

モデル生成型定理証明器を用いたアブダクションの計算における効率化手法

A Pruning Technique for Abduction on Model Generation Theorem Provers

太田 好彦*¹ 井上 克己*² 長谷川 隆三*³ 中島 誠*⁴
 Yoshihiko Ohta Katsumi Inoue Ryuzo Hasegawa Makoto Nakashima

- * 1 職業能力開発大学校情報工学科
 Dept. of Information Engineering, Polytechnic University, Kanagawa 229, Japan.
- * 2 豊橋技術科学大学情報工学系
 Dept. of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi 441, Japan.
- * 3 (財)新世代コンピュータ技術開発機構
 Institute for New Generation Computer Technology, Tokyo 108, Japan.
- * 4 大分大学工学部知能情報システム工学科
 Dept. of Computer Science and Intelligent Systems, Faculty of Engineering, Oita University, Oita 870-11, Japan.

1992年12月17日 受理

Keywords: abduction, model generation.

Summary

It has been presented that we can translate an abduction framework into a model generation problem. The translation method is called "Skip". The MGTP prover can generate the minimal models of a set of first-order clauses. A large amount of OR parallelism appears in the translated program. However, the number of model candidates generated by the MGTP prover is often exponential-order. Therefore, we need a pruning technique for the translated program. Since we usually require only the minimal explanations of a query from an abduction framework, non-minimal explanations are redundant. In parallel processing, it is difficult for the MGTP prover to find that model candidates are redundant at intermediate steps.

This paper describes a pruning technique for the MGTP programs translated by the Skip method. Here, it is proved that we may omit model candidates containing multiple instances of an abducible predicate if no predicates in the given theory are multiple-dependent on the abducible predicate. This condition can be checked with information from dependency analysis among predicates. We can also use the MGTP prover as the dependency analyzer. If we make program containing negative clauses (called cut rules) for abducible predicates which satisfy the condition, then these cut rules automatically prune redundant model candidates on the MGTP prover. We illustrate that the cut rules dramatically reduce the number of model candidates generated by the MGTP prover with the translated program. This pruning technique involves no overheads in parallel processing and allows us good performance on the parallel inference machine PIM/m.

1. ま え が き

アブダクションは、与えられた理論から観測の説明を生成する枠組みである。これは、仮説推論[國藤 87]と呼ばれることもあり、診断などの問題を非常にうまく定式化できる[Poole 87]とともに、不完全な知識[石塚 88]に基づく問題解決に利用できる[井上 92]。しかしながら、このような説明の生成には仮説の組合

せの爆発が生じる可能性があるとともに、導入した仮説の無矛盾性の検査のための計算コストが高いという問題がある。このようなアブダクションの計算時間を短縮するために、並列計算機を利用することは有効であると考えられる。

筆者らは、疎結合型並列計算機 PIM/m 上で OR 並列性を容易に抽出できるモデル生成法[Manthey 88]に基づく一階述語論理の定理証明器 MGTP[Fujita 91]を利用した、アブダクションの種々の計算方式に関

する研究を行っている。この結果、[井上 93]によれば、仮説生成部と無矛盾性検査部とを別々のモジュールとして構成し、それらを結合した方式（例えば、MGTP に ATMS[de Kleer 86]などを結合したシステム）では、二つのコンポーネント間の通信が動的に必要であり、大きな台数効果を得ることが困難である。そこで、生成された仮説集合に対する無矛盾性検査部から新たな仮説生成を行う部分を分離しない方式（Skip 方式と呼ぶ）が提案されている[井上 93]。この方法によれば、MGTP において、仮説を導入するかしないかの場合分けによって生じる OR 並列性をうまく利用することができ、より大きな台数効果が得られる[井上 93]。

この Skip 方式では、並列に動作しているプロセス間にほとんど通信はなく、並列処理に有利であるという特長を持つ。しかしながら、冗長な仮説集合を生成する可能性があり、仮説の数が多い場合に並列計算可能なプロセスの数が爆発するという問題点がある。ここに、冗長な仮説集合とは、観測を説明する極小の仮説集合を包含するものであり、これは、むだな場合分けを行うことによって生じる。

本稿では、Skip 方式によるアブダクションの計算で、このような問題点を解決するために、与えられたアブダクションの枠組みを静的に解析し、冗長であるモデル候補を刈るためのルール（枝刈りルール：*cut rule*）を生成して、与えられたアブダクションの枠組みに追加するという方法を示す。これによれば、Skip 方式の並列計算の効果を損なうことなく、冗長な仮説集合の生成を抑制することができるようになる。

本稿の構成は次のようである。2章で、モデル生成型定理証明に関するいくつかの定義と MGTP の動作の概略について示す。3章で、本稿で扱うアブダクションの問題を定義する。4章で、MGTP 上でアブダクションの問題解決を行う Skip 方式の概略と、その問題点を示す。5章で、それによるアブダクションの計算を効率化する一手法を提案し、その効果を例によって示す。さらに、6章で並列推論マシン PIM/m 上の実験結果を示し、本手法を評価する。

