制約付き書換え帰納法におけるラグランジュ補間を用いた補題生成

比嘉 慎哉　西田 直樹　酒井正彦

書換え帰納法などの定理証明では，証明を成功させるために，発散する等式を一般化した補題等式を生成することが有効であることが多い．命令型プログラムの等価性を制約付き書換え帰納法に基づいた証明法で検証する研究においては，局所変数の宣言時の初期値に注目した補題生成法が提案されている．しかし，この手法で必ず適切な補題等式を生成できるとは限らず，実際に適切な補題等式の生成に失敗する例が存在する．本稿では，ラグランジュ補間を用いた補題生成法を提案し，制約付き書換え帰納法に基づく定理証明系に導入する．提案手法ではまず，一般化の対象とした等式からの発散に出現するどの等式よりも一般的な等式をラグランジュ補間を用いて補題等式の候補として推定する．残念ながら，推定された等式が実際には補題等式ではない場合がある．補題等式であるかどうかは書換え帰納法により証明が試みられることで確認され，反証された時点で補題等式ではないことが確定する．提案手法では反証された際にその過程を利用して推定した等式を適切な補題等式に修正することを試みる．既存手法では適切な補題等式を生成できない二重ループを含むプログラムの等価性証明の例に対して，提案手法が適切な補題等式の生成に成功することを報告する．

1　はじめに

近年，プログラムの等価性を論理制約付き項書換え系（LCTRS）[3] における制約付き等式の恒真性に帰着させて検証する手法が研究されている [2]．その手法ではプログラムを LCTRS に変換し，プログラムの等価性を表す制約付き等式がその LCTRS のもとで恒真であること，つまりその LCTRS の帰納的定理であることを書換え帰納法 [4] で証明する．

書換え帰納法に基づく定理証明では，書換え系 $\mathcal{R}$ と等式の集合 $\mathcal{E}$ が与えられ，$\mathcal{E}$ のすべての等式が $\mathcal{R}$ の帰納的定理であることを検証する．等式の集合と帰納法の仮定の集合の対を証明状態と呼ぶ．推論規則は証明状態から新たな証明状態への推論を定義する．初期証明状態 $(\mathcal{E}, \emptyset)$ に推論規則を適用していき，証明状態 $(\emptyset, \mathcal{H})$ に到達できれば証明は成功である．$\mathcal{R}$ の合流性などの特定の条件のもとでは，$\mathcal{E}$ の等式の 1 つが帰納的定理でないことを証明するため，すなわち反証するための推論規則も存在する．

書換え帰納法による定理証明では推論規則の適用が無限に続き証明成功しないことがある．証明状態から推論が無限に続くことは（証明の）発散と呼ばれる．発散の主な原因は等式に帰納法の仮定に相当する書換え規則を適用して証明を収束させられず，新たな帰納法の仮定を生成し続けることである．発散を回避するために，証明の途中で発散を引き起こす可能性がある等式を一般化することで補題等式を生成する手法が研究されている．項書換え系（TRS）上の定理証明における一般化の手法として発散鑑定法 [7] や健全一般化法 [5] などが知られている．命令型プログラムを変換して得られる LCTRS を対象とした一般化の手法として，等式に含まれる元のプログラムの局所変数の初期値を変数に置き換える手法が提案されている [2]．

既存手法 [2] では適切な補題等式を生成できない例が存在する．図 1 における関数 `sum`，`sum3` を定義する C プログラムは同じく図 1 の LCTRS $\mathcal{R}_1$ に変換される [2]．関数 `sum` と関数 `sum3` の等価性を表す等式 $\mathsf{sum}(n) \approx \mathsf{sum}_3(n)$ が $\mathcal{R}_1$ の帰納的定理であるこ

Lemma Generation by Lagrange Interpolation in Constrained Rewriting Induction.
Shinya Higa, Naoki Nishida, Masahiko Sakai, 名古屋大学大学院情報学研究科, Graduate School of Informatics, Nagoya University.

```
int sum(int n) {
  if (n < 0) return 0;
  return n + sum(n - 1);
}

int sum3(int n) {
  int ret = 0;
  for (int i = 0; i <= n; i++)
    for (int j = 0; j < i; j++)
      ret++;
  return ret;
}
```

$$\mathcal{R}_1 = \left\{ \begin{array}{rll} \mathsf{sum}(n) & \to \mathsf{return}(v_3) & [n{<}0 \wedge v_3{=}0], \\ \mathsf{sum}(n) & \to \mathsf{u}_2(n, \mathsf{sum}(n-1)) & [n \geq 0], \\ \mathsf{u}_2(n, \mathsf{return}(w)) & \to \mathsf{return}(n+w), & \\ \mathsf{u}_2(n, \mathsf{error}) & \to \mathsf{error}, & \\ \mathsf{sum}_3(n) & \to \mathsf{u}_0(n, v_0, v_1) & [v_0{=}0 \wedge v_1{=}0], \\ \mathsf{u}_0(n, ret, i) & \to \mathsf{u}_1(n, ret, i, v_2) & [i{\leq}n \wedge v_2{=}0], \\ \mathsf{u}_1(n, ret, i, j) & \to \mathsf{u}_1(n, ret+1, i, j+1) & [j < i], \\ \mathsf{u}_1(n, ret, i, j) & \to \mathsf{u}_0(n, ret, i+1) & [j \geq i], \\ \mathsf{u}_0(n, ret, i) & \to \mathsf{return}(ret) & [i > n] \end{array} \right\}$$

図 1　0 から n までの総和を計算する関数 sum3，sum を定義する C プログラムとそれを変換して得られる LCTRS $\mathcal{R}_1$

$$\begin{array}{l} (\{\mathsf{sum}(n) \approx \mathsf{sum}_3(n)\}, \emptyset) \\ \vdash^+_{ri} \left(\begin{array}{l} \{(\mathrm{A}) : \mathsf{u}_0(n, v_0+1, v_1+2) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, v_0, v_1+1))\ [v_1 = 0 \wedge v_1 = 0 \wedge n \geq 0 \wedge v_1 + 1 \leq n]\}, \\ \{\ldots\} \end{array} \right) \\ \vdash^+_{ri} \left(\begin{array}{l} \{(\mathrm{B}) : \mathsf{u}_0(n, v_0+3, v_1+3) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, v_0+1, v_1+2))\ [v_0 = 0 \wedge v_1 = 0 \wedge n \geq 0 \wedge v_1 + 2 \leq n]\}, \\ \{\ldots\} \end{array} \right) \\ \vdash^+_{ri} \left(\begin{array}{l} \{(\mathrm{C}) : \mathsf{u}_0(n, v_0+6, v_1+4) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, v_0+3, v_1+3))\ [v_0 = 0 \wedge v_1 = 0 \wedge n \geq 0 \wedge v_1 + 3 \leq n]\}, \\ \{\ldots\} \end{array} \right) \\ \vdash^+_{ri} \left(\begin{array}{l} \{(\mathrm{D}) : \mathsf{u}_0(n, v_0{+}10, v_1+5) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, v_0+6, v_1+4))\ [v_0 = 0 \wedge v_1 = 0 \wedge n \geq 0 \wedge v_1 + 4 \leq n]\}, \\ \{\ldots\} \end{array} \right) \\ \vdash^+_{ri} \left(\begin{array}{l} \{(\mathrm{E}) : \mathsf{u}_0(n, v_0{+}15, v_1+6) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, v_0{+}10, v_1+5))\ [v_0 = 0 \wedge v_1 = 0 \wedge n \geq 0 \wedge v_1 + 5 \leq n]\}, \\ \{\ldots\} \end{array} \right) \\ \vdash^+_{ri} \cdots \end{array}$$

図 2　$\mathcal{R}_1$ における帰納的定理の証明の発散

とを既存手法 [2] で証明する際には，図 2 に記した発散が起こり，証明に成功しない †1．ここで，$\vdash_{ri}$ は証明状態への書換え帰納法の推論規則の適用を表す．図 2 の発散において等式 (D) を含む証明状態に至った時点で，(D) を既存手法 [2] で一般化する場合を考える．このとき，以下が補題等式の候補として生成される．

$$(\mathrm{D}_1) : \mathsf{u}_0(n, v_0+10, v_1+5) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, v_0+6, v_1+4)) \quad [n \geq 0 \wedge v_1 + 4 \leq n]$$

†1　証明状態に出現する等式や書換え規則は見やすさのために簡単化されている．例えば，図 2 の (A)–(D) は実際には図 5 の (A′)–(D′) である．

上述の等式は $\mathcal{R}_1$ の帰納的定理ではない．すなわち，既存手法 [2] が生成する補題候補は実際には適切な補題等式ではなく，証明に成功しない．

本稿では，ラグランジュ補間を用いた補題生成法を提案し，制約付き書換え帰納法に基づく定理証明系に導入することで，制約付き書換え帰納法の証明能力の強化を目指す．提案手法ではまず，一般化の対象とした等式からの発散に出現するどの等式よりも一般的な等式をラグランジュ補間を用いて補題等式の候補として推定する．残念ながら，推定された等式が実際には補題等式ではない場合がある．補題等式であるかど

うかは書換え帰納法により証明が試みられることで確認され，反証された時点で補題等式ではないことが確定する．提案手法では反証された際にその過程を利用して推定した等式を適切な補題等式に修正することを試みる．

