論文

質問変換に基づく並列サーカムスクリプションの計算手法
A Computation Method for Parallel Circumscription Based on Equivalent Transformation of Queries

岩沼 宏治*1 原尾 政輝*2 野口 正一*3
Kouji Iwanuma Masateru Harao Shoichi Noguchi

*1 山梨大学工学部電子情報工学科
Dept. of Electrical Eng. and Computer Science, Faculty of Eng., Yamanashi University, Kofu 400, Japan.
*2 九州工業大学情報工学部知能情報工学科
Dept. of Artificial Intelligence, Faculty of Information Eng., Kyushu Institute of Technology, Iizuka 820, Japan.
*3 東北大学電気通信研究所
Research Institute of Electrical Communication, Tohoku University, Sendai 980, Japan.

1989年9月13日 受理

Keywords: parallel circumscription, query transformation, automated deduction, equivalent transformation.

Summary

In this paper, we give a new method for computing parallel circumscription based on an equivalent transformation of queries. Circ {\[ A : \Gamma : \Delta \]} \vdash B is equivalent to \( A \vdash B \), if \( B \) has no negative occurrences of predicates of \( \Gamma \) and no occurrences of predicates of \( \Delta \). Therefore, if \( B \) satisfies these conditions, then the computation of Circ {\[ A : \Gamma : \Delta \]} \vdash B can be reduced into the first-order computation of \( A \vdash B \). In general, the queries \( B \) don’t satisfy the above conditions, but, sometimes, they can be transformed into the above computational sentences.

In this paper, we present an equivalent transformation method of queries into the above computational sentences, and formalize it in the form of the resolution. This transformation method replaces a negative occurrence of a minimized predicate with the formula representing its minimal extension. It is similar to the unfolding in logic programming. This transformation method is intuitationally very clear, and can deal with theories consisting of both recursive and non-definite clauses.

1. はじめに

サーカムスクリプション \(^{11, 12}\) は一般に高階論理式として定式化されるが、そのためサーカムスクリプション上の論理式 \( B \) の真偽計算は極めて難しい、これも容易にする手法として、これまではサーカムスクリプション自体を一階論理式 \(^{31}\) や論理型プログラム \(^{12}\) へ変換する方法が研究されてきた。一方、質問 \( B \) の変換を計算を容易にする手法も考えられるが、これまでほとんど研究されていない。本論文では、質問の等価変換という新しい手法に基づいた並列サーカムスクリプションの計算手法について考察する。並列サーカムスクリプション上の真偽計算、すなわち Circ {\[ A : \Gamma : \Delta \]} \vdash B の計算は、もし \( B \) が等価であるならえば、\( A \vdash B \) の計算に帰着できる \(^{11, 12}\)。制約 \( A \) は一階論理式であるから、この場合の \( B \) の真偽は一階論理上で計算できる。当然、一般的な質問 \( B \) は上の条件を満たさない、しかし、上のような論理式へ等価変換できるものも数多く存在すると考えられる。問題はその具体的な変換方法である。

本論文では以上の様々な推論処理を可能にするために、並列サーカムスクリプション上での質問の等価変換手法を考察する。われわれは文献 (7) で極小定義式を導出し、それが極小化した述語の定義として見なし
ことを示した。ここで導入する変換手法は、質問中の述語を極小定義式へ書き換えるものである。ある種のUnfold手法(6)であり、サーマススクリプションの部分計算手法と見なせる。本論文ではサーマススクリプションの制約と質問を節集合に限定する。すべての質問が前述の論理式へ変換できるわけではないが、再帰的な節と非再帰的な節双方を取り扱い、有用であると思われる。

質問変換に関する研究はまことに少なく、筆者の知る限りでは、PrzymusinskiのMILO導出法(Minimal model Linear Ordered resolution)(5)しかみないが、MILO導出法は制約と質問両方を基盤的要素と仮定して定義されている。Przymusinskiは一般の節を含むように自然に一般化できると述べているが、実際には容易ではない。すなわち、MILO導出法の一階理論への自然な一般化は健全とならない(9)。本手法は一階理論に対して健全である。したがって、MILO導出法とは違って定義変換であり、全く違った観点から構成されている。

本論文の構成は以下のとおりである。§2で並列サーマススクリプションの基本的な性質を述べる。§3で質問変換について考察する。3.1でパラメータが存在しない場合の質問変換について考察する。3.2ではパラメータの取扱い手法を考察し、並列サーマススクリプションの質問の定義変換手法を定式化する。

2. 並列サーマススクリプションの性質

以下では一階語彙に述語変数、P,Q,...を導入した二階の語彙Lを考慮する。関数変数は考えない。述語変数を含まない式を一階の式と呼ぶ。Lは関数変数を持たないから、Lの節はすべて一階である。一般論理式や節は特に断らない限り、すべて一階と約束する。Lの意味の通常の二階の論理の意味論(13)に従って定める。文集合Sの中のすべてのモデルで文Aが真であることをS→Aと略記し、S→φのときにS→Aと略記する。さらに、次の論理式をα→βと略記する。

∀x_{1},...,x_{r}(a(x_{1},...,x_{r})→β(x_{1},...,x_{r}))

【定義1】

Γ=\{p_{1},...,p_{m}\}と\Delta=\{q_{1},...,q_{r}\}を素

な述語変数の集合、A[\{p_{1},...,p_{m}\},\{q_{1},...,q_{r}\}]を節集

合とする。このとき以下二つの文を並列サーマススクリ

プションと呼び、Circ\{A;Γ;Δ\}と略記する。

A \wedge Q_{1},...,Q_{r},R_{1},...,R_{s},[A[Q_{1},...,Q_{r},R_{1},...,R_{s}]]

\wedge Γ_{1}\wedge \Delta_{1}(p_{i}\leq p_{j})\wedge Γ_{2}\wedge \Delta_{2}(q_{i}\leq q_{j})

