3J1-3

停止性証明に基づく項書換え系多重完備化手続き

Multi-Completion Procedures for Term Rewriting Systems with Termination Checking

佐藤 晴彦　　栗原 正仁
Haruhiko Sato　Masahito Kurihara

北海道大学大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University

In equational theorem proving, convergent term rewriting systems play a crucial role. In order to compute convergent term rewriting systems, the standard completion procedure (KB) was proposed by Knuth and Bendix and has been improved in a various way. The *multi-completion* system MKB developed by Kurihara and Kondo accepts as input a set of reduction orders and efficiently simulates parallel execution of KB procedure. Wehrman and Stump also developed a new variant of completion procedure, *constraint-based completion*, in which reduction orders need not be given by using automated modern termination checker. As a result, the constraint-based procedures simulate the execution of parallel KB processes in a sequential way, but naive search algorithms sometimes cause serious inefficiency. In this paper, we present a new procedure, called a *constraint-based multi-completion* procedure MKBcs, by augmenting the constraint-based completion with the framework of the multi-completion for suppressing the combinatorial explosion by sharing inferences among the processes.

1. はじめに

項書換え系 [2] は書換え型の計算モデルの一つであり，定理自動証明や代数的仕様記述など，計算機科学における様々な分野で広く用いられている．項書換え系における重要な性質に停止性と合流性がある．停止性は無限の書換え列が存在しないという性質であり，合流性はある項から複数の書換え結果が得られる場合，それらが同一の項に書換え可能であるという性質である．これらの 2 つの性質を共に満たす項書換え系を完備な項書換え系という．

多くの応用において，項書換え系は完備である事が望ましいため，書換え規則の追加や変更を行う事で，元々の系と等価かつ完備である新たな系を構成する完備化手続き (KB) が Knuth と Bendix によって提案された [4]．完備化手続きは常に成功するとは限らず，成功しない場合には手続きが終了しないことがある半アルゴリズムである．手続きが成功するかどうかは，簡約順序と呼ばれる，完備化手続きに与えるパラメータが大きく影響するが，手続きが成功するような適切な簡約順序を事前に特定することは困難である．このため，完備化手続きが成功しない場合，利用者が手続きを打ち切り，与える簡約順序を変更して再度手続きをやり直すという作業が必要になる．

この問題を解決するため，栗原らによって多重完備化手続き MKB が提案された [5]．多重完備化手続きは複数の完備化手続きを並列に実行するプロセス群の動作を単一のプロセスで効率良くシミュレートすることで，複数の簡約順序を同時に取り扱うことを可能とした手続きである．これにより，考え得る簡約順序が複数存在する場合に，それらを全て試すことが容易となった．

また，簡約順序の入力を不要とした完備化手続きの実現として Wehrman らによる停止性証明に基づく完備化手続きが提案された [8]．この手続きは生成される項書換え系の停止性を外部の停止性証明器を用いて証明する．これにより，停止性の証明により複雑な手法を用いることが可能となり，これまでの完備化手続きの多くで用いられていた簡約順序のクラスでは完備化が不可能であった系の完備化が可能となった．しかし，新しい手続きでは停止性証明方法の自由度が増したため，ある等式が両方向に向き付け可能な場合が考えられる．このとき，できるだけ完備化を成功させるためには，可能な全ての推論を試すことが望ましいため，複数の完備化手続きの推論を並列に進める必要があるが，現在の実装ではこれらの複数の手続きを独立に動作させており，複数の手続きで同じ推論が何度も行われるため無駄が多い．

本稿では，停止性証明に基づく完備化手続きの並列実行を多重完備化手続きの枠組みを用いて効率よくシミュレートする新しい手続き MKBcs を提案する．

連絡先: 北海道大学 大学院情報科学研究科,
札幌市北区北 14 条西 9 丁目，Tel: 011-706-6815,
E-mail: haru@complex.eng.hokudai.ac.jp

