極大完備化に基づく等式定理の自動証明

萩原 崇央　青戸 等人

項書き換えシステムの代表的な自動証明法として，クヌース・ベンディクス完備化がある．失敗無し完備化 (Bachmair ら，1989) はクヌース・ベンディクス完備化を利用した等式の定理自動証明法である．一方，完備化の新しい手法として，極大完備化 (Klein&広川，2011) が提案されている．極大完備化は，パラメータとして簡約順序を必要としない，SMT ソルバを活用した完備化手法である．本報告では，極大完備化を失敗無し極大完備化へと拡張する．具体的には，等式判定手続きへの変更，変数のスコーレム定数への置き換え，向き付け不能な等式による書き換え手続きの追加，柔軟な危険対生成手続きの導入を行い，極大完備化に基づく等式定理の自動証明システムを実現した．また，実装したシステムによる定理証明実験の結果を報告する．

1　はじめに

等式論理の定理自動証明法として，項書き換えシステム (TRS) による完備化は代表的なものである．完備な TRS とは，停止性と合流性をもった TRS のことであり，完備化とは，与えられた等式集合から，等価で完備な TRS を構成することである．等価で完備な TRS を用いると，等式定理の判定を行うことができる．

完備化の代表的な手法として，クヌース・ベンディクス (KB) 完備化 [4] が知られている．KB 完備化はパラメータとして簡約化順序を用いるが，KB 完備化が成功するかどうかは，用いる簡約順序に依存する．

Klein と広川 [3] によって提案された極大完備化は，改良の著しい SMT ソルバを活用した完備化手法であり，KB 完備化と遜色ない強力さを持つことが実験的にも確認されている．極大完備化では，最終的な完備な TRS を発見するまで，簡約順序を固定しないまま動作する．

KB 完備化においては，与えられた等式の証明を目標に完備化手続きを走らせる，失敗無し完備化 [1] が知られている．失敗無し完備化は，完備化自体が失敗する場合であっても，等式定理証明に成功する場合もあることが知られており，実際には重要な定理証明法である．しかしながら，極大完備化に基づく失敗無し完備化はまだ実現されていない．

本報告では，極大完備化に基づく失敗無し完備化を考案し，KB 完備化の代わりに極大完備化を用いて等式定理の自動証明法を実現する．具体的には，等式判定手続きへの変更，変数のスコーレム定数への置き換え，向き付け不能な等式による書き換え手続きの追加，柔軟な危険対生成手続きの導入を行い，極大完備化に基づく等式定理の自動証明システムを実現した．また，実装したシステムによる定理証明実験の結果を報告する．

2　準備

2.1　項書き換えシステム

本報告で用いる定義や記法について説明する．

関数の集合を F で表し，変数の集合を V で表す．$V(t)$ は項 t 中の変数の集合を表す．項の全体集合を $T(F,V)$ で表す．$|t|$ は項 t のサイズを表す．$t\sigma$ は項 t に対して代入 σ を適用して得られた項を表す．

文脈とは，特別な定数 $\Box$(ホール) をただ一つ含む

Automated Equational Theorem Proving based on Maximal Completion

Takahisa Hagihara,Takahito Aoto, 新潟大学大学院自然科学研究科, Graduate School of Science and Technology, Niigata University.

$$\textit{Deduce}: \quad \frac{\langle E, R\rangle}{\langle E \cup \{s \approx t\}, R\rangle} \quad s \leftarrow_R u \rightarrow_R t \qquad \textit{Orient}: \quad \frac{\langle E \cup \{s \stackrel{\cdot}{\approx} t\}, R\rangle}{\langle E, R \cup \{s \rightarrow t\}\rangle} \quad s \succ t$$

$$\textit{Delete}: \quad \frac{\langle E \cup \{s \approx s\}, R\rangle}{\langle E, R\rangle} \qquad \textit{Simplify}: \quad \frac{\langle E \cup \{s \stackrel{\cdot}{\approx} t\}, R\rangle}{\langle E \cup \{u \approx t\}, R\rangle} \quad s \rightarrow_R u$$

$$\textit{Compose}: \quad \frac{\langle E, R \cup \{s \rightarrow t\}\rangle}{\langle E, R \cup \{s \rightarrow u\}\rangle} \quad t \rightarrow_R u \qquad \textit{Collapse}: \quad \frac{\langle E, R \cup \{s \rightarrow t\}\rangle}{\langle E \cup \{u \approx t\}, R\rangle} \quad s \rightarrow_{R, l \rightarrow r} u$$