Circ\{A;Γ;Δ\}のAを制約式。Γの元を極小化述

語、Δの元をパラメータ述語と呼ぶ。Γ=φのときの

Circ\{A;Γ;Δ\}定義サーマススクリプションと呼

び、Circ\{A;Γ;Δ\}と略記する。また適宜ΓとΔの元

を用いて、Circ\{A;p_{1},...,p_{s};q_{1},...,q_{r}\}のように表

記する。

【定義2】

Lの項に関する単名言語(Unique Name Assumption,以下UNAと略す)を、Clark

の等式理論(14)と定める。

【補題1】Aは節集合、ΓとΔを互いに素な述語変

数の有限集合とする。またBをΓの元が負に出現せず、

Δの元が全く出現しない任意の節集合とする。このとき以下の等価である。

1. UNA、Circ\{A;Γ;Δ\}=B

2. UNA、A=B

【証 明】文献(1)の定理6.12を用いれば、文献

(1)の定理5.6と全く同様に証明できるため省略

する。

Bが補題1の条件を満たすとき、UNA、Circ\{A;Γ;Δ\}=Bの計算はUNA、A=Bの一階計算に帰着
できる。当然、一般のBの条件を満たさない。しか

し、満たすように変換できるのも数多く存在する。

問題はその具体的な変換方法である。本論文ではその

ための手法として、質問Aに負に出現する極小化述語

を以下の極小定義式で置き換えて消去する手法を考

察する。

【定義3】

Aを節集合、pを述語変数とする。この

ときI-simpl,(A)p=Aを、pが正に出現するA中の

節の集合と定める。またT-simpl,(A)をI-simpl

(A)の補集合、すなわち(A\setminus I-simpl,(A))と定め

る。

【定義4】

Aを節集合、pをn項述語変数、t_{1},...,t_{r}

を項とする。このときAにおけるpの正の出現p(s_{1},

...,s_{r})をすべて、それぞれ以下の式で置き換える。

\{(s_{1},...,s_{r})\wedge(s_{1}\neq t_{1}\wedge\ldots\wedge s_{r}\neq t_{r})

置き換えたAをBと定めよう、\neg Bをpの極小定義

式と呼び、Def\{A;p:t_{1},...,t_{r}\}と表記する。

【補題2】以下が成り立つ。

Circ\{A;Γ;Δ\} ∫

\wedge Γ_{1}\wedge \Delta_{1}(p(x_{1},...,x_{r})=\neg \wedge Γ_{1}\wedge \Delta_{1}(p(x_{1},...,x_{r}))

文献(14)の定理3の(i)より明らか。B

を質問、B'をp∈ΓのB中の負の出現p(t_{1},

...,t_{r})を\neg Def\{I-simpl,(A):p:t_{1},...,t_{r}\}で置き

換えた式とする。このときBとB'は、補題2よりCirc

\{A;Γ;Δ\}上で等しい。しかし、pの負の出現を単に

極小定義式で置き換えても、pの出現はかから消去

Sept. 1990
できない、以下の節集合Aに対する Circ \{A; \text{even}\} を考える。

\begin{align*}
\text{even} (0), \\
\text{even} (x) &\supset \text{even} (\text{suc}^2 (x))
\end{align*}

質問Bとして以下の節からなる節集合を考える。

\begin{align*}
\exists \text{even} (0) \\lor \\
\exists \text{even} (x) &\supset \exists \text{even} (\text{suc}^2 (x))
\end{align*}

このとき、Def \{I\text{-simpl}_n ,.., A; \text{even} : y\} は

\begin{align*}
\exists \text{even} (0) \\lor y = 0 \\
\exists \text{even} (x) &\supset y = \text{suc}^2 (x)
\end{align*}

となる。\( B' \)をBの負リテラル\( \neg \text{even} (0) \\lor \neg \text{even} (x) \)をDef \{I\text{-simpl}_n ,.., A; \text{even} : \text{suc}^2 (0)\} で置き換えたものとする。\( B' \)は、Aを前提とすれば以下の等価である。

\[ \exists \text{suc} (0) \\not= 0 \]

\[ \forall x \exists \text{suc} (0) \not= \text{suc}^2 (x) \]


\( B \)と\( B' \)はCirc \{A; \text{even}\} の上で等しい。しかし、変換した\( B' \)にも\( even \)が負に出現してしまう。

本論文では、このような極小化述語の再帰的出現在を防ぐために等号に着目する。すなわちUNAを仮定して、(極小定義式中の) 等号をニュフィケーションで置き換えることを考える。このように置き換えた場合、例えば上の\( B' \)では\#が真であり、\( B' \)全体がTrueとなる。よって、evenの再度の出現を防ぐことができる。一般に等号をニュフィケーションで置き換えると、(ニュフィケーションの失敗により) 短かくの再帰計算を停止させることが可能である。それに伴う極小化述語の再帰的出現在も防ぐことができる。以下では、極小定義式を用いた質問変換の、UNAを用いた導出原理風の定式化について考察する。

3. 質問変換

3.1 伝語サーマスクリプション上の質問変換

まず\( \Delta = \phi \)、すなわち伝語サーマスクリプションCirc \{A; \Gamma\} 上の質問の等価変換について考察する。最も単一化作用素（以下mguと略す）等は通常のよう

【定義5】 Cを節、\( p \)を述語定数とする。このときLit_i(p')Lit_i(p'')をC中の\( p \)の正（負）リテラルの集合と定める。

【定義6】 CとDを同じ名前の変数を持たない節、\( L = \neg p (t_1, \ldots , t_n) \)をCの負リテラル、\( L = \{ M_1, \ldots , M_n \} \)をLit_i(p')の空でない部分集合とする。このとき\( L = (\neg M_1, \theta) = \cdots = (\neg M_n, \theta) \)なるmguが存在するならば、

\[ (C \theta - L \theta) \cup (D \theta - L \theta) \]

を、Cの負リテラルLに対してDの\( L \)を用いたCからの導出節を呼び、Res_{C,i}(D)Lと略記する。

また、節集合Res_{C,i}(D) \( \equiv \{ \text{Res}_{C,i}(Ls); Ls \subseteq \text{Lit}_i(p') \} \)を定める。さらにDSをCと同じ名前の変数を持たない節の集合とするとき、節集合Res_{C,i}(DS)をdef. or Res_{C,i}(D)と定める。

