人　工　知　能　77－20－3
（1991．7．18）

並列オブジェクト指向による制約処理

橋田 浩一　ICOT 第3研究室

1　導言

認知過程においては、認知主体よりも圧倒的に複雑な世界が与える多様な文脈に応じた多様な情報の流れが生ずる。従って、AIシステムの設計には制約(constraint)を用いるべきである。制約を用いるとは、情報の流れる方向を設計の段階で明示的に限定しないという意味である。予め指定された大域的な順序等に従ってではなく、制約の各部分における時々刻の文脈に応じて、情報の流れを動的に制御しなければならない。それには、制約の各部分が計算能力を持つ必要があろう。

そこで、並列オブジェクト指向を制約の実装レベルとして用いることを考える。つまり、メソッドまたはその一部として制約を用いるのではなく、制約の各部分をオブジェクトとすることによりオブジェクト指向計算のマクロなパターンとして制約の処理を実現する。ここで並列オブジェクト指向を用いるのは、オブジェクト間のメッセージ伝達に基づいて情報の流れを局所的に制御し、自然な並列分散処理を行なうためである。

本稿で扱う制約は論理プログラムとして表現され、ある力の場に置かれている。処理の制御はこの力の場に関する力学によって規定される。以下では、そのような処理が局所的な計算として定式化でき、従って並列オブジェクト指向計算として自然に実現可能であることを論ずる。

2　制約ネットワーク

節形式の論理プログラムとして書かれた制約をネットワークと見なす。たとえば、下記のプログラムは図1のようなネットワークである。

```
  (i) true :- member(a,X).
 (ii) member(A,[A|S]).
(iii) member(A,[B|S]) :- member(A,S).
```

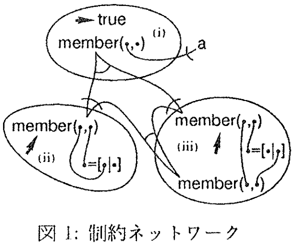

図 1: 制約ネットワーク

ネットワーク中の節点には、実要素制約(要素式と束縛)およびそれらの項の2種類がある。項はしばしば•で示す。各結線は2つの項の間を結び、両者の間の等式を表わす。実要素制約と等式を総称して要素制約と言う。定項への束縛(たとえば •=a)はその定項そのもの(つまりa)によって示す。図1に見るように、各々の節の中の要素制約で定項への束縛以外のものは1つの閉曲線で囲まれている。太い矢印は、それが指している要素制約を、その節が正リテラルとして参照していることを示す。

節外の結線は、2つの実要素制約の対応する項を結ぶ。2つの実要素制約の間に節外の結線があるとき、それらの述語(束縛 X=f(Y) の述語は =f とする)は等しく、かつ各々の対応する項の組が節外の結線で結ばれている。2つの実要素制約の間にある全ての節外の結線を束ねてあたかも1本の結線であるかのように示す。初期状態において、各述語に対してそれを持つ実要素制約の間の節外の結線は図2のような形をしている。左側は p が member

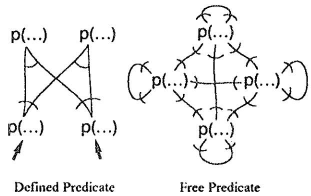

図 2: 節外の結線

のような定義された述語の場合、右側はそうでない述語(自由述語と呼ぶ)の場合である。図1においては、自由述語を持つ実要素制約 •=[•|•] の間の節外の結線は明示されていない。

一般に結線は、両端の項(節外の結線の場合はそれらの項を含む2つの実要素制約全体)が具現例(instance)を共有し得るということを表わす。結線の端点の近くに端点の方に凹んだ弧があるとき、その弧と交わる異なる結線はその端点の同一の具現例に言及していない。また、その弧が自分と交わる結線たちの外に突き出ていないときは、その端点の具現例は全てそれらの結線のいずれかによって言及されている。ちょうど1本の結線と交わり、かつ外に突き出ていない弧は明示されない。たとえば図1では、節(i)の member(•,•) の具現例は全て、節(ii)の member(•,•) または(iii)の上の方の member(•,•) の具現例でもある。ただし、この3つの member(•,•) に共通の具現例は存在しない。また、図1の場合には、節内の結線は、両端の項の具現例の集合が等しいことを表現している。