2. モデル生成型定理証明器 MGTP

2.1 定義

α_i ($1 \leq i \leq n$; $n \geq 0$) と β_{j,l_j} ($1 \leq j \leq m$; $m \geq 0$; $l_j \geq 1$) とを一階述語論理のアトム、“ \rightarrow ”を含意、“ \wedge ”を連言、“ \vee ”を選言とすると、

$$\alpha_1 \wedge \cdots \wedge \alpha_n \rightarrow (\beta_{1,1} \wedge \cdots \wedge \beta_{1,l_1}) \vee \cdots \vee (\beta_{m,1} \wedge \cdots \wedge \beta_{m,l_m}) \quad (1)$$

を MGTP の節（以下、単に節）といい、MGTP の節の集合を MGTP プログラム（以下、単にプログラム）という。また、節の含意記号の左辺を前件、右辺を後件という。プログラムの節は、後件に出現する変数がその前件に出現していなければならない。この条件を値域限定 (range-restrictedness) と呼ぶ。なお、 $m=1$ の節を確定節、 $m=0$ の節を負節、 $m \geq 2$ の節を非ホーン節という。また、 $n=0$ の節は、正節とも呼ばれ、含意記号を省略して記述することもある。

与えられたプログラム P のエルブラン基底の任意の部分集合を解釈という。解釈 I 、基礎代入 θ に対して、節(1)が以下の条件を満たすとき、この節は I において θ で満たされないという。

条件： $\alpha_1 \theta, \dots, \alpha_n \theta \in I$ であり、かつ、いかなる j ($1 \leq j \leq m$) についても $\beta_{j,1} \theta, \dots, \beta_{j,l_j} \theta \in I$ でない。

I で満たされない節が P に存在しないとき、 P は充足可能であるといい、 I を P のモデルという。また、 P のモデルが存在しないとき P は充足不能であるという。

2.2 動作

MGTP は、与えられたプログラム P の充足可能性を判定するために、 P のモデルを生成することを試みる。モデルを生成する方法として、ある解釈 I を設定して、 I で満たされない節が P に存在しないかどうかを検査する。以下に示す手続き、すなわち MGTP によって生成される P の解釈 I はモデル候補と呼ばれる。

まず、最初のモデル候補 I を空集合と置き、以下を実行する。

モデル候補 I において、 P の節(1)が基礎代入 θ で満たされないとき、 $\beta_{j,1}, \dots, \beta_{j,l_j}$ ($1 \leq j \leq m$; $m \geq 1$) に対して、モデル候補 I を I_1, \dots, I_m に分岐する。ここに、 I_j は $I \cup \{\beta_{j,1} \theta, \dots, \beta_{j,l_j} \theta\}$ である。このようにモデル候補の要素を増加させることをモデル候補拡張と呼ぶ。拡張された各モデル候補 I_j を I とし、同様にこれを繰り返す。もし、 I で満たされない節が P に存在しなければ、 I は P のモデルであり、このモデル候補の拡張は終了する。また、 I で満たされない負節が P に存在する場合、このモデル候補 I は、モデル候補拡張によってモデルとなり得ないので棄却される。

3. アブダクションの定義

本稿で扱うアブダクションの定義を示す。

【定義1】 確定節の集合 T 、負節の集合 N 、および仮定可能な述語（仮定可能述語：abducible predicate）の集合 A の組 $\langle T, N, A \rangle$ をアブダクションの枠組み

(abduction framework) [Eshghi 89]という。また、仮定可能述語からなるアトムを仮説と呼ぶ。ここで、 T の節の後件に仮定可能述語が出現しないものとする。

【定義2】 $\langle T, N, A \rangle$ をアブダクションの枠組み、 Q をその述語が仮定可能述語でないアトム (質問) とする。このとき、次の条件を満たす仮説の基礎例の集合 E が存在するとき、 Q は説明可能であるといい、 E を $\langle T, N, A \rangle$ に関する Q の説明という。

- $T \cup E \models Q$.
- $T \cup E \cup N$ は充足可能。

【定義3】 アブダクションの枠組み $\langle T, N, A \rangle$ に関する質問 Q の任意の説明を E として、いかなる E の真部分集合も Q の説明でないときに、その説明 E は $\langle T, N, A \rangle$ に関する Q の極小説明であるという。

【定義4】 アブダクションの枠組み $\langle T, N, A \rangle$ に関する質問 Q のすべての極小説明の集合を $\langle T, N, A \rangle$ に関する Q のアブダクションの解という。

4. MGTP によるアブダクションの解法

4.1 基本原理

MGTP によって、アブダクションの問題を解くために、アブダクションの枠組み $\langle T, N, A \rangle$ を以下のような MGTP のプログラムに変換する。この変換方法は Skip 方式 [井上 93] に基づいている。

