項書き換えシステム自動変換のための 2階一般化アルゴリズムについて

On a Second-Order Generalization Algorithm for Automatic Transformation of Term Rewriting Systems

千葉 勇輝　青戸等人　外山芳人

Yuki Chiba　Takahito Aoto　Yoshihito Toyama

東北大学 電気通信研究所

RIEC, Tohoku University

{chiba,aoto,toyama}@nue.riec.tohoku.ac.jp

パターンによるプログラム自動変換 (Chiba et al. 2006) では，2 階変換パターンをもちいて，項書き換えシステムの自動変換を実現している．具体的な変換例から 2 階変換パターンを自動生成する方法として，Chiba et al. (2007) は 2 階一般化アルゴリズムを提案し，その健全性を示すとともに，変換パターンの自動生成実験を行った．本発表では，2 階一般化アルゴリズムの改良を試みるとともに，その完全性について考察する．

1 はじめに

パターンに基づくプログラム変換は Huet と Lang により提案されたプログラム効率化の手法である [4]．著者らは文献 [2, 3] において，項書き換えを計算モデルに用いたパターンに基づくプログラム変換の枠組みを提案した．この枠組みでは，変換パターン $\mathcal{P} \Rightarrow \mathcal{P}'$ と項書き換えシステム $\mathcal{R}$ が与えられたとき，項書き換えパターン $\mathcal{P}$ と項書き換えシステム $\mathcal{R}$ のパターン照合を行い，得られた項準同型写像 φ を用いて，$\varphi(\mathcal{P}')$ を出力する．

パターンに基づくプログラム変換を行うためには，適切な変換パターンを前もって用意する必要がある．このため，パターンに基づくプログラム変換の適用範囲を広げるためには，さまざまな種類の変換パターンを準備することが重要である．しかしながら，我々が調査した限りでは，変換パターンの構築に関する研究はあまり知られていない．

そこで，著者らは，文献 [1] において，変換パターンを構築する手法の 1 つとして，具体的なプログラム変換例から変換パターンを構築する手法を提案した．この手法では，類似のプログラム変換を一般化することにより，変換パターンを生成する．個別のプログラム変換の相違点を一般化することにより，他のプログラムに対しても有効な変換パターンが得られることが期待できる．

実際，文献 [1] では，ヒューリスティックスクを用いた実装によって，いくつかの具体例に対し，プログラム変換例から変換パターンを構築できることを示すとともに，2 つの具体的なプログラム変換 $\mathcal{R}_1 \Rightarrow \mathcal{R}'_1$ と $\mathcal{R}_2 \Rightarrow \mathcal{R}'_2$ から生成した変換パターン $\mathcal{P} \Rightarrow \mathcal{P}'$ を用いて，別のプログラム変換 $\mathcal{R}_3 \Rightarrow \mathcal{R}'_3$ に成功する場合があることを示した．

プログラム変換の一般化手続きは，項の 2 階一般化アルゴリズムを原理とする．著者らは，Plotkin による 1 階一般化アルゴリズム [5] を拡張することにより，2 階一般化アルゴリズムを得た．項の 1 階一般化とは，2 つの項 s, t の異なる部分を変数とすることにより，$u\theta_1 = s$，$u\theta_2 = t$ となるような代入 θ_1, θ_2 の存在するような項 u を構成することである．例えば，項 $+(\mathsf{s}(x_1), y_1)$ と項 $+(x_2, \mathsf{s}(y_2))$ の一般化は $+(x_3, y_3)$ となる．これに対して，2 階一般化では，関数部分もパターン変数により一般化されることが特徴である．例えば，項 $+(\mathsf{s}(x_1), y_1)$ と項 $\times(\mathsf{s}(x_2), y_2)$ の一般化はパターン $\mathsf{p}(\mathsf{s}(x_3), y_3)$ となる．ここで，p は ($+$ や $\times$ により具体化される) パターン変数である．