制約ネットワークの各節点と各結線をオブジェクトと考え、ネットワーク中で結合しているオブジェクト同士はメッセージを直接遣り取りできるとする。制約に関する離散的な情報処理は制約ネットワークの構造変換、つまりオブジェクトおよびオブジェクト間の通信路の複写および消去として定式化できる。次節で述べるように、制約の置かれた力の場に関するアナログ的な情報処理もある種のメッセージの伝達として実現される。

ここでは記号的演算として包摂化(subsumption)を考える。項 ι から項 ξ への等式 δ 上の包摂化は、図3に示したような処理である[1]。詳しい定義[1]は省略するが、包

[1]破線の弧は、それが凹んだ側にある項の具現例が全て、その弧と交わる結線のいずれかによって言及されていることを表わす。ξ と α および β の間の2本の結線は各々両端点の具現例の部分集合の間の1対1の対応関係を示しており、ξ と γ の間の結線は必ずしもそうではないとする。

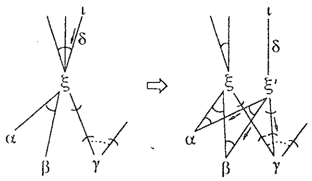

図 3: ι から ξ への δ 上の包摂化

摂化は制約ネットワークの具現化の可能性を保存する。

図3の中の細い矢印は探針 (probe) と呼ばれる。探針は原点 (origin) を持つ。原点は項であり、α を原点とする探針の後方の項は α に包摂 (subsume) される。項 μ が ν を包摂するとは、ν の具現例が全て μ の具現例であるとことである。図3からわかるように、包摂化によって探針の原点が包摂する項が増える。こうして項 μ が ν を包摂することがわかったとき、たとえばもし μ と ν が互いに単一化不可能な束縛を受けているならば、ν とその束縛を消去し、必要に応じて消去を伝播させる、などの処理が考えられる。たとえば Prolog における処理は、このような包摂化と消去の組合せである。Horn 節に限らない場合の処理も実現可能である。

包摂化は局所的な演算であり、図3の包摂化は、δ から ξ にメッセージを伝えて ξ を ξ' に複写し、さらに ξ に繋がっていて δ が言及していたのと同一の ξ の具現例に言及している可能性のある各々の結線にメッセージを伝えてそれらの結線を複写する、というオブジェクト指向計算として捉えることができる。ξ を含む実要素制約を複写することも同様に定式化可能である。また、原点からのメッセージが探針を通じて直接 ξ' に伝達可能であるとすれば、原点と ξ' が互いに単一化不能な束縛を受けている場合の処理も同様に扱える。

3 エネルギーと力の場

各要素制約は (0,1) 区間の実数である**活性値** (activation value) を持っており、さらに制約全体はそれらの活性値の関数である**ポテンシャル・エネルギー** U を持つとする。U の値は制約が満たされていない度合いを表わす。U は力の場 $-\nabla U$ を生む。$\vec{x}$ は活性値のベクトルである。また、その他に力の場 $\vec{f}$ があるとすれば、力の場全体は $\vec{F} = \vec{f} - \nabla U$ となる。

$\vec{F} = \vec{0}$ の安定平衡解を求めるために、$\vec{F}$ に従って活性値を変更してゆくことを**活性拡散** (spreading activation) と言う。活性拡散において一時に考慮すべき範囲は、$\vec{F}$ の各成分の各項が含む活性値を持つ制約の部分である。たとえば、¬p ∨ q. という節に U の項 $x(1-y)$ が割り当てられているとすると、$\vec{F}$ の y 成分は x という項を持つ。x と y はそれぞれ p と q の活性値であり、$x(1-y)$ の意味は、x か $1-y$ が 0 に近い、つまり p が偽か q が真だということである。このような局所的な力の処理は、オブジェクト指向計算として自然に扱うことができる。