2. 停止性証明に基づく多重完備化手続き

本節では，従来の手続き MKB を拡張する形で，新しい手続き MKBcs を定義する．まず，プロセスの分岐を簡潔に取り扱うため，新しいプロセスの表現を導入する．次に，停止性証明に基づく手続きにおいて重要となる，制約系と呼ばれる書換え規則の集合を保持するため，ノードを拡張する．

まず，新しいプロセスの表現について述べる．MKBcs では，あるプロセス p においてある等式が両方向に向き付け可能である場合，そのプロセスは 2 つのプロセス $p0, p1$ に分岐するものとする．これにより，プロセスは 0 と 1 からなる列，すなわちビット列として表現される．プロセス $p0(p1)$ は等式を左から右 (右から左) に向き付ける．P, Q をビット列の集合とする．分岐関数 $split_P(Q)$ を次のように定義する．

$$split_P(Q) = Q \setminus P \cup \{p0, p1 \mid p \in P \cap Q\}$$

分岐関数 $split_P$ は，引数として与えられた集合に含まれるプロセスのうち，P に含まれるプロセス p のみを分岐後の 2 つのプロセス $p0, p1$ で置き換え，それ以外のプロセスはそのまま残すことで得られるプロセスの集合を表す．

次に，制約系を保持するためのノードの拡張について述べる．MKBcs におけるノードを 6 つ組 $\langle s : t, R_1, R_2, E, C_1, C_2 \rangle$ と定義する．ここで $s : t$ は項の順序対であり，R_1, R_2, E, C_1, C_2

$$
\begin{array}{ll}
\textsc{Orient}: & N \cup \{n\} \vdash split_P(N) \cup \{n'\} \\
& \text{if } n = \langle s:t, R_1, R_2, E, C_1, C_2 \rangle, \\
& n' = \langle s:t, R_1 \cup R_{lr}, R_2 \cup R_{rl}, \\
& E', C_1 \cup R_{lr}, C_2 \cup R_{rl} \rangle, \\
& E_{lr}, E_{rl} \subseteq E, E_{lr} \cup E_{rl} \neq \emptyset, \\
& P = E_{lr} \cap E_{rl}, E' = E \setminus (E_{lr} \cup E_{rl}), \\
& C[N,p] \cup \{s \to t\} \text{ terminates for all } p \in E_{lr}, \\
& C[N,p] \cup \{t \to s\} \text{ terminates for all } p \in E_{rl}, \\
& R_{lr} = (E_{lr} \setminus E_{rl}) \cup \{p0 \mid p \in P\}, \\
& \text{and } R_{rl} = (E_{rl} \setminus E_{lr}) \cup \{p1 \mid p \in P\}
\end{array}
$$

図 1: Orient rule of MKBcs

は以下の条件を満たすプロセスの集合であり，ラベルと呼ばれる．

- $(R_1 \cup C_1) \cap (R_2 \cup C_2) = \emptyset$
- $E \cap (R_1 \cup R_2 \cup C_1 \cup C_2) = \emptyset$

以降では n をノードとし，$n = \langle s:t, R_1, R_2, E, C_1, C_2 \rangle$ とする．n は $\langle t:s, R_2, R_1, E, C_2, C_1 \rangle$ と等価であり，区別しない．ラベル R_1, R_2, E は MKB におけるノードのラベル L_1, L_2, L_3 に対応する．MKB に存在しないラベル $C_1(C_2)$ は，規則 $s \to t(s \leftarrow t)$ を制約系に持つプロセスの集合を表現する．MKB と同様，MKBcs はノードの集合 N に対する推論系として定義される．推論中のある時点におけるノードの集合 N に対し，その時点においてプロセス p が保持する制約系 $C[N,p]$ を以下のように定義する．

$$
C[N,p] = \bigcup_{n \in N} C[n,p], \quad C[n,p] = \begin{cases} \{s \to t\}, & \text{if } p \in C_1, \\ \{t \to s\}, & \text{if } p \in C_2, \\ \emptyset, & \text{otherwise.} \end{cases}
$$