Plotkin による 1 階一般化アルゴリズムは，健全かつ完全であることが示されている [5]．一方，著者らが提案した 2 階一般化アルゴリズムについて健全性は成立する [1] が，完全性についてはまだ明らかになっていない．本論文では，2 階一般化アルゴリズムの完全性について検討する．

Gen$_1$

$$\frac{\langle C[s \wedge t], \phi\rangle}{\langle C[x], \phi \cup \{s \wedge t \mapsto x\}\rangle}\ \mathrm{root}(s) \neq \mathrm{root}(t),\ s \wedge t \notin \mathrm{dom}(\phi).\ \text{ただし，}x\text{ は新しい変数}$$

Gen$_2$

$$\frac{\langle C[s \wedge t], \phi\rangle}{\langle C[x], \phi\rangle}\ \mathrm{root}(s) \neq \mathrm{root}(t),\ \phi(s \wedge t) = x$$

Divide

$$\frac{\langle C[f(s_1,\ldots,s_n) \wedge f(t_1,\ldots,t_n)], \phi\rangle}{\langle C[f(s_1 \wedge t_1,\ldots,s_n \wedge t_n)], \phi\rangle}$$

Ident

$$\frac{\langle C[x \wedge x], \phi\rangle}{\langle C[x], \phi\rangle}$$

図 1: 1 階一般化アルゴリズム Gen の推論規則

2 1 階一般化アルゴリズムとその健全性・完全性

我々は，Plotkin による 1 階一般化アルゴリズムを我々の体系に合うように形式化し，健全性，完全性を示した．以下では，簡単に，1 階一般化アルゴリズムとその健全性・完全性について説明する．

まず，組項の概念を導入する [1]．組項とは，特別な 2 引数関数 $\wedge$ を用いた項で，$\wedge$ のネストしない項をいう．正確には，組項の集合 $\mathrm{T}_\wedge(\mathcal{F}, V)$ は以下のように定義される．(1) $\mathrm{T}(\mathcal{F}, V) \subseteq \mathrm{T}_\wedge(\mathcal{F}, V)$; (2) $s, t \in \mathrm{T}(\mathcal{F}, V)$ ならば $s \wedge t \in \mathrm{T}_\wedge(\mathcal{F}, V)$; (3) $s_1, \ldots, s_n \in \mathrm{T}_\wedge(\mathcal{F}, V)$ かつ $f \in \mathcal{F}^n$ ならば，$f(s_1, \ldots, s_n) \in \mathrm{T}_\wedge(\mathcal{F}, V)$．$\wedge$ の出現を持たない組項 t を $\wedge$-自由，ルート記号が $\wedge$ の組項 t を $\wedge$-トップとよぶ．

図 1 に 1 階一般化アルゴリズム Gen の推論規則を示す．推論規則は，組項と，$\wedge$-トップ組項から変数への写像の対を推論の対象とする．$\langle s, \phi\rangle$ が Gen の適用により $\langle s', \phi'\rangle$ となるとき，$\langle s, \phi\rangle \leadsto \langle s', \phi'\rangle$ と記す．

次に，組項に対する射影を定義する．組項 t の射影 $\pi_i(t)$ $(i = 1, 2)$ は次のように定義される：(1) $\pi_i(t) = t$ $(t \in \mathrm{T}(\mathcal{F}, V)$ のとき)，(2) $\pi_i(f(s_1, \ldots, s_n)) = f(\pi_i(s_1), \ldots, \pi_i(s_n))$ $(f \in \mathcal{F}$ のとき)，(3) $\pi_i(s_1 \wedge s_2) = s_i$．

以下のように，1 階一般化アルゴリズムの停止性と健全性が成立する．

定理 1 (1 階一般化アルゴリズムの停止性) $\leadsto$ は整礎.