一方、制約ネットワーク中のかなり広い範囲にわたる $\vec{F}$ の項もあり得る。たとえば、**依存巡回路** (dependency cycle) $\delta_0 \cdots \delta_{n-1}$ の持つ**推移エネルギー** (transitivity energy) は、δ_i の活性値 e_i で $e_i < \theta$ となるものが高々1つのときは以下の式で定義され、そうでないときは0である。

$$-E\prod_i (e_i - \theta)$$

$\delta_0 \cdots \delta_{n-1}$ が依存巡回路であるとは、$\delta_{i \bmod n}$ と $\delta_{i+1 \bmod n}$ が端点 $\alpha_{i \bmod n}$ を共有し、$\alpha_{i \bmod n}$ の同一の具現例に言及している可能性があり、かついずれかが弱い結線(自由述語を持つ要素制約の間の節外の結線で包摂化が行なわれていないもの)以外の結線である、ということである。E は正の定数、θ は (0,1) 区間の定数とする。各結線は、その活性値 x を含む最大の正の推移エネルギーと最小の負の推移エネルギーの和を U_T とすると、それによる力 $-\frac{\partial U_T}{\partial x}$ を受ける[2]。これは等号の推移律に相当する。

この力を求める計算も、実は局所的に行なうことができる。それには、各結線にその活性値を含む最大の正の推移エネルギーと最小の負の推移エネルギーの値を記憶し、さらにそれらの推移エネルギーを持つ依存巡回路を結線の間の関係の中に記憶しておいて、結線の活性値が変化するに従ってそうした情報を更新してゆけばよい。ここで、その情報を記憶しておくために各結線において必要な記憶領域の大きさは定数で抑えられる。つまり各オブジェクトの大きさは有界であるから、自然な局所的な処理が行なわれることになる。

推移エネルギーは、**結合エネルギー** (connection energy) と組合わさって様々な連想推論を実現する。結合エネルギーとは、2つの実要素制約の間の節外の結線の活性値が高いほど両者の活性値を近付ける力を生ずるようなエネルギーである。たとえば図4のような場合、もし項 A と

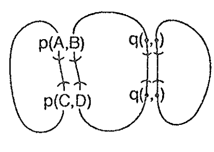

図 4: 統語的類似性に基づく連想

C を通る節外の結線を含む依存巡回路上の結線の活性値が揃って高いとすると、推移エネルギーによってそれらの活性値は一層高まり、この節外の結線の活性値も高くなるので、p(A,B) と p(C,D) の活性値が近くなる。すると、B と D を結ぶ節外の結線の活性値が高くなり易くなるので、この結線を含む依存巡回路の中の各結線の活性値も高くなる傾向が生ずる。従って2つの q(∘,∘) の活性値が近付き、同様の情報処理がさらに伝播してゆく。こうして、制約ネットワーク中の統語的に似ている2つの部分は、同様の活性化パターンを持ち易いことになる。

ここで詳述する余裕はないが、記号計算の優先度は、ある種の最急降下法によって定式化され、誤差逆伝播法の一般化に当たる手法を用いてやはり局所的な処理によって計算できる。そこにこのような連想が自然に反映され、こうして事例に基づく推論や類推が創発 (emerge) すると考えられる。

4 結言

並列オブジェクト指型計算によって制約処理を実現する枠組について述べた。現在 ICOT において、KL1 によるこの理論の実装が進行中である。

参考文献

[1] Hasida, K. (1991) 'Common Heuristics for Parsing, Generation, and Whatever ...,' presented at the Workshop on Reversible Grammar in Natural Language Processing, held in connection with the 29th Annual Meeting of ACL, Berkeley.

[2] 推移エネルギーは U には含まれない。