3E－3

制約表現を用いた空間的関係の獲得機構について

開 一夫， 安西祐一郎
慶應義塾大学

1. はじめに

空間的対象やそれらの空間的関係といった、空間に関する知識を計算機上においていかに表現し利用するかということは、人工知能の問題としてばかりでなく、コンピュータグラフィックス・コンピュータビジョン・CAD等多くの計算機科学の分野において重要な課題である。しかし、空間的関係の文脈依存性・漠然性等の性質から、空間に関する知識をアプリオリに記述することはかなりのコストを要すると考えられる[1]。設計支援システムを例にとると、設計対象があらかじめ厳密に分からないような場合、一般性・汎用性を持たせてシステムを作成することは一般に困難である。

これらのことから、空間に関する知識の記述を自動（半自動）的に獲得する機構の必要性が示唆される。そこで、ここでは空間的関係が代数的制約表現で記述できるという仮定のもと、left_of(X,Y), next_to(A,B)といった対象間の空間的関係を表す述語に対応した代数的定義を獲得するための方法について述べる。ここで述べる方法は、従来の機械学習にみられる意味ネットレベルの記述に対して処理を行うもの[2]とは異なり、これらの研究における意味ネット等の構造記述中の空間的関係を表したリンクに相当する部分に関する記述を獲得するものである。

2. 空間的関係と制約表現

近年、制約論理プログラミングというプログラミングパラダイム[3][4]が提唱されているが、我々は対象の形状モデルや対象間の空間的関係は代数的制約式を用いて表現できることを図の生成を例に示してきた[5]。制約論理プログラミングの特徴は、変数間の関係を宣言的に記述しておけば、変数の値を求める手続きを陽に記述しなくても良いとうい点にある。この特徴は、局所的な空間的関係の記述を宣言的に与えて、図に於ける全体の位置関係を求めるのに有効である。そこで、逆にある関係を表しているいくつかのシーンの事例から、その空間的関係を示す制約式を帰納的に獲得することができれば、関係を表すシーンを与えるだけで、制約プログラムが半自動的に生成されることになる。

図1は制約論理型プログラミング言語と制約表現を生成するConstraint Generatorとの関係を表したものである。Constraint Generatorは与えられたシーンの記述から制約式を生成する。この際、生成される制約式は、制約論理型言語のプログラムと考える事がでる。また、当然のことながらConstraint Generatorが生成可能な制約は、対象言語に埋め込まれた制約解消系(Constraint Solver)の機能に依存している。つまり、制約解消系が解くことのできないような、制約は生成されないものとする。現在、我々が開発した線形不等式・等式を扱うことができる制約論理型言語C-PIG[5]の制約解消系を、Constraint Generatorによって生成されうる制約表現に対して柔軟に対応できるよう修正・改良中である。

Constraint Generatorに入力される事例としては、画像のようなパターンも考えられるが、ここでは、初期処理が終了した段階における記述がConstraint Generatorに入力されることを仮定している。

一般に知識獲得システムあるいは機械学習システムにおいて、獲得された記述がはたす役割は大まかに分類タスクと再生タスクの2つのタスクに分けられる。分類タスクとは、与えられた事例があるカテゴリーに属すか属さないかを決定することであり、再生タスクは、獲得された記述からあるカテゴリーに対応した事例を生成することを意味する。こうした観点からみると、制約論理プログラミングの枠組みは、分類・再生タスクを実行するためのモジュールを別に用意する必要はなく、制約解消系にこれらの処理が埋め込まれているので知識獲得システムの構築を検討する上で適していると考えられる。

また、Geometric Reasoning[6]におけるヒューリスティクスの導入を考えても制約論理プログラミングのパラダイムは空間に関する知識を取り扱うのに適していると言える。

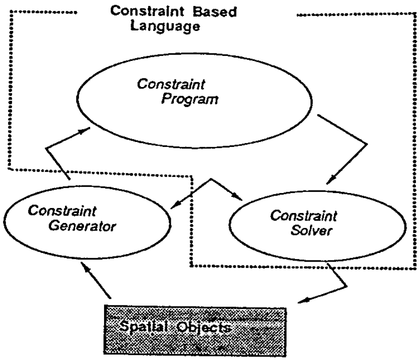

図1 制約論理型言語と空間に関する知識表現

3. 空間的関係の獲得機構

ここでは、特に取り扱う対象を極座標(γ, θ) $(0 \leqq \gamma \leqq \infty, 0 \leqq \theta \leqq 2\pi)$ 上の点に限定し、2つの対象間の空間的関係がγとθに関する代数的制約式で表される場合について、空間的関係をいかにして獲得していくか、その方法について述べる。