定理 2 (1 階一般化アルゴリズムの健全性) $\langle s \wedge t, \emptyset\rangle \overset{*}{\leadsto} \langle u, \phi\rangle$ かつ u は $\wedge$-自由とする．このとき，$\phi_1^{-1}(u) = s$ かつ $\phi_2^{-1}(u) = t$ が成立する．ただし，ここで，$\phi_i^{-1} = \{x \mapsto v_i \mid v_1 \wedge v_2 \mapsto x \in \phi\}$.

完全性の証明に次の補題を利用する．

補題 3 $i = 1, 2$ について $\pi_i(e) = \pi_i(e')$ が成立するとする．このとき，$\langle e, \phi\rangle \overset{*}{\leadsto} \langle \tilde{e}, \phi\rangle$ かつ $\langle e', \phi\rangle \overset{*}{\leadsto} \langle \tilde{e}, \phi\rangle$ となる $\tilde{e}$ が存在する．

項 u,s,t に対して，$u\theta_1 = s$ かつ $u\theta_2 = t$ となる代入 θ_1 と θ_2 が存在するとき，u を s と t の 1 階一般化とよぶ．

補題 4 $\langle e, \phi\rangle$ に対して，u は $\pi_1(e)$ と $\pi_2(e)$ の 1 階一般化かつ e は $\wedge$-自由でないとする．このとき，u は $\pi_1(e')$ と $\pi_2(e')$ の 1 階一般化かつ $\langle e, \phi\rangle \overset{*}{\leadsto} \langle e', \phi'\rangle$ となる $\langle e', \phi'\rangle$ が存在する．

証明: $e = C[s \wedge t]_p$ とする．$\mathrm{root}(s) = \mathrm{root}(t)$ のときは，Ident か Divide が適用できる．適用後の結果を $\langle e', \phi\rangle$ とすると，$\pi_i(e) = \pi_i(e')$ $(i = 1, 2)$ であるので，u は $\pi_1(e')$ と $\pi_2(e')$ の 1 階一般化となる．

$\mathrm{root}(s) \neq \mathrm{root}(t)$ のとき，$p' \leq p$ かつ $u|_{p'} = y \in \mathscr{V}$ となるポジション $p' \in \mathrm{Pos}(u)$ が存在する．ここで，$u\theta_1 = \pi_1(e)$，$u\theta_2 = \pi_2(e)$，$\{p_1, \ldots, p_n\} = \{q \in \mathrm{Pos}(u) \mid u|_q = y\} \setminus \{p'\}$ とする．

$u\theta_i = \pi_i(e)$，$u|_{p'} = y$ より，$\theta_i(y) = \pi_i(e|_{p'}) = \pi_i(C[s \wedge t]|_{p'})$．ここで $C[s \wedge t]|_{p'} = C'[s \wedge t]$ とおく．このとき

$$
\begin{aligned}
\langle e, \phi \rangle &= \langle C[C'[s \wedge t], e|_{p_1}, \ldots, e|_{p_n}]_{p', p_1, \ldots, p_n}, \phi \rangle \\
&\overset{*}{\rightsquigarrow} \langle C[\hat{C}[s \wedge t], \hat{C}[s \wedge t], \ldots, \hat{C}[s \wedge t]]_{p', p_1, \ldots, p_n}, \phi \rangle \\
&\qquad\qquad \text{補題 3 より} \\
&\overset{+}{\rightsquigarrow} \\
&\langle C[\hat{C}[x], \hat{C}[x], \ldots, \hat{C}[x]]_{p', p_1, \ldots, p_n}, \phi \cup \{s \wedge t \mapsto x\} \rangle \\
&\qquad\qquad \text{Gen}_1 \text{ もしくは } \text{Gen}_2 \text{ を適用}
\end{aligned}
$$