図 1　KB 完備化の推論規則

項のことをいう．ホールの位置が p であるような文脈 C を $C[\,]_p$ と記す．$t[]_p$ は項 t の位置 p をホールに置き換えて得られる文脈を表す．$C[t]$ は文脈 C のホールを項 t で置き換えて得られる項を表す．$t|_p$ は項 t の位置 p の部分項を表す．$t \rhd s$ は $\exists\sigma.t|_p = s\sigma$ かつ $t \neq s$ を表す．

2 つの項 l, r が $l \notin V, V(r) \subseteq V(l)$ を満たすとき，$l \rightarrow r$ を書き換え規則とよぶ．さらに，書き換え規則の集合を項書き換えシステム (TRS) という．R を項書き換えシステムとする．書き換え規則 $l \rightarrow r \in R$，代入 σ，文脈 $C[\,]_p$ が存在して，$s = C[l\sigma]_p$ かつ $t = C[r\sigma]_p$ となるとき，$s \rightarrow_R t$ と記す．$s \rightarrow_R t$ において，用いた書き換え規則 $l \rightarrow r \in R$ を明示する場合には，$s \rightarrow_{R, l\rightarrow r} t$ と記す．$s \stackrel{*}{\rightarrow}_R t$ は $\rightarrow_R$ の反射推移閉包を，$\stackrel{*}{\leftrightarrow}_R$ は $\rightarrow_R$ の等価閉包を表す．

項書き換えシステム R に対して，項 s が正規形であるとは，$s \rightarrow_R t$ となるような t が存在しないときをいう．正規形の集合を NF と記す．$s \stackrel{*}{\rightarrow}_R t \in NF$ となるとき，項 t を項 s の正規形とよび，$s{\downarrow_R}$ で表わす．また，項の集合 T について，$T{\downarrow_R} = \{s{\downarrow_R} \mid s \in T\}$ と定義する．

無限書き換え列 $t_0 \rightarrow_R t_1 \rightarrow_R \cdots$ が存在しないとき，項書き換えシステム R は停止性を持つといい，$SN(R)$ と記す．代入及び文脈に閉じている整礎な項上の半順序を簡約順序という．

R を項書き換えシステム，$l_1 \rightarrow r_1 \in R, l_2 \rightarrow r_2 \in R$ を書き換え規則とする (一般性を失うことなく，共通変数を持たないものとする)．l_2 のある非変数部分項 $l_2|_p$ と l_1 が単一化可能であるとき，位置 p で $l_1 \rightarrow r_1$ が $l_2 \rightarrow r_2$ へ重なるという．代入 σ を $l_2|_p$ と l_1 の最汎単一化子とするとき，項の対 $\langle l_2[r_1]_p\sigma, r_2\sigma\rangle$ を危険対とよび，R の危険対の集合を $CP(R)$ で表す．ただし，$l_1 \rightarrow r_1 = l_2 \rightarrow r_2$ のとき，p は根位置でないものとする．

任意の項 s, t, u について，$s \stackrel{*}{\leftarrow}_R u \stackrel{*}{\rightarrow}_R t$ ならば，ある項 w が存在して，$s \stackrel{*}{\rightarrow}_R w \stackrel{*}{\leftarrow}_R t$ となるとき，項書き換えシステム R は合流性を持つという．項書き換えシステム R が停止性と合流性を持つとき，項書き換えシステム R は完備であるという．

