

マルチエージェント自己認識論理に基づく図形知識の形式化

2P-01

山村 和寿†

外山 勝彦†

稲垣 康善†

†名古屋大学大学院工学研究科

yamamura@inagaki.nuie.nagoya-u.ac.jp

1 はじめに

図形は情報処理の対象として基本的である。従来、図形の表現や操作の種々の手法が開発されてきた。それらは、個別的かつ経験的に記述されていることが多く、論理的な操作や処理の対象にするのが難しい。

そこで本稿では、それらの問題を解決するために、多エージェント系自己認識論理 (MAEL) [2] を用いて、図形や図形間の関係、および図形操作に対する知識表現手法を提案する。本手法では、形式論理を用いることにより、図形に関する知識の表現に必要な基本概念の構造を体系化する。

図形処理システムの一例として Sync/Draw [1] を取り上げる。これは、発話入力と同時進行的に作図を行う図形エディタとして提案されている。Sync/Draw では、図形知識をオブジェクトと呼ぶ構造により表しているが、図形自体に関する知識と図形操作に関する知識が混在していたり、図形間の関係の表現が曖昧であるなどの問題がある。

本稿では、Sync/Draw のオブジェクト表現や書き換え規則と、本手法による表現とを対応付けることにより、Sync/Draw で扱う図形知識や図形操作の意味付けを行う。

2 図形と図形操作の表現

Sync/Draw で扱う図形には、点、線分、四角形、円などがあり、それらをオブジェクトと呼ぶ。オブジェクトはいくつかの属性とその値を持つ。例えば、点は座標という属性を持ち、その値は実数値である。また、線分は始点、終点などの属性を持つ。一方、図形に対する操作には、移動、拡大・縮小、生成、消去などがあり、Sync/Draw では、それらもオブジェクトとして扱う。例えば、移動は対象図形、移動の開始点、終了点などの属性を持つ。しかし、図形自体とそれに対する操作とを同じ次元で表現することは本質的に不自然であり、このままでは Sync/Draw の動作の解析や意味付けを困難にする。

この問題は、作図操作中のある時点における種々の図形間の関係や、操作実行前後において対応する図形間の関係を明確に表現することによって解決できると考えられる。ある時点における図形間の関係には、四角形とその一種である長方形の間や、点全般と具体的な点の間などに成り立つ IS-A (上位 - 下位) 関係、線分とその始点、終点の間などに成り立つ HAS-A (全体 - 部分) 関係がある。

そこで本稿では、次のような図形や図形操作の表現手法を提案する。まず、図形クラスおよび図形インスタンスのみをエージェントとみなし、図形の各属性が値を持つことは、各エージェントにおいて各知識が成立することである

と考える。例えば、点というエージェントは「x 座標は 1 である」などの知識を持つとする。

また、ある時点における図形間の関係をエージェント間の関係として捉える。例えば、線分の始点、終点も点であるので、それぞれエージェントとみなす。このとき、始点、終点は点に関する属性を継承する。そこで、IS-A 関係にある上位エージェントから属性を継承することをメタ知識として表し、それを下位エージェントに与える。

さらに、図形に対する操作は、操作前の図形と操作後に得られる図形を別エージェントとみなし、それらの間の関係として捉える。このとき、操作によって変化しない属性を表現するために、操作前のエージェントの知識は矛盾が無い限り継続されることを表すメタ知識を操作後のエージェントに与える。また、例えば線分を移動する場合、線分の始点、終点も移動するというように、HAS-A 関係にあるエージェント間には、「全体に対する操作は部分にも適用される」という性質が成り立つ。そこで、部分エージェントに対して、この性質を表すメタ知識を与えることにより図形操作を表現する。言い換えれば、図形操作の性質の表現を通じて図形間の HAS-A 関係を表現する。