Dの中にLと唯一化できる正のリテラルが存在しないとき、Res_{C,i}(D) = \phi とすることに注意されたい。

【例1】以下節を考える。

\begin{align*}
\text{block} (z) &\supset \text{rectan} (z) \\
\text{on} (x, y) &\supset \text{block} (x) \lor \text{block} (y) \\
\text{block} (b) &
\end{align*}

\[ D_1 \]

また、以下の節集合DSを考える。

\begin{align*}
\text{block} (z) &\supset \text{rectan} (z) \\
\text{on} (x, y) &\supset \text{block} (x) \lor \text{block} (y) \\
\text{block} (b) &
\end{align*}

\[ D_2 \]

\[ D_3 \]

\[ D_4 \]

\[ D_5 \]

\[ D_6 \]

したがって、Res_{C,i}(D_1) = \{ D_1, D_3, D_5 \} である。

次にRes_{C,i}(D_2)を求める。Lit_i(block') = \{ block (b) \} だから、Res_{C,i}(D_2)は、節\( \text{rectan} (b) \)だけが変換である。よって、Res_{C,i}(DS) = \{ D_1, D_3, D_5, D_7 \} である。

【補題3】 Aを節集合、\( \Gamma \)を節、\( L = \neg p (t_1, \ldots , t_n) \)をCの負リテラルとする。論理式C'を、Cの\( L = \neg p (t_1, \ldots , t_n) \)をDef \{I\text{-simpl}_n ,.., A; \text{p} : t_1, \ldots , t_n\} で置き換えた式とするとき、以下の定理が成立立つ。

UNA, I\text{-simpl}_n (A) \supset

\[ C' = \text{Res}_{C,i}(I\text{-simpl}_n (A)) \]

＜証明＞ 付録参照。

以上でC中の負リテラルLの極小化定義式への置換が、Res_{C,i}(I\text{-simpl}_n (A))に定式化できた。

（定理1） AとBを節集合、\( \Gamma \)を互いに素な述

語定数の有限集合とする。また\( p \in \Gamma \)なる\( n \)項述語定数CをBの中の節、\( L = \neg p (t_1, \ldots , t_n) \)をCの負リテラルとする。このとき節集合\( B' \)を

\[ B' = (B - \{ C \}) \cup \text{Res}_{C,i}(I\text{-simpl}_n (A)) \]

とすれば、以下の定理が成立立つ。
UNA, Circ \[ A : \Gamma : \Delta \] \[ \equiv B \equiv B' \]

＜証 明＞ 補題２と３より明らか。

定理１の \( B \) と \( B' \) は論理的には同じであることに注意されたい。C の Res \((\neg\text{simp1}(A))\) への置換（以下，
\( L \) の展開と呼ば）はある種の Unfold \((6)\) と見なせる。

定理１によれば，Res \((\neg\text{simp1}(A)) \equiv \phi \) の場合，
\( B \) と \( B - \{C\} \) が等しく，\( B \) から \( C \) が削除できる。

Res \((\neg\text{simp1}(A)) \equiv \phi \) は，正リテラル \( \neg L \) を真
と判定計算しようとして失敗したことを意味し，\( B \) か
らの \( C \) の削除は負リテラル \( L \) を真，すなわち \( C \) を真と
判定することにほかならない。よって，この展開は論
理型プログラムの Negation as Failure \((4)\) に対応す
ることに注意されたい。

（例２） 以下の節集合 \( B \) に対する Circ \([ A : \text{block, on} ]\) を考える。

\[
\begin{align*}
on(x, y) & \equiv \text{block}(x) \lor \text{block}(y) \quad \cdots \cdot D_2 \\
\text{block}(b), \quad \cdots \cdot D_3 \\
on(a, b), \quad \cdots \cdot D_8 \\
\text{rectan}(b)
\end{align*}
\]

このとき以下の節からなる質問節集合 \( B \) を考える。

\[
\begin{align*}
\text{block}(z) & \equiv \text{rectan}(z) \quad \cdots \cdot D_1 \\
D_1 \text{ に極小化述語 block が負に出現しているので展開す}
\end{align*}
\]

\[
\begin{align*}
\text{on}(z, y) & \equiv \text{block}(z) \lor \text{block}(y) \quad \cdots \cdot D_4 \\
\text{on}(z, y) & \equiv \text{block}(z) \lor \text{block}(y) \quad \cdots \cdot D_5 \\
\text{on}(x, z) & \equiv \text{block}(x) \lor \text{rectan}(z) \quad \cdots \cdot D_6 \\
\text{rectan}(b) & \quad \cdots \cdot D_7
\end{align*}
\]

\( D_3, D_5, D_6 \) の極小化述語 \( \text{on} \) をさらに展開する。\( D_4, D_7 \) の
\( \text{on}(z, z) \) は \( D_3(\equiv \neg\text{simp1}(A)) \) と単一化でき
ないもの削除される。最終的に \( B \) は以下のよう
に変換される。

\[
\begin{align*}
\text{rectan}(a) & \lor \text{block}(b) \\
\text{block}(a) & \lor \text{rectan}(b)
\end{align*}
\]

上の節集合を \( B' \) とするとき，定理１より，UNA，Circ
\([ A : \text{block, on} ] \equiv B \equiv B' \).

\[ \text{arc}(x, y) \equiv \text{path}(x, y), \quad \cdots \cdot D_9 \]

\[ \text{arc}(a, b), \quad \cdots \cdot D_{10} \]

\[ \text{arc}(b, c), \quad \cdots \cdot D_{11} \]

このとき，以下の節からなる質問節集合 \( B \) を考え
る。

\[ \neg\text{path}(w, w) \]

以下同様にして変換を施すと，最終的に \( B \) は節の空集
合 \( \phi \) に等価変換される。\( \phi \) は恒真式
True と同じだから，明らかに \( A \) で \( B \) は真である。