ここで $\langle C[\hat{C}[x], \hat{C}[x], \ldots, \hat{C}[x]]_{p', p_1, \ldots, p_n}, \phi \cup \{s \wedge t \mapsto x\} \rangle = \langle e', \phi' \rangle$ とし，θ'_1 と θ'_2 を以下で定義する．

$$
\begin{aligned}
\theta'_1 &= \{\theta_1 \setminus \{y \mapsto \pi_1(C[s \wedge t]|_{p'})\}\} \cup \{y \mapsto \pi_1(\hat{C}[x])\} \\
\theta'_2 &= \{\theta_1 \setminus \{y \mapsto \pi_2(C[s \wedge t]|_{p'})\}\} \cup \{y \mapsto \pi_2(\hat{C}[x])\}
\end{aligned}
$$

このとき $u\theta'_1 = \pi_1(e')$，$u\theta'_2 = \pi_2(e')$ であるので，u は $\pi_1(e')$ と $\pi_2(e')$ の 1 階一般化となる． □

補題 4 より以下の定理が導かれる．ここで，項 s と t に対して，$\theta(s) = t$ となる代入 θ が存在するとき，$s \precsim t$ と書く．

定理 5 (1 階一般化アルゴリズム *Gen* の完全性) u を s と t の 1 階一般化とする．このとき，$\langle s \wedge t, \emptyset \rangle \overset{*}{\rightsquigarrow} \langle v, \phi \rangle$ かつ $u \precsim v$ となる $\wedge$-自由な項 v が存在する．

3 2 階一般化アルゴリズムとその健全性・完全性

我々は図 1 で示した 1 階一般化アルゴリズム *Gen* を拡張し，2 階一般化アルゴリズム *2nd-Gen* を提案した．また，その停止性，および，健全性を示した [1]．以下では，文献 [1] で提案されたアルゴリズム *2nd-Gen* を修正し，アルゴリズムの完全性について議論する．

図 2 は，図 1 で示した 1 階一般化アルゴリズムを拡張した 2 階一般化アルゴリズムの推論規則である．我々の枠組みでは 2 階一般化された記号を表現するためにパターン変数の概念を導入した [2, 3]．パターン変数を含む項は項パターンとよぶ．推論規則は組項パターンと $\wedge$-トップの組インデックス付き文脈からパターン変数への写像の対を推論の対象とする．図 2 のアルゴリズムは，1 階一般化アルゴリムの拡張であることを明確にし，完全性に関する議論の見通しをよくするため，[1] で提案されたアルゴリズムにおいて局所変数の概念を無効にしている．

$\langle s, \Phi \rangle$ が *2nd-Gen* の適用により $\langle s', \Phi' \rangle$ となるとき，$\langle s, \Phi \rangle \rightsquigarrow \langle s', \Phi' \rangle$ と記す．また，組項パターンに対する射影は 1 階一般化と同様に定義する．項パターン u, s, t に対して，$\varphi_1(u) = s$ かつ $\varphi_2(u) = t$ となる項準同型写像 φ_1, φ_2 が存在するとき，u を s と t の 2 階一般化とよぶ．項パターン s, t に対して $\varphi(s) = t$ となる項準同型写像 φ が存在するとき，$s \precsim t$ と書く．

図 2 のアルゴリズムは [1] で提案されたアルゴリズムを制限したものなので，以下の通り，停止性と健全性が成立する．

定理 6 (2 階一般化アルゴリズムの停止性) $\rightsquigarrow$ は整礎.

定理 7 (2 階一般化アルゴリズムの健全性) $\langle s \wedge t, \emptyset \rangle \overset{*}{\rightsquigarrow} \langle u, \Phi \rangle$ かつ u は $\wedge$-自由とする．このとき，$\Phi_1^{-1}(u) = s$ かつ $\Phi_2^{-1}(u) = t$ が成立する．ここで，$\Phi_i^{-1} = \{p \mapsto C_i \mid C_1 \wedge C_2 \mapsto p \in \Phi\}$.