提案手法は等式の列に対して以下のように補題等式の候補を生成する．

1. ラグランジュ補間を用いて補題等式の候補を生成する．
2. 生成した補題等式の候補が帰納的定理であることの証明を書換え帰納法を用いて試みる．
3. 生成した補題等式の候補が反証された場合は，反証までの過程を参照して生成した補題等式の候補の制約部分を修正し，改めて帰納的定理であることの証明を試みる．

上記の手法については，生成した補題等式の候補を書換え帰納法により検証する際にも発散が発生する可能性があり，その場合には上記の手法を用いて補題等式の候補の生成が試みられる．

既存手法が生成した補題等式の候補 (D_1) は等式 (E) よりも一般的ではない．なぜなら，(D_1) において成立する以下の関係は (E) においては成立しないからである．

$$(\text{左辺の } \mathsf{u}_0 \text{ の第 2 引数}) = (\text{右辺の } \mathsf{u}_0 \text{ の第 2 引数}) + 4$$

提案手法では，一般化の対象とする等式を導出するまでに現れた等式 $(\mathrm{A}), \ldots, (\mathrm{D})$ を参照し，$(\mathrm{D}), (\mathrm{E}), \ldots$ すべてにおいて成り立つ引数間の関係式をラグランジュ補間により推定する．提案手法で等式 (D) を一般化すると以下の等式を補題候補として得る．

$$(\mathrm{D}_2):\ \mathsf{u}_0(n, r_1, i_1) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, r_0, i_0)) \quad \begin{bmatrix} i_0 \leq n \wedge 1 \leq n \wedge i_1 = i_0 + 1 \\ \wedge\, r_1 = i_1 * (i_1 - 1)/2 \\ \wedge\, r_0 = i_1 * (i_1 - 1)/2 \end{bmatrix}$$

提案手法は補題等式の候補を生成した後に，その等式が帰納的定理であることの証明を試みる．残念ながら，上述のように生成された (D_2) は帰納的定理ではなく，反証される．提案手法では反証に至るまでの場合分けを排除する論理式を，生成された補題等式の候補の制約に連接することにより修正する．等式 (D_2) については反証された場合分けである $0 \geq i_0$ の否定に相当する論理式 $0 < i_0$ を (D_2) の制約部に連接することにより，以下の等式を新たな補題等式の候補として得る．

$$(\mathrm{D}_3):\ \mathsf{u}_0(n, r_1, i_1) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, r_0, i_0)) \quad \begin{bmatrix} i_0 \leq n \wedge 1 \leq n \wedge i_1 = i_0 + 1 \\ \wedge\, r_1 = i_1 * (i_1 - 1)/2 \\ \wedge\, r_0 = i_1 * (i_1 - 1)/2 \\ \wedge\, 0 < i_0 \end{bmatrix}$$

上述の等式が帰納的定理であることを証明する際には発散が生じる．そこで，途中で提案手法を新たに適用して，(D_3) が帰納的定理であることを証明するために補題生成を行う．その際に生成される補題等式の候補について書換え帰納法でさらに証明を試みた際には発散は生じずに，証明に成功する．よって，(D_3) が帰納的定理であることが証明される．

本稿の構成は次の通りである．2 章では，準備として本稿で用いる概念や記法の定義を記す．3 章では，等式が別の等式よりも一般的であるという概念を定義する．4 章では，提案手法の概要について述べる．5 章では，ラグランジュ補間を用いた補題生成法について述べる．6 章では，反証過程を用いて補題等式の候補を修正する手法について述べる．7 章では，まとめと今後の課題について述べる．

2 準備

本節では，本稿で用いる概念や記法を記す [1,2].

2.1 論理制約付き項書換え系

本稿を通じて，$\mathcal{S}$ を**基本型**の集合，$\mathcal{V}$ を型付き**変数**の無限集合とする．そして，**シグネチャ**Σ を**関数記号**の集合と定義し，$\mathcal{V}$ と Σ は互いに素であるとする．各関数記号 $\mathsf{f} \in \Sigma$ は**型** $[\iota_1 \times \cdots \times \iota_n] \Rightarrow \kappa$（ただし，$\iota_1, \ldots, \iota_n, \kappa \in \mathcal{S}$）を持つ．読みやすさのため，型 $[] \Rightarrow \kappa$ を κ と略記する．本稿では型の整合性が取れている項のみを扱い，$T(\Sigma, \mathcal{V})$ を Σ，$\mathcal{V}$ 上の集合とおく．項 s の型が ι であるときに，ι を明示するときは s を $s : \iota$ と書く．$Var(s)$ は s に含まれている変数の集合を表す．$Var(s) = \emptyset$ のとき，s は**基底項**であるという．さらに，複数の項 $s_1, \ldots, s_n$ に対しては

$Var(s_1,\ldots,s_n) = \bigcup_{1\le i\le n} Var(s_i)$ と定義する．

項 s の**位置** p は正整数の列である．項 s の位置 p における**部分項**を $s_{|p}$ と表記する．項 t と $s_{|p}$ の型が等しいとき， $s[t]_p$ は s の位置 p の項を t で置き換えて得られる項を表す．文脈 C は型が付いた 1 つ以上の**ホール** $\square_i : \iota_i$ を含む項である．項 $s_1 : \iota_1,\ldots,s_n : \iota_n$ に対して，$C[s_1,\ldots,s_n]$ は C に含まれる各 $\square_i : \iota_i$ を s_i に置き換えて得られる項である．

代入 γ を $\mathcal{V}$ から $T(\Sigma,\mathcal{V})$ への型を保存する写像と定義する．γ を項 s に適用して得られる項 $s\gamma$ は s に出現する変数 x を $\gamma(x)$ に置き換えて得られる項である．γ の**定義域** $Dom(\gamma)$ は $\gamma(x) \neq x$ であるような変数 x の集合である．$\gamma(x_i) = s_i$ かつ $Dom(\gamma) = \{x_1,\ldots,x_n\}$ であるような代入 γ を $\{x_1 \mapsto s_1,\ldots,x_n \mapsto s_n\}$ と表すこともある．代入 γ,δ の合成 $\gamma\delta$ は，任意の変数 x に対して $\gamma\delta(x) = \delta(\gamma(x))$ である代入である．代入 γ と集合 $\mathcal{V}' \subseteq \mathcal{V}$ に対して，$\gamma|_{\mathcal{V}'}$ を $\{x \mapsto x\gamma \mid x \in Dom(\gamma) \cap \mathcal{V}'\}$ と定義する．項 s,t に対して，ある代入 γ が存在して $s\gamma = t\gamma$ が成立するとき s と t は**単一化可能**であるといい，γ は s と t の**単一化子**であるという．単一化可能である s と t の単一化子 γ に対して，s,t の任意の単一化子 γ' についてある代入 δ が存在して $\gamma' = \gamma\delta$ が成立するとき，γ を s と t の**最汎単一化子**であるといい，最汎単一化子を $mgu(s,t)$ と表す．

論理制約付き項書換え系 [3] を定義するために，次のシグネチャと解釈を扱う：(1) $\Sigma = \Sigma_{terms} \cup \Sigma_{theory}$ を満たすシグネチャ Σ_{terms} と Σ_{theory}，(2) Σ_{theory} に含まれる型 ι を集合 $\mathcal{I}_\iota$ に割り当てる写像 $\mathcal{I}$，(3) 各関数記号 $\mathsf{f} : [\iota_1 \times \cdots \times \iota_n] \Rightarrow \kappa \in \Sigma_{theory}$ を $\mathcal{I}_1 \times \cdots \times \mathcal{I}_n \Rightarrow \mathcal{I}_\kappa$ の関数 $\mathcal{J}_\mathsf{f}$ に割り当てる写像 $\mathcal{J}$，(4) 各型 ι に対する**値**の集合 $Val_\iota \subseteq \Sigma_{theory}$．ただし，$\mathcal{J}$ は各基本型 ι に関しては Val_ι から $\mathcal{I}_\iota$ への全単射写像とする．さらに，$\Sigma_{terms} \cap \Sigma_{theory} \subseteq Val = \bigcup_\iota Val_\iota$ とする．Σ_{theory} に出現する型を**理論型**，関数記号を**理論記号**と呼ぶ．$T(\Sigma_{theory},\mathcal{V})$ に含まれる項を**理論項**と呼ぶ．

基底理論項に対する**解釈** $[\![\cdot]\!]$ を $[\![\mathsf{f}(s_1,\ldots,s_n)]\!] := \mathcal{J}_\mathsf{f}([\![s_1]\!],\ldots,[\![s_n]\!])$ と定義する．すべての基底理論項 s に対して $[\![s]\!] = [\![c]\!]$ となる唯一の値 c が存在し，c を s **の値**という．$\mathcal{J}_\mathsf{bool} = \{\top,\bot\}$ であるようなある型 bool に対して，型が bool である項を**制約**と呼ぶ．制約 ϕ に対して，$Ran(\gamma|_{Var(\phi)}) \subseteq Val$ を満たす任意の代入 γ に対して $[\![\phi\gamma]\!] = \top$ であるとき，ϕ は**恒真**であるという．$Ran(\gamma|_{Var(\phi)}) \subseteq Val$ かつ $[\![\phi\gamma]\!] = \top$ を満たす代入 γ が存在するとき，ϕ は**充足可能**であるという．代入 γ と制約 ϕ に対して，$Ran(\gamma|_{Var(\phi)}) \subseteq Val$ かつ $[\![\phi\gamma]\!] = \top$ であるとき，γ は ϕ を**満たす**という．

