1F2-04

自動類推証明のための射影ラベル付けに基づく効率的スキーママッチング

Efficient Schema Matching Algorithm Based on Projection Point Labeling for Automatic Analogical Theorem Prover

山田 敬三[*1] 尹 淑萍[*2] 原尾 政輝[*1] 平田 耕一[*1]
Keizo YAMADA Shuping YIN Masateru HARAO Kouichi HIRATA

[*1]九州工業大学 情報工学部 知能情報工学科
Depertment of Artificial Intelligence, Kyushu Institute of Technology

[*2]九州工業大学大学院 情報工学研究科 情報科学専攻
Graduate School of Computer Science and Systems Engineering, Information Science, Kyushu Institute of Technology

We discuss a second order predicate schema matching. We have designed a schema matching algorithm without free individual variables. In this paper, we introduce an extended matching which is a matching between a schema allowed renaming the constants and a formula. Firstly, We design the extended schema matching algorithm which constracts judgment terms for deciding the matchability. Then, we discrib a method deriving matchers form the judgment terms.

1. はじめに

知識のテンプレートであるスキーマを，証明制御情報として用いる知識処理をスキーマ誘導知識処理という [3]．特に，応用の観点から，スキーマとして 2 階の表現を用いた知識処理に関する研究として，プログラム変換 [8]，プログラム自動合成 [3]，類推 [2, 4] などがある．これらの研究においては，スキーマ誘導のための 2 階マッチングが重要な技術となる．

本論文では，2 階の表現である述語論理式スキーマと閉じた論理式とのマッチングであるスキーママッチングについて述べる．このスキーママッチングは一般には，そのマッチング問題が NP-完全であることが知られている [1]，2 階マッチングとみなすことができる．そこで，[6] では，様々な構文的な条件の下で 2 階マッチング問題が NP-完全となるか，多項式時間で解けるかを詳細に調べた．我々は，さらに，スキーマに個体自由変数を含まないスキーママッチング問題が効率的に解けることを示し [5, 11]，Huet と Lang[7, 8] の書き換え系を拡張したマッチングアルゴリズムを設計した．本論文では，このスキーママッチングを拡張し，スキーマに出現する定数の名前換えを許すマッチングを扱う．

2. 準備

IC を個体定数の集合とし，$a, b, c, \cdots$ で表す．IV を個体変数の集合とし，$x, y, z, \cdots$ で表す．FC を関数定数の集合とし，$f, g, h, \cdots$ で表す．PC を述語定数の集合とし，$p, q, r, \cdots$ で表す．PV を述語変数の集合とし，$P, Q, R, \cdots$ で表す．個体変数が出現しない論理式を閉じているといい，閉じた論理式を $\phi, \psi, \cdots$ で表す．スキーマとは，論理式に述語変数を加えた閉じた 2 階論理式であり，$\Phi, \Psi, \cdots$ で表す．例えば，以下の式 Φ はスキーマである．

$\Phi_1 = P(c_S) \wedge \forall x.(P(x) \supset P(f_S(x))) \supset P(f_S(f_S(c_S)))$.

ここで，c_S，f_S はスキーマに出現する定数を表しており，それぞれスキーマ個体定数 (SIC)，スキーマ関数定数 (SFC) という．以降，スキーマ個体定数とスキーマ関数定数は，特に区別の必要がない場合，単にスキーマ定数という．また，スキーマや論理式は型付 2 階の項として表すことができる．以降では，スキーマ，論理式，1 階項を特に区別しない場合，これらを単に項と呼ぶ．項 t の最も左側に現れる記号を $hd(t)$ と表す．例えば上の Φ_1 に対して，$hd(\Phi_1) = \supset$ となる．

$V = IV \cup PV \cup SIC \cup SFC$ とする．代入 θ とは，有限の $v \in V$ に対して $\theta(v) \neq v$ となる，V から論理式への写像である．代入 θ が，$v_i \in V$, 項 t_i $(1 \leq i \leq n)$ に対して，$\theta(v_i) = t_i$ となるとき θ を $\{v_1 := t_1, \cdots, v_n := t_n\}$ と表す．

n 個の項の並び $t_1, \cdots, t_n$ を $\overline{t_n}$ と表す．項 t に代入 θ を適用した結果 $t\theta$ は，次のように帰納的に定義される．