いま、 α_i ($1 \leq i \leq n$; $n \geq 0$) および β を仮定可能述語でない述語からなるアトム、 γ_j ($1 \leq j \leq m$; $m \geq 0$) を仮定可能述語からなるアトムとし、 T の確定節

$$\alpha_1 \wedge \dots \wedge \alpha_n \wedge \gamma_1 \wedge \dots \wedge \gamma_m \rightarrow \beta \quad (2)$$

は、

$$\alpha_1 \wedge \dots \wedge \alpha_n \rightarrow (\gamma_1 \wedge \dots \wedge \gamma_m \wedge \beta) \vee \neg K \gamma_1 \vee \dots \vee \neg K \gamma_m \quad (3)$$

に変換される ($m=0$ の場合はもとの節のままである)。ここに、 $\neg K X$ は K リテラルと呼ばれ、仮説 X がモデルに含まれないと仮定することを表している。MGTP は、このような K リテラルを単に新しいアトムとして扱う。また、変換された節が MGTP の値域限定の条件を満たさなければならないので、 β あるいは各 γ_j に出現する変数は $\alpha_1 \wedge \dots \wedge \alpha_n$ に出現しなければならないとする*1。

このように、 T の各確定節を変換した節の集合を

*1 この条件を満たさない場合には、言語を構成する定数をもとに対象領域を生成する *dom* 述語 [Manthey 88] を前件に付加することにより値域限定の条件を満たすようにできる。

*2 [井上 93] に示されているように、 N の要素に対しても T の要素と同様に変換してもよい。

$tr(T, A)$ と記述する。すると、与えられたアブダクションの枠組み $\langle T, N, A \rangle$ に対応する変換プログラムは、

$P(T, N, A) \equiv tr(T, A) \cup N \cup \{\neg K X \wedge X \rightarrow\}$ である*2。ここに、 $(\neg K X \wedge X \rightarrow)$ は、 $\neg K X$ および X を含んでいるモデル候補を棄却するためのスキーマである。

MGTP によって、アブダクションの枠組み $\langle T, N, A \rangle$ に関する与えられた質問 Q の解を得る手順は、次のようである。

1. MGTP によって得られる $P(T, N, A)$ のモデルの集合を計算し \mathcal{M} とする。
2. $M_Q \equiv \{M \mid M \models Q, M \in \mathcal{M}\}$.
3. $\mathcal{E} \equiv \{E \mid E \text{ は } M \text{ に含まれる仮定可能述語からなる基礎アトムの集合, } M \in M_Q\}$.
4. \mathcal{E} の要素のうち極小な要素の集合が、アブダクションの枠組み $\langle T, N, A \rangle$ に関する質問 Q の解である。

4.2 問題点

前節で示したアブダクションの枠組みの変換方式によって得られるプログラムは、次の例に示すように、モデル候補の爆発を引き起こす可能性がある。

【例題1】

$$T = \{\neg q(c_1), \neg q(c_2), \dots, \neg q(c_n), \\ p(X) \wedge q(X) \rightarrow r(X), r(X) \rightarrow g(X)\},$$

$N = \phi, A = \{p/1\}$ とする。このとき、アブダクションの枠組み $\langle T, N, A \rangle$ に関する質問 $Q = g(X)$ の解を求めよ。この変換プログラム

$$P(T, N, A) = \{\neg q(c_1), \neg q(c_2), \dots, \neg q(c_n), \\ q(X) \rightarrow (p(X) \wedge r(X)) \\ \vee \neg K p(X), r(X) \rightarrow g(X)\}$$

の MGTP 上の動作の概略は次のようである。モデル候補の初期値 I_0 を ϕ とする。この I_0 で、 n 個の正節は満たされない節であるので、新しいモデル候補 I_1 を、 $\{q(c_1), q(c_2), \dots, q(c_n)\}$ とする。次に、このモデル候補 I_1 で基礎代入 $\{X := c_1\}$ に対して $q(X) \rightarrow (p(X) \wedge r(X)) \vee \neg K p(X)$ なる非ホーン節は満たされない。ここで、次の二つのモデル候補に分岐する。

- $I_{1,1} = I_1 \cup \{p(c_1), r(c_1)\}$
- $I_{1,2} = I_1 \cup \{\neg K p(c_1)\}$

この各モデル候補で基礎代入 $\{X := c_2\}$ に対して、その非ホーン節は満たされない。ここで、 $I_{1,1} \cup \{p(c_2), r(c_2)\}, I_{1,1} \cup \{\neg K p(c_2)\}, I_{1,2} \cup \{p(c_2), r(c_2)\}$ および $I_{1,2} \cup \{\neg K p(c_2)\}$ の四つのモデル候補に分岐する。このようなモデル候補分岐がおのおのについて n 段行われ、モデル候補の数は 2^n となる。以下、モデル候補分岐は