よって

\[ \text{UNA, Circ} [ A : \text{path, arc} ] \equiv B \]

一般に変換すべき節集合 \( B \) には極小化述語が複数出
現する。よって，可能な変換も複数存在し，一意には
定まらない。このため変換の戦略が問題となる。極小
化述語が負に出現しない節集合（以下，正規化節集
合）と略に本来変換できる質問 \( B \) も適当な変換戦略
を用いないと変換できなくなる。例えば，事例 \( 3 \) の節
\( D_1 \) の変換において path を先に展開し，その後も path
の展開を続けていくと，展開は有限に続き，空集合を
得ることはできない（path の展開では変数への定数
の束縛が生じないことに注意）。よって，正規化節集
合に変換できる質問 \( B \) を，必ずそのように変換する戦
略，すなわちある種の正規化戦略を明らかにすること
が極めて重要となる。この正規化戦略は，本変換手法
と論理型プログラムの否定計算との関連を考察するこ
とによって明らかにできる。結果を述べると，ある種
の公開な戦略が本変換の正規化戦略となることが示
せる。詳細は省略する。文献（10）を参照されたい。

3.2 併列サーカムスクリプション上での質問変換
Circ \([ A : \Gamma : \Delta ] \) での質問変換を考える。ただし，
本論文では質問 \( B \) に出現する \( \Delta \) の元を直接消去する手
法は考察しない。$B$ に $\Delta$ の元が出現しない場合でも、前節の手法をそのまま適用すると、変換された $B$ には $\Delta$ の元が出現することがある。以下それを防ぐ手法を考察する。

【例 4】以下に節集合 $A$ に対する $\text{Circ}[A; \text{heavy}, \text{on}; \text{block}]$ を考える。block はパラメータである。

\[
\begin{align*}
\text{on} \,(x, y) & \supset \text{block}(x) \lor \text{block}(y) \quad \cdots \quad D_{15} \\
\text{block}(b), & \quad \cdots \quad D_{16} \\
\text{on}(a, b), & \quad \cdots \quad D_{17} \\
\text{rectan}(b), & \quad \cdots \quad D_{18} \\
\text{block}(z) & \supset \text{heavy}(z) \quad \cdots \quad D_{19} \\
\text{block}(c), & \quad \cdots \quad D_{20}
\end{align*}
\]

以下の箇処から節定理節合$B$を考える。

\[
\begin{align*}
\text{heavy} \,(w) & \supset \text{rectan}(w) \quad \cdots \quad D_{21}
\end{align*}
\]

と変換される。実の上にはパラメータ block が出現するので、$A$ 上で真偽判定を行うことはできない。

例 4 の heavy の展開はパラメータ block が出現する極小定義式を用いての置換えに相当する。節定理節集合 $B$ にパラメータが現れないしないこと、変換された $B$ もパラメータを出現させないためには、パラメータが出現しない極小定義式での置換えを考えればよい。そのためには、もとの割合を書き換えパラメータを消去するほですねはよい。割合中の中パラメータ消去には文献 (8) の手法が適用でき、結果として以下の補題 4 が成り立つ。

【定義 7】節 $C$ が $p$ に関して正 (負) に確定的とは、$C$ が $p$ に関して正 (負) なリテラルがただ一つ存在する場合をいう。節集合 $A$ が $p$ に関して正 (負) に確定的とは、$A$ の各節に $p$ の正 (負) なリテラルがただ一つしか存在しない場合である。節集合 $A$ が $p$ に関しても再帰的とは、$A$ の各節に $p$ の正と負のリテラルが同時に存在しない場合をいう。

【定義 8】$A$ を節集合、$p$ を述語定数とする。このとき $\text{U-simpl}_p(A) \subseteq A$ を、$p$ が負に出現する $A$ 中の節の集合と定める。また $F-\text{simp}_p(A)$ を $\text{U-simpl}_p(A)$ の補集合、すなわち集合 $(A - F-\text{simp}_p(A))$ と定める。

【定義 9】$A$ を節集合、$p$ を述語定数、$t_1, \ldots, t_n$ を頂とする。このとき

1. $I-\text{simp}_p(A)$ の $p$ の正の出現 $p(s_1, \ldots, s_n)$ すべてを、それぞれ $s_i \neq t_1 \lor \cdots \lor s_i \neq t_n$ で置き換え、置き換えた $I-\text{simp}_p(A)$ を $B$ とするとき、$\neg B$ を $\text{Pd}[A; p; t_1, \ldots, t_n]$ と略記する。
2. $U-\text{simp}_p(A)$ の $p$ の負の出現 $\neg p(s_1, \ldots, s_n)$ すべてを、それぞれ $s_i \neq t_1 \lor \cdots \lor s_i \neq t_n$ で置き換える。置き換えを施した $U-\text{simp}_p(A)$ を $\text{Nd}[A; p; t_1, \ldots, t_n]$ と略記する。

【定義 10】$p$ を述語定数、$B[x_1, \ldots, x_n]$ を真理値とする。論理式 $A$ がその表現する節を $\Theta_{B[x_1, \ldots, x_n]}$ と表記し、置き換えた結果を $\Theta_{B[x_1, \ldots, x_n]}^+$ と略記する。ただし、変数の変換を避けるために、必要ならば $B[x_1, \ldots, x_n]$ 中の束縛変数を適宜付け換えられる。

【補題 4】$q \equiv \Delta$ とするとき、以下が成り立つ。

1. $A$ が $p$ に関して正に確定的ならば、

\[
\text{UNA} \equiv \text{Circ}[A; \Gamma; \Delta] \equiv A \land \text{Circ}[A; \Theta_{B[x_1, \ldots, x_n]}^+; \Gamma; \Delta]
\]

2. $A$ が $q$ に関して負に確定的ならば、

\[
\text{UNA} \equiv \text{Circ}[A; \Gamma; \Delta] \equiv A \land \text{Circ}[A; \Theta_{B[x_1, \ldots, x_n]}^+; \Gamma; \Delta]
\]