1 階一般化と同様，以下の予想を示すことにより 2 階一般化アルゴリズムの完全性を示すことが可能となる．

予想 8 $\langle e, \Phi \rangle$ に対して，u は $\pi_1(e)$ と $\pi_2(e)$ の 2 階一般化かつ e は $\wedge$-自由でないとする．このとき，u は $\pi_1(e')$ と $\pi_2(e')$ の 2 階一般化かつ $\langle e, \Phi \rangle \overset{+}{\rightsquigarrow} \langle e', \Phi' \rangle$ となる．$\langle e', \Phi' \rangle$ が存在する．

予想 8 から以下の定理が導出される．

定理 9 (2 階一般化アルゴリズム *2nd-Gen* の完全性) u を s と t の 2 階一般化とする．このとき，$\langle s \wedge t, \emptyset \rangle \overset{*}{\rightsquigarrow} \langle v, \Phi \rangle$ かつ $u \precsim v$ となる $\wedge$-自由な項パターン v が存在する．

一般化アルゴリズムの完全性を証明する際に重要なことは，同じ変数に代入される部分の一般化を同期させることである．例えば，$s = \mathsf{f}(\mathsf{g}(x), \mathsf{g}(x))$ と $t = \mathsf{f}(\mathsf{h}(y, z), \mathsf{h}(y, z))$ の 1 階一般化の 1 つとしては $u = \mathsf{f}(v, v)$ があげられる．本論文で示した 1 階一般化アルゴリズムを用いて s と t の一般化を求める際，以下の推論が実行される．

$$
\begin{aligned}
&\langle s \wedge t, \emptyset \rangle \\
&\rightsquigarrow \langle \mathsf{f}(\mathsf{g}(x) \wedge \mathsf{h}(y, z), \mathsf{g}(x) \wedge \mathsf{h}(y, z)), \emptyset \rangle \\
&\rightsquigarrow \langle \mathsf{f}(v, \mathsf{g}(x) \wedge \mathsf{h}(y, z)), \{\mathsf{g}(x) \wedge \mathsf{h}(y, z) \mapsto v\} \rangle
\end{aligned}
$$

$u = \mathsf{f}(v, v)$ が s と t の一般化であるので，$\mathsf{f}(\mathsf{g}(x) \wedge \mathsf{h}(y, z), \mathsf{g}(x) \wedge \mathsf{h}(y, z))$ における f の 2 つの引数を同

Gen$_1$

$$\frac{\langle C[C_1\langle s_1,\ldots,s_n\rangle \wedge C_2\langle t_1,\ldots,t_n\rangle],\Phi\rangle}{\langle C[p(\alpha_1,\ldots,\alpha_n)],\Phi\cup\{C_1\wedge C_2\mapsto p\}\rangle}\quad \begin{array}{l} C_1\neq C_2,\\ C_1\wedge C_2\notin \mathrm{dom}(\Phi),\\ \mathscr{H}(C_1)\cup\mathscr{H}(C_2)=\{\Box_1,\ldots,\Box_n\}.\\ \text{ただし，}p\text{ は新しいパターン変数.}\\ \alpha_i=\begin{cases} s_i\wedge t_i & \text{if } \Box_i\in\mathscr{H}(C_1)\cap\mathscr{H}(C_2)\\ s_i & \text{if } \Box_i\in\mathscr{H}(C_1)\setminus\mathscr{H}(C_2)\\ t_i & \text{if } \Box_i\in\mathscr{H}(C_2)\setminus\mathscr{H}(C_1)\end{cases}\end{array}$$

Gen$_2$

$$\frac{\langle C[C_1\langle s_1,\ldots,s_n\rangle \wedge C_2\langle t_1,\ldots,t_n\rangle],\Phi\rangle}{\langle C[p(\alpha_1,\ldots,\alpha_n)],\Phi\rangle}\quad \begin{array}{l} C_1\neq C_2,\\ \Phi(C_1\wedge C_2)=p\\ \alpha_i=\begin{cases} s_i\wedge t_i & \text{if } \Box_i\in\mathscr{H}(C_1)\cap\mathscr{H}(C_2)\\ s_i & \text{if } \Box_i\in\mathscr{H}(C_1)\setminus\mathscr{H}(C_2)\\ t_i & \text{if } \Box_i\in\mathscr{H}(C_2)\setminus\mathscr{H}(C_1)\end{cases}\end{array}$$