3 拡張多エージェント系自己認識論理

前節で述べた手法を MAEL を用いて表現する。MAEL はマルチエージェント系に対する非単調論理であり、各エージェントが他エージェントとの間の知識獲得関係に基づいて推論する知識を形式化する。MAEL を用いた IS-A 関係や知識の状態継続の表現は [2] で示されているので、本稿では、HAS-A 関係および図形操作の表現を与える。

HAS-A 関係にあるエージェントや操作後に得られる図形を指すエージェントを表現するために、エージェントの表現を拡張する。図形の全体を指すエージェント *whole* に対して、その部分 *part* を指すエージェントを *whole.part* と表す。また、操作 *act* の実行前のエージェント *obj* に対して、操作実行後のエージェントを *act(obj)* と表す。例えば、線分 1 を指すエージェント *line1* に対して、その始点、終点を指すエージェントをそれぞれ *line1.start*, *line1.end* と表す。また、線分 1 を x 軸方向に 2, y 軸方向に 3 だけ移動した線分を指すエージェントを *move_{2,3}(line1)* と表す。

本体系では、命題論理において通常用いる記号に加えて、様相記号 *L*, ピリオド ' ', 属性記号 *start, end, ...*, 操作記号 *move_{a,b}, ...*, 原子エージェント記号 *line, point, ...* を用いる。また、エージェントとは、(1) 原子エージェント記号、(2) α, β , (3) $\gamma(\alpha)$ (α はエージェント, β は属性記号, γ は操作記号) のいずれかの形の表現である。論理式は、通常の命題論理式に様相演算子 L_α (α はエージェント) を加えて得られる表現である。

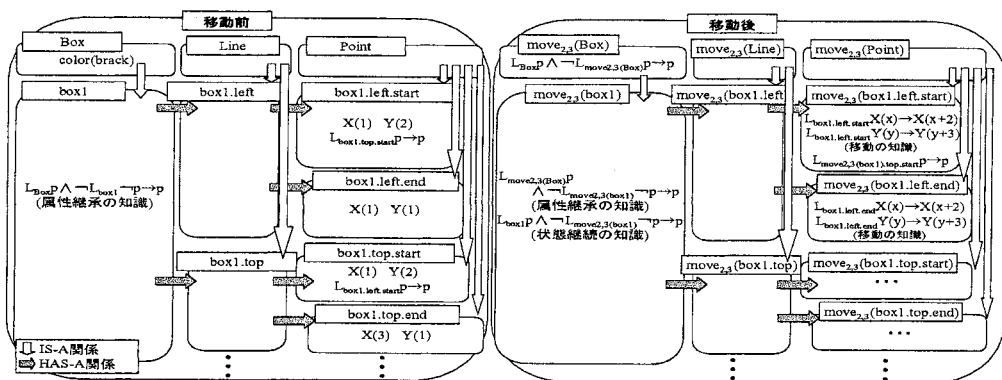


図 1: 知識表現例: 四角形の移動

4 知識表現例

図1に四角形 $box1$ の移動の表現例を示す。ここで $box1$ は、その部分として、左辺 $box1.left$ 、上辺 $box1.top$ を持つとする。まず、四角形、その辺、および辺の始点、終点を指すエージェントに、属性継承を表す知識を与えることにより、それぞれの上位エージェントとの IS-A 関係を表す。例えば、 $box1$ に属性継承知識 $L_{box1.p} \wedge \neg L_{box1}.p \rightarrow p$ を与える。これにより、四角形全般を表すエージェント Box に $color(black)$ (色は黒である) という初期知識を与えれば、 $box1$ も $color(black)$ という知識を持つことを推論できる。

また、エージェント $move_{2,3}(box1)$ にエージェント $box1$ の知識の状態を継続することを表すメタ知識 $L_{box1.p} \wedge \neg L_{move_{2,3}(box1)}.p \rightarrow p$ を与えることにより、移動操作によって変化しない知識 $color(black)$ を推論できる。さらに、移動後の四角形の左辺 $move_{2,3}(box1).left$ 、および上辺 $move_{2,3}(box1).top$ それぞれの始点、終点に移動を表す知識を与える。例えば、エージェント $move_{2,3}(box1).left.start$ に「移動前の x 座標に2を足す」を表す知識 $L_{box1.left.start} X(x) \rightarrow X(x+2)$ を与えることにより、四角形に対する移動操作が、その部分である左辺にも適用され、さらにその部分である始点にも適用される。これにより、移動操作を表すとともに、左辺とその始点の間、および四角形とその左辺の間の HAS-A 関係を表す。また、移動後の始点、終点の座標を具体的に推論することができる。