生じないが、各モデル候補で、もし $r(X)$ の基礎例が含まれていれば、対応する $g(X)$ の基礎例を含むようにモデル候補が拡張される。最終的には、満たされない節がなくなり、結局、 2^n 個のモデルが生成される。しかしながら、この問題の解は $\{p(c_1)\}, \dots, \{p(c_n)\}$ であり、極小説明の個数は n である。以下では、このような不要な爆発を抑止する手法について述べる。

5. 枝刈りルールによる効率化

前記問題点を解決するために、与えられたアブダクションの枠組みを静的に解析し、冗長であるモデル候補を刈るためのルールを生成して、与えられたアブダクションの枠組みに追加しておく方法を示す。

5.1 原理

与えられた確定節の集合 T の任意の述語と仮定可能述語との依存関係を以下のように定義する。

【定義 5】 与えられた確定節の集合 T 中の任意の確定節は、後件の述語の定義節という。

【定義 6】 仮定可能述語は、その述語自身にのみ依存する。

【定義 7】 ある述語は、与えられた確定節の集合 T のその定義節の前件の述語が依存する仮定可能述語に依存する。

【定義 8】 ある仮定可能述語に依存する述語が、ある確定節の前件に複数出現する場合、その確定節の後件の述語は、その仮定可能述語に重複して依存する (*multiple-dependent*) という。

例題 1 において、述語の依存関係は次のようである。

- $p/1$ は、 $p/1$ に依存する。
- $q/1$ は、 $p/1$ に依存しない。
- $r/1$ は、 $p/1$ に依存するが、重複して依存しない。
- $g/1$ は、 $p/1$ に依存するが、重複して依存しない。

よって、 T の任意の述語は、仮定可能述語 $p/1$ に重複して依存しない。

【補題】 与えられた確定節の集合 T 中の任意の述語が、ある仮定可能述語 γ に重複して依存しなければ、

$$TU\{\gamma(t_1), \gamma(t_2)\}$$

の最小モデルは、 $TU\{\gamma(t_1)\}$ の最小モデルと $TU\{\gamma(t_2)\}$ の最小モデルとの和集合に等しい。ここに、 t_1 および t_2 は、任意の基礎項の組とする。

【証明】 ここで、 $TU\{\gamma(t_1), \gamma(t_2)\}$ の最小モデル M が、 $TU\{\gamma(t_1)\}$ の最小モデル M_1 と $TU\{\gamma(t_2)\}$ の最小モデル M_2 との和集合に等しくないと仮定する。 T は

確定節の集合であるから、 $M \supset (M_1 \cup M_2)$ である。よって、次のような節、およびアトム a_i, a_j が存在する。

- $(\alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \beta) \in T$ であり、
- $\alpha_i \theta \in M_1, \alpha_i \theta \notin M_2$ ($1 \leq i \leq n$) かつ
- $\alpha_j \theta \notin M_1, \alpha_j \theta \in M_2$ ($1 \leq j \leq n$) かつ
- $\beta \theta \in M, \beta \theta \notin M_1, \beta \theta \notin M_2$

となる (θ は基礎代入)。ここで、 $TU\{\gamma(t_1)\} \vdash \alpha_i \theta, TU\{\gamma(t_2)\} \vdash \alpha_j \theta$ である α_i, α_j は T から直接導けず、 α_i および α_j に対応する述語はそれぞれ仮定可能述語 γ に依存していることになる。よって、与えられた確定節の集合 T 中の β の述語が仮定可能述語 γ に重複して依存していることになり、前提に反するので補題が証明された。 ■

〈定理 1〉 $\langle T, N, A \rangle$ を与えられたアブダクションの枠組みとする。もし、 T 中の任意の述語がある仮定可能述語 γ に重複して依存しなければ、負節

$$C = (\gamma(X) \wedge \gamma(Y) \wedge X \neq Y \rightarrow)$$

として、 $\langle T, N, A \rangle$ に関する質問 Q の解は、 $\langle T, NU\{C\}, A \rangle$ に関するその質問 Q の解と等しい。なお、このような負節 C を γ に関する枝刈りルール (*cut rule*) と呼ぶ。

【証明】 $\langle T, N, A \rangle$ に関する Q の解が $\langle T, NU\{C\}, A \rangle$ に関する Q の解と等しくないと仮定する。すると、 $TUNU\{C\} \cup E$ が充足不能なく $\langle T, N, A \rangle$ に関する Q の極小説明 E が存在する。 $TUNU\{C\} \cup E$ は充足可能であり、さらに N が負節の集合であることから

$$TUE \models \gamma(X) \wedge \gamma(X) \wedge X \neq Y.$$

このとき、ある基礎例 $\gamma(t_1)$ および $\gamma(t_2)$ ($t_1 \neq t_2$) が存在して、 $TUE \models \gamma(t_1) \wedge \gamma(t_2)$ 。