1. $t = c$ のとき，$c \in IC$ ならば，$t\theta = c$ であり，$c \in SIC$ かつ $\theta(c) = d \quad (\in IC)$ ならば，$t\theta = d$.
2. $t = x \in IV$ かつ $\theta(x) = t'$ ならば，$t\theta = t'$ であり，そうでなければ $t\theta = x$.
3. $t = f(\overline{t_n})$ かつ $f \in FC \cup PC$ ならば，$t\theta = f(\overline{t_n\theta})$.
4. $t = P(\overline{t_n})$，$P \in PV$ かつ $(P := \lambda\overline{v_n}.t') \in \theta$ ならば，$t\theta = t'\{\overline{v_n := t_n\theta}\}$ であり，そうでなければ $t\theta = P(\overline{t_n})$.
5. $t = t_1 \# t_2$，$\# \in \{\wedge, \vee, \supset\}$ ならば $t\theta = t_1\theta \# t_2\theta$ であり，$t = \neg t'$ ならば，$t\theta = \neg(t'\theta)$.
6. $t = \natural x.t'$，$\natural \in \{\forall, \exists\}$ ならば $t\theta = \natural y.((t'\{x := y\})\theta$. ただし，$y$ は新しい変数とする．

例 1 Φ_1 を上の例で用いたスキーマとする．Φ_1 に代入 $\theta = \{P := \lambda x_1.\forall z.p(z, u),\ c_S := 0,\ f_S := f\}$ を適用した結果 $\Phi\theta$ は次の論理式になる．

$$
\begin{aligned}
& \Phi\theta \\
= \; & (P(c_S) \wedge \forall x.(P(x) \supset P(f_S(x))) \supset P(f_S(f_S(c_S))))\theta \\
= \; & \forall z.p(z, 0) \wedge \forall x.(\forall z.p(z, x) \supset \forall z.p(z, f(x))) \\
& \quad \supset \forall z.p(z, f(f(0))).
\end{aligned}
$$

スキーマと論理式の組の有限集合をマッチング表現という．特に，述語変数 P_i，論理式 φ_i $(i \in I)$ に対して，$\{\langle P_i(t^i_1, \cdots, t^i_{n_i}), \varphi_i\rangle \mid i \in I\}$ の形のマッチング表現を簡約化マッチング表現という．

連絡先: 山田 敬三，九州工業大学情報工学部，郵便 820-8502 福岡県飯塚市川津 680-4，Tel:0948-29-7884，Fax:0948-29-7601，e-mail:yamada@ai.kyutech.ac.jp

3. 射影ラベル付けスキーママッチング

本節では，[8] の 2 階マッチングの拡張である，射影ラベル付けに基づくスキーママッチングアルゴリズムについて述べる．本アルゴリズムは，与えられたマッチング表現を簡約化マッチング表現に変換する簡単化手続き Simple，マッチング代入の集合をラベル付き項で判定項を求める手続きからなる，更に，判定項からマッチング代入を取り出すことができる．

E をマッチング表現とし，$\langle s,t\rangle \in E$ とする．このとき，簡単化手続き Simple は以下の規則からなる．

1. $s = t$ のとき，$s \in IC$ または $s = w$ ならば，$E = E' \cup \{\langle s,t\rangle\} \Rightarrow E'$.
2. $hd(s) = hd(t) = @ \in FC \cup PC \cup \{\wedge, \vee, \supset, \neg\}$ のとき，$E = E' \cup \{\langle @(\overline{s_m}), @(\overline{t_m})\rangle\} \Rightarrow E' \cup \{\overline{\langle s_m, t_m\rangle}\}$.
3. $s = \natural x.\Phi$，$t = \natural y.\varphi$，($\natural \in \{\forall, \exists\}$) ならば，$E = E' \cup \{\langle s,t\rangle\} \Rightarrow E' \cup \{\langle \Phi\{x := w\}, \varphi\{y := w\}\rangle\}$．ただし，$w$ は束縛変数が出現していたことを表す特別な定数とする．

上の簡単化規則を，E に適用することができなくなるまで、繰り返し適用することにより，E の簡約化マッチング表現を得ることができる．

例 2 E を以下のマッチング表現とする．

$$E = \left\{ \begin{array}{l} \langle \forall x.(P(a_S, x, b_S) \wedge \exists y.Q(x, y, a_S), \\ \forall x.(p(b, a, x) \wedge \exists y.q(a, y, x, b)) \rangle \end{array} \right\}$$