Σ^{core}_{theory}（$\subseteq \Sigma_{theory}$）を関数記号 $\mathsf{true},\mathsf{false} : \mathsf{bool}$, $\wedge,\vee \Rightarrow: [\mathsf{bool},\mathsf{bool}] \Rightarrow \mathsf{bool}$, $\neg : [\mathsf{bool}] \Rightarrow \mathsf{bool}$，すべての理論型 ι に対する関数記号 $=_\iota, \neq_\iota$ のみを含むシグネチャと定義する．$=_\iota$ と $\neq_\iota$ の添字である型 ι が文脈上明らかであるときは ι を省略し，それぞれ $=$, $\neq$ と記述する．$\mathcal{J}$ は Σ^{core}_{theory} を一般的な解釈のもとで評価することとする．

上述の定義では制約で量化子を直接扱うことができない．しかし，量化子を含む論理式ごとに新たな関数記号を Σ_{theory} に追加することで間接的に扱うことができる [2]．よって，以降では制約で量化子を扱えるものとして議論を進める．

整数理論を表現する標準的なシグネチャ Σ^{int}_{theory} は $\Sigma^{core}_{theory} \cup \{+,-,*,/,\mathsf{exp},\mathsf{mod} : [\mathsf{int} \times \mathsf{int}] \Rightarrow \mathsf{int},\ \le,< : [\mathsf{int},\mathsf{int}] \Rightarrow \mathsf{bool}\} \cup \{\mathsf{n} \mid n \in \mathbb{Z}\}$ と定義される．n は整数 n に対応する関数記号である．$\mathcal{J}_\mathsf{f}$ は全域関数でなければならないため，任意の $n,k < 0$ に対して，$\mathcal{J}_/(n,0) = \mathcal{J}_\mathsf{mod}(n,0) = \mathcal{J}_\mathsf{exp}(n,k) = 0$ とする．その他の場合は $\mathcal{J}$ は各関数記号を通常の意味に沿って解釈する．

制約付き書換え規則 $l \to r\ [\phi]$ は同じ型の項 l,r と制約 ϕ の三つ組である．ただし，l は $\mathsf{f}(l_1,\ldots,l_n)$ という形をしており，理論項ではない．$\phi = \mathsf{true}$ であるときは，この書換え規則を $l \to r$ と表記することがある．$LVar(l \to r\ [\phi])$ を $Var(\phi) \cup (Var(r) \setminus Var(l))$ と定義する．代入 γ と書換え規則 $l \to r\ [\phi]$ について，$Ran(\gamma|_{LVar(l\to r\ [\phi])}) \subseteq Val$ かつ $[\![\phi\gamma]\!] = \top$ であるとき，γ は $l \to r\ [\phi]$ を**例化する**という．

$\mathcal{R}_{calc}$ を $\{\mathsf{f}(x_1,\ldots,x_n) \to y\ [y = \mathsf{f}(x_1,\ldots,x_n)] \mid \mathsf{f} : [\iota_1 \times \cdots \times \iota_n] \Rightarrow \kappa \in \Sigma_{theory} \setminus Val\}$ と定義する．左辺 $\mathsf{f}(x_1,\ldots,x_n)$ が理論項であるため $\mathcal{R}_{calc}$ の要素は正確には書換え規則ではないが，$\mathcal{R}_{calc}$ の要素も書換え規則と呼ぶ．$\mathcal{R}$ の**書換え関係** $\to_\mathcal{R}$ を次のように

定義する：$s \to_{\mathcal{R}} t$ が成り立つとき，かつそのときに限り，制約付き書換え規則 $l \to r\ [\phi] \in \mathcal{R} \cup \mathcal{R}_{calc}$，代入 γ，文脈 $C[\,]_p$ が存在し，$s = C[l\gamma]$，$t = C[r\gamma]$，γ が $l \to r\ [\phi]$ を満たす．このとき，その書換えは位置 p で起こるという．$\to_{\mathcal{R}}$ の対称閉包を $\leftrightarrow_{\mathcal{R}}$ と記す．

$\mathcal{R}$ を制約付き書換え規則の集合とする．抽象書換え系 $(T(\Sigma, \mathcal{V}), \to_{\mathcal{R}})$ を**論理制約付き項書換え系**（LCTRS）と呼び，単に $\mathcal{R}$ で LCTRS を表す．$\mathcal{S} \supseteq \{\mathsf{bool}, \mathsf{int}\}$ であり，Σ_{theory} が Σ_{theory}^{int} である LCTRS を**整数 LCTRS** と呼ぶ．

関数記号 $\mathsf{f} \in \Sigma$ に対して，$\mathsf{f}(l_1, \ldots, l_n) \to r\ [\phi]$ という形の書換え規則が $\mathcal{R}$ に含まれるとき，f を $\mathcal{R}$ の**被定義記号**と呼び，$\mathcal{R}$ の被定義記号の集合を $\mathcal{D}_{\mathcal{R}}$ と表す．$\Sigma_{terms} \setminus \mathcal{D}_{\mathcal{R}}$ に含まれる関数記号とすべての値を**構成子**と呼び，$\mathcal{R}$ の構成子の集合を $\mathcal{C}_{\mathcal{R}}$ と表す．$T(\mathcal{C}_{\mathcal{R}}, \mathcal{V})$ に含まれる項を $\mathcal{R}$ の**構成子項**と呼ぶ．任意の $x \in Dom(\gamma)$ について $\gamma(x)$ が基底構成子項であるとき，γ は**基底構成子代入**であるという．

制約付き項は項 s と制約 ϕ の組であり，$s\ [\phi]$ と表される．直観的には $s\ [\phi]$ は ϕ を満たす代入 γ について $s\gamma$ をすべて集めた集合を表す．制約付き項 $s\ [\phi]$ と $t\ [\psi]$ に対して，ϕ を満たす任意の代入 γ に対して ψ を満たす代入 δ が存在して $s\gamma = t\delta$ が成立し，かつ逆もまた同様のとき，$s\ [\phi]$ と $t\ [\psi]$ は等しいといい，$s\ [\phi] \sim t\ [\psi]$ と書く．制約付き書換え規則 $\rho = l \to r\ [\psi] \in \mathcal{R} \cup \mathcal{R}_{calc}$ と位置 q に関する制約付き項上の書換え関係を次のように定義する：$s\ [\phi] \to_{\rho,q} t\ [\phi]$ が成り立つとき，かつそのときに限り，ある代入 γ が存在して，$s_{|q} = l\gamma$ かつ $t = s[r\gamma]_q$ かつ $Ran(\gamma|_{LVar(\rho)}) \subseteq Val \cup Var(\phi)$ であり，$\phi \Rightarrow (\psi\gamma)$ が恒真である．制約付き書換え規則 ρ と位置 q が存在し $s\ [\phi] \to_{\rho,q} t\ [\phi]$ が成立するとき，$s\ [\phi] \to_{base} t\ [\phi]$ と表す．制約付き項上の二項関係 $\to_{\mathcal{R}}$ を $\sim \cdot \to_{base} \cdot \sim$ と定義する．

2.2 論理制約付き書換えにおける書換え帰納法

本稿では文献 [2] と同様に，以降で扱う LCTRS $\mathcal{R}$ について以下の性質を仮定する．

- $\mathcal{R}$ は停止性を持つ．
- $\mathcal{R}$ は十分完全性を持つ．
- Σ に出現する型 ι それぞれについて，型が ι である基底項 s が存在する．

項 s, t と制約 ϕ に対して，三つ組 $s \approx t\ [\phi]$ を（制約付き）**等式**と呼ぶ．$s \simeq t\ [\phi]$ は $s \approx t\ [\phi]$ または $t \approx s\ [\phi]$ を表す．代入 γ と等式 $s \approx t\ [\phi]$ について，γ が ϕ を満しかつ $Var(s) \cup Var(t) \subseteq Dom(\gamma)$ であるとき，γ は $s \approx t\ [\phi]$ を**例化する**という．

定義 2.1 (帰納的定理 [2])**.** $\mathcal{R}$ を LCTRS とする．等式 $s \approx t\ [\phi]$ について，$s \approx t\ [\phi]$ を例化する任意の基底構成子代入 γ ついて $s\gamma \leftrightarrow_{\mathcal{R}}^{*} t\gamma$ が成立するとき，$s \approx t\ [\phi]$ は $\mathcal{R}$ の**帰納的定理**であるという．

等式の集合 $\mathcal{E}$，制約付き書換え規則の集合 $\mathcal{H}$，フラグ $b \in \{\textsc{cmplt}, \textsc{incmplt}\}$ の三つ組 $(\mathcal{E}, \mathcal{H}, b)$ を**証明状態**という．$\mathcal{E}$ は帰納的定理であることを証明したい等式の集合を表し，$\mathcal{H}$ はその証明状態以降にて使用できる帰納法の仮定を表し，b は反証が健全であるかどうかを表す．反証が起こった状況を表す $\bot$ も証明状態と呼ぶ．