＜証明＞付録参照。

【定義 9】$A$ が $q$ に関して再帰的場合、$\text{Pd}[A; q; t_1, \ldots, t_n]$ と $\text{Nd}[A; q; t_1, \ldots, t_n]$ には $q$ が出現しないので、$A \Theta_{B[x_1, \ldots, x_n]}^+$ と $A \Theta_{B[x_1, \ldots, x_n]}^-$ にも $q$ が出現しない。よって、節集合 $A$ が $p \equiv \Delta$ に関して非再帰的かつ正または負に確定的ならば、補題 4 より、$A$ は $q$ が出現しない式で置き換えることができる。さらにこのとき、$A$ は次の二つの式は簡単的にできる。

【補題 5】以下が成り立つ。

1. $A$ が $p$ に関して非再帰的かつ正に確定的ならば、

\[
\land A \Theta_{B[x_1, \ldots, x_n]}^+ \equiv T-\text{simp}_p(A) \Theta_{B[x_1, \ldots, x_n]}^+
\]

2. $A$ が $p$ に関して非再帰的かつ負に確定的ならば、

\[
\land A \Theta_{B[x_1, \ldots, x_n]}^- \equiv F-\text{simp}_p(A) \Theta_{B[x_1, \ldots, x_n]}^-
\]

＜証明＞付録参照。

以下、$T-\text{simp}_p(A)$ と $F-\text{simp}_p(A)$ に対する節総合を、UNA を用いて導出原理型に定式化する。

【定義 11】$C$ を節、$p$ を述語定数とする。

1. $\text{Lit}_c(p) = \{L_i, \ldots, L_n; (n \equiv 1) \}$ と仮定し、$p$ に関し正に確定的な$n$ 個の節 $D_1, \ldots, D_n$ を考え、このとき $C$ の負リテラル $L_1, \ldots, L_n$ に対して、それぞれ $D_1, \ldots, D_n$ を用いて並列出節

\[
\text{P-Res}_{C, p} \prec L_i, D_1, \ldots, \prec L_n, D_n, \succ
\]

を以下のように定める。まず $D_1, \ldots, D_n$ の変数を、$C$ および $D_1, \ldots, D_n$ が互いに同じ名前の変数を持たないように適当に付け換える。付け換えた節を
$D_1', \ldots, D_r'$とすると、$D_1', \ldots, D_r'$のそれぞれの
唯一の $p$ の正リテラルを $M_1', \ldots, M_r$ とするとき、
$L_1' \theta = (\neg M_1) \theta, \ldots, L_r' \theta = (\neg M_r) \theta$
なる $mgu \theta$ が存在するならば、節
$(C \theta \text{-Lit}(p')) \cup \cup_{i=1}^n (D_i' \theta \text{-Lit}(M_i \theta))$
を $P \text{-Res}_{c,r}, (<L_1, D_1>, \ldots, <L_r, D_r>)$ と定める。

2. $DS$ を $p$ に関して正に確定的な節集合とするとき、
節集合 $P \text{-Res}_{c,r}, (DS)$ を、
$(P \text{-Res}_{c,r}, (<L_1, D_1>, \ldots, <L_r, D_r>) \cup D_1', \ldots, D_r \in DS)$
と定める。特に $C$ に $p$ の正リテラルが存在しな
い場合は、便宜上、$P \text{-Res}_{c,r}, (DS) = \{C\}$ と定める。

3. $C$ の $p$ の正リテラル $L_1, \ldots, L_r$ に対して $D_1, \ldots, D_r$ も
用いる並列導出節 $P \text{-Res}_{c,r}, (<L_1, D_1>, \ldots, <L_r, D_r>)$ および節集合 $P \text{-Res}_{c,r}, (DS)$
を、①と②の双対として定める。すなわち①と②
の $p$ を $p'$、正（負）を負（正）へ読み換えたもの
として定義する。

【定義 12】 $A$ を $p$ に関して正に確定的な節集合と
するととき、以下の節集合を $\text{Red}[A ; p]$ と略記する。

\[ \cup_{\text{Res}_{c,r}, (A)} P \text{-Res}_{c,r}, (I \text{-simpl}_{c,s}(A)) \]
また $A$ を $p$ に関して負に確定的な節集合と
するととき、以下の節集合を $\text{Red}[A ; p]$ と略記する。

\[ \cup_{\text{Res}_{c,r}, (A)} P \text{-Res}_{c,r}, (U \text{-simpl}_{c,s}(A)) \]

【補題 6】 以下が成り立つ。

① $A$ が $p$ に関して正に確定的ならば、
$\text{UNA} \vdash \text{Red}[A ; p] = T \text{-simpl}_{c,s}(A) \theta_{x \rightarrow A, p_1, p_2, \ldots, p_n}$

② $A$ が $p$ に関して負に確定的ならば、
$\text{UNA} \vdash \text{Red}[A ; p] = F \text{-simpl}_{c,s}(A) \theta_{x \rightarrow A, p_1, p_2, \ldots, p_n}$

【証明】 補題 3 同様に示証できる。省略。

(定理 2) $q \in \Delta$ とするとき以下が成り立つ。

① $A$ が $q$ に関して非再帰的かつ正に確定的ならば、
$\text{UNA} \vdash \text{Circ}[A : \Gamma \Delta] = A \wedge \text{Circ}[\text{Red}[A ; q'] : \Gamma \Delta]$

② $A$ が $q$ に関して非再帰的かつ負に確定的ならば、
$\text{UNA} \vdash \text{Circ}[A : \Gamma \Delta] = A \wedge \text{Circ}[\text{Red}[A ; q'] : \Gamma \Delta]$

【証明】 $S$ を $\Gamma$ と $\Delta$ の元が出現しない文集合、
$A$ と $B$ を $S \vdash A \equiv B$ なる文とするとき、明らかに
$S \vdash \text{Circ}[A : \Gamma \Delta] \equiv \text{Circ}[B : \Gamma \Delta]$
よって、$\text{UNA}$ に $\Gamma$ と $\Delta$ の元が出現しないことに注意
すれば、補題 4, 5, 6 より明らか。