項の等式 $s \approx t$ と $t \approx s$ を同一視するときは $s \stackrel{\cdot}{\approx} t$ と記す．E を等式集合とする．等式 $s \stackrel{\cdot}{\approx} t \in E$，代入 σ，文脈 $C[]_p$ が存在して，$s = C[l\sigma]_p$ かつ $t = C[r\sigma]_p$ となるとき，$s \stackrel{*}{\leftrightarrow}_E t$ と記す．等式 $s \approx t$ が等式集合 E の定理であるとは，$s \stackrel{*}{\leftrightarrow}_E t$ となることをいう．

2.2　クヌース・ベンディクス完備化

完備化の標準的な手法であるクヌース・ベンディクス (KB) 完備化手続き [1] [4] を定義する．KB 完備化では図 1 の推論規則を用いて完備化を行う．図 1 の推論規則による導出を $\vdash$ と表す．$\vdash^*$ と表記した場合，0 回以上の導出を表す．$\langle E, \emptyset\rangle = \langle E_0, R_0\rangle \vdash^* \langle E_n, R_n\rangle = \langle \emptyset, R\rangle$ なる導出が公平であるとは，$\exists i.\langle s, t\rangle \in \bigcap_{i \leq j} CP(R_j)$ ならば，$\exists k.s \approx t \in E_k$ であることをいう．KB 完備化手続きとは，等式集合 E，簡約順序 $\succ$ を入力とし，$\langle E, \emptyset\rangle \vdash^* \langle \emptyset, R\rangle$ となる公平な導出を構成する手続きをいう [4]．

命題 2.1 (KB 完備化手続きの正しさ) 等式集合 E，簡約順序 $\succ$ を KB 完備化手続きの入力とする．このとき，項書き換えシステム R が出力されるならば，R は完備かつ $\stackrel{*}{\leftrightarrow}_E = \stackrel{*}{\leftrightarrow}_R$ となる．

KB 完備化は簡約順序を固定して完備化を行うため，手続きの成功や効率は，用いる簡約順序に大きく依存する．

$$Deduce_2: \quad \frac{\langle E, R\rangle}{\langle E \cup \{s \approx t\}, R\rangle} \quad s \leftrightarrow_{E\cup R} u \leftrightarrow_{E\cup R} t \wedge s \not\succ u \wedge t \not\succ u$$

$$Simplify_2: \quad \frac{\langle E \cup \{s \stackrel{\cdot}{\approx} t\}, R\rangle}{\langle E \cup \{u \approx t\}, R\rangle} \quad s \rightarrow_{E_\succ, l\sigma\to r\sigma} u \wedge s \rhd l$$

$$Compose_2: \quad \frac{\langle E, R \cup \{s \to t\}\rangle}{\langle E, R \cup \{s \to u\}\rangle} \quad t \rightarrow_{E_\succ} u$$

$$Collapse_2: \quad \frac{\langle E, R \cup \{s \to t\}\rangle}{\langle E \cup \{u \approx t\}, R\rangle} \quad s \rightarrow_{E_\succ, l\sigma\to r\sigma} v \wedge s \rhd l$$

図 2　失敗無し完備化の推論規則

2.3　失敗無し完備化

失敗無し完備化[1] は KB 完備化を等式定理証明に拡張した手続きである．

失敗無し完備化は KB 完備化に図 2 の推論規則を追加する．ここで，$E_\succ$ は $E_\succ = \{l\sigma \to r\sigma \mid l \approx r \in E, l\sigma \succ r\sigma\}$ と定義される項書き換えシステムである．また，$Deduce_2$ は $Deduce$ の拡張となっていることに注意する．失敗無し完備化では，危険対の代わりに拡張危険対[1] を用いる．項書き換えシステム R，等式集合 E とし，σ を最汎単一化子，書き換え規則 $l_1 \to r_1 \in R$ もしくは $l_1 \stackrel{\cdot}{\approx} r_1 \in E$ は，位置 p で $l_2 \to r_2 \in R$ もしくは $l_2 \stackrel{\cdot}{\approx} r_2 \in E$ へ重なるとする．このとき，$r_2\sigma \not\succ l_2\sigma$ で $r_1\sigma \not\succ l_1\sigma$ となる等式 $r_2\sigma \approx l_2\sigma[r_1\sigma]_p$ を拡張危険対と呼ぶ．等式集合 $E \cup R$ の間の全ての拡張危険対の集合を $ECP_\succ(E \cup R)$ と表す．

