ICOT Technical Report: TR-654

TR-654

超集合制約充足問題としての素性構造形成

向井　国昭

June, 1991

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

超集合制約充足問題としての素性構造形成*

On Synthesis of Feature Structures as Constraint Satisfaction Problems over Hypersets

向井 国昭
Kuniaki Mukai

新世代コンピュータ技術開発機構
Institute for New Generation Computer Technology (ICOT)
Mita Kokusai Bldg. 21F, 1-4-28 Mita, Minato-ku, Tokyo 108 JAPAN
Tel: +81-3-456-3069, E-mail: mukai@icot.or.jp

あらまし

素性構造を超集合 (nonwellfounded set) とみなし，素性構造の形成過程を超集合の上の制約充足問題 (= 単一化問題) とみなす．またその制約充足問題のあるクラスに対して自然な決定手続きを与える．これは，extensional な素性構造の包摂 (subsumption) 制約充足問題 —Dörre[4] が肯定的に解決 (1990 年 4 月)— の超集合論的な別解である．
次に有限オートマトンを以下の方法で超集合に変換して両者の間の関係を明確にした．まず，高々有限個の繰り返し構造を持つ超集合の一種として有理木を導入し，有理木の上の制約として 3 型文法にも似た正則等式系を，超集合の部分制約クラスとして導入した．そしてこの有理木制約クラスが有限オートマトンと，したがって正則言語の概念とも，等価であること示す自然でかつ直接的な翻訳を与えた．この翻訳は，有限オートマトン理論が，有理木のクラスに対応する制約理論であることを示している．

ABSTRACT:

*電子情報通信学会 「言語獲得と概念形成過程の工学的モデル化時限研究専門委員会」第 3 回研究会，第二種研究会資料 LA90－14(1991－2)，1991 年 2 月 14 日．

Identifying feature structures with nonwellfounded sets, process of feature structure synthesis is modeled as constraint satisfaction problems, i.e., unification problems, over hypersets. A decision procedure for a class of extensional subsumption constraint satisfaction problems over feature structures is given. The procedure is a hyperset–theoretical version of positive answer to the problem class, which was first solved by Dörre (1990 April).
A relation between automata and hypersets is made clear by a translation from automata into hepersets. First, rational trees are introduced as a kind of hypersets which have only finite repetitive substructures and also systems of regular equations are introduced as a subclass of hyperset constraints which have similar forms to grammars of type 3. Second, a natural translation is given from automata into systems of regular equations, which shows that concepts of finite automata, regular languages, and systems of regular equations are equivalent to each other. Also the translation shows that the theory of finite automata is a theory which corresponds exactly to the subclass of rational trees in the universe of hypersets.

1 はじめに

Colmerauer[3] は与えられた例からオートマトンを合成する簡単な Prolog プログラムを示している. 例は受理すべき文字列の他に, 受理してはならない文字列も含む. オートマトンは有限の有向グラフの一種であり, そして有向グラフは無限木の部分クラスである有理木 (rational tree) であることから, Colmerauer のプログラムでは, 目標のオートマトンを, 有理木上の (有限の) 等式系の解として表現されている. このオートマトン合成プログラムは無限木の単一化の応用のデモンストレーションとして 示された. Prolog のバックトラッキングのメカニズムにしたがって状態数が小さいオートマトンから順に生成され, 与えられた例を満たす最小状態数のオートマトンが生成されている.

近年, 計算言語学における素性構造 (feature structure) の研究を中心に, 知識表現における概念記述, プログラミング言語におけるレコード型の推論など, データベースにおける複合オブジェクトなど, 構造を持った情報の研究が盛んである. 次の文献はその一部である [13, 7, 8, 6, 9, 15, 14, 12, 16, 17].

これらの研究は, 情報の等価性や情報量の大小関係に関する制約理論の研究と見ることができる. 基本的な情報構造として, 多くは, 木 (tree) あるいはより一般に有向グラフが採用されている. つまり素性構造は有向グラフで表現される. すなわち, 情報構造 — 有向グラフ — 有限オートマトン — 無限木の単一化 — 論理プログラミング　とある種の '親密' な関連性があることが理解できる. この関連性および Colmerauer[3] にしたがい, 素性構造の形成メカニズムの基本を, 有限オートマトンの合成問題として進めることもできるであろう. いずれにせよ, 素性構造の基礎として, 有向グラフが重要な役割を果たしてしていることをここで確認しておく.

一方, Barwise[2] は超集合論 [1] を用いて, 超集合上の制約理論に関する一つの定理 (Barwise の単一化定理) を示している. Barwise の単一化定理は, bisimulation を表す等式と subsumption を表す不等式とからなる制約の可解性を, simulation 対への拡張可能性として特徴付けるものである. ここで subsumption とは hireditary な部分集合関係であり, また simulation 対とは bisimulation と subsumption の規則をともに満たし, 無矛盾でかつ自由変数を含まない制約である. しかし Barwise[2] は素性構造への応用を示さなかった. Mukai[10] は, Barwise の単一化定理が simulation 対に自由変数を許した場合でも成り立つことを示した. これは本稿で示すアルゴリズムの正当性の上で本質的な拡張である.