以上で制約式 $A$ におけるパラメータ $q$ の消去が、
Red $[A ; q]$ と Red $[A ; q]$ で定式化できた。

【例 5】 例 4 の $A$ は block 関して負に確定的で
あるから、Red $[A ; \text{block}]$ が定義できる。以下こ
れを求める。F-simpl $\_	ext{soc}(A)$ 以下の節集合である。
$\text{on}(x, y) \supset \text{block}(x) \vee \text{block}(y), \ldots, D_{15}$
block $(b), \ldots, D_{16}$
$\text{on}(a, b), \ldots, D_{17}$
$\text{rectan}(b), \ldots, D_{18}$
また $U \text{-simpl}_{c,s}(A)$ 以下の節集合である。
$\text{block}(z) \equiv \text{heavy}(z), \ldots, D_{19}$
block $(c), \ldots, D_{20}$
まず $D_{15}$ に対する $P \text{-Res}_{c,s} \text{res}_{c,s}(U \text{-simpl}_{c,s}(A))$ を求める。$D_{15}$ の block の単のリテラルは $L_1 = \text{block}(x), L_2 = \text{block}(y)$ である。そこで $L_1, L_2, D_{19}, D_{20}$ の四つの組合せを考える。まず $P \text{-Res}_{c,s} \text{res}_{c,s}(L_1, D_{19}, L_2, D_{20})$ は以下のもの
となる。

$\text{on}(x, y) \supset \text{heavy}(x) \vee \text{heavy}(y) \ldots, D_{22}$
$\text{on}(x, c) \equiv \text{heavy}(x) \ldots, D_{23}$
$\text{on}(c, y) \equiv \text{heavy}(y) \ldots, D_{24}$
$\text{on}(c, c) \equiv \text{heavy}(c) \ldots, D_{25}$

したがって、$P \text{-Res}_{c,s} \text{res}_{c,s}(U \text{-simpl}_{c,s}(A)) =
\{D_{22}, D_{23}, D_{24}, D_{25}\}$ となる。

次に $D_{16}$ に対する $P \text{-Res}_{c,s} \text{res}_{c,s}(U \text{-simpl}_{c,s}(A))$ を求める。$P \text{-Res}_{c,s} \text{res}_{c,s}(\text{block}(b), D_{19})$
は

$\text{heavy}(b), \ldots, D_{26}$
となる。block $(b)$ と block $(c)$ は単一化できないので、
$P \text{-Res}_{c,s} \text{res}_{c,s}(\text{block}(b), D_{26})$ は定義されな
い。よって、$P \text{-Res}_{c,s} \text{res}_{c,s}(U \text{-simpl}_{c,s}(A)) =
\{D_{26}\}$ である。

また、$D_{17}$ には block が出現しないので、$P \text{-Res}_{c,s} \text{res}_{c,s}(U \text{-simpl}_{c,s}(A)) =
\{D_{17}\}$ であり、$D_{18}$ も同様である。以上から Red $[A ; \text{block}] =$
$\{D_{22}, D_{23}, D_{24}, D_{25}, D_{26}, D_{17}, D_{18}\}$ である。以上から、以下の質問変換アルゴリズムが構成できる。

《質問変換アルゴリズム U》

Input Cir [A : Γ Δ] と質問節集合 B
Output 節集合 TB
step 1 : TA = A, TB = B
step 2 : TB にΓの元が負に出現しないとき
① TBにΔが元の出現しないならば success し、
TBを出力する。
アルゴリズム U は TB に出現するΔの元を直接消去することはできない。また、アルゴリズム全体の停止性も保証されていない。

[補題 7] アルゴリズム Uに、Circ[A; Γ; Δ] と Bを入力したときの計算過程に現れる任意の TAに対して以下が成り立ち。

\[
\text{UNA ⊨ Circ}[A; Γ; Δ] ⊨ Circ'[TA; Γ; Δ] \]

＜明＞ step3に入った時点で TA を E 更新された TA を E' とするとき、以下の定理を示す。\( \vdash \)

UNA ⊨ Circ[\( E; Γ; Δ \)] ⊨ Circ'[E'; Γ; Δ]

定理 2 より、UNA ⊨ Circ[\( E; Γ; Δ \)] ⊨ E ⊨ Circ'[E'; Γ; Δ] が成り立つから、これは明らか。

（定理 3）アルゴリズム Uに、Circ[A; Γ; Δ] と Bを入力したときの計算過程に現れる任意の TBに対して以下が成り立ち。

UNA ⊨ Circ[A; Γ; Δ] ⊨ B ≡ TB

＜明＞ step 6 が等価変換であることを示せの場合、step 6 で使用する TA に対して、補題 7 より

UNA ⊨ Circ[A; Γ; Δ] ⊨ Circ'[TA; Γ; Δ]

また、step6 に入った時点で TB を F 更新された TB を F' とするとき、定理 1 より

UNA, Circ[TA; Γ; Δ] ⊨ F ≡ F'

よって、以下が明らかに成り立つ。

UNA, Circ[A; Γ; Δ] ⊨ F ≡ F'

[例 6] 例 4 の Circ[A; heavy, on; block] と質問節集合 B に対して、アルゴリズム U を適用する。Bには極小化語 heavy が負に出現する。I-simpl, (A) = \{D₁₈\} にパラメータ block が出現するので、step5 に進む。TA は block に関して非再帰的かつ負に確定的であるから、Red [TA; block'] で更新する。例 5 より、TA は以下のようになる。

\[
\begin{align*}
on(x, y) & \circ heavy(x) \land \text{heavy(y)} &\cdots & D_{22} 
on(x, c) & \circ heavy(x) &\cdots & D_{23} 
on(c, y) & \circ heavy(y) &\cdots & D_{24} 
onon(c, c), & 
\end{align*}
\]

\[
\begin{align*}
\text{heavy(b),} & \cdots & D_{26} 
on(a, b), & \cdots & D_{17} 
\end{align*}
\]

rectan(b) \[= D_{18}\]