以下では，項 t 中の変数全てを定数化 (スコーレム化) した項を $\hat{t}$ と表す．また，$\mathsf{eq}, \mathsf{True}, \mathsf{False}$ を新しい関数記号とする．$\langle E \cup \{\mathsf{eq}(x,x) \approx \mathsf{True}, \mathsf{eq}(\hat{s},\hat{t}) \approx \mathsf{False}\}, \emptyset\rangle = \langle E_0, R_0\rangle \vdash^* \langle E_n, R_n\rangle = \langle \emptyset, R\rangle$ なる導出が公平であるとは，$\exists i. \langle s,t\rangle \in \bigcap_{i\le j} ECP_\succ(R_j)$ ならば，$\exists k. s \approx t \in E_k$ であることをいう．失敗無し完備化手続きとは，等式集合 E，等式 $s \approx t$，完全な簡約順序 $\succ$ を入力とし，$\langle E, \cup\{\mathsf{eq}(x,x) \approx \mathsf{True}, \mathsf{eq}(\hat{s},\hat{t}) \approx \mathsf{False}\}, \emptyset\rangle = \langle E_0.R_0\rangle \vdash^* \langle E_n, R_n\rangle$,$\mathsf{True} \approx \mathsf{False} \in E_i$ なる公平な導出を構成する手続きをいう．

失敗無し完備化手続きの正しさは文献[1] に与えられている．

命題 2.2 (失敗無し完備化手続きの正しさ) 等式集合 E，等式 $s \approx t$，完全な簡約順序 $\succ$ を失敗無し完備化手続きの入力とする．失敗無し完備化手続きが成功するとき，等式 $s \approx t$ は E の定理である．

3　極大完備化に基づく等式定理証明

3.1　極大完備化

極大完備化は Klein と広川[3] によって提案された完備化手法である．極大完備化では，入力として等式集合 E だけを与え，$\stackrel{*}{\leftrightarrow}_R = \stackrel{*}{\leftrightarrow}_E$ かつ完備な項書き換えシステム R を発見する．極大完備化手続きを以下で与える．

入力：　等式集合 E

出力：　完備かつ $\stackrel{*}{\leftrightarrow}_E = \stackrel{*}{\leftrightarrow}_R$ な項書き換えシステム R

Step 1　$C := E$

Step 2　$R_1, ..., R_k \in max\, T(C)$ を見つける．

Step 3　それぞれの $1 \le i \le k$ について，

(3-1)　$NF_i := (CP(R_i) \cup E)\downarrow_{R_i}$

(3-2)　$NF_i = \emptyset$ なら完備化に成功し，$R := R_i$ を出力して終了．

(3-3)　$NF_i \neq \emptyset$ なら $C_i := NF_i \setminus Id$

Step 4　$C := C \cup \bigcup_{1\le i\le k} C_i$ とし，Step 2 へ戻る．

ここで，

$$Id = \{u \approx u \mid u \in T(F,V)\}$$

$$T(C) = \{R \mid R \subseteq C \cup C^{-1},\, SN(R)\}$$

$$max\{R_1, \ldots, R_n\} = \{R_i \mid \neg\exists R_j.\ R_i \subsetneq R_j\}$$

極大完備化手続きの正しさは文献[3] で与えられている．

命題 3.1 (極大完備化手続きの正しさ) 等式集合 E

を極大完備化手続きの入力とする．項書き換えシステム R が出力されるならば，R は完備かつ $\overset{*}{\leftrightarrow}_E = \overset{*}{\leftrightarrow}_R$ となる．

3.2 失敗無し極大完備化

この節では極大完備化を等式定理証明に拡張する．失敗無し極大完備化手続きを以下に示す．

入力： 等式集合 E，等式 $s \approx t$

出力： 成功

Step 1 $C := E \cup \{\mathsf{eq}(x,x) \approx \mathsf{True}, \mathsf{eq}(\hat{s},\hat{t}) \approx \mathsf{False}\}$