本稿で述べるシステムは、獲得したい空間的関係の正事例・負事例の有限集合から、距離及び方向に関する代数的制約式で表される空間的関係の定義を出力する一種の帰納システムと考えられる。具体的には、入力としてとして極座標上にお空間的関係Rを有する2点の組を要素とする正事例の集合：

$\{ (P_1, P_2), \cdots, (P_{i-1}, P_i), \cdots, (P_{2m-1}, P_{2m}) \}$

Aquisition of Spatial Relations Using Constraint Expressions

Kazuo Hiraki, Yuichiro Anzai
Faculty of Science and Technology, Keio Univ.

$P_i = (\gamma_i, \theta_i)$　$(1 \leqq i \leqq 2m)$
及び、空間的関係Rが成り立たない２つの点の組を要素とする負事例の集合：
$\{ (N_1, N_2), \cdots, (N_{j-1}, N_j), \cdots, (N_{2n-1}, N_{2n}) \}$
$N_j = (\gamma_j, \theta_j)$　$(1 \leqq j \leqq 2n)$
をとる。また、距離及び方向に関してそれぞれp, q個の制約式が生成されたとすると、出力として：
$$C_1 \cup C_2 \cup \cdots \cup C_i \cup \cdots \cup C_n$$
$C_i \in \{ (C\gamma_1 \cap C\theta_1), (C\gamma_1 \cap C\theta_2), \cdots, (C\gamma_1 \cap C\theta_q) (C\gamma_2 \cap C\theta_1), \cdots, (C\gamma_i \cap C\theta_j), \cdots, (C\gamma_p \cap C\theta_q) \}$
の形をとる制約表現をRの定義として出力する。ここで$C\gamma_i$、$C\theta_j (0 \leqq i \leqq p, 0 \leqq j \leqq q)$は、それぞれ距離、方向に関する制約式である。

正事例・負事例の集合から上のような空間的関係Rの定義を生成するため、次のような手順で処理を行う。
まず、正事例・負事例の集合の全ての要素から：
$DP = \{dp_1, \cdots dp_i, \cdots, dp_m\}$,　$GP = \{gp_1, \cdots, gp_i, \cdots, gp_m\}$
$DN = \{dn_1, \cdots dn_j, \cdots, dn_n\}$,　$GN = \{gn_1, \cdots, gn_j, \cdots, gn_n\}$
$dp_i = \| \gamma_{2i-1}, \gamma_{2i} \|$,　$gp_i = \| \theta_{2i-1}, \theta_{2i} \|$
$dn_j = \| \gamma_{2j-1}, \gamma_{2j} \|$,　$gn_j = \| \theta_{2j-1}, \theta_{2j} \|$
を生成する。次に集合DP, GP, DN, GNに対し、一般化を行い制約表現を生成する。以下に２つの空間的対象の距離を表すパラメータをγとし、γに関する制約式を生成するためDp, DNに適用される一般化規則を示す。

[1] $a \leqq b$, $a \notin DN$, $b \in DP$なる性質を満たすbの集合をB $(\subseteq DP)$とし、$B \neq \phi$ならば
$\alpha = \max(B)$として　$0 \leqq \gamma \leqq \alpha$　を生成。
[2] $c \leqq e$, $e \notin DN$, $c \in DP$なる性質を満たすcの集合をC $(\subseteq DP)$とし、$C \neq \phi$ならば
$\beta = \min(C)$として$\beta \leqq \gamma \leqq \infty$を生成。
[3] $f \leqq g \leqq h$, $g \notin DN$, $f, h \in Dp$なる性質を満たすf, hにおいて$o \leqq r \leqq f$, $o \in Dp$, $r \notin DN$, $f \leqq s \leqq t$, $t \in DP$, $s \notin DN$となるo, r, s, tが存在しないならば
$f \leqq \gamma \leqq h$を生成。

