

「思考」をするプログラムを作っていく時、一番の問題はやはり、「自律とオンデマンド解析・推論」をいかに実現するかということですね。知識にない数式や図面の属性を、思考しながら把握していかなければなりません。

【例文】 xy 平面上に 3 点 $O(0,0), A(1,0), B(0,1)$ がある。

(1) $a > 0$ とする。 $OP:AP=1:a$ を満たす点 P の軌跡を求めよ。

(2) $a > 0, b > 0$ とする。 $OP:AP:BP=1:a:b$ を満たす点 P が存在するための a, b に対する条件を求め、 ab 平面上に図示せよ。

この問題 (1) は特にオンデマンド解析をする必要はなく、既存の知識 (円の公式) で解けます。問題は (2) です。

(1) 「点 P が存在するための a, b に対する条件」を求めろ・・・とから、そもそも、 a と b によって、2つの図形が描かれることを実際に図形を描いてその図の解析によって得ることになります。この2つの図形を持つという知識はもともとは無かったものです。例文 (1) からこの2つの図形は円であることが式の形から推論できます。これも、オンデマンド解析で行います (センサーですね)。

(2) 「存在する」は、文脈から、「2つの円が重なる」という解釈を連想しなくては成りません。東ロボの「含意関係認識」ですね。その能力が必要です。

(3) 円の中心と点 O が直角三角形をなすから、 a, b によって、2つの円の中心間の距離がピタゴラスの定理で求まることが分かります。これを求めればいいねという、自律的判断とオンデマンドの解析・推論が展開されねばなりません。この問題には中心点間距離を求めればよいという知識はないですから、自分で得るしかないのです。

(4) 円の中心点間の距離よりも、2つの円の半径の和が大きければ「2つの円は重なる」ということを、あらかじめ知っていると良いですが、なければ、試行錯誤で仮説検定して、この知識を得ることになります。で、その方程式を組むと、目標の立式になります。あとは一直線に不等式を解いて図示すればいいのです。

この例のように、オンデマンドで知識を獲得していくことが必須であることが分かります。オンデマンド解析・推論は将棋と同じようなアルゴリズムになると思います。いくつかの可能な手を解析・連想で得て、実際に試してみる。何手か推移を観て、目標にどれだけ近づくか (評価点を