定義 2.2 (制約付き書換え帰納法 [2])**. 書換え帰納法**において証明状態に適用する推論規則を図 2.1 に示す．この推論規則のどれかを適用して $(\mathcal{E}, \mathcal{H}, b)$ から $(\mathcal{E}', \mathcal{H}', b')$ を導出するとき，以下のように表記する．

$$(\mathcal{E}, \mathcal{H}, b) \vdash_{ri} (\mathcal{E}', \mathcal{H}', b')$$

本稿では，制約において $x = n$（ただし，$n \in Val$）のように固定の値が格納される変数の集合を $\mathcal{V}_{init}$ と表記し，$\mathcal{V}_{init}$ に含まれる変数を v_i と表記する．Expansion や Simplification が適用される際に使用される書換え規則の変数のうち，$\mathcal{V}_{init}$ に含まれる変数については名前替えをしない．

以下の定理は書換え帰納法の正しさを保証する．

定理 2.3 ([2])**.** $\mathcal{R}$ を LCTRS，$\mathcal{E}$ を等式の集合，$flag \in \{\textsc{cmplt}, \textsc{incmplt}\}$ とする．このとき，ある $\mathcal{H}', flag'$ が存在して $(\mathcal{E}, \emptyset, flag) \vdash_{ri}^{*} (\emptyset, \mathcal{H}', flag')$ が成り立つならば，$\mathcal{E}$ に含まれるすべての等式は $\mathcal{R}$ の帰納的定理である．また，$\mathcal{R}$ が合流性を持ち，かつある $\mathcal{E}', \mathcal{H}'$ が存在して $(\mathcal{E}, \emptyset, \textsc{cmplt}) \vdash_{ri}^{*} \bot$ であるならば，$\mathcal{E}$ は $\mathcal{R}$ の帰納的定理でない等式を含む．

2.3 値割当てを表す論理式の削除による一般化

本節では，文献 [2] の等式の一般化を定義する．

Simplification

$$(\mathcal{E} \uplus \{s \simeq t\ [\phi]\}, \mathcal{H}, b) \vdash_{simp} (\mathcal{E} \cup \{u \approx t\ [\psi]\}, \mathcal{H}, b)$$

ただし，$\approx$ を構成子とみなし，$(s \approx t\ [\phi]) \rightarrow_{\mathcal{R}\cup\mathcal{H}} (u \approx t\ [\psi])$ が成立する．

Expansion

$$(\mathcal{E} \uplus \{s \simeq t\ [\phi]\}, \mathcal{H}, b) \vdash_{expd} (\mathcal{E} \cup \mathit{Expd}(s \approx t\ [\phi], p), \mathcal{H} \cup \mathcal{H}'', b)$$

ただし，$s_{|p} = \mathsf{f}(s_1, \ldots, s_n)$，$\mathsf{f} \in \mathcal{D}_{\mathcal{R}}$，$s_1, \ldots, s_n \in T(\mathcal{C}_{\mathcal{R}}, \mathcal{V})$，$\mathcal{H}' \subseteq \mathcal{H}''$，$\mathcal{H}''$ と $\mathit{Expd}(s \approx t\ [\phi], p)$ は以下のように定義される集合である．

$$\mathcal{H}'' = \begin{cases} \{s \rightarrow t\ [\phi]\} & (\mathcal{R} \cup \mathcal{H} \cup \{s \rightarrow t\ [\phi]\} \text{ が停止性を持つとき}) \\ \emptyset & (\text{そうでないとき}) \end{cases}$$

$$\mathit{Expd}(s \approx t\ [\phi], p) = \left\{ s' \approx t'\ [\phi'] \;\middle|\; \begin{array}{l} \rho = l \rightarrow r\ [\psi] \text{ は } \mathcal{R} \text{ に含まれる規則の名前替え,} \\ \mathit{Var}(\rho) \cap \mathit{Var}(s, t, \phi) = \emptyset,\ \gamma = \mathit{mgu}(s_{|p}, l), \\ (s\gamma \approx t\gamma\ [\phi\gamma \wedge \psi\gamma]) \sim \cdot \rightarrow_{\rho,1.p} \cdot \sim (s' \approx t'\ [\phi']) \end{array} \right\}$$

Deletion

$$(\mathcal{E} \uplus \{s \approx t\ [\phi]\}, \mathcal{H}, b) \vdash_{del} (\mathcal{E}, \mathcal{H}, b)$$

ただし，$s = t$ または ϕ が充足不能である．

Postulate

$$(\mathcal{E}, \mathcal{H}, b) \vdash_{pos} (\mathcal{E} \cup \mathcal{E}', \mathcal{H}, \textsc{incmplt})$$

ただし，$\mathcal{E}'$ は任意の等式の集合である．

EQ-Deletion

$$\begin{aligned} &(\mathcal{E} \uplus \{C[s_1, \ldots, s_n] \simeq C[t_1, \ldots, t_n]\ [\phi]\}, \mathcal{H}, b) \\ &\qquad \vdash_{EQdel} (\mathcal{E} \cup \{C[s_1, \ldots, s_n] \approx C[t_1, \ldots, t_n]\ [\phi \wedge \neg(\bigwedge_{i=1}^{n} s_i = t_i)]\}, \mathcal{H}, b) \end{aligned}$$

ただし，各 i（$1 \leq i \leq n$）について，$s_i, t_i \in T(\Sigma_{theory}, \mathit{Var}(\phi))$ である．

Disproof

$$(\mathcal{E} \uplus \{s \simeq t\ [\phi]\}, \mathcal{H}, \textsc{cmplt}) \vdash_{disprove} \bot$$

ただし，s の型を ι としたとき，以下のどれかが成立する．

- $s, t \in T(\Sigma_{theory}, \mathcal{V})$ であり，かつ，ι が理論型であり，かつ，$\phi \wedge s \neq t$ が充足可能である．
- $s = \mathsf{f}(\vec{s})$，$t = \mathsf{g}(\vec{t})$，かつ，$\mathsf{f} \neq \mathsf{g}$ である構成子であり，ϕ は充足可能である．
- $s \in \mathcal{V} \setminus \mathit{Var}(\phi)$ であり，かつ，ϕ が充足可能であり，かつ，返り値の型が ι であるような構成子が少なくとも 2 つ存在し，以下のうちどちらかが成立する．
  - t は s と異なる変数である．
  - ある $\mathsf{g} \in \mathcal{C}_{\mathcal{R}}$ が存在して t が $\mathsf{g}(\vec{t})$ という形をしている．

図 3　制約付き書換え帰納法の推論規則

変数の集合 X（$\subseteq \mathcal{V}$）に含まれる変数 x について，制約 ϕ が満たされるときに x に割り当てられる値を一意に決定する論理式 $x = v$（ただし，$n \in \mathit{Val}$）を制約から除去して等式を一般化する処理 RmInit_X を以下のように定義する．

$$\mathit{RmInit}_X(s \approx t\ [\phi \wedge \bigwedge_{x \in X} x = n_x]) = (s\sigma \approx t\sigma\ [\phi\sigma])$$

ただし，各 $x \in X$ に対して n_x は値であり，σ は X の変数をこの処理を適用する時点でどこにも現れていない変数に名前変えする代入である．等式 e に RmInit を適用する際には，$X = \mathit{Var}(e) \cap \mathcal{V}_{init}$ とし，X を省略して $\mathit{RmInit}(e)$ と書く．

2.4 ラグランジュ補間式

ラグランジュ補間式は以下のように定義される．

定義 2.4 ([6])**.** F を体とし，そして $x_1, \ldots, x_n$ を F の相異なる要素，$y_1, \ldots, y_n$ を F の要素とし，$\mathcal{P} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$，$z$ を変数とする．このとき，以下のように定義される多項式 $L_{\mathcal{P}}(z)$ を $\mathcal{P}$ の**ラグランジュ補間式**と呼ぶ．

$$L_{\mathcal{P}}(z) := \sum_{(x,y)\in\mathcal{P}} \frac{y}{\prod_{(x',y')\in\mathcal{P}, x\neq x'}(x-x')} \prod_{(x',y')\in\mathcal{P}, x\neq x'} (z-x')$$

この $L_{\mathcal{P}}$ は任意の $(x, y) \in \mathcal{P}$ に対して，$L_{\mathcal{P}}(x) = y$ をみたす次数最小の多項式である [6]．

例 2.5. $\mathcal{P} = \{(1, 2), (2, 4), (3, 6)\}$ に対してラグランジュ補間式 $L_{\mathcal{P}}(z)$ は以下のようになる．

$$\begin{aligned} L_{\mathcal{P}}(z) &= \frac{1}{(1-2)(2-3)}(z-2)(z-3) \\ &\quad + \frac{4}{(2-1)(2-3)}(z-1)(z-3) \\ &\quad + \frac{6}{(3-1)(3-2)}(z-1)(z-2) \\ &= 2z \end{aligned}$$

3 制約付き等式の一般性

本章では等式が別の等式より一般的であるという概念を定義する．そして，その定義において一般化的である等式が帰納的定理ならばその等式よりも具体的である等式も帰納的定理であるということを示す．

制約がない等式においては，$s \approx t$ が $s' \approx t'$ より一般的であるという概念は，等式を項とみなして $s \approx t$ が $s' \approx t'$ または $t' \approx s'$ よりも一般的であることと定義される．本稿ではこの考え方を制約付き等式に自然に拡張する．