さて, 超集合論 [1] は — 私見によれば — 集合論の標準である ZFC 集合論を有向グラフ論的な対象の記述にも便利なように拡張したものである. 超集合論の観点から素性構造, 有限オート

マトンなどを見直したらどうなるだろうか? 実際に見直しをやってみた限り, 素性構造理論や有限オートマトン理論は容易にかつ自然に超集合論に翻訳できることが分かった. 特に, 素性構造の上の extensional な subsumption 問題と呼ばれる未解決問題に対する決定手続きを具体的に与えることができた. この未解決問題は, Dörre[4] が非決定性オートマトンを決定性オートマトンに変換するアルゴリズムに帰着させることですでに解いていたものであるが, 本稿の結果はそれと独立であり, 超集合に対するより一般的な制約理論 [10] の応用として得られるものである. 本稿の主目標は, このアルゴリズムの概要を示すことである. 非決定性オートマトンを決定性オートマトンに変換する良く知られた手続きが, 超集合制約の変換の形で現れる. 「オートマトンは超集合」であることを示唆していると考えられる.

有限オートマトン — 正則言語 — 正則表現, これらが等価な概念であることはよく知られている. そこで超集合論の等式制約の一種として 3 型文法にも似た正則等式系, そして有限個の繰り返し構造をもつ超集合としての有理木をそれぞれ定義する. 有理木は正則等式系の解として特徴付けられる. また有理木の中を走る有限長の極大パスの全体集合は正則言語であることも容易に分かる. 逆に正則言語は, 有理木のパス全体としても特徴付けられる. 有限オートマトンと正則等式系の間にそれぞれに対応する正則言語を保存するような自然な翻訳が示される. まとめると, 有限オートマトン — 正則言語 — 正則表現, そして有理木 — 正則等式系, これら 5 つはすべて同等な概念であることが困難なく分かる. これらを明確に指摘することが本稿第 2 の目的である.

有限オートマトンには決定性と非決定性の 2 種類あるが, それに応じて, 本稿でも決定性 / 非決定性正則等式系あるいは決定性 / 非決定性有理木が自然に導入される. また非決定性正則等式系を決定性正則等式系へ変換するアルゴリズムも与えられる.

さて, [10] は一般の超集合制約として等式 (bisimulation) と不等式 (subsumption), 否定, 論理和を議論している. その中で, 否定と論理和を持たない場合の制約の可解性の構成的な特徴付けを与えている. つまり, 決定手続きも含んでいる. ただしそこでの subsumption は hireditary な部分集合関係であるので, 本来の素性構造の意味での決定手続きとはなっていない. ちなみに, 否定と論理和を含む一般の超集合制約は Colmerauer の無限木の等式 (=) 非等式 ($\neq$) の制約理論の理論の保守拡大 (conservative extension) であることが証明されている [11].

そこで本稿では, その決定手続きを素性構造に適用できるように拡張した. 本稿の超集合制約に基づく定式化の利点は, 素性構造の subsumption 決定問題を有限オートマトンの subsumption 問題に変換するすることなく、概念的により一般的で柔軟な集合 subsumption 問題の一種として直接的に扱えることである.

本稿の構成は次のとうりである. 次節で超集合論からのやや詳しい準備を行う. Aczel[1] に従う. 超集合の概念, bisimulation, subsumption, coinductive な定義法などが重要である. 第 3 節で素性構造制約充足問題の決定手続きを与える. 第 4 節で有限オートマトンと正則等式系の翻訳規則を与える. 第 5 節は結論である.

2 超集合論からの準備

2.1 AFA と Solution Lemma

P.Aczel[1] の集合論の発端は, R.Milner の SCCS(Synchronous Calculus of Communicating System) 理論である. SCCS はプロセスを状態遷移ネットを使って表す. Aczel は, 状態間の遷移を集合の間のメンバシップに対応させることによりプロセスの状態を集合と解釈できることを見いだした. プロセスの等価性は, 状態遷移に関して合同な, 状態間の同値関係で表わされるが, P.Aczel は, 等価なプロセスは同一の集合を表すと考えた. ただし, 無限に状態遷移を続けるプロセスをも取り扱うので, 次のような無限のシーケンスを許す集合論が必要となる:

$$x_0 \ni x_1 \ni \cdots \ni x_{n-1} \ni x_n \ni \cdots$$

これを nonwellfounded 集合論という. 本稿では, 短く, 超集合論 (hyperset theory) と呼ぶ.

超集合論の世界では，たとえば集合方程式

$$x = \{x\}$$

は解を持つ．この方程式の解は

$$x = \{\{\{\cdots\}\}\}$$

である．この x は無限に深い底無しの繰り返し構造を持つ．一方，通常の ZFC 集合の世界では，自分自身を含む集合は許されない．あるいは，もっと一般に上のような無限のシーケンスは存在しないと公理 (FA, Foundation Axiom) で宣言されている．FA は，通常は次の形で述べられる：

FA (Axiom of Foundation) すべての集合 x は $x \cap y = \emptyset$ なる要素 $y \in x$ を持つ．

超集合論では FA が AFA (Anti–Foundation Axiom) に置き換わる．残りの公理はそのままである． AFA は，有向グラフの接点が，矢をメンバシップとみなすことにより，集合と解釈できるということを主張する公理である．この AFA をより正確に述べよう．ノードからなるクラス N と $N \times N$ の部分クラス E の順序対 $G = (N, E)$ を有向グラフという．$(x, y) \in E$ のとき x を y の親，y を x の子と呼ぶ．x の子の全体は集合を成すと仮定する．G の**デコレーション**とは各ノードに集合を割り当てる関数 d であり，各ノード x に割り当てる集合 $d(x)$ はその子に割り当てる集合の全体である：$d(x) = \{d(y) \mid y \text{ は } x \text{ の子}\}$．**超集合論**は，ZFC から FA を取り除き，次の反基礎の公理 AFA で置き換えた集合論である．