さらに、仮定可能述語 γ が T のいかなる節の後件にも出現しないことから、

$$\gamma(t_1) \in E \quad \text{かつ} \quad \gamma(t_2) \in E$$

である。ここで、

$$E' \equiv E \setminus \{\gamma(t_1), \gamma(t_2)\}$$

と置く。

一方、 T 中の任意の述語が γ に重複して依存しないことと補題により、 TUE の最小モデルは、 $TUE' \cup \{\gamma(t_1)\}$ の最小モデルと $TUE' \cup \{\gamma(t_2)\}$ の最小モデルとの和集合に等しい。また、 E が Q の $\langle T, N, A \rangle$ に関する極小説明であることから、 TUE の最小モデルは、 Q のある基礎例を含んでいる。よって、 Q のその基礎例は、 $TUE' \cup \{\gamma(t_1)\}$ の最小モデルに属するかまたは、 $TUE' \cup \{\gamma(t_2)\}$ の最小モデルに属する。これは、 E が Q の極小説明であることに反するので、定理が証明された。 ■

上の定理に基づいて、枝刈りルール C を導入するこ

とにより、解とならないような説明を含む不要なモデル候補を棄却することができるようになる。

5・2 検出方法

与えられたアブダクションの枠組み $\langle T, N, A \rangle$ のうち、 T および A を下記のような MGTP の節に変換して、定理 1 の条件部を満たす仮定可能述語を検出できる。

以下で、 $d(\hat{\beta}, \hat{E})$ をアトム β の述語 $\hat{\beta}$ が \hat{E} の要素に依存する (dependent) と読むこととする。ここに、 \hat{E} は、 A の部分集合とする。 A の要素、すなわち各仮定可能述語 $\hat{\beta}$ は、定義 6 により、

$$\rightarrow d(\hat{\beta}, \{\hat{\beta}\}) \quad (4)$$

と記述できる。また、 T の確定節 $\alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \beta$ は、定義 7 により、

$$d(\hat{\alpha}_1, \hat{E}_1) \wedge \dots \wedge d(\hat{\alpha}_n, \hat{E}_n) \rightarrow d(\hat{\beta}, \bigcup_{i=1}^n \hat{E}_i) \quad (5)$$

と記述できる。また、 $m(\hat{\beta}, \hat{E})$ を述語 $\hat{\beta}$ が仮定可能述語の集合 \hat{E} の要素に重複して依存する (multiple-dependent) と読むことにすると、 T の各確定節 $\alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \beta$ ($n \geq 2$) は、定義 8 により、 $(i \neq j) \wedge (1 \leq i \leq n) \wedge (1 \leq j \leq n)$ なる任意の i, j に関して、

$$d(\hat{\alpha}_i, \hat{E}_i) \wedge d(\hat{\alpha}_j, \hat{E}_j) \rightarrow m(\hat{\beta}, \hat{E}_i \cap \hat{E}_j) \quad (6)$$

と記述できる。そこで、 A の各要素に対する変換節(4)と T の各要素に対する変換節(5)、および変換節(6)の集合をプログラムとして MGTP によってそのモデルを生成する。この場合、プログラムが確定節の集合なので、最小モデル M が得られる。ここで、仮定可能述語 γ が任意の述語 X に対して $m(X, \hat{E}) \in M$ かつ $\gamma \notin \hat{E}$ なる条件を満たせば、任意の述語は仮定可能述語 γ に重複して依存しない。

なお、前の前件のアトム数が n ならば、変換された節(6)の数は、 $n \cdot (n-1)/2$ である。そこで、 $n \geq 3$ なる確定節 $\alpha_1 \wedge \dots \wedge \alpha_n \rightarrow \beta$ に対して、(5)および(6)の代わりに、次の $n-1$ 個の節を用いてもよい。

$$\bullet d(\hat{\alpha}_1, \hat{E}_1) \wedge d(\hat{\alpha}_2, \hat{E}_2) \rightarrow c_{k,2}(\hat{E}_1 \cup \hat{E}_2) \wedge m(\hat{\beta}, \hat{E}_1 \cap \hat{E}_2),$$

$$\bullet 2 \leq i \leq n-2 \text{ に対して,}$$

$$c_{k,i}(\hat{E}_i) \wedge d(\hat{\alpha}_{i+1}, \hat{E}_{i+1}) \rightarrow c_{k,i+1}(\hat{E}_i \cup \hat{E}_{i+1}) \wedge m(\hat{\beta}, \hat{E}_i \cap \hat{E}_{i+1}),$$

$$\bullet c_{k,n-1}(\hat{E}_{n-1}) \wedge d(\hat{\alpha}_n, \hat{E}_n) \rightarrow d(\hat{\beta}, \hat{E}_{n-1} \cup \hat{E}_n) \wedge m(\hat{\beta}, \hat{E}_{n-1} \cap \hat{E}_n).$$