定義 3.1 (一般性)**.** $s \approx t\ [\phi], s' \approx t'\ [\phi']$ を制約付き等式とし，$s'' \approx t''\ [\phi'']$ を $s' \approx t'\ [\phi']$ に含まれる変数を $s \approx t\ [\phi]$ と変数を共有しないように名前変えした制約付き等式とする．ある σ が存在して以下のすべてが成立するならば，$s \approx t\ [\phi]$ は $s' \approx t'\ [\phi']$ より**一般的である**といい，$s' \approx t'\ [\phi']$ は $s \approx t\ [\phi]$ より**具体的である**という．

- $s'' = s\sigma$ が成り立つ．
- $t'' = t\sigma$ が成り立つ．
- 任意の $x \in Var(\phi) \cap Var(s, t)$ に対して，$\sigma(x) \in T(\Sigma_{theory}, Var(\phi''))$ が成り立つ．
- $\phi'' \Rightarrow \exists x_1, \ldots, x_n.\ \phi\sigma$ が恒真である．ただし，$\{x_1, \ldots, x_n\} = Var(\phi) \setminus Var(s, t)$ である．

一般的である制約付き等式について以下の性質が自明に成り立つ．

定理 3.2. $s \approx t\ [\phi]$ を $s' \approx t'\ [\phi']$ より一般的な制約付き等式とする．$s \approx t\ [\phi]$ が $\mathcal{R}$ の帰納的定理であるならば，$s' \approx t'\ [\phi']$ も $\mathcal{R}$ の帰納的定理である．

定義 3.1 の一般性は文献 [2] の推論規則 Generalization で用いられる一般性を含む概念である．

4 提案手法の概要

本章では，提案手法の概要を例を用いて述べる．なお，提案手法は以下の性質を満たす整数 LCTRS を対象とする．

- 任意の書換え規則 $l \to r\ [\psi]$ について，r は int 型の値ではなく，かつ，r の任意の部分項 $\mathsf{f}(r_1, \ldots, r_n)$（f は被定義記号）においてどの r_i も int 型の値ではない．

上記の性質は C プログラムを文献 [2] の手法で変換して得られる LCTRS に，さらに値割当てを表す論理式の削除による一般化（2.3 節）を用いるための変換 [2, 5.2 節] を適用して得られる LCTRS が必ず満たす性質である．よって，本稿は C プログラムの検証を目的としているため，対象とする LCTRS が上記の性質を満たす仮定は目的に対して制限にならない．

提案手法では等式の列に対して以下のように補題等式の候補を生成する．

1. ラグランジュ補間を用いて補題等式の候補を生成する．
2. 生成した補題等式の候補が帰納的定理であることの証明を書換え帰納法を用いて試みる．証明に成功した際にはその証明で得られた帰納法の仮定の集合を元の等式の証明状態に追加して，元の等式の証明を再開する．
3. 生成した補題等式の候補が反証された場合は，反証までの過程を参照して生成した補題等式の候補の制約部分を修正し，改めて帰納的定理であることの証明を試みる．

以降ではまず，ラグランジュ補間を用いた補題生成

法の概要を述べる（4.1 節）．次に，補題等式の候補の制約部分を修正する方法を述べる（4.2 節）．

4.1 ラグランジュ補間を用いた一般化の概要

例として，図 2 の発散において (D) が出現する証明状態に至った際に (D) を一般化した補題等式 (D_2) と同様の等式をどのように既出の等式の列 (A), . . . , (D) から得たかを説明する．

1. 等式 (D) に 2.3 節の *RmInit* を適用し以下の等式を生成する．

$$\mathsf{u}_0(n, w_0 + 15, w_1 + 6) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, w_0 + 10, w_1 + 5)) \quad [n \geq 0 \wedge w_1 + 5 \leq n]$$

2. 1. で生成した制約付き等式の等式部の部分項のうち，列において変化しているものを変数に置き換え，“置換え後の変数 = 置換え前の項” という等式を制約部に連接する．これにより，生成した等式が等式 (A)–(D) すべての一般化となる範囲で置換えの一部を可能な限りキャンセルする．

$$\mathsf{u}_0(n, r_0, i_0) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, r_1, i_1)) \quad \begin{bmatrix} n \geq 0 \wedge w_1 + 5 \leq n \wedge r_0 = w_0 + 15 \\ \wedge\, i_0 = w_1 + 6 \wedge i_1 = w_1 + 5 \end{bmatrix}$$

3. 導入した任意の 2 つの変数について，各等式からその 2 つの変数に相当する項の対の集合を対象としてラグランジュ補間を適用することで，導入した変数間の関係を求めて制約に連接する．例えば，等式 (A)–(D) において左辺の u_0 の第 2 引数 i_0 の値は $2, 3, 4, 5$ であり第 3 引数 r_0 の値は $1, 3, 6, 10$ となる．この列から整数の組 $(1, 2), (3, 3), (4, 6), (5, 10)$ を取り出し，ラグランジュ補間を行い，$r_0 = i_0 * (i_0 - 1)/2$ という関係を推定する．同様に $r_1 = i_1 * (i_1 - 1)/2$ という関係も推定し，制約部に連接することで補題等式の候補を推定する

$$\mathsf{u}_0(n, r_0, i_0) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, r_1, i_1)) \quad \begin{bmatrix} n \geq 0 \wedge w_1 + 5 \leq n \wedge r_0 = w_0 + 15 \\ \wedge\, i_0 = w_1 + 6 \wedge i_1 = w_1 + 5 \\ \wedge\, r_0 = i_0 * (i_0 - 1)/2 \\ \wedge\, r_1 = i_1 * (i_1 - 1)/2 \end{bmatrix} \tag{1}$$

なお，この例では等式の列において変数が具体的な値をとっているが，実際には変数を使って表される式となっている．よって，実際の処理で連接される制約は $r_0 = i_0 \times (i_0 - 1)/2 \wedge r_1 = i_1 \times (i_1 - 1)/2$ より複雑なものとなる．

上述のように生成された等式が補題等式，すなわち帰納的定理であることは保証されない．よって，生成した補題等式の候補が帰納的定理であることの証明を書換え帰納法を用いて試みる．残念ながら等式 (1) は $\mathcal{R}_1$ の帰納的定理ではなく，反証される．よって，その反証過程を利用して等式 (1) の制約部を修正する．

4.2 反証された補題等式の候補の修正

等式 (1) に対して，反証過程からどのように修正するかを述べる．等式 (1) は図 4 のように Expansion によって $(i_0 \leq n \wedge v_2 = 0) \wedge v_2 \geq i_0 \wedge i_2 > n$ という場合分けが行われた後に反証される．ここで，制約に含まれる $v_2 = 0$ と $i_2 = i_0 + 1$ を使い等式 (1) に出現しない変数を除去して，$i_0 \leq n$, $0 \geq i_0$, $i_0 + 1 > n$ という制約の列を生成する．提案手法は反証過程から $i_0 \leq n$, $0 \geq i_0$, $i_0 + 1 > n$ という制約の列を入力として受け取る．そして，この列の接頭辞を連接したものを否定し等式 (1) の制約部に連接して得られる等式のうち，等式 (D) より一般的であるものの中で接頭辞が最短である等式を修正結果として生成する．

例えば，長さ 1 の接頭辞を連接した論理式 $i_0 \leq n$ の否定を等式 (1) の制約部分に連接して得られる等式は等式 (D) よりも一般的でないため (D) を証明するための補題等式として適切ではない．長さ 2 の接頭辞を連接した論理式 $i_0 \leq n \wedge 0 \geq i_0$ の否定を連接した以下の等式は等式 (D) よりも一般的である．

$$\mathsf{u}_0(n, r_0, i_0) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, r_1, i_1)) \quad \begin{bmatrix} n \geq 0 \wedge v_1 + 4 \leq n \wedge i_0 = v_1 + 5 \\ \wedge\, i_1 = v_1 + 4 \wedge r_0 = i_0 * (i_0 - 1)/2 \\ \wedge\, r_1 = i_1 * (i_1 - 1)/2 \\ \wedge\, \neg(i_0 \leq n \wedge 0 \geq i_0) \end{bmatrix} \tag{2}$$

よって，上述の等式 (2) を等式 (1) の修正結果として返す．この等式 (2) は $\mathcal{R}_1$ の帰納的定理であり，書換え帰納法により提案手法を使用せずに証明できる．

$$
\begin{aligned}
&(\{\text{等式 (1)}\}, \emptyset, \mathrm{CMPLT}) \\
&\vdash_{expd} \\
&\Big(\big\{(\mathrm{E}) : \mathsf{u}_1(n, r_0, i_0, v_2) \approx \mathsf{u}_2(n, \mathsf{u}_0(n-1, r_1, i_1))\ \big[\text{等式 (1) の制約部} \wedge \underline{i_0 \leq n \wedge v_2 = 0}\big], \ldots\big\}, \{\ldots\}, \mathrm{CMPLT}\Big) \\
&\vdash^*_{simp} \cdot \vdash_{expd} \\
&\left(\begin{array}{l}\big\{(\mathrm{F}) : \mathsf{u}_0(n, r_0, i_0+1) \approx \mathsf{u}_2(n, \mathsf{u}_0(m, r_1, i_1, v_2))\ \big[(\mathrm{E}) \text{ の制約部} \wedge \underline{v_2 \geq i_0}\big], \ldots\big\}, \\ \quad \{\ldots\}, \mathrm{CMPLT}\end{array}\right) \\
&\vdash^*_{simp} \cdot \vdash_{expd} \\
&\Big(\big\{\mathsf{return}(r_0) \approx \mathsf{u}_2(n, \mathsf{u}_0(m, r_1, i_3))\ \big[(\mathrm{F}) \text{ の制約部} \wedge i_3 = i_1 + 1 \wedge i_2 = i_0 + 1 \wedge \underline{i_2 > n}\big], \ldots\big\}, \{\ldots\}, \mathrm{CMPLT}\Big) \\
&\vdash^*_{simp} \cdot \vdash_{EQdel} \cdot \vdash_{disprove} \bot
\end{aligned}
$$