公理 1 (AFA(Anti–Foundation Axiom)) 任意の有向グラフはデコレーションを常にしかも唯一持つ．

超集合論は，wellfounded な集合のみからなる従来の ZFC 集合論の世界に nonwellfounded な集合を導入して広げたものと考えられる．このような例は，過去にも，実数から複素数へ拡張など多くの例がある．以後，単に集合と言えば超集合論の意味での集合を意味し，ZFC の意味での集合を well-founded 集合と呼ぶ．

AFA は，使用にとても便利な Solution Lemma と呼ぶ定理と同値である．その定理を述べる．X はパラメータのクラスで，proper クラスであってもよいとする．X– 集合とは要素としてパラメータをも許した集合である．たとえば，$X = \{x, y\}$ として，$\{x, \{x, y\}\}$ は X– 集合である．**等式系**とは 等式のクラス

$$\{x = b_x \mid x \in X\}$$

である．ここで X はインデックス集合，$\{b_x\}_{x \in X}$ は X– 集合族である．b_x に現れる変数は X からのものに限られているので，等式系には‘自由変数’は現れないことに注意する．そのとき：

定理 1 (Solution Lemma) 等式系は解を常にしかも唯一持つ．

例 1 Solution Lemma によれば，次の集合方程式が解をただ一つ持つ．$x = \{a, x\}$, $y = \{b, y\}$．ここで a, b はアトムとする． □

超集合論の世界では帰納的 (inductive) な定義は一般にはできない．帰納法がメンバシップ関係の well-founded 性に基礎を置くからである． しかし帰納法の代わりに coinductive な定義が使える．たとえば無限 2 進木の領域 は次の条件を満たす**最大**の超集合領域 B として定義できる：

(1) $x \in B$ ならば $x = \emptyset$ か，あるいはある $y, z \in B$ が存在して

(2) $x = (y \cdot z)$ である．

これは，クラスオペレータ $X \mapsto \{(x \cdot y) \mid x, y \in X\}$ の**最大**不動点でもある．一方，有限 2 進木の領域 D は次を満たす**最小**の超集合領域として定義できる：

(1) $\emptyset \in D$.

(2) $x \in D$ かつ $y \in D$ ならば $(x \cdot y)$ は D の要素である.

これは, クラスオペレータ $X \mapsto \{(x \cdot y) \mid x, y \in X\}$ の最小不動点でもある.

無限2進木と同様に, ストリームも定義できる. 実際, A を基本データの集合として, A の要素からからなる無限のストリームの全体集合 A^* は, 次の条件を満たす最大の超集合領域として定義される: x が A^* の要素ならばある $a \in A, y \in A^*$ が存在して $x = \langle a, y \rangle$ と書ける. これは, クラスオペレータ $X \mapsto A \times X$ の最大不動点でもある.

また超集合のユニーバース V 自体も, $x \in V$ ならば x は集合でかつ $x \subseteq V$ であるような最大のクラスとして特徴つけられる. V は power クラスオペレータ $pow: X \mapsto pow(X)$ の最大不動点でもある.

'最大不動点法' を使って色々な性質を示すことができる. たとえば, Ω を $\Omega = \{\Omega\}$ を満たす解とするとアトムを含まない任意の超集合 x に対して $x \sqsubseteq \Omega$ であることを最大不動点法で証明してみよう.

証明 超集合の間の2項関係 R を $R(x, y) \iff x \in V \wedge y = \Omega$ と定義する. R は subsumption の最大性を除いた定義条件を満足することはすぐ確かめられる. $\sqsubseteq$ の最大性から $R \subseteq \sqsubseteq$. これは証明すべき命題そのものであった. □

2.2 Bisimulation と Subsumption

M を有向グラフとする. ノード a に対して, a_M で a の (直接の) 子の全体集合を示す. 集合論の場合 $a_M = a$ である. Aczel は有向グラフ上の関係として, bisimulation を定義した.

定義 1 [Bisimulation] M の上の2項関係 R は $R \subseteq R^+$ のとき M の上の *bisimulation* という. ここで $a, b \in M$ に対して

$$aR^+b \iff \forall x \in a_M \exists y \in b_M xRy \quad \& \quad \forall y \in b_M \exists x \in a_M xRy.$$

□

bisimulation とはお互いに他を摸倣するということであり, つまり等価であることを意味する. 代数でいう合同な同値関係 (congruence relation) の一種であり, いわば集合上のメンバシップ関係に関する合同関係である. 集合論の場合, bisimulation は集合の外延性 (extensionality), すなわち, 等価な要素を等しく持つ集合は等しいとする性質を述べていると考えられる.

次に subsumption は一般にある種の情報量の大小関係を表す関係である. bisimulation を一方向にしたものである. $\sqsubseteq$ で表す. 集合論においては hereditary な部分集合関係のことである. オートマトン理論においてはオートマトン間の準同型写像を関係に一般化したものである. オートマトンの受理能力の大小関係を表す.