新しい TA にはパラメータ block が出現しない。よって、step6 に進み、この TA 上で TB を更新する。TBは

\[
\begin{align*}
on(x, x) & \circ \text{rectan(x),} &\cdots & D_{27} 
on(x, y) & \circ \text{rectan(x) \lor \text{heavy(y),} } &\cdots & D_{28} 
onon(x, y) & \circ \text{heavy(x) \lor \text{rectan(y),} } &\cdots & D_{29} 
on(x, c) & \circ \text{rectan(x),} &\cdots & D_{30} 
on(c, y) & \circ \text{rectan(y),} &\cdots & D_{31} 
\end{align*}
\]

\[
\begin{align*}
\text{rectan(b),} & \cdots & D_{32} 
\end{align*}
\]

さらに、on の負リテラルを展開すると、D_{27}, D_{28}, D_{29} が消去される。最終的に TB は以下のようになる。

\[
\begin{align*}
\text{rectan(a) \lor \text{heavy(b),} } & \cdots & D_{33} 
\text{heavy(a) \lor \text{rectan(b),} } & \cdots & D_{34} 
\text{rectan(b),} & \cdots & D_{35} 
\end{align*}
\]

上の TB を B' とするとき、UNA, Circ[A; heavy, on; block] ⊨ B ≡ B' である。B' は補題 1 の条件を満たす。また UNA, A ⊨ B' である。よって

UNA, Circ[A; heavy, on; block] ⊨ B

5. む す び

本論文では、質問の等価変換という新しい手法に基づいた並列サーカスクリプションの計算手法を示した。サーカスクリプション自身の変換による計算手法との比較は紙面の都合上省略した。大ざっぱに言うと、サーカスクリプションを一階論理式変換する手法(3)では、再帰的な節取り扱いが必要ない、また、論理型プログラム変換する手法(4)では、非確定的な節取り扱いが必要ない。本手法では、両方の統一的に取り扱い容易である。特に、語句サーカスクリプションの計算手法としてはかなり強力である。欠点として\( \exists \)記号や、パラメータそのものに関する質問、優先順位の取扱いが難しい点があげられる。
謝辞

山形大学情報工学科，および東北大学野口研究室の皆さんに感謝いたします。

◇ 参考文献 ◇

（9）岩沼・原尾：並列サーカムスクリプションの計算手法，信学技，COMP 89-42 (1989).

[担当編集委員・査読者：中川裕志]

◇ 付 録 ◇

以下、論理式Aの全称閉包式と存在閉包式を∀A、∃Aと略記する。

【定義 13】代入θ＝{x[i,l],…,x[i,l]}に対して、論理式x[i]=t[i]∧…∧x[i]=t[l]を、Eq(θ)で表す。

《命題 1》θを代入するとき、以下が成り立つ。

UNA ⊨ ∀(Eq(θ) ⊃ A) ≡ ∀(Aθ) □

【補題 8】原子論理式p(s[1],…,s[i])とp(t[1],…,t[i]),…,p(t[1],…,t[n])の単一化に関して、

p(s[1],…,s[i])θ⇔p(t[1],…,t[i])θ…,⇔p(t[1],…,t[n])θ

なるmguθが存在するならば、以下が成り立つ。

UNA ⊨ ∀[(∧[i=1](AND s[i]=t[i])] ≡ Eq(θ)]

また、単一化不可能ならば、以下が成り立つ。

UNA ⊨ □∃(∧[i=1](AND s[i]=t[i])) □

Lをリテラル、S={M[1],…,M[n]}をリテラルの集合とするとき、LとSのmguθをLθ=M[1],θ,…,θ=M[n]としてmguθを定める。LとSがmguを持つとき、空でない部分集合S'⊂SもLとmguを持ち、

＜補題 3 の証明＞ 論理の単一化のため、pは1項述語定数。Cは∀x(p(x)⊃B)の形をなし、さらにCとAには同じ変数が出現しないと仮定する。

(I) I-simpl(A) = φの場合を考える。このときResc(I)(I-simpl(A)) = φとなり、定義によりDef[I-simpl(A)] : p : tもTrueであるから、CもTrueである。


E[1] : ∀y(F[1] ⊃ p(s[1])∨p(s[2]))