ここで $box.left.start$ と $box.top.start$ のように同じ対象を表すが、表現の異なるエージェントが存在するという問題が起るが、それに対しては、知識を互いに交換することを表すメタ知識をそれら2つのエージェント与えることで対処した。

5 Sync/Draw における表現との対応

Sync/Draw のオブジェクトには、図形自体を表すオブジェクトと図形操作を表すオブジェクトがあり、本手法の表現では、前者は図形を表すエージェント、後者は操作前後のエージェント間の関係に対応する。また、オブジェクトの持つ属性は、オブジェクトを属性値とする属性とそれ以外の属性値を持つ属性があり、前者は部分を表すエージェント、後者はエージェントの持つ知識に対応する。

また、Sync/Draw では、書き換え規則を用いてオブジェクトを更新することにより作図処理を行っている。書き換え規則には、オブジェクト追加規則とオブジェクト変更規則がある。オブジェクト追加規則には、(1) 図形自体を追加

するもの、(2) 図形の持つ属性とその属性値を追加するもの、(3) オブジェクトにある操作を施すものがある。例えば、(1) “ $\Rightarrow Line[]$ ” はオブジェクトが線分であることを規定し、(2) “ $\Rightarrow Color : black$ ” は色が黒であることを規定し、(3) “ $\Rightarrow Move : []$ ” はオブジェクトを移動することを表す。本手法の表現では、(1) は、エージェント $line$ に対応し、(2) は、エージェントが $color(black)$ という知識を持つことに対応し、(3) は、操作前後の2つのエージェントの知識の変化に対応する。

一方、オブジェクト変更規則には、(1) あるオブジェクトが別のオブジェクトの属性値となることを表すもの、(2) 描画、消去を表すものがある。例えば、(1) “ $Object : \phi \Rightarrow Center : \phi$ ” はオブジェクト ϕ が属性 $Center$ の属性値になることを規定し、(2) “ $\phi \Rightarrow Draw : \phi$ ” は、オブジェクト ϕ を描画することを表す。本手法の表現では、(1) は、2つのエージェント間に HAS-A 関係があることに対応する。また、(2) は、エージェント ϕ が可視であることを表す知識 $visible$ を持つことに対応すると捉える。

このように、Sync/Draw のオブジェクト表現および書き換え規則は、本手法の表現と対応付けることができ、これにより、Sync/Draw で扱う図形知識や図形操作を意味付けることができる。

6 おわりに

本稿では、MAEL に基づく図形知識の表現手法を提案した。本手法では、図形をエージェントと捉え、図形間の関係、図形に対する操作をエージェント間の関係と捉えることにより、図形知識の持つ構造を明らかにした。さらに、Sync/Draw のオブジェクト表現や書き換え規則と、本手法による表現とを対応付けることにより、Sync/Draw で扱う図形知識や図形操作への意味付けを行った。

他の図形表現手法に対しても、本手法による表現との対応付けができると考え、代表的な図形エディタから基本的な図形操作 36 種類を抽出し、それらの意味を本手法により記述した。

参考文献

- [1] 河口信夫, 松原茂樹, 外山勝彦, 稲垣康善: 発話同時理解に基づくマルチモーダル図形エディタ, 情報処理学会 音声言語情報処理研究会, 22-1, pp1-6 (1998).
- [2] 外山勝彦, 稲垣康善, 福村晃夫: 多エージェント系自己認識論理に基づく状態継続と因果関係の表現, 人工知能学会誌, vol.12, No.3, pp.466-474 (1997).