ここに、 k は元の節を他の元の節と区別するための識別子である。 $c_{k,i}(\hat{E}_i)$ における \hat{E}_i は、 $\hat{\alpha}_1, \dots, \hat{\alpha}_i$ のいずれかが依存する仮定可能述語の集合である。したがって、 $\hat{E}_{n-1} \cup \hat{E}_n$ は、 $\hat{\beta}$ が依存する仮定可能述語の集合である。また、 \hat{E}_i を $\hat{\alpha}_1, \dots, \hat{\alpha}_i$ のいずれかが依存する仮定可能述語の集合、 \hat{E}_{i+1} を $\hat{\alpha}_{i+1}$ が依存する仮定可

能述語の集合とすれば、 $\hat{\alpha}_{i+1}$ が $\gamma \in \hat{E}_i \cap \hat{E}_{i+1}$ に依存し、かつ $\hat{\alpha}_1, \dots, \hat{\alpha}_i$ のいずれかがその γ に依存する。よって、 $\hat{\beta}$ は、 $\gamma \in \hat{E}_i \cap \hat{E}_{i+1}$ ($2 \leq i \leq n-1$) に重複して依存する。各 i ($2 \leq i \leq n-1$) に対する $m(\hat{\beta}, \hat{E}_i \cap \hat{E}_{i+1})$ は、このことを表現している。

5・3 適用例

例題 1 について、仮定可能述語 $p/1$ に重複依存する述語は T に存在しないから、仮定可能述語 $p/1$ に関する枝刈りルール ($p(X) \wedge p(Y) \wedge X \neq Y \rightarrow$) を導入することができる。

この枝刈りルールを加えた場合の変換プログラムの MGTP の動作の概略は次のようである。初期モデル候補 ϕ から、次の四つのモデル候補に分岐するまでは、枝刈りルールを導入しない場合と同様である。

- $I_{2,1} = I_1 \cup \{p(c_1), r(c_1)\} \cup \{p(c_2), r(c_2)\}$
- $I_{2,2} = I_1 \cup \{p(c_1), r(c_1)\} \cup \{\neg Kp(c_2)\}$
- $I_{2,3} = I_1 \cup \{\neg Kp(c_1)\} \cup \{p(c_2), r(c_2)\}$
- $I_{2,4} = I_1 \cup \{\neg Kp(c_1)\} \cup \{\neg Kp(c_2)\}$

ここに、 $I_1 = \{q(c_1), \dots, q(c_n)\}$ である。MGTP は、満たされない節の検出に関して、負節を他の節よりも優先的に処理するように実装されているから、 $p(c_1)$ と $p(c_2)$ とを含むモデル候補 $I_{2,1}$ は枝刈りルールにより即座に棄却される。残りのモデル候補は、たかだか 1 個の基礎仮説を含んでいる。同様にして、以降 n 段まで実行されるが、モデル候補の数は $(n+1)$ 個で抑えられる。すなわち、この例題では、MGTP によって生成されるモデル候補の数を枝刈りルールを導入することにより、基礎仮説の数の指数オーダから多項式オーダに減少させることができる。

5・4 枝刈りルールの変形

この枝刈りルールの適用範囲を広げるために、定理 1 の変形について示す。

T の任意の述語がある複数の仮定可能述語 $\gamma_1, \dots, \gamma_m$ に重複して依存しなければ、定理 1 を繰り返し適用できる。すなわち、1 から m の j について枝刈りルール

$$C_j = (\gamma_j(X) \wedge \gamma_j(Y) \wedge X \neq Y \rightarrow)$$

を導入し、次のアブダクションの枠組みに対して問題を解けばよい。

$$\langle T, N \cup \bigcup_{j=1}^m \{C_j\}, A \rangle.$$

また、 T 中の任意の述語が相異なる仮定可能述語 γ_1 および γ_2 の両者に依存しないことがわかれば、負節 $C_{1,2} = (\gamma_1(X) \wedge \gamma_2(Y) \rightarrow)$ として、アブダクションの枠組み $\langle T, N \cup \{C_{1,2}\}, A \rangle$ に対して問題を解けばよい。この例を次にあげる。

[例題 2]

$T = \{ \rightarrow \text{人(太郎)},$
 $\text{人}(X) \wedge \text{カゼ}(X) \rightarrow \text{くしゃみ}(X),$
 $\text{人}(X) \wedge \text{花粉症}(X) \rightarrow \text{くしゃみ}(X) \},$

$N = \phi, A = \{ \text{カゼ}/1, \text{花粉症}/1 \}$ とし、アブダクションの枠組み $\langle T, N, A \rangle$ に関する質問 $Q = \text{くしゃみ(太郎)}$ の解を求めよ。

T 中に “カゼ/1” かつ “花粉症/1” の両方に依存する述語はない。よって、枝刈りルール

$\text{カゼ}(X) \wedge \text{花粉症}(Y) \rightarrow$

が適用できて、 $\{ \text{カゼ(太郎)}, \text{花粉症(太郎)} \}$ に対応するモデル候補の生成を防ぐことができる。

6. 実験結果

次のような例題を用いて、要素プロセッサの数が 64 台の並列推論マシン PIM/m 上の MGTP による実行時間を計測した。この実験結果を用いて、本手法を適用した場合としない場合との比較を行う。