例えば、空間的関係next_toに対して正事例の集合：
$\{ ((0,0), (1, 7/4\pi)), ((0,0), (2, 1/4\pi)), ((0,0), (9, 3/4\pi)), ((0,0), (5, 5/4\pi)) \}$
負事例の集合：
$\{ ((0,0), (0.5, 5/2\pi)), ((0,0), (4, 3/2\pi)), ((0,0), (11, 1/2\pi)) \}$
が与えられたとすると、２つの対象の位置を表すパラメータ$(\gamma 1, \theta 1), (\gamma 2, \theta 2)$の制約表現として：
$((1 \leqq \| \gamma 1 - \gamma 2 \| \leqq 2) \cap (5/4\pi \ll \| \theta 1 - \theta 2 \| \ll 1/4\pi)) \cup$
$((1 \leqq \| \gamma 1 - \gamma 2 \| \leqq 2) \cap (3/4\pi \ll \| \theta 1 - \theta 2 \| \ll 7/4\pi)) \cup$
$((5 \leqq \| \gamma 1 - \gamma 2 \| \leqq 9) \cap (5/4\pi \ll \| \theta 1 - \theta 2 \| \ll 1/4\pi)) \cup$
$((5 \leqq \| \gamma 1 - \gamma 2 \| \leqq 9) \cap (3/4\pi \ll \| \theta 1 - \theta 2 \| \ll 7/4\pi))$
が獲られる。ここで、"$5/4\pi \ll \theta \ll 1/4\pi$"は$\theta$が$5/4\pi$から反時計回りに$1/4\pi$までの閉区間上にあることを示す。図２はこの制約表現に対応する領域を示したものである。

4.　制約表現の修正

３節で述べた手続きを適用することで、全ての入力事例を説明する空間的関係に対応した制約表現を生成することができる。しかし、先の空間的関係の例next_toでは、生成された制約式が図２に示すように同心円状に分断された形になっている。これは、制約表現を生成する際、与えられた負事例が、方向に関してだけでなく、距離に関しても影響を及ぼしていることに起因する。つまり、事例は極座標上の点として与えられるため、負事例の持つ距離と方向２つの属性のうちどれがターゲットとしている空間的関係の制約式に反しているか判定できないのである。このような問題に対応するため、次に示す規則によって質問事例(inquirring expamples)を生成し、質問事例の正・負をシステム側が能動的に問い合わせることにより望まれる制約表現を獲得することができる。

[1] DNの各要素$di (\in DN)$と、$C\theta_j$を満たす任意の値uから極座標上の点(di, u)を質問事例として生成し、これが正事例であればdiをDNから削除する。
[2] GNの各要素$gj (\in GN)$と、$C\gamma_i$を満たす任意の値vから極座標上の点(v, gj)を質問事例として生成し、これが正事例であればgiをGNから削除する。

先の例next_toの場合、上の規則で問い合わせを行うことにより、最終的に求められる制約表現は次のようになる。
$((0 \leqq \| \gamma 1 - \gamma 2 \| \leqq \infty) \cap (5/4\pi \ll \| \theta 1 - \theta 2 \| \ll 1/4\pi)) \cup$
$((0 \leqq \| \gamma 1 - \gamma 2 \| \leqq \infty) \cap (3/4\pi \ll \| \theta 1 - \theta 2 \| \ll 7/4\pi))$

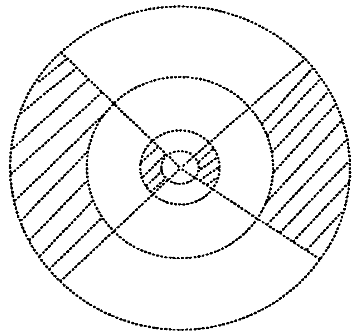

図２　next_toに対し生成される制約表現の領域

5.　まとめと展望

本論文では空間的関係の正事例・負事例から、その関係に対応した制約表現を生成する方法について述べた。生成された制約表現は制約論理型言語のプログラムとしてみることができ、制約論理型言語における自動プログラミングとの関連性も深い。今後、ここで述べた方法を基に、取り扱う空間的対象が構造を持つ場合に対応できるよう、Constraint Generatorの機能を拡張していく方針である。

[参考文献]
[1] 例えば Winston, P. H: Learning structural descriptions from examples, The Psychology of Computer Vision, McGraw-Hill, 1975.
[2] 開, 安西：　文脈に依存した定性的空間表現とその自然言語対話処理システムへのインプリメンテーション, コンピュータソフトウェア, Vol.6, No1, 1989.
[3] Lassez, C.: Constraint Logic Programming, IBM tech. report, March 16th 1987.
[4] 溝口, 古川, Lassez編：　制約論理プログラミング, 共立出版, 1989.
[5] 開, 安西：　空間的関係の表現, 日本ソフトウェア科学会第５回大会論文集, 1988.
[6] D. Kapur, J. L. Mundy (ed.): Special Volume on Geometric Reasoning, Artifical Intelligence, Vol.37, 1988.