のラベル中の環境の和集合を計算し，$\{Env2\} \cup \{Env3\}$ と $\{Env2\} \cup \{Env4\}$ が$\gamma_{s=t}$のラベルに加わる．さらに$\gamma_{s=t}$を前件に持つ正当化によって正当化されているノードがあれば，$\gamma_{s=t}$に新たに加えられた環境が伝播し，そのノードのラベルが更新される．

5 おわりに

本稿では，完備化手続きの自動化の問題点とその解決法及び RMS 利用の一方法として各推論規則の適用による RMS ノードの操作について述べた．今後の課題として

1. パラレルに処理するための効率的な文脈の切り替え戦略．
2. インプリメント

が挙げられる．

参考文献

[1] Knuth,D.E. and Bendix,P.B., Simple words problems in universal algebras, in: J.Leech, ed., *Computational problems in abstract algebra*, Pergamon Press, Oxford, pp.263-297(1970).

[2] Dershowitz,N., Completion and its applications, in:Ait-Kaci and M.Nivar, eds., *Resolution of Equations in Algebraic Structures, Vol.II: Rewriting Techniques*, Academic Press, pp.31-85(1989).

[3] Dershowitz,N., Termination of rewriting, *J.Symbolic Computation* **3**, pp.69-116(1987).

[4] de Kleer,J., Assumption-based TMS *Artificial Intelligence* **28** pp.127-162(1986).