このとき，簡単化規則を適用することにより，以下のようにして簡約化マッチング表現を得ることができる．

$$\begin{array}{rl} & \left\{ \begin{array}{l} \langle P(a_S, w_1, b_S) \wedge \exists y.Q(w_1, y, a_S), \\ p(b, a, w_1) \wedge \exists y.q(a, y, w_1, b)\rangle \end{array} \right\} \\ \Rightarrow & \left\{ \begin{array}{l} \langle P(a_S, w_1, b_S) \wedge \exists y.Q(w_1, y, a_S), \\ p(b, a, w_1) \wedge \exists y.q(a, y, w_1, b)\rangle \end{array} \right\} \\ \Rightarrow & \left\{ \begin{array}{l} \langle P(a_S, w_1, b_S), p(b, a, w_1)\rangle, \\ \langle \exists y.Q(w_1, y, a_S), \exists y.q(a, y, w_1, b)\rangle \end{array} \right\} \\ \Rightarrow & \left\{ \begin{array}{l} \langle P(a_S, w_1, b_S), p(b, a, w_1)\rangle, \\ \langle Q(w_1, w_2, a_S), q(a, w_2, w_1, b)\rangle \end{array} \right\} \\ & \left(= \left\{ \begin{array}{l} \langle s_1, t_1\rangle, \\ \langle s_2, t_2\rangle \end{array} \right\} = E' \right) \end{array}$$

マッチング表現 $E = \{\overline{\langle s_n, t_n\rangle}\}$ に対して，$s_i\theta = t_i$ ($1 \leq i \leq n$) となる代入 θ が存在するとき，E はマッチング可能といい，θ を E のマッチング代入という．

定理 1 マッチング表現 E がマッチング可能なとき，また，そのときに限り E の簡約化マッチング表現はマッチング可能となる．

定義 1 s をスキーマ $P(s_1, \cdots, s_m)$，t を閉じた論理式とする．集合 $\{i : \theta \mid s_i\theta = t,\ 1 \leq i \leq m\}$ を $\rho(t)$ と表す．このとき，s, t の判定項 $J(s,t)$ は以下のように帰納的に定義される．

1. $J(s,w) = *^{\rho(w)}$，$J(s,c) = c^{\rho(c)}$ ($c \in IC$)，$J(s,x) = x^{\emptyset}$ ($x \in IV$).
2. $t = f(\overline{t_m})$，$f \in FC$ のとき，$J(s,t_i)$ ($1 \leq i \leq m$) を，それぞれ s に対する t_i の判定項とする．このとき，$J(s,t) = f^{\rho(\theta)}(\overline{J(s,t_m)})$.
3. $t = p(\overline{t_m})$，$p \in PC \cup \{\wedge, \vee, \supset\}$ のとき，$J(s,t_i)$ ($1 \leq i \leq m$) を，それぞれ s に対する t_i の判定項とする．このとき，$J(s,t) = p^{\emptyset}(\overline{J(s,t_m)})$.
4. $t = \natural x.t_1$ ($\natural \in \{\forall, exists\}$) のとき，$t_1$ の判定項を $J(s,t_1)$ とする．このとき，$J(s,t) = \natural x.J(s,t_1)$.

ここで，$*$ は模倣規則が適用できないことを表す新しい記号である．

例 3 例 2 における，簡約化マッチング表現 E' に対する判定項は，以下のようになる．

$J(s_1,t_1) = p(b^{1:a_S := a, 3:b_S = b}, a^{1:a_S := a, 3:b_S = a}, *^2)$,
$J(s_2,t_2) = q(a^{3:a_S := a}, *^2, *^1, b^{3:a_S := b})$

次に，この判定項の共通部分を表す共通判定項を求める．

定義 2 簡約化マッチング表現 $E = \{\overline{\langle s_n, t_n\rangle}\}$ $hd(s_i) = P$ に対して，s_i に対する t_i の判定項を $J(s_i,t_i)$ ($1 \leq i \leq n$) とする．このとき，E に対する共通判定項 $J(E) =$ を以下のように帰納的に定義する．

1. $J(s_i,t_i) = c^{\rho_i}$，$c \in IC \cup IV \cup \{*\}$ ($i \in I$) ならば，$J(E) = c^{\rho_1 \times \cdots \times \rho_n}$．ただし $\rho_1 \times \rho_2 = \{i : \theta_{1i} \cup \theta_{2i} \mid i : \theta_{1i} \in \rho_1,\ i : \theta_{2i}\}$.
2. $J(s_i,t_i) = f^{\rho_i}(t^i_1, \cdots, t^i_m)$，$f \in FC$ ならば，$J(E) = f^{\rho_1 \times \cdots \times \rho_n}(J(\{\overline{J(s_n, t^n_1)}\}), \cdots, J(\{\overline{J(s_n, t^n_m)}\}))$
3. $J(s_i,t_i) = p^{\emptyset}(t^i_1, \cdots, t^i_m)$，$p \in PC \cup \{\wedge, \vee, \supset, \neg\}$ ならば，$J(E) = p^{\emptyset}(J(\{\overline{J(s_n, t^n_1)}\}), \cdots, J(\{\overline{J(s_n, t^n_m)}\}))$
4. $J(s_i,t_i) = \natural x_i.t'_i$，$\natural \in \{\forall, \exists\}$ ならば，$J(E) = \natural x^{\emptyset} J(\{\overline{J(s_n, t^n)}\})$
5. $hd(J(s_k,t_k)) \neq hd(J(s_j,t_j))$ となる $1 \leq j,k \leq n$ が存在するとき $J(E) = *^{\rho_1 \times \cdots \times \rho_n}$.