Step 2 $(R_1,\succ_1),\ldots,(R_k,\succ_k) \in maxT(C)$ を見つける．

Step 3 それぞれの $1 \le i \le k$ について

(3-1) $NF1_i := (ECP_{\succ_i}(C \cup R_i) \cup E)\downarrow_{R_i}$

(3-2) $NF2_i := \{(u' \approx v \mid u \rightarrow_{C_{\succ_i},l\sigma\rightarrow r\sigma} u', u \rhd l, u \mathrel{\dot{\approx}} v \in C\}$

(3-3) $\mathsf{True} \mathrel{\dot{\approx}} \mathsf{False} \in NF1_i \cup NF2_i$ なら証明成功．

(3-4) $\mathsf{True} \mathrel{\dot{\approx}} \mathsf{False} \notin NF1_i \cup NF2_i$ なら $C_i := (NF1_i \cup NF2_i) \setminus Id$

Step 4 $C := C \cup \bigcup_{1\le i\le k} C_i$ とし，Step 2 へ戻る．

ここで，$(R_i,\succ_i) \in maxT(C)$ は，$R_i \in maxT(C)$ かつ $\succ_i$ が $R_i \subseteq \succ_i$ なる簡約順序であることを表す．極大完備化手続きとの差分は以下の通りである．

- Step 1 の C の初期値を $C := E$ から $C := E \cup \{\mathsf{eq}(x,x) \approx \mathsf{True}, \mathsf{eq}(\hat{s},\hat{t}) \approx \mathsf{False}\}$ に変更．
- Step 3 における危険対を拡張危険対に変更．
- $Simplify_2$，$Compose_2$，$Collapse_2$ の規則追加のための手続き (3-2) を追加．
- 手続きの終了条件を $\mathsf{True} \mathrel{\dot{\approx}} \mathsf{False} \in NF1_i \cup NF2_i$ かどうかに変更．

以下の定理を容易に示すことが出来る．

定理 3.2 (失敗無し極大完備化手続きの正しさ) 等式集合 E，等式 $s \approx t$ を失敗無し極大完備化手続きの入力とする．このとき，手続きが成功するならば，等式 $s \approx t$ は E の定理である．

4 失敗無し極大完備化の実装

4.1 極大完備化のヒューリスティックス

極大完備化手続きは素朴に実装すると，等式集合 C がとても大きくなるため，計算時間が増え，成功に至らない場合が多い．そこで，極大完備化では，Step 2 で発見する項書き換えシステムの個数 k に上限を設ける．さらに，Step 4 で C に追加する等式の制限を行うことで，効率の良い完備化手続きを実現する [3]．

定義 4.1 (k の上限 [3]) K の初期値を 1 とする．Step 2 で K を用いるように変更し，$k \le K$ とする．Step 4 で K を以下の様に変更する．

$$K = \begin{cases} K+1 & (\bigcup_{1\le i\le k} C_i \subseteq C \text{ のとき}) \\ K & (\text{その他の場合}) \end{cases}$$

定義 4.2 (C のフィルタリング [3]) 項の大きさに関する C のフィルタリング $C^{<d}$ を以下のように定義する．

$$C^{<d} = \{s \approx t \in C \mid |s| + |t| < d\}$$

さらに，C の中からサイズの小さい順に n 個抜き出したものを $C{\uparrow_n}$ と記す．これらのパラメータを用いて Step 4 を以下の様に変更する．

$$C := C \cup (\bigcup_{1\le i\le k} C_i^{<d}){\uparrow_n}$$

我々の実装でも，これらのパラメータは Klein と広川の実装 [3] に従い，$n = 7$，$d = 20$ の値を用いる．

4.2 失敗無し極大完備化のヒューリスティック (1)

極大完備化では，d は定数であったが，失敗無し極大完備化では d を以下の様に変更する．

$$d = \begin{cases} 20 & (m \le 20 \text{ のとき}) \\ m+8 & (\text{その他の場合}) \end{cases}$$