定義 2 [Subsumption] M の上の2項関係 R は $R \subseteq R^-$ のとき M の上の *bisimulation* という. ここで $a, b \in M$ に対して

$$aR^-b \iff \forall x \in a_M \exists y \in b_M xRy.$$

□

2.3 超集合の上の制約論理プログラミング

超集合論がなぜ制約論理プログラミング (CLP)[5] 図式と相性が良いと考えられるか，その理由を一つだけ説明しよう．なお [10] により詳しく説明されている．そのため，制約としては等式のみからなる集合を考えよう．超集合の世界で [1] により次の 3 性質がすべて成り立つことが，それぞれ自明あるいは容易に示される．

(1) すべてのオブジェクトは制約の解として表される (制約定義可能性).

(2) 制約が解を持つのはその任意の有限部分系が解を持つときおよびそのときに限る (コンパクト性).

(3) 制約が解を持つのはそれが bisimulation に拡張できるときおよびそのときに限る (充足完全性).

超集合論が CLP 図式の要請を自然な形で満たしていることは両者の相性が良い根拠であろう．

3 有理木，正則等式系，素性構造

3.1 有理木

Σ をアトムからなる有限集合とする．次の条件を満たす最大のクラス T を $T(\Sigma)$ で表しその元を (Σ 上の) (非決定性) 木と呼ぶ: $x \in T$ ならば

(1) x は有限集合.

(2) $x \subseteq \{\lambda\} \cup \Sigma \times T$.

次の条件を満たす最大のクラス T' を $T'(\Sigma)$ で表し，その元を (Σ 上の) 決定性木と呼ぶ:

(1) $x \subseteq \Sigma \times T'$ は Σ から T' への部分関数.

(2) $x = \{\lambda\} \cup y$, ただし $y \in T'$.

$T'(\Sigma) \subseteq T(\Sigma)$ である．

定義 3 メンバシップ ($\in$) に関する推移閉包が有限集合であるような木を (非決定性) 有理木と呼びその全体を $RT(\Sigma)$ と表す．決定性有理木の全体 $RT'(\Sigma)$ も同様に定義する． □

有理木および，決定性有理木はそれぞれ正則等式系 ($\in E(\Sigma, X)$) および決定性正則等式系 ($\in E'(\Sigma, X)$) の解として得られる超集合として特徴付けられる．

定義 4 [木上のパス] 関数 $L{:}\,T(\Sigma) \to pow(\Sigma^*)$ を次のように定義する．$t \in T(\Sigma)$ とする．

$$t \ni (a_1, u_1), u_1 \ni (a_2, u_2), \cdots, u_{n-1} \ni (a_n, u_n), u_n \ni \lambda$$

なる有限列 $a_1 \cdots a_n \in \Sigma^*$ の全体を $L(t)$ と置く．$L(t)$ の元を t のパスと呼ぶ． □

3.2　木のマージ演算

$T'(\Sigma)$ の上のマージ演算 $*$ を coninductive に定義する．p を決定性木として $(a,x) \in p$ となるよう x は高々ひとつである．この x を p_a と書く．このことに注意して $r \stackrel{\text{def}}{=} p * q$ を次の条件で定義する．

(1) $\lambda \in p \cup q$ かつそのときに限り $\lambda \in r$.

(2) p_a かつ q_a の両方が存在するならば $(a, p_a * q_a) \in r$.

(3) p_a のみが存在するならば $(a, p_a) \in r$.

(4) q_a のみが存在するならば $(a, q_a) \in r$.

命題 2 $L(x * y) = L(x) \cup L(y)$.

$T(\Sigma)$ から $T'(\Sigma)$ へのマージ変換 τ を coinductive に定義する．$p \in T(\Sigma)$ とする．$a \in \Sigma$ として $p(a) = \{x \mid (a,x) \in p\}$ とおく．$q \stackrel{\text{def}}{=} \tau(p)$ を次の条件で定義する．

(1) $\lambda \in p$ かつそのときに限り $\lambda \in q$.

(2) $p(a)$ が空ならば $q(a)$ も空である．

(3) $p(a)$ が空でなければ $q(a) = \{\tau(x_1) * \cdots * \tau(x_n)\}$ である．ただし $p(a) = \{x_1, \cdots, x_n\}$ とする．

マージ変換を $T'(\Sigma)$ に制限すると恒等写像が得られる．
次の命題は困難なく証明される．

命題 3 $t \in RT(\Sigma)$ ならば $\tau(t) \in RT'(\Sigma)$ である．

命題 4 $L(t) = L(\tau(t))$

3.3　素性構造

Σ と C を互いに素な二つの有限集合とする．次の条件を満たす最大のクラス F を $F(\Sigma, C)$ と表しその元を素性構造 (feature structure) と呼ぶ: $x \in F$ ならば

(1) $x \in C$，または

(2) x は Σ から F への部分関数.

有理木と同様に有理素性構造を定義し，その全体を $RF(\Sigma, C)$ と書く．ただしすべて‘決定性’とする．‘非決定性’素性構造は考えない．

4 素性構造制約決定問題

4.1 集合制約

[10] において, (超) 集合に関する制約理論が扱われている. 本稿の素性構造の制約アルゴリズムの基礎として必要な部分をここに要約して利用する. L を次のような制約言語とする. 制約言語 L の基本制約形は, 集合等式 $x = y$, 集合不等式 $x \sqsubseteq y$ である. 等式は集合としての同一性を, 集合不等式は hireditary な部分集合関係しての subsumption 関係を表す. 等式は bisimulation の制約規則に, 不等式は subsumption の制約規則にそれぞれ従う.