このようにして構成された共通判定項には，以下の性質が成り立つ．

定理 2 E を簡約化マッチング表現とする．このとき，$J(E)$ は $O(|E|^2)$ 時間で構成することができ，E がマッチング可能となるとき，またそのときに限り $J(E) \neq *^{\emptyset}$ となる．

4. マッチング代入の構成

前節で構成した共通判定項から，以下のようにしてマッチング代入を構成することができる．

定義 3 $E = \{\langle s_i, t_i\rangle \mid 1 \leq i \leq m\}$ を $hd(s_i) = P \in PV$ ($1 \leq i \leq m$) となる簡約化マッチング表現とする．このとき，E に対する代入の集合 $S_{J(E)}$ を，以下のように帰納的に定義する．

1. $J(E) = *^{\rho}$ ならば，$S_{J(E)} = \{\lambda\overline{v_n}.v_j, \theta \mid j : \theta \in \rho\}$.
2. $J(E) = c^{\rho}$ ならば，$S_{J(E)} = \{\lambda\overline{v_n}.c\} \cup \{\lambda\overline{v_n}.v_j \mid j : \theta_j \in \rho\} \cup \theta_j$.
3. $J(E) = f^{\rho}(J(E_1), \cdots, J(E_m))$ ならば，$S_{J(E)} = \{\lambda\overline{v_n}.f(t_1, \cdots, t_m) \mid \lambda\overline{v_n}.t_i \in S_{J(E_i)},\ 1 \leq i \leq m\} \cup \{\lambda\overline{v_n}.v_j \mid j : \theta_j \in \rho\} \cup \theta_j$.

例 4 例 2 において，$S_{J(\{\langle s_1,t_1\rangle\})}, S_{J(\{\langle s_2,t_2\rangle\})}$ は以下のようになる．

$$S_{J(\{\langle s_1,t_1\rangle\})} = \left\{ \begin{array}{l} \{P := \lambda\overline{x_3}.p(b,a,x_2)\}, \\ \{P := \lambda\overline{x_3}.p(x_1,a,x_2), a_S := b\}, \\ \{P := \lambda\overline{x_3}.p(x_3,a,x_2), b_S := b\}, \\ \{P := \lambda\overline{x_3}.p(b,x_1,x_2), a_S := a\}, \\ \{P := \lambda\overline{x_3}.p(x_3,x_1,x_2), \\ \quad a_S := a, b_S := b\}, \\ \{P := \lambda\overline{x_3}.p(b,x_3,x_2), b_S := a\}, \\ \{P := \lambda\overline{x_3}.p(x_1,x_3,x_2), \\ \quad a_S := b, b_S := a\} \end{array} \right\}$$

$$S_{J(\{\langle s_2,t_2\rangle\})} = \left\{ \begin{array}{l} \{Q := \lambda\overline{x_4}.q(a,x_2,x_1,b)\}, \\ \{Q := \lambda\overline{x_4}.q(x_3,x_2,x_1,b), a_S := a\}, \\ \{Q := \lambda\overline{x_4}.q(a,x_2,x_1,x_3), a_S := b\} \end{array} \right\}$$

一般のマッチング表現 E のマッチング代入を求めるには，まず，E の簡約化マッチング表現 E' を求め，次に，E' を対に含まれる述語変数毎に分解する．すなわち，$E' = \bigcup E_{P_i}$，$E_{P_i} = \{\langle s,t\rangle \mid hd(s) = P_i,\ \langle s,t\rangle \in E\}$．さらに，代入の集合 $S_E = \{\bigcup \theta_i \mid \theta_i \in \theta_i \in S_{J(E_{P_i})}$ただし，$(v_1 := t_1), (v_2 := t_2) \in \bigcup \theta_i$かつ $v_1 = v_2$ならば $t_1 = t_2\}$