ここで，$m = max\{|s| + |t| \mid s \approx t \in E\}$ とする．これは，$|s| + |t| > 20$ となる等式を含む E に対応するためである．Step 4 で，$\bigcup_{1\le i\le k} C_i \subseteq C$ が 3 回連続した場合，d と K の値を以下の様に変更する．

- $d := d + 2$
- $K := K - 3$

これは，手続きの途中で d 以上の等式しか生成されない可能性に対応するためである．

4.3 失敗無し極大完備化のヒューリスティック (2)

失敗無し極大完備化ではさらに，Step 4 で追加されなかった等式を保存するリストを導入した．これは，小さい等式から C へ追加する機能を強化するためである．失敗無し極大完備化手続きの Step 4 を以下に変更する．

Step 4

$$P := P \cup (\bigcup_{1 \le i \le k} C)^{<d}$$
$$C := C \cup (\overline{P\uparrow_n})$$
$$P := (P \setminus (P\uparrow_n))\uparrow_m$$

ただし，$m = 75$ を用いる．

5 実験結果と考察

提案手法に基づく定理自動証明システム unfMax を実装した．unfMax は関数型言語 SML を用いて実装しており，プログラムの行数は約 2500 行である．また，手続き内で SMT ソルバである Yices1.0 [2] を用いた．

実験には，定理証明の問題集である TPTP 6.4.0 [6]Unit Equality Problem(UEQ) を用いた．極大完備化を基にした定理自動証明システムである maedmax [7] および，多重 KB 完備化を基にした定理自動証明システムである mkbTT [5] と比較した．表 1 に実験結果を示す．unfMax は timeout を 7 分とし，CPU：AMD phenomII x6 2.80GHz，RAM：16GB の PC 上で実験結果である．maedmax，mkbTT の実験結果はウェブページ [7] の引用である．

表 1　等式定理証明の実験結果

	unfMax	maedmax	mkbTT
success	340	522	223
timeout	675	504	762
error	26	15	56

成功個数は mkbTT < unfMax < maedmax となった．maedmax では定理証明に失敗し，unfMax では成功した例は COL003-12，COL003-13，REL49-1，RNG008-4 の 4 例である．

6 まとめと今後の課題

極大完備化を失敗無し完備化に拡張した等式定理証明法である失敗無し極大完備化を提案した．最近同様な極大完備化を基にした等式定理証明システム maedmax が提案されているが，maedmax で用いられている手法との比較や，より強力な極大完備化に基づく等式定理証明法は今後の課題の検討課題である．我々の手法を実装した unfMax は maedmax に比べて成功数が少なかったが，maedmax では成功しなかった例について，unfMax は成功した．

今後の課題としては，maedmax では定理証明に成功し unfMax では失敗した問題や，unfMax で定理証明に成功し，maedmax では失敗した問題についてその要因を探る必要がある．また，より効率の良いヒューリスティクスを発見することにより，証明成功率の向上を図ることが挙げられる．

参 考 文 献

[1] Bachmair, L., Dershowitz, N., and Plaisted, D. A.: Completion without failure, *Resolution of Equations in Algebraic Structures*, Vol. 2, Academic Press Inc., 1989.

[2] Dutertre, R. and de Moura, L.: The Yices SMT Solver. `http://yices.csl.sri.com/`.

[3] Klein, D. and Hirokawa, N.: Maximal completion, *Proc. of 22nd RTA*, LIPIcs, Vol. 10, 2011, pp. 71–80.

[4] Knuth, D. E. and Bendix, P.: Simple word problems in universal algebras, *Computational Problems in Abstract Algebra*, Leech, J.(ed.), Pergamon Press, 1970, pp. 263–297.

[5] Sato, H., Winkler, S., Kurihara, M., and Middeldorp, A.: Multi-completion with termination tools (system description), *Proc. of 4th IJCAR*, LNAI, Vol. 5195, 2008, pp. 306–312.

[6] Sutcliffe, G. and Suttner, C.: The TPTP Problem Library for Automated Theorem Proving. `http://www.cs.miami.edu/~tptp/`.

[7] Winkler, S.: Maximal Ordered Completion. `http://cl-informatik.uibk.ac.at/software/maedmax/`.