図 4　等式 (1) の反証過程

5　ラグランジュ補間を用いた一般化

本章では，ラグランジュ補間を用いた一般化による補題生成法について述べる．この補題生成法は，発散しそうな有限の証明過程から取り出した等式列 $s_1 \approx t_1\ [\phi_1], \ldots, s_k \approx t_k\ [\phi_k]$ を入力として受け取り，ラグランジュ補間を用いて等式 $s_k \approx t_k\ [\phi_k]$ の一般化を行い補題等式の候補を生成する．具体的には以下の手続きで補題等式の候補を生成する．

1. 等式 $s_k \approx t_k\ [\phi_k]$ の部分項を変数に置き換えて等式部分を一般化する．
2. 1. の置換えで導入した変数間の関係を表す制約をラグランジュ補間により求め，1. で生成した等式の制約部分に連接する．

以降では，上記の手続きそれぞれの詳細を説明する．

5.1　変数への置換えによる等式部分の一般化

本節では，等式 $s_k \approx t_k\ [\phi_k]$ の部分項を変数に置換えて等式部を一般化する手続きについて述べる．この手続きは等式列 $s_1 \approx t_1\ [\phi_1], \ldots, s_k \approx t_k\ [\phi_k]$ が与えられ，以下のように等式部分を一般化する．

1. 末尾の等式 $s_k \approx t_k\ [\phi_k]$ を 2.3 節の *RmInit* により一般化し，$s \approx t\ [\phi]$ を得る．その際に導入された名前替え代入を σ（2.3 節の σ）とする．便宜上，以降では $s_i\sigma \approx t_i\sigma\ [\phi_i\sigma]$ を $s_i \approx t_i\ [\phi_i]$，$s \approx t\ [\phi]$ を $s_k \approx t_k\ [\phi_k]$ とする．
2. 等式部分 $s_k \approx t_k$ の部分項のうち，等式列 $s_1 \approx t_1\ [\phi_1], \ldots, s_k \approx t_k\ [\phi_k]$ に沿って変化しているものをそれまでに使用されていない変数に置き換え，$s'_k \approx t'_k\ [\phi_k]$ を得る．
3. 生成した等式が $s_1 \approx t_1\ [\phi_1], \ldots, s_k \approx t_k\ [\phi_k]$ すべてより一般的になるように，2. の変数の置換えを可能な限りキャンセルする論理式 ϕ' を ϕ_k に連接し，$s'_k \approx t'_k\ [\phi_k \wedge \phi']$ を得る．

以降では，上記の 2. と 3. の詳細を順に説明する．

5.1.1　変化する部分項の一般化

2.3 節の *RmInit* を適用して得られた $s_k \approx t_k\ [\phi_k]$ の等式部分 $s_k \approx t_k$ の部分項のうち，等式列 $s_1 \approx t_1\ [\phi_1], \ldots, s_k \approx t_k\ [\phi_k]$ に沿って変化しているものをそれまでに使用されていない新しい変数 $z_1, z_2, \ldots$ に置き換える．そのために，等式列 $s_1 \approx t_1\ [\phi_1], \ldots, s_k \approx t_k\ [\phi_k]$ に沿って変化している部分項を定義する．

まず，$\approx$ を 2 引数関数記号とみなして，各 $i \in \{1, \ldots, k\}$ について以下が成立する文脈 E，自然数 m を発見する．

- 項 $u_{i,1}, \ldots, u_{i,m} \in T(\Sigma_{theory}, Var(\phi_i))$ が存在して，$(s_i \approx t_i) = E[u_{i,1}, \ldots, u_{i,m}]$ が成り立つ．

E を求めることは決定可能であり，E として最大のものをとるとすれば E は一意に定まる．発見できなかった場合は補題等式の候補を生成せずに提案手法自体を終了する．

次に，$u_{k,1}, \ldots, u_{k,m}$ をそれまでに使用されていない新しい変数 $z_1, \ldots, z_m$ に置き換えて以下の等式を生成する．

$$E[z_1, \ldots, z_m]\ [\phi_k]$$

最後に，生成した等式 $E[z_1,\ldots,z_m]\ [\phi_k]$ が $s_1 \approx t_1\ [\phi_1],\ldots,s_k \approx t_k\ [\phi_k]$ すべてより一般的になるように，変数の置き換えを可能な限りキャンセルする制約を以下のように ϕ_k に連接し等式を生成する．

$$E[z_1,\ldots,z_m]\ [\phi_k \wedge \bigwedge_{j\in I} z_j = u_{k,j}]$$

ただし，$I \subseteq \{1,\ldots,m\}$ は上述の等式が各等式 $s_i \approx t_i\ [\phi_i]$ より一般的である最大の集合とし，求め方は後述する．論理式 $\bigwedge_{j\in I} z_j = u_{k,j}$ が ϕ_k に連接される論理式 ϕ' に相当する．

提案手法では I を次のように求める．$b_1,\ldots,b_m$ を 0 もしくは 1 をとるメタ変数とする．このとき，以下の等式が $s_1 \approx t_1\ [\phi_1],\ldots,s_k \approx t_k\ [\phi_k]$ より一般的であり，かつ $[\![b_1+\cdots+b_m]\!]$ が最大になる $I = \{i \mid b_i = 1\}$ を求める．

$$E[z_1,\ldots,z_m]\ [\phi_k \wedge \bigwedge_{1\le j\le m} b_j * z_j = b_j * u_{k,j}]$$

I の候補は有限である．また，制約が線形整数制約である場合には等式が一般的であるかどうかは決定可能である．よって，制約が線形整数制約である場合には，I は存在するならば必ず求められる．

例 5.1. 図 1 の $\mathcal{R}_1$ について，等式 $\mathsf{sum}(n) \approx \mathsf{sum}_3(n)$ が帰納的定理であることを証明する際に，1 章で説明したように証明が発散する．このとき，図 5 の等式列が得られる．

1. (D′) を 2.3 節の *RmInit* により一般化し，以下の等式を得る．
$$\mathsf{u}_0(n,r_7,i_7) \approx \mathsf{u}_2(n,\mathsf{u}_0(m,r_6,i_6)) \quad [\psi]$$
ただし，ψ を (D′) の制約から $v_0 = 0 \wedge v_1 = 0$ を取り除き，変数 v_0, v_1 を変数 w_0, w_1 に名前替えして得られる以下の制約とする．
$$\psi = \begin{pmatrix} 0 \le n \wedge m = n-1 \wedge r_0 = w_0 \wedge i_0 = w_1 + 1 \\ \wedge\, r_1 = w_0 + 1 \wedge i_1 = w_1 + 2 \wedge w_1 + 1 \le n \\ \wedge\, r_2 = r_0 + 1 \wedge i_2 = i_0 + 1 \wedge i_1 \le n \\ \wedge\, r_4 = r_2 + 2 \wedge i_4 = i_2 + 1 \wedge i_3 \le n \\ \wedge\, r_6 = r_4 + 3 \wedge i_6 = i_4 + 1 \wedge i_5 \le n \end{pmatrix}$$
2. 上述の制約付き等式の等式部分の部分項のうち変化しているものをそれまでに使用されていない変数 $z_1,\ldots,z_4$ に置き換えて以下を得る．
$$\mathsf{u}_0(n,z_1,z_2) \approx \mathsf{u}_2(n,\mathsf{u}_0(m,r_6,i_6)) \quad [\psi]$$
3. 生成した等式が等式 (A′)–(D′) すべてより一般的であるように，変数の置換えを可能な限りキャンセルする制約 $z_1 = r_7 \wedge z_2 = i_7 \wedge z_4 = i_6$ を制約部に連接し以下の等式を得る．
$$\text{(L)}\colon\ \mathsf{u}_0(n,z_1,z_2) \approx \mathsf{u}_2(n,\mathsf{u}_0(m,z_3,z_4)) \quad \begin{bmatrix} \psi \wedge z_1 = r_7 \wedge z_2 = i_7 \\ \wedge\, z_4 = i_6 \end{bmatrix}$$

前述したように生成した等式は帰納的定理ではないため，適切な補題等式ではない．次節の手法により，この等式の制約部を修正することで，最終的には適切な補題等式を得る．

5.2 ラグランジュ補間による制約の生成

本節では，等式部分の一般化で変数に置き換えられた任意の 2 つの位置について，入力として与えられている等式列 $s_1 \approx t_1\ [\phi_1],\ldots,s_k \approx t_k\ [\phi_k]$ から対の列を抽出し，その列にラグランジュ補間を適用することで，その 2 つの位置にある変数の間に成り立つ関係を多項式の等式として推定する．推定された制約すべてを制約部に連接し，生成する等式が発散の過程で以降に出現する等式の一般化になるように制約部を推定する．なお，次数の大きい多項式関係は発散の過程で以降に出現する等式において成り立たないと予想し，採用しない．