定理 3 E をマッチング表現, E に対する代入の集合を S_E とする．このとき，$\theta \in S_E$ は E のマッチング代入となる．

例 5 例 2 におけるマッチング表現 E に対するマッチング代入の集合 S_E は，以下の 19 個の代入よりなる．

$\{P := \lambda\overline{x_3}.p(b,a,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,b)\}$
$\{P := \lambda\overline{x_3}.p(b,a,x_2),\ Q := \lambda\overline{x_4}.q(x_3,x_2,x_1,b), a_S := a\}$
$\{P := \lambda\overline{x_3}.p(b,a,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,x_3), a_S := b\}$
$\{P := \lambda\overline{x_3}.p(x_1,a,x_2), a_S := b,\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,b)\}$
$\{P := \lambda\overline{x_3}.p(x_1,a,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,x_3), a_S := b\}$
$\{P := \lambda\overline{x_3}.p(x_3,a,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,b), b_S := b\}$
$\{P := \lambda\overline{x_3}.p(x_3,a,x_2),\ Q := \lambda\overline{x_4}.q(x_3,x_2,x_1,b), a_S := a, b_S := b\}$
$\{P := \lambda\overline{x_3}.p(x_3,a,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,x_3), a_S := b, b_S := b\}$
$\{P := \lambda\overline{x_3}.p(b,x_1,x_2)\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,b), a_S := a\}$
$\{P := \lambda\overline{x_3}.p(b,x_1,x_2),\ Q := \lambda\overline{x_4}.q(x_3,x_2,x_1,b), a_S := a\}$
$\{P := \lambda\overline{x_3}.p(x_3,x_1,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,b), a_S := a, b_S := b\}$
$\{P := \lambda\overline{x_3}.p(x_3,x_1,x_2),\ Q := \lambda\overline{x_4}.q(x_3,x_2,x_1,b), a_S := a, b_S := b\}$
$\{P := \lambda\overline{x_3}.p(b,x_3,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,b), b_S := a\}$
$\{P := \lambda\overline{x_3}.p(b,x_3,x_2),\ Q := \lambda\overline{x_4}.q(x_3,x_2,x_1,b), a_S := a, b_S := a\}$
$\{P := \lambda\overline{x_3}.p(b,x_3,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,x_3), a_S := b, b_S := a\}$
$\{P := \lambda\overline{x_3}.p(x_1,x_3,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,b), a_S := b, b_S := a\}$
$\{P := \lambda\overline{x_3}.p(x_1,x_3,x_2),\ Q := \lambda\overline{x_4}.q(a,x_2,x_1,x_3), a_S := b, b_S := a\}$

5. まとめ

本論文では，スキーマ誘導型自動証明への応用を目的とした 2 階スキーママッチングアルゴリズムを，スキーマ定数の名前換えを行なうことができるよう拡張した．なお，提案したマッチングアルゴリズムは，研究室で開発しているスキーマ誘導型自動証明器に組み込まれている．

参考文献

[1] L. D. Baxter: The complexity of unification, Doctoral Thesis, Department of computer Science, University of Waterloo, 1977.

[2] B.Brock, S.Cooper and W.Pierce: Analogical reasoning and proof and proof discovery, LNCS, No.310, pp. 454–468, 1988.

[3] P. Flener: Logic program synthesis form incomplete information, (Kluwer Academic Press, 1995)

[4] M. Harao: Proof discovery in LK system by analogy, in: Proc. 3rd Asian Computing Science Conf., LNCS 1345 (Springer, 1997) pp.197–211.

[5] M. Harao, K. Yamada, K. Hirata: Efficient second order predicate matching algorithm, in: Proc. 5th annual Inter. Computing and Combinatorics Conf., LNCS 1627 (Springer, 1999) pp.432–442.

[6] K. Hirata, K. Yamada, M. Harao: Tractable and intractable second-order matching algorithm, in: Proc. 5th Annual Intern. Computing and Combinatorics Conf., LNCS 1627 (Springer, 1999), pp.432–442.

[7] G. P. Huet: A unification algorithm for typed λ-calculus, Theoretical Computer Science 1, pp.27–57, 1975.

[8] G. P. Huet and B. Lang: Proving and applying program transformations expressed with second-order patterns, Acta Inform. 11, pp.31–55, 1978.

[9] P. Kilpelainen: Tree matching problems with applications to structured text databases, University of Helsinki Department of Computer Science, Series of Publications, No.A-1992-6, 1992.

[10] D. A. Miller: A logic programming language with lambda-abstraction, function variables and simple unification, Journal of Logic and Computation, 1(4), pp.494-536, 1991.

[11] 山田，平田，原尾: スキーママッチングとその計算量，信学技報 J82-D-I, pp.1307–1316, 1999.