制約の一般形としてそれらのブール結合: 否定 $\neg$, 論理積 $\wedge$, 論理和 $\vee$ を許す. しかし, $\forall$ と $\exists$ は許さない. 次は L の集合制約の例である.

$$x = \{u, y\} \vee (y \neq \{v, x\} \wedge u \sqsubseteq v).$$

集合制約が解を持つための条件を support の存在条件で特徴付けることが基本的方針である. ここで support とはリテラルからなる, 無矛盾でかつ制約規則で閉じている集合である. support は自由変数を含んで良い. 実際はさらに付帯条件が付くが本稿は否定と論理和を扱わないのでこれで充分である.

否定制約を含まない制約に関して Barwise[2] は次の単一化定理を証明した. ただし, 実際の述べ方は異なる.

Barwise の単一化定理　制約は, 全ての変数が束縛されているような support に拡張できれば解を持つ.

次の定理は [10] に与えられている.

束縛化定理　すべての support は自由変数を持たない support に拡張できる.

有限のグラフとして与えられている制約に対して, 単純な列挙法により, すべての極小 support を構成できることが容易に分かる. したがって, support を持つか否かは決定可能である. 解を持てば support が存在することは明かであり, 逆に support が存在すれば, 上の束縛化定理と Barwise の単一化定理を順に適用することにより, 制約が解を持つことが言える. したがって (超) 集合の bisimulation–subsumption 制約問題は決定可能である. この結果を使って素性構造の単一化問題を解く.

Σ から $X \cup C$ への部分関数を項と呼ぶ. p, q を項として式 $p = q$ を等式, $p \sqsubseteq q$ を subsumption という. 等式と subsumption からなる有限集合を (素性構造) 制約と呼ぶ. 制約の全体を $Con(\Sigma, C, X)$ で示す. c を制約とし θ を X から $F(\Sigma, C)$ への部分関数で $\mathcal{V}(c) \subseteq \mathrm{dom}(\theta)$ とする. このような θ を c の割り当てと呼ぶ. また θ が制約のすべての等式あるいは subsumption を満たす時 θ を c の**解**であるという. c が解を持つとき c は**可解**であるという. c が可解かどうかを判定することを**決定問題**という.

アルゴリズムの記述の工夫として, $pow(X \cup C)$ の元も変数と見なす. これを**拡張変数**と呼ぶ. $x, y \subseteq X \cup C$ のとき (拡張) 変数 xy を $xy \stackrel{\mathrm{def}}{=} x \cup y$ と規約する. 変数 xy に対する解が変数 x と y の解のマージになるように制約規則が設計されている: $\theta(xy) = \theta(x) * \theta(y)$, ここで θ は制約の解で, $*$ は有理木のマージと同様に定義される素性構造のマージ演算である. また x と y をオートマトンの状態と見れば, xy はそれらを'マージ'した状態を表すようになっている.

定義 5 次の条件を満たす制約 $c \in Con(\Sigma, C, X)$ を *support* と呼ぶ.

(1) $u = v \in c \Longrightarrow v = u \in c.$

(2) $u = v, v = w \in c \Longrightarrow u = w \in c$.

(3) $p = q \in c \Longrightarrow \mathrm{dom}(p) = \mathrm{dom}(q)$.

(4) $p = q \in c,\ a \in \mathrm{dom}(p) \Longrightarrow p(a) = q(a) \in c$.

(5) $x = p, y = q \in c \Longrightarrow xy = p * q \in c$.

(6) $u \sqsubseteq v, v \sqsubseteq w \in c \Longrightarrow u \sqsubseteq w \in c$.

(7) $p \sqsubseteq q \in c \Longrightarrow \mathrm{dom}(p) \subseteq \mathrm{dom}(q)$.

(8) $p \sqsubseteq q \in c, a \in \mathrm{dom}(p) \Longrightarrow p(a) \sqsubseteq q(a) \in c$.

(9) $p \sqsubseteq x \in c, q \sqsubseteq x \in c \Longrightarrow p * q \sqsubseteq x \in c$.

(10) $u \in x, v \in x \Longrightarrow u = v \in c$. ただし，$x$ は c に現れる拡張変数: $x \subseteq X \cup C$.

□

規則 5と 9が重要である． 規則の意味は拡張変数 xy の意味と素性構造のマージ演算 $*$ の意味から明らかであろう. 他の規則は, bisimulation と subsumption の規則を素直に述べたものに過ぎない. 非決定性オートマトンを決定性オートマトンに変換する, 良く知られたアルゴリズムにおける状態のマージに対応している操作と考えられる.
項に関する 2 項演算 $*$ を定義する.

定義 6 p, q を項とする. ただし $p, q \notin X \cup C$ とする. $r \stackrel{\mathrm{def}}{=} p * q$ を次の条件を満たす項として定義する.

(1) $\mathrm{dom}(r) = \mathrm{dom}(p) \cup \mathrm{dom}(q)$.

(2) $a \in \mathrm{dom}(p) \cap \mathrm{dom}(q) \Longrightarrow r(a) = p(a) * q(a)$.

(3) $a \in \mathrm{dom}(p), a \notin \mathrm{dom}(q) \Longrightarrow r(a) = p(a)$.