ラグランジュ補間によって，以下の論理式を 5.1.1 節で生成した補題候補の制約部に連接する．

$$\phi_k\gamma \wedge \bigwedge_{p\in\mathcal{P}} p \wedge \bigwedge_{x\in Var(E)\cap Var(\phi_k)} x\sigma = x\gamma \qquad (3)$$

ただし，γ は ϕ_k に含まれている変数をそれまでに使用されていない変数に名前替えする代入であり，$\mathcal{P}$ は

$$z_j = L_{\{(u_{1,i}\gamma,u_{1,j}\gamma),\ldots,(u_{k,i}\gamma,u_{k,j}\gamma)\}}(z_i)$$

という形の論理式のうちで以下のすべてを満たすものの集合である．

- $i,j \in \{1,\ldots,m\}$ である．
- 任意の $\phi_k\gamma$ を満たす代入 δ について，$u_{1,i}\gamma\delta,\ldots,u_{k,i}\gamma\delta$ の値は異なる．
- 任意の $\phi_k\gamma$ を満たす代入 δ について，多項

$$(\mathrm{A}') : \mathsf{u}_0(n, r_1, i_1) \approx \mathsf{u}_2(n, \mathsf{u}_0(m, r_0, i_0)) \quad \begin{bmatrix} v_0 = 0 \wedge v_1 = 0 \wedge 0 \leq n \wedge m = n-1 \wedge r_0 = v_0 \\ \wedge\, i_0 = v_1 + 1 \wedge r_1 = v_0 + 1 \wedge i_1 = v_1 + 2 \wedge v_1 + 1 \leq n \end{bmatrix}$$

$$(\mathrm{B}') : \mathsf{u}_0(n, r_3, i_3) \approx \mathsf{u}_2(n, \mathsf{u}_0(m, r_2, i_2)) \quad \begin{bmatrix} \text{等式 } (\mathrm{A}') \text{ の制約部} \wedge r_3 = r_1 + 2 \wedge i_3 = i_1 + 1 \\ \wedge\, r_2 = r_0 + 1 \wedge i_2 = i_0 + 1 \wedge i_1 \leq n \end{bmatrix}$$

$$(\mathrm{C}') : \mathsf{u}_0(n, r_5, i_5) \approx \mathsf{u}_2(n, \mathsf{u}_0(m, r_4, i_4)) \quad \begin{bmatrix} \text{等式 } (\mathrm{B}') \text{ の制約部} \wedge r_5 = r_3 + 3 \wedge i_5 = i_3 + 1 \\ \wedge\, r_4 = r_2 + 2 \wedge i_4 = i_2 + 1 \wedge i_3 \leq n \end{bmatrix}$$

$$(\mathrm{D}') : \mathsf{u}_0(n, r_7, i_7) \approx \mathsf{u}_2(n, \mathsf{u}_0(m, r_6, i_6)) \quad \begin{bmatrix} \text{等式 } (\mathrm{C}') \text{ の制約部} \wedge r_7 = r_5 + 4 \wedge i_7 = i_5 + 1 \\ \wedge\, r_6 = r_4 + 3 \wedge i_6 = i_4 + 1 \wedge i_5 \leq n \end{bmatrix}$$

図 5　等式 $\mathsf{sum}(n) \approx \mathsf{sum}_3(n)$ が $\mathcal{R}_1$ の帰納的定理であることを証明する際に得られる等式列

式 $\delta(L_{\{(u_{1,i}\gamma, u_{1,j}\gamma), \ldots, (u_{k,i}\gamma, u_{k,j}\gamma)\}}(z_i))$ の次数が $k-1$ 未満である．

なお，ラグランジュ補間の性質より，多項式 $L_{\{(u_{1,i}\gamma, u_{1,j}\gamma), \ldots, (u_{k,i}\gamma, u_{k,j}\gamma)\}}(z_i)$ の次数は $k-1$ 以下である．

$\bigwedge_{p \in \mathcal{P}} p$ はラグランジュ補間式から生成した多項式関係を連接した制約である．$\bigwedge_{x \in Var(E) \cap Var(\phi_k)} x\sigma = x\gamma$ は変化していない変数の値を固定する制約である．$\phi_k\gamma$ に対してラグランジュ補間式を連接する理由は，ラグランジュ補間式の係数に $s_1 \approx t_1, \ldots, s_k \approx t_k$ に含まれる変数が含まれており，その情報を補完するためであり，γ を適用する理由は，5.1.1 節で生成した補題候補の制約部に含まれる変数と，変数名が衝突することを避けるためである．

$\mathcal{P}$ を求める際の 2 番目の条件が満たされるかどうかの判定は以下の以下の論理式の恒真性を判定することにより行える．

$$\phi_k\gamma \Rightarrow \bigwedge_{i', j' \in \{1, \ldots, k\}, i' \neq j'} u_{i',i}\gamma \neq u_{j',i}\gamma$$

また，3 番目の条件が満たされるかどうかの判定は以下の論理式の恒真性を判定することにより行える．

$\phi_k\gamma \Rightarrow$ (展開した式の $k-1$ 次の項の係数) $= 0$

例 5.2. 例 5.1 で生成した等式にさらにラグランジュ補間により生成した制約を追加する．等式 (D') の制約に含まれるすべての変数をこれまでに現れていない新しい変数に名前替えして得られる制約を ϕ とおく．

$$\phi = \begin{pmatrix} v'_0 = 0 \wedge v'_1 = 0 \wedge 0 \leq n' \wedge m' = n'-1 \\ \wedge\, r'_0 = v'_0 \wedge i'_0 = v'_1 + 1 \wedge r'_1 = v'_0 + 1 \\ \wedge\, i'_1 = v'_1 + 2 \wedge v'_1 + 1 \leq n' \\ \wedge\, r'_2 = r'_0 + 1 \wedge i'_2 = i'_0 + 1 \wedge i'_1 \leq n' \\ \wedge\, r'_4 = r'_2 + 2 \wedge i'_4 = i'_2 + 1 \wedge i'_3 \leq n' \\ \wedge\, r'_6 = r'_4 + 3 \wedge i'_6 = i'_4 + 1 \wedge i'_5 \leq n' \end{pmatrix}$$

また，各 $\mathcal{P}_{i,j}$ を以下の集合とする．

- $\mathcal{P}_{2,1} = \{(i'_1, r'_1), (i'_3, r'_3), (i'_5, r'_5), (i'_7, r'_7)\}$
- $\mathcal{P}_{4,1} = \{(i'_0, r'_1), (i'_2, r'_3), (i'_4, r'_5), (i'_6, r'_7)\}$
- $\mathcal{P}_{2,3} = \{(i'_1, r'_0), (i'_3, r'_2), (i'_5, r'_4), (i'_7, r'_6)\}$
- $\mathcal{P}_{4,3} = \{(i'_0, r'_0), (i'_2, r'_2), (i'_4, r'_4), (i'_6, r'_6)\}$
- $\mathcal{P}_{2,4} = \{(i'_1, i'_0), (i'_3, i'_2), (i'_5, i'_4), (i'_7, i'_6)\}$
- $\mathcal{P}_{4,2} = \{(i'_0, i'_1), (i'_2, i'_3), (i'_4, i'_5), (i'_6, i'_7)\}$

このとき，各 $\mathcal{P}_{i,j}$ のラグランジュ補間式は以下のようになる．

- $L_{\mathcal{P}_{2,1}}(z) =$

$$\begin{aligned} &\frac{(z - i'_3) * (z - i'_5) * (z - i'_7)}{(i'_1 - i'_3) * (i'_1 - i'_5) * (i'_1 - i'_7)} * r'_1 \\ &+ \frac{(z - i'_1) * (z - i'_5) * (z - i'_7)}{(i'_3 - i'_1) * (i'_3 - i'_5) * (i'_3 - i'_7))} * r'_3 \\ &+ \frac{(z - i'_1) * (z - i'_3) * (z - i'_7)}{(i'_5 - i'_1) * (i'_5 - i'_3) * (i'_5 - i'_7)} * r'_5 \\ &+ \frac{(z - i'_1) * (z - i'_3) * (z - i'_5)}{(i'_7 - i'_1) * (i'_7 - i'_3) * (i'_7 - i'_5)} * r'_7 \end{aligned}$$

- $L_{\mathcal{P}_{4,1}}(z) =$

$$
\begin{aligned}
&\frac{(z-i'_2)*(z-i'_4)*(z-i'_6)}{(i'_0-i'_2)*(i'_1-i'_4)*(i'_1-i'_6)}*r'_1\\
&+\frac{(z-i'_0)*(z-i'_4)*(z-i'_6)}{(i'_2-i'_0)*(i'_2-i'_4)*(i'_2-i'_6))}*r'_3\\
&+\frac{(z-i'_0)*(z-i'_2)*(z-i'_6)}{(i'_4-i'_0)*(i'_4-i'_2)*(i'_4-i'_6)}*r'_5\\
&+\frac{(z-i'_0)*(z-i'_2)*(z-i'_4)}{(i'_6-i'_0)*(i'_6-i'_2)*(i'_6-i'_4)}*r'_7
\end{aligned}
$$