Divide

$$\frac{\langle C[q(s_1,\ldots,s_n)\wedge q(t_1,\ldots,t_n)],\Phi\rangle}{\langle C[q(s_1\wedge t_1,\ldots,s_n\wedge t_n)],\Phi\rangle}\quad q\in\mathscr{F}\cup\mathscr{X}$$

図 2: 2 階一般化アルゴリズム *2nd-Gen* の推論規則

期させる必要がある．すなわち，f の第 2 引数 $\mathsf{g}(x)\wedge\mathsf{h}(y,z)$ が v となる必要がある．1 階一般化では補題 3 が成立しているので，この同期を実現することができる．

一方，2 階一般化では，代入される変数が項パターンの葉の位置以外にも出現するため，1 階一般化と同様，同じパターン変数へ代入される部分の同期には，項の同期ではなく文脈の同期が必要となる．また，2 階一般化では Gen$_1$ と Gen$_2$ によってもあたらしい $\wedge$-トップの項パターンが出現するため，1 階一般化における補題 3 と同様の補題を，2 階一般化で示すことは 1 階のときほど簡単ではない．

また，“u は s と t の 2 階一般化” と言ったときに，$\varphi_1(u)=s$，$\varphi_2(u)=t$ となる項準同型写像 φ_1，φ_2 は一般に複数存在する．我々の実験から，複数存在する項準同型写像のなかから，適切なものを選択しないと完全性の証明に失敗することが分かっている．2 階一般化アルゴリズムの完全性を示すには，上記の複数ある項準同型写像から適切なものを選択する基準をあきらかにする必要がある．

4　おわりに

本論文では一般化アルゴリズムの完全性に関する議論を行った．はじめに，Plotkin による 1 階一般化アルゴリズムを我々の体系のもとで形式化し，その完全性の証明を与えた．また，1 階一般化アルゴリズムを拡張し，2 階一般化を実現するアルゴリズムを構築した．完全性の証明と，1 階一般化アルゴリズムからの拡張であることをあきらかにするために，[3] で提案されたアルゴリズムから局所変数の概念を無効にした，より簡明なアルゴリズムを提案した．さらに，アルゴリズムの停止性，健全性を与え，完全性証明を実現するための問題をあきらかにした．

今後の課題は，前節で示した完全性を示すための問題を解決し，2 階一般化アルゴリズムの完全性証明を与えることである．また，本論文では議論の見通しを明快にするために局所変数を無効にした枠組みでアルゴリズムを提案し，その完全性の議論を行った．しかしながら，局所変数はプログラム変換への応用を考慮した際に重要な概念となる．したがって，局所変数を考慮した枠組みにおける，簡明な 2 階一般化アルゴリズムの提案と，その完全性を示すことも今後の課題である．

謝辞

本研究の一部は，科学研究費 17700002, 19500003 の補助を受けて行われた．

参考文献

[1] Y. Chiba, T. Aoto, and Y. Toyama. Automatic construction of program transformation template. Revised version has been submitted to *IPSJ Transactions on Programming*.

[2] Y. Chiba, T. Aoto, and Y. Toyama. Program transformation by templates based on term rewriting. In *Proceedings of the 7th ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP 2005)*, pp. 59–69. ACM Press, 2005.

[3] Y. Chiba, T. Aoto, and Y. Toyama. Program transformation by templates: A rewriting framework. *IPSJ Transactions on Programming*, 47(SIG 16 (PRO 31)):52–65, 2006.

[4] G. Huet and B. Lang. Proving and applying program transformations expressed with second order patterns. *Acta Informatica*, 11:31–55, 1978.

[5] G. D. Plotkin. A note on inductive generalization. In *Machine Intelligence*, Vol. 5, chapter 8, pp. 153–163. Edinbrgh University Press, 1969.