また，プロセスの集合に対する分岐関数をノードの各ラベルに適用した，ノードに対する分岐関数を以下のように定義する．

$$
\begin{aligned}
split_P(n) = \langle s:t, split_P(R_1), split_P(R_2), \\
split_P(E), split_P(C_1), split_P(C_2) \rangle
\end{aligned}
$$

$$
split_P(N) = \{split_P(n) \mid n \in N\}
$$

以上の定義を用いた，MKBcs の Orient 規則を図 1 に示す．他の推論規則については，MKB と同様に定義される．

3. 実験結果

本節では前節で定義した手続きを実装し，典型的な問題 [6, 8, 7] について従来の手続きと比較した結果について述べる．停止性の自動証明には，依存対法 [1] に基づき適切な多項式順序を SAT ソルバを用いて求める手法 [3] を用いた．実験結果を表 1 に示す．表の「naive」の列が停止性証明に基づく手続きを単純に並列動作させた場合の結果に,「MKBcs」の列が前節で提案した手続きを用いた場合の結果に対応する．それぞれの列において「all」が完備化全体に要した秒数を,「re/de」が書換え及び危険対の生成に要した秒数を表す．結果から全ての問題において MKBcs はより高速であり，特に長い時間を要する難しい問題に対して効果的であることが分かる．

表 1: 単純な並列動作と多重完備化手続きとの比較

Problem	naive		MKBcs	
	all	re/de	all	re/de
SK90_3.04	190.3	150.1	55.9	30.2
SK90_3.07	4.1	2.3	2.3	0.6
SK90_3.09	146.6	115.1	29.4	7.2
SK90_3.27	21.1	3.2	19.1	1.7
SK90_3.28	410.5	133.4	207.9	2.1
WSW06_CGE$_2$	435.9	272.4	126.4	10.7
WS06_PR	28074.7	14690.5	10752.1	25.7

4. おわりに

本稿では，停止性証明に基づく多重完備化手続き MKBcs を提案した．実験により，MKBcs が従来の手続きより高速であることを示した．今後の課題として，向き付け不能な等式を扱うための拡張完備化手続きや，基礎項上の完備な系を得るための無向完備化手続きといった，より難しい問題を取り扱うための拡張を導入することが挙げられる．

参考文献

[1] T. Arts and J. Giesl, "Termination of term rewriting using dependency pairs", Theoretical Computer Science, vol. 236, pp. 133-178, 2000.

[2] F. Baader and T. Nipkow, Term Rewriting and All That, Cambridge University Press, 1998.

[3] C. Fuhs, J. Giesl, A. Middeldorp, P. Schneider-Kamp, R. Thiemann, and H. Zankl, "SAT solving for termination analysis with polynomial interpretations", Proc. 10th International Conference on Theory and Applications of Satisfiability Testing, vol. 4501 of Lecture Notes in Computer Science, pp. 340-354, 2007.

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

[5] M. Kurihara and H. Kondo, "Completion for multiple reduction orderings", Journal of Automated Reasoning, vol.23, no.1, pp. 25-42, 1999.

[6] J. Steinbach and U. Kühler, "Check your ordering - termination proofs and problems", Technical Report SR-90-25, Universität Kaiserslautern, 1990.

[7] I. Wehrman and A. Stump, "Mining propositional simplification proofs for small validating clauses", Electronic Notes in Theoretical Computer Science, vol. 144, no. 2, pp. 79-91, 2006.

[8] I. Wehrman, A. Stump, and E. Westbrook, "Slothrop: Knuth-Bendix completion with a modern termination checker", Proc. 17th International Conference on Rewriting Techniques and Applications, vol. 4098 of Lecture Notes in Computer Science, pp. 287-296, 2006.