まず、Resc(I)(I-simpl(A))を求める。論理の単一化のため、p(t)とLitc(p')(j=1,2)の空でない部分集合との単一化に以下仮定を置く。

(a) p(t)とLitc(p')がmguθを持ち。
(b) p(t)は{p(u[1])}とmguθを持ち、その他のLs⊂Litc(p')とはmguを持たない。

Sept. 1990 質問変換に基づく並列サーカムスクリプションの計算手法 585
以上の仮定の下では、Lit₂(ρ)+ の部分集合 {p(s₁)} と {p(s₂)} は p(t) とある mgu θ₁₁, θ₁₂を持つ。また Res₃, p₁, p₂, (p(u₃), (1)) と Res₂, p₁, p₂, (p(u₁), p(u₂)) は定義されないことに注意すれば、Res₃, (I-simpl, (A)) は以下の論理式となる。
\[\forall ((F₁ \sqsupset p₁) \sqcap B) \theta₁₁ \land\]
\[\forall ((F₂ \sqsupset p₂) \sqcap B) \theta₁₂ \land\]
\[\forall ((F₁ \sqsupset p₁) \sqcap B) \theta₁₁ \land\]
\[\forall ((F₁ \sqsupset p₁) \sqcap B) \theta₁₂ \land\]
\[\text{次に } C' \text{を求める。Def } [I-simpl₃(A): p : t] \text{は}\]
\[\exists y (F₁ \land t=s₁) \land (p(s₁) \sqsupset t=s₁) \land\]
\[\exists z (F₂ \land (p(u₁) \sqsupset t=u₁) \land (p(u₂) \sqsupset t=u₂)) \land\]
なる。部分式②は次の式と同値である。
\[\exists y (F₁ \land t=s₁) \land (p(u₁) \sqsupset t=u₁) \land\]
\[\exists y (F₁ \land (p(u₁) \sqsupset t=u₁) \land (p(u₂) \sqsupset t=u₂)) \land\]
よって、以下が明らかに成り立つ。
\[\text{I-simpl₃}(A) \vdash \text{②} \equiv (\exists y (F₁ \land t=s₁) \land t=s₁) \land\]
\[\exists y (F₁ \land (p(u₁) \sqsupset t=u₁) \land t=u₁) \land\]
\[\text{したがって、単一化の仮定(a)を考慮すれば、補題8より}\]
UNA, I-simpl₃(A) \vdash \text{②} \equiv (\exists y (F₁ \land (p(u₁) \sqsupset t=u₁) \land t=u₁) \land\]
\[\exists y (F₁ \land (p(u₁) \sqsupset t=u₁) \land (p(u₂) \sqsupset t=u₂)) \land\]
同様にして部分式③に対して、まず以下が成り立つ。
\[\text{I-simpl₃}(A) \vdash \text{③} \equiv (\exists z (F₂ \land (p(u₁) \sqsupset t=u₁) \land\]
\[\exists z (F₂ \land (p(u₁) \sqsupset t=u₁) \land (p(u₂) \sqsupset t=u₂)) \land\]
よって、単一化の仮定(b)を考慮すれば、補題8より
UNA, I-simpl₃(A) \vdash \text{③} \equiv (\exists z (F₂ \land (p(u₁) \sqsupset t=u₁) \land (p(u₂) \sqsupset t=u₂)) \land\]
\[\text{C'は } \forall x (\text{②} \lor \text{③} \lor B) \text{であるから、以下の式}\]
\[\forall x ((\exists y (F₁ \land (p(u₁) \sqsupset t=u₁) \land\]
\[\exists y (F₁ \land (p(u₁) \sqsupset t=u₁) \land (p(u₂) \sqsupset t=u₂)) \land\]
\[\text{を④と置けば、以下が成り立つ。}\]
UNA, I-simpl₃(A) \vdash \text{C' \equiv ④}
④は明らかに以下の式と同値である。
\[\forall xy (\text{Eq}(θ₁₀) \sqsupset F₁ \lor B) \land\]
\[\forall xy (\text{Eq}(θ₁₁) \sqsupset F₁ \lor p₁ \lor B) \land\]
\[\forall xy (\text{Eq}(θ₁₂) \sqsupset F₁ \lor p₁ \lor B) \land\]
\[\forall xy (\text{Eq}(θ₁₁) \sqsupset F₁ \lor p₁ \lor B) \land\]
命題1より上式は①と UNA の下で同値である。 □
<補題5の証明> (1)を証明する。(2)も全く同様に証明できる。議論の簡略化のため、"p"は1項述語定数と仮定する。節集合"A"は"T-simpl1(A)∧I-simpl1(A)"の形をなすので、以下の証明すれば十分である。

\[\vdash I-simpl1(A) \Theta_{p/p(A,A,p)} \equiv \text{True}\]
そのためには、"I-simpl1(A)"が"C_1∧⋯∧C_n"の形をなすので、各"C_i"に対して以下を証明すれば十分である。

\[\vdash C_i \Theta_{p/p(A,A,p)} \equiv \text{True}\]

議論の簡略化のため、"I-simpl1(A) = C_1 ∧ C_2 "と仮定する。条件より、"A"が"p"に関して正に確定的かつ非再帰的であるから、"I-simpl1(A) = C_1 ∧ C_2 "は以下の形をなす。

\[\forall x [D_1 ⊃ p(s)] ∧ \forall y [D_2 ⊃ p(t)]\]
ただし、"D_1 ∧ D_2 "は"p"が出現しない論理式である。したがって、"Pd [A:p; z]"は以下の形をなす。

\[\exists u (D_1' ∧ z = s') ∧ \exists v (D_2' ∧ z = t')\]
ただし、"u テ は"A"に出現しない変数で、"D_1 ' "と"s "は"D_1 " s "の自由変数"x "を"u "に置き換えたもの、"D_2 ' "と"t "は"D_2 " t "の自由変数"y "を"v "に置き換えたものである。このとき"C_1 Θ_{p/p(A,A,p)} "は以下のものとなる。

\[\forall x [D_1 ⊃ (\exists u (D_1 ' ∧ z = s') ∨ \exists v (D_2 ' ∧ z = t'))]\]
これを簡単化すると以下を得る。

\[\forall x [D_1 ⊃ (\exists u (D_1 ' ∧ z = s')) ∨ \exists v (D_2 ' ∧ z = t'))]\]
よって、以下の証明が成立する。

\[\vdash C_i \Theta_{p/p(A,A,p)} \equiv \text{True}\]

同様にして、\[\vdash C_2 \Theta_{p/p(A,A,p)} \equiv \text{True}\]も成り立つ。□

著者紹介
岩波 宏治 (正会員)
1983年東北大学通信工学科卒業。1985年同大学院修士課程修了。同年山形大学情報工学科助手。1990年山形大学電子情報工学科講師、主としてソフトウェア基礎論、人工知能基礎論等の研究に従事。電子情報通信学会、情報処理学会、ソフトウェア科学会各会員。

野口 正一 (正会員)
1954年東北大学工学部電気工学科卒業。1960年同大学院博士課程修了。1971年東北大学電気通信研究所教授。1984年東北大学大型計算機センター長。工学博士。主として情報システム構成論、知識処理に関する研究に従事。著書「情報ネットワークの理論」(岩波)「知識工学基礎論」(オーム社)など。

原尾 政輝 (正会員)
1966年九州工業大学電気工学科卒業。1972年東北大学博士課程修了。山形大学情報工学科教授などを経て、1989年九州大学情報工学科教授。工学博士。1980~82年西ドイツブランシュバイン工科大学客員研究員(フンプルト奨学生)。この間セルオートマトン、並列処理、アルゴリズム論などを研究し、現在、論理に基づく知識処理の研究に従事。電子情報通信学会、情報処理学会、ソフトウェア科学会、IEEE各会員。

Sept. 1990 質問変換に基づく並列サーカススクリプションの計算手法 587