[例題 3][Maruyama 88] 二つの整数の最大公約数を計算する機能を有し、チップ面積と性能の制約条件を満たす回路を設計せよ。ここに、チップ面積の制約条件として基本セル数の上限値を与え、性能の制約条件として遅延時間の上限値を与える。

この問題を定義 1 の枠組みで定式化した。確定節の集合 T の要素数は 116 である。次は、その確定節の一例である。

$\text{control_box}(X1) \wedge \text{inverter}(X2) \wedge \text{register}(X3) \wedge$
 $\text{register}(X4) \wedge \text{multiplexer}(X5) \wedge$
 $\text{multiplexer}(X6) \wedge \text{comparator}(X7) \wedge$
 $\text{subtractor_with_MUX}(X8) \wedge$
 $\text{satisfy_constraints_for_GCD}(\{X1, \dots, X8\})$
 \rightarrow
 $\text{calculator_of_GCD}(\{X1, \dots, X8\}).$

これは、 $X1$ から $X8$ までを組み合わせた回路が回路の制約を満たすならば、その回路が最大公約数を計算する回路であることを示している。各部分回路は、基本セル数が決まっている CMOS 標準セルによって、マッピングされる。このマッピングのための知識もまた T の確定節として表現されている。また、負節の集合 N の要素数は 5 である。次は、その一例である。

$\text{calculator_of_GCD}(X) \wedge \text{delay_time}(X, T) \wedge$
 $\text{time_limit}(L) \wedge T > L \rightarrow.$

これは、最大公約数を計算する回路 X の遅延時間 T が、遅延時間の上限値 L よりも大きくはないことを示している。また、仮定可能述語の集合 A の要素数は 4

表 1 要素プロセッサ数 64 の PIM/m 上の MGTP での実行時間

基本セル数	遅延時間	枝刈りルールなし		枝刈りルールあり	
		実行時間 [s]	モデル候補数	実行時間 [s]	モデル候補数
300	40	83	5536	7	1087
400	50	1800 以上	$2^{44} + 63$ 以上	23	3239
500	60	1800 以上	$2^{54} + 63$ 以上	26	3955

であり、 $\text{satisfy_constraints_for_GCD}/1$ という述語は、仮定可能述語の一例である。

このようなアブダクションの枠組みに、Skip 方式による変換を施したプログラムを生成した。また、擬似並列マルチ PSI システム上の MGTP により 1.4 秒の計算時間を費やし、重複依存しない仮定可能述語をすべて検出した。その後、枝刈りルールを導入したアブダクションの枠組みに Skip 方式による変換を施したプログラムを生成した。このような枝刈りルールを、導入しない場合と導入した場合のそれぞれのプログラムの実行時間を表 1 に示す。実行時間は、要素プロセッサ数 64 の PIM/m 上の MGTP によるものである。この実験結果によると、制約が緩い場合(表 1, 下 2 段)、枝刈りルールを導入しないと 30 分以内に解を得ることができなかった。これは、MGTP によって生成されるモデルの数が、表 1 に示すように、莫大な数(理論値)になってしまうからである。一方、枝刈りルールを導入すると、二十数秒で解を得ることができた。MGTP によって生成されるモデルの数は、表 1 に示すように、枝刈りルールを導入しない場合と比較して、極めて少数に抑えられている。これは、特に、 $\text{satisfy_constraints_for_GCD}/1$ という仮定可能述語に関する枝刈りルールが極めて有効に作用している。

7. む す び

並列モデル生成型定理証明器 MGTP 上でアブダクションの問題解決を行う Skip 方式において、説明の極小性から冗長とみなされるモデル候補を刈るための負節を導入して、計算の効率化を行う手法を提案した。このような負節は、与えられたアブダクションの枠組みの静的解析により生成され、その理論的正当性を証明した。また、本手法を適用しない場合の MGTP によって生成されるモデル候補の数が基礎仮説の数の指数オーダであるのに対して、本手法を適用することにより、多項式オーダにすることができることを例示した。一方、本手法はすべてのアブダクションの枠組みに適用できるわけではなく、その適用範囲を広げるための変形例を示した。

また、本手法は、逐次型のモデル生成型定理証明器 SATCHMO[Manthey 88]を用いてアブダクション

の問題を解くときにも有効である。一方、トップダウン情報を付加させることによるボトムアップ型のアブダクションの高速化[Ohta 92]もまた、本手法を適用したプログラムに合わせて用いることができる[井上 93]。

謝 辞

本研究の機会を与えて下さり、常に御指導いただい

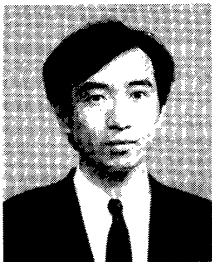
た元 ICOT 所長の東京大学 瀧 一博教授に深く感謝致します。また、並列アブダクションに関して討論していただいた ICOT PAB ワーキンググループの主宰を務められた慶應義塾大学 古川康一教授、ならびに東京大学 石塚 満教授をはじめ、各委員の方々に深く感謝致します。さらに、インプリメントに関する助言を与えて下さった ICOT の越村三幸氏、貴重なコメントを与えて下さった査読者の方に感謝致します。