(4) $a \in \mathrm{dom}(q), a \notin \mathrm{dom}(p) \Longrightarrow r(a) = q(a)$.

□

定義 7 次の式を**コンフリクト**と呼ぶ. ただし $c, c' \in C$, $c \neq c'$, $p \notin C \cup X$ とする.

(1) $c \sqsubseteq c'$.

(2) $c = c'$.

(3) $p \sqsubseteq c$.

(4) $c \sqsubseteq p$.

□

命題 5 $Con(\Sigma, C, X)$ に属する素性構造の単一化問題は決定可能である.

証明 証明は難しくない. 証明の概要を記す. 与えられた制約 d とおく: $d \in Con(\Sigma, C, X)$. d から出発して, support の規則にしたがって, 新しい等式と不等式を生成して d に加えて行く. この過程を新しい式が生成されなくなるまで繰り返す. この過程は制約と規則の形から有限ステップで停止することは明らかである. 結果の有限集合を c とおく: $d \subseteq c$. 上で説明したように, Barwise の単一化定理と Mukai[10] の束縛定理がこの場合にも適用できる. したがって c は次の性質を持つ. もし c がコンフリクトを持たない support のときおよびそのときに限り d は解を持つ. □

例 2 $x, y, z \in X$ を相異なる変数, $a, b, c \in \Sigma$ を相異なる属性, $a', b' inC$ を定数とする. 次の (1) から (5) までが与えられた素性構造制約として与えられたとする:

$$
\begin{aligned}
x &= \{(a, a'), (b, b'), (c, x)\} & (1)\\
y &= \{(a, a'), (c, z)\} & (2)\\
z &= \{(b, b'), (c, y)\} & (3)\\
\{(c, y)\} &\sqsubseteq x & (4)\\
\{(c, z)\} &\sqsubseteq x & (5)
\end{aligned}
$$

この制約に上の決定手続きを適用する. まず, (4) と (5) に 規則 9 を適用して 下の (6) を得る. ここで拡張変数 yz が生じた. そこで拡張変数 yz について, 規則 5を (2) と (3) に適用して (7) を得る.

$$
\begin{aligned}
\{(c, yz)\} &\sqsubseteq x & (6)\\
yz &= \{(a, a'), (b, b'), (c, yz)\} & (7)
\end{aligned}
$$

こうして得られた (1)~(7) 対して, 他の規則についての自明な閉包演算を適用することにより, すべての制約規則 1 ~ 10に関して閉じてかつコンフリクトのない制約すなわち support を得る. ゆえに, 与えられた制約は充足可能である. 実際, x, y, z は (1) ~ (3) から solution lemma によりユニークに決まり, またそれらが (4) と (5) を満たすことは容易に確かめられる. □

4.2 素性構造から木へ

素性構造 $f \in F(\Sigma, C)$ に表われる (a, b) $(b \in C)$ を $(a, (b, \{\lambda\}))$ に置き換えることにより $F(\Sigma, C)$ を $T(\Sigma \cup C)$ に, また $RF(\Sigma, C)$ を $RT(\Sigma \cup C)$ に自然に埋め込むことができる. こうして, 正則等式系で定義された素性構造の等価性や subsumption の判定問題はそれぞれ有限オートマトンの等価性問題や受理言語の包含問題として翻訳できることが分かる.

5 有限オートマトンと正則等式系

5.1 正則等式系

Σ と X を互いに素な二つの集合とする. そのとき次の条件を満たす族 $\{b_x\}_{x \in I}$ を (Σ, X) の上の正則等式系と呼ぶ.

(1) $I \subseteq X$ は有限集合.

(2) $b_x \subseteq \{\lambda\} \cup \Sigma \times I$.

ここで λ は特別なアトムとする. 正則等式系の全体を $E(\Sigma, X)$ で表す. 正則等式系は, $(a, y) \in b_x$ かつ $(a, z) \in b_x$ ならば $y = z$ という条件が常に成り立っているとき決定性正則等式系と呼ぶ. 決定性正則等式系の全体を $E'(\Sigma, X)$ で表す. 正則等式系 $\{b_x\}_{x \in I}$ は便宜上, 等式の集合 $\{x = b_x \mid x \in I\}$ として書く.

5.2 決定性正則等式系への変換

変換 ψ: $E(\Sigma, X) \to E'(\Sigma X)$ を定義する. $E = \{x = b_x \mid x \in I\}$ とおく. 簡単のため $pow(I)$ から X への単射 i で $i(\{x\}) = x$ $(x \in I)$ なるものが存在すると仮定する.

$$\psi(E) \stackrel{\text{def}}{=} \{i(u) = b'_{i(u)} \mid u \subseteq I, u \neq \emptyset\}$$

と定義する. ここで $b'_{i(u)}$ は次の条件で定義される集合である.

(1) ある $x \in u$ に対して $\lambda \in b_x$ かつそのときにかぎり $\lambda \in b'_{i(u)}$.

(2) $v = \{y \mid (a, y) \in b_x, x \in u\}$ として $v \neq \emptyset$ かつそのときにかぎり $(a, i(v)) \in b'_{i(u)}$.

命題 6 E を正則等式系とし, x を E のルート変数とする. そのとき $\theta(x) = \theta'(x)$ である. ただし θ, θ' はそれぞれ E, $\psi(E)$ の解である.