- $L_{\mathcal{P}_{2,3}}(z) =$

$$
\begin{aligned}
&\frac{(z-i'_3)*(z-i'_5)*(z-i'_7)}{(i'_1-i'_3)*(i'_1-i'_5)*(i'_1-i'_7)}*r'_0\\
&+\frac{(z-i'_1)*(z-i'_5)*(z-i'_7)}{(i'_3-i'_1)*(i'_3-i'_5)*(i'_3-i'_7))}*r'_2\\
&+\frac{(z-i'_1)*(z-i'_3)*(z-i'_7)}{(i'_5-i'_1)*(i'_5-i'_3)*(i'_5-i'_7)}*r'_4\\
&+\frac{(z-i'_1)*(z-i'_3)*(z-i'_5)}{(i'_7-i'_1)*(i'_7-i'_3)*(i'_7-i'_5)}*r'_6
\end{aligned}
$$

- $L_{\mathcal{P}_{4,3}}(z) =$

$$
\begin{aligned}
&\frac{(z-i'_2)*(z-i'_4)*(z-i'_6)}{(i'_0-i'_2)*(i'_1-i'_4)*(i'_1-i'_6)}*r'_0\\
&+\frac{(z-i'_0)*(z-i'_4)*(z-i'_6)}{(i'_2-i'_0)*(i'_2-i'_4)*(i'_2-i'_6))}*r'_2\\
&+\frac{(z-i'_0)*(z-i'_2)*(z-i'_6)}{(i'_4-i'_0)*(i'_4-i'_2)*(i'_4-i'_6)}*r'_4\\
&+\frac{(z-i'_0)*(z-i'_2)*(z-i'_4)}{(i'_6-i'_0)*(i'_6-i'_2)*(i'_6-i'_4)}*r'_6
\end{aligned}
$$

- $L_{\mathcal{P}_{2,4}}(z) =$

$$
\begin{aligned}
&\frac{(z-i'_3)*(z-i'_5)*(z-i'_7)}{(i'_1-i'_3)*(i'_1-i'_5)*(i'_1-i'_7)}*i'_0\\
&+\frac{(z-i'_1)*(z-i'_5)*(z-i'_7)}{(i'_3-i'_1)*(i'_3-i'_5)*(i'_3-i'_7))}*i'_2\\
&+\frac{(z-i'_1)*(z-i'_3)*(z-i'_7)}{(i'_5-i'_1)*(i'_5-i'_3)*(i'_5-i'_7)}*i'_4\\
&+\frac{(z-i'_1)*(z-i'_3)*(z-i'_5)}{(i'_7-i'_1)*(i'_7-i'_3)*(i'_7-i'_5)}*i'_6
\end{aligned}
$$

- $L_{\mathcal{P}_{4,2}}(z) =$

$$
\begin{aligned}
&\frac{(z-i'_2)*(z-i'_4)*(z-i'_6)}{(i'_0-i'_2)*(i'_1-i'_4)*(i'_1-i'_6)}*i'_1\\
&+\frac{(z-i'_0)*(z-i'_4)*(z-i'_6)}{(i'_2-i'_0)*(i'_2-i'_4)*(i'_2-i'_6))}*i'_3\\
&+\frac{(z-i'_0)*(z-i'_2)*(z-i'_6)}{(i'_4-i'_0)*(i'_4-i'_2)*(i'_4-i'_6)}*i'_5\\
&+\frac{(z-i'_0)*(z-i'_2)*(z-i'_4)}{(i'_6-i'_0)*(i'_6-i'_2)*(i'_6-i'_4)}*i'_7
\end{aligned}
$$

以上のラグランジュ補間式のうち，ϕ を満たす任意の代入 γ に対して $\gamma(L_{\mathcal{P}_{2,1}}(z))$，$\gamma(L_{\mathcal{P}_{4,1}}(z))$，$\gamma(L_{\mathcal{P}_{2,3}}(z))$，$\gamma(L_{\mathcal{P}_{4,3}}(z))$ の次数は 2 であり，$\gamma(L_{\mathcal{P}_{2,4}}(z))$，$\gamma(L_{\mathcal{P}_{4,2}}(z))$ の次数は 1 である．提案手法では次数 3 未満の式を $\mathcal{P}$ として採用するため，以下の等式が補題候補として生成される．

$$
\begin{aligned}
&\mathsf{u}_0(n,z_1,z_2) \approx \mathsf{u}_2(n,\mathsf{u}_0(m,z_3,z_4))\\
&\qquad\begin{bmatrix}
\psi \wedge z_1 = r_7 \wedge z_2 = i_1\\
\wedge\, z_4 = i_6 \wedge \phi\\
\wedge\, z_1 = L_{\mathcal{P}_{2,1}}(z_2) \wedge z_1 = L_{\mathcal{P}_{4,1}}(z_4)\\
\wedge\, z_3 = L_{\mathcal{P}_{2,3}}(z_2) \wedge z_3 = L_{\mathcal{P}_{4,3}}(z_4)\\
\wedge\, z_4 = L_{\mathcal{P}_{2,4}}(z_2) \wedge z_2 = L_{\mathcal{P}_{4,2}}(z_4)\\
\wedge\, n = n' \wedge m = m'
\end{bmatrix}
\end{aligned}
$$

6 反証された等式の修正

本章では生成した補題等式の候補が反証された場合にその等式の制約部を修正する手続きについて述べる．

この手続きは以下を入力として受け取る．

- 5 章で生成した補題候補の等式 $s' \approx t'\ [\phi']$
- 5 章で一般化の対象とした等式 $s_k \approx t_k\ [\phi_k]$
- 制約の列 $\psi_1, \ldots, \psi_n$

ただし，ψ_i は $s' \approx t'\ [\phi']$ が反証されるまでに生じた n 回の場合分けのうちで i 回目の場合分けで制約部に連接された論理式である．そして，以下の等式を修正結果として返す．

$$s' \approx t'\ [\phi' \wedge \neg(\psi_1 \wedge \cdots \wedge \psi_j)]$$

ただし，j は $s' \approx t'\ [\phi' \wedge \neg(\psi_1 \wedge \cdots \wedge \psi_i)]$ が $s_k \approx t_k\ [\phi_k]$ より一般的である数 i $(1 \leq i \leq n)$ のうちで最小の値とする．

一見，j として n を取ればよい，すなわち，$\neg(\psi_1 \wedge \cdots \wedge \psi_n)$ を連接すればよいと考えられる．しかし，その場合には帰納的定理ではない補題等式に修正される可能性がある．よって，修正して得られる等式が $s_k \approx t_k\ [\phi_k]$ より一般的であるように，$\psi_1, \ldots, \psi_n$ の接頭辞を連接した論理式を否定したものを ϕ' に連接する．一方，この接頭辞を短く取りすぎると $s_k \approx t_k\ [\phi_k]$ よりも一般的ではなくなる場合がある．

例 6.1. 以下を本節の手続きの入力として受け取った場合を考える．

- 4 章の等式 (1)
- 図 2 の等式 (D)
- $i_0 \leq n, 0 \geq i_0, i_0 + 1 > n$

このとき，等式 (2) が修正結果として返される．

7 おわりに

本稿では，整数 LCTRS を対象として，ラグランジュ補間を用いた補題生成法を提案し，さらにそれが反証された場合に修正する方法を提案した．以上の手法を用いて，既存手法 [2] において適切な補題等式の生成に失敗する二重ループを含むプログラム例の等価性証明に対して，提案手法では適切な補題の生成に成功することを報告した．

今後の課題として，本手法を実装し，制約付き書換え帰納法に基づく定理証明系に組み込むことが挙げられる．また，提案手法では二重ループを含むプログラム例に対しては適切な補題を生成できるが，その他の例において提案手法がどの程度有効なのかは明らかではない．よって，提案手法の有効性の評価を行うことも今後の課題として挙げられる．

謝辞 本研究は JSPS 科研費 JP18K11160 の助成を受けたものである．

参考文献

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

[2] Fuhs, C., Kop, C., and Nishida, N.: Verifying Procedural Programs via Constrained Rewriting Induction, *ACM Transactions on Computational Logic*, Vol. 18, No. 2(2017), pp. 14:1–14:50.

[3] Kop, C. and Nishida, N.: Term Rewriting with Logical Constraints, *Proceedings of the 9th International Symposium on Frontiers of Combining Systems*, Fontaine, P., Ringeissen, C., and Schmidt, R. A.(eds.), Lecture Notes in Computer Science, Vol. 8152, Springer, 2013, pp. 343–358.

[4] Reddy, U. S.: Term rewriting induction, *Proceedings of the 10th International Conference on Automated Deduction*, Stickel, M. E.(ed.), Lecture Notes in Computer Science, Vol. 449, Springer, 1990, pp. 162–177.

[5] Urso, P. and Kounalis, E.: Sound generalizations in mathematical induction, *Theoretical Computer Science*, Vol. 323, No. 1(2004), pp. 443–471.

[6] Victor, S.: *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press, 2008.

[7] Walsh, T.: A Divergence Critic for Inductive Proof, *Journal of Artificial Intelligence Research*, Vol. 4, No. 1(1996), pp. 209–235.