◇ 参 考 文 献 ◇

- [de Kleer 86] de Kleer, J.: An Assumption-based TMS, *Artif. Intell.*, Vol. 28, pp. 127-162 (1986).
- [Eshghi 89] Eshghi, K. and Kowalski, R. A.: Abduction Compared with Negation by Failure, *Proc. ICLP-89*, pp. 54-70 (1989).
- [Fujita 91] Fujita, H. and Hasegawa, R.: A Model Generation Theorem Prover in KL1 Using a Ramified-stack Algorithm, *Proc. ICLP-91*, pp. 535-548 (1991).
- [井上 92] 井上克己: アブダクションの原理, *人工知能学会誌*, Vol. 7, No. 1, pp. 48-59 (1992).
- [井上 93] 井上克己, 太田好彦, 長谷川隆三, 中島 誠: モデル生成に基づく並列アブダクション, *人工知能学会誌*, Vol. 8, No. 6, pp. 786-796 (1993).
- [石塚 88] 石塚 満: 不完全な知識の操作による次世代知識ベース・システムへのアプローチ, *人工知能学会誌*, Vol. 3, No. 5, pp. 552-562 (1988).
- [國藤 87] 國藤 進: 仮説推論, *人工知能学会誌*, Vol. 2, No. 1, pp. 22-29 (1987).
- [Manthey 88] Manthey, R. and Bry, F.: SATCHMO: A Theorem Prover Implemented in Prolog, *Proc. CADE-88, Lecture Notes in Computer Science*, Vol. 310, pp. 415-434, Springer Verlag (1988).
- [Maruyama 88] Maruyama, F., Kakuda, T., Masunaga, Y., Minoda, Y., Sawada, S. and Kawato, N.: co-LODEX: A Cooperative Expert System for Logic Design, *Proc. FGCS-88*, pp. 1299-1306 (1988).
- [Ohta 92] Ohta, Y. and Inoue, K.: A Forward-chaining Hypothetical Reasoner Based on Upside-down Meta-interpretation, *Proc. FGCS-92*, pp. 522-529 (1992).
- [Poole 87] Poole, D., Goebel, R. and Aleliunas, R.: Theorist: A Logical Reasoning System for Defaults and Diagnosis, Cercone, N. and McCalla, G. (eds.), *The Knowledge Frontier: Essays in the Representation of Knowledge*, pp. 331-352, Springer-Verlag (1987).

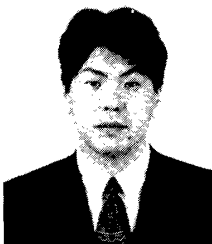
(担当編集委員: 飯田敏幸, 査読者: 牧野俊朗)

著 者 紹 介



太田 好彦 (正会員)

1983年千葉大学工学部電子工学科卒業。1985年千葉大学大学院工学研究科電子工学専攻修士課程修了。同年、三菱電機(株)入社。同社コンピュータ製作所勤務。1988年7月、(財)新世代コンピュータ技術開発機構へ出向。1992年10月より職業能力開発大学校情報工学科講師。人工知能の研究に従事。電子情報通信学会、情報処理学会各会員。



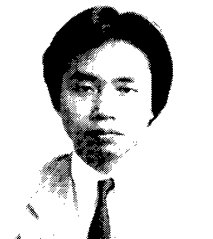
井上 克己 (正会員)

1982年京都大学工学部数理工学科卒業。1984年京都大学大学院工学研究科数理工学専攻修士課程修了。同年、松下電器産業(株)入社。同社東京情報システム研究所勤務、および(財)新世代コンピュータ技術開発機構への出向を経て、1993年4月より豊橋技術科学大学情報工学系講師。京都大学博士(工学)。人工知能および論理プログラミングに関する研究に従事。1988年3月学術奨励賞(情報処理学会)受賞。情報処理学会会員。



長谷川 隆三

1972年九州大学工学部通信工学科卒業。1974年九州大学大学院工学研究科通信工学専攻修士課程修了。同年、日本電信電話公社入社。同社武蔵野電気通信研究所勤務。1987年(財)新世代コンピュータ技術開発機構へ出向。現在に至る。九州大学博士(工学)。ポリプロセッサシステム、データフローマシン、関数型言語、論理プログラミングおよび定理証明に関する研究に従事。電子情報通信学会、情報処理学会各会員。



中島 誠 (正会員)

1987年図書館情報大学図書館情報学部卒業。1989年図書館情報大学大学院修士課程修了。(財)日本情報処理開発協会勤務を経て、現在、大分大学工学部知能情報システム工学科助手。認知モデルについての研究に従事。情報処理学会、ACM、IEEE各会員。