5.3 有限オートマトン

bisimulation と subsumption は有限オートマトンにも自然に適用される. ふたつの (決定性) 有限オートマトンは bisimulation 関係の存在するときおよびそのときに限り受理言語に関して等価である. 非決定性有限オートマトンの場合, 受理能力に関して等価でも bisimulation が存在しない例はすぐ作れるから, この特徴付けは成り立たない.

一方, subsumption は言語の受理能力の大小に関係している. 有限オートマトンは 5- 組 $(\Sigma, S, s_0, \delta, F)$ である. ここで $\delta: \Sigma \times S \to S$ は部分関数, $s_0 \in S$, $F \subseteq S$ である. 次に δ を記号列に対しても拡張定義して それを δ^* とおく. すなわち, $\delta^*: \Sigma^* \times S \to S$ (部分関数), $\delta(\epsilon, s) = s$, $\delta^*(a * \alpha, s) = \delta(a, \delta^*(\alpha, s))$.

$A(\Sigma, S)$ で (非決定性) 有限オートマトンの全体集合を表す. $A'(\Sigma, S)$ で決定性有限オートマトンの全体集合を表す. $A'(\Sigma, S) \subseteq A(\Sigma, S)$. 単に有限オートマトンと言えば非決定性有限オートマトンを意味することとする.

オートマトン α が受理する言語を $L(\alpha)$ と書く. 非決定性オートマトン α をそれと等価な決定性オートマトンに変換するアルゴリズムは良く知られている. この変換を ϕ とおく: $L(\alpha) = L(\phi(\alpha))$.

オートマトンの間の subsumption 関係を定義する. ふたつのオートマトンを $A_i = (\Sigma_i, S_i, s_i, \delta_i, F_i)$ $(i = 1, 2)$ とおく. それぞれの状態空間 S_1 と S_2 の間に次の性質を満たす 2 項関係 $\sqsubseteq \subseteq S_1 \times S_2$ が存在すれば A_1 は A_2 を subsume するという.

(1) $s_1 \sqsubseteq s_2$.

(2) $s \in F_1$ かつ $s \sqsubseteq s'$ ならば $s' \in F_2$.

(3) $s \sqsubseteq s'$ かつ $\delta_1(a, s) \ni u$ ならば $\delta(a, s') \ni u'$ かつ $u \sqsubseteq u'$ なる $u' \in S_2$ が存在する.

5.4 正則言語

$R(\Sigma)$ で Σ 上の正則言語全体を表す. すなわち $R(\Sigma)$ は, Σ^* の任意の有限部分集合を含みかつ次の条件を満たす最小の集合 R である: $L, L' \in R$ ならば次の各条件がすべて成り立つ:

(1) $L \cup L' \subset R$.

(2) $L \cap L' \in R$.

(3) $\Sigma^* \setminus L \in R$.

(4) $L \cdot L' \in R$. (言語の連接)

(5) $L^* \in R$. (Kleene 閉包)

正則言語が有限オートマトンで受理される言語でありその逆も真であるあることはよく知られている.

5.5 有限オートマトンから正則等式系へ

有限オートマトン $\alpha = (\Sigma, I, x_0, \delta, Y) \in A(\Sigma, X)$ から正則等式系 $E_\alpha = \{b_x \mid x \in I\}$ を次の条件で定義する．ここで $x_0 \in I, Y \subseteq I$ とする．

(1) $x \in Y$ かつそのときに限り $b_x \ni \lambda$.

(2) $\delta(a, x) \ni y$ かつそのときに限り $b_x \ni (a, y)$.

この変換を φ とおく: $\varphi(\alpha) = E_\alpha$. この変換は正則等式系に'ルート'変数 x_0 を指定すれば正確に 1 対 1 もれなく対応する．このことから次の命題も明きらかである．与えられた正則式 (regular expression) r が表す正則言語を $L(r)$ とおく．正則式は帰納的に定義される．正則式に関する構造帰納法により，次の命題は容易である:

命題 7 r から $L(E_r) = L(r)$ となるような正則等式系 E_r を構成することができる．

命題 8 α を上述のオートマトンとして $L(\alpha) = \theta(x_0)$. ここで θ は正則等式系 $\varphi(\alpha)$ の (ユニーク) 解とする．

例 3 E_1 を，ルート変数 x を持つ正則等式系とする:

$$x = \{\lambda, (a, x)\}.$$

E_2 を，ルート変数 y を持つ正則等式系とする:

$$y = \{\lambda, (a, z)\},$$

$$z = \{\lambda, (a, y)\}.$$

Solution lemma より両正則等式系は解を持つ．また等式系 $x = y$, $y = z$, $x = z$ は bisimulation に拡張できることが定義から容易に確かめられる．ゆえに超集合として $x = y$ であることが分かる．実際，$x = y = \{a^n \mid n \geq 0\}$ である． □

一般に，変数 x, y の連接 xy はマージ演算の結果 $x * y$ を表す新変数，$\overline{x}$ はマージ変換 $\tau(x)$ を表す新変数という意味を持たせておけば両演算の性質から，次の例は分かり良い．

例 4 E_3 を，ルート変数 x を持つ正則等式系とする:

$$x = \{\lambda, (a, y)\},$$

$$y = \{(b, x)\}.$$

E_4 を，ルート変数 z を持つ正則等式系とする:

$$z = \{\lambda, (a, u)\},$$

$$u = \{\lambda, (a, u), (b, u), (b, z)\}.$$

E_4 を決定性正則等式系してそれを E_4' とおく．ルート変数は $\overline{z}$ である．

$$\overline{z} = \{\lambda, (a, \overline{u})\},$$

$$\overline{u} = \{\lambda, (a, \overline{u})), (b, \overline{uz})\},$$

$$\overline{uz} = \{\lambda, (a, \overline{u}), (b, \overline{uz})\}.$$

制約 $E_3 \cup E'_4$ に, 次の 4 つを加えて E_5 とおく.

$$
\begin{aligned}
x &\sqsubseteq z \\
\overline{u} &= \overline{uz}, \\
x &\sqsubseteq \overline{z} \quad (\sqsubseteq \overline{uz}), \\
y &\sqsubseteq \overline{u} \quad (\sqsubseteq \overline{uz}).
\end{aligned}
$$

E_5 は, 自明な閉包演算を加えて容易に support に拡張できることが分かる. したがって, x が z を subsume する: $x \sqsubseteq z$. 実際 $L(x) = \{(ab)^n \mid n \geq 0\}$, $L(z) = \{(a\alpha b)^n \mid n \geq 0, \alpha \in \{a, b\}^*\}$ であるから $L(x) \subseteq L(z)$ である. ここで, 便宜上, 変数でその解として得られる超集合を表している. □

6 おわり

主要な結果として, 超集合に基づく素性構造と subsumption の定式化およびその決定手続きが示された. また有限オートマトンの理論が超集合の有理木クラスの制約理論として自然に埋め込まれていることも示された. 正則言語の基本的重要性から推測して, これらの結果は計算言語学における素性構造の単一化理論, 知識表現言語の概念記述に関する推論, プログラミング言語における型推論などに応用を持つと期待される.

本原稿を書き終えて気がついたことであるが, 「拡張変数」のアイデアはマージばかりでなく, 他のタイプの演算にも一般化できそうである. ポイントは, ある代数規則で生成される拡張変数の全体が有限的に閉じていることである. この点は機会を改めて確認したい.

謝辞 素性構造の subsumption 問題を理解する上で J.Dörre 博士 (Stuttgart 大), F. Guenthner 教授 (Münich 工科大), B. Rounds 教授 (Michigan 大), および今村誠氏 (三菱電機) から有用なコメントや情報をいただきました. また, 超集合の計算言語への応用を進める過程で田中穂積先生 (東工大) からは多くの示唆と激励を頂きました. 記して感謝します.

References

[1] P. Aczel. *Non-well-founded Sets.* CSLI lecture notes Number 14. CSLI Publications, Stanford University, 1988.

[2] J. Barwise. *The Situation in Logic.* CSLI Lecture Notes 17. Standord: CSLI Publications, 1989.

[3] A. Colmerauer. Prolog and infinite trees. In S.A. Tarnlund, editor, *Logic Programming,* APIC Studies in Data Processing No. 16. Academic Press, 1982.

[4] J. Dörre. Feature logic with weak subsumption constraints. Technical Report IWBS tech. Report 101, IBM Deutschland,Stuttgart, April 1990.

[5] J. Jaffar and J.-L. Lassez. Constraint logic programmig. In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages,* 1987.

[6] M. Johnson. *Attribute-Value Logic and the Theory of Grammar.* CSLI Lecture Notes 16. Stanford: CSLI Publications, University of Chicago Press, 1988.

[7] R.T. Kasper and W. Rounds. A logical semantics for feature structures. In *Proceedings of the 24th Annual Meeting of the ACL, Columbia University.* ACL, 1986. New York, N.Y.

[8] M.D. Moshier and W. Rounds. A logic for partially specified data structures. In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages*, 1987.

[9] L.S. Moss. Completeness theorems for logics of feature structures. In Yiannis N. Moschovakis, editor, *Proceedings of The MSRI Workshop on Logic From Computer Science*. Springer-Verlag, 1990. to appear.

[10] K. Mukai. Coinductive semantics of Horn clause with compact constraint. Technical Report TR-562, ICOT, 1990. Presented at the second conference on Situation Theory and its Application at Scotland, September 1990.

[11] K. Mukai. Infinite tree unification in hyperset theory. Technical report, ICOT, 1991. 準備中.

[12] M. Schmidt-Schauss and G. Smolka. Attributive concept descriptions with unions and complements. Technical Report SEKI Report SR-88-21, FB Informatik, University of Kaiserlautern, West Germany, 1988. to appear in Artificial Intelligence.

[13] S.M. Shieber. *An Introduction to Unification–based Approaches to Grammar*. CSLI Lecture Notes No.4. CSLI publications, Stanford University, 1986.

[14] G. Smolka. Feature constraint logics for unification grammars. Technical Report IWBS report 93, IBM Deutschland GmbH, 1989.

[15] G. Smolka. Feature logic with subsorts. Technical Report LILOG Report 33, IWBS, IBM Deutschland, Postfach 80 08 80, 7000 Stuttgart 80, W. Germany, May 1989.

[16] H. Yasukawa and K. Yokota. Labeled graphs as semantics of objects. Technical report, ICOT, November 1990. SIGDBS& SIGAI of IPSJ.

[17] 今村 誠. 素性構造の単一化. 情報処理学会誌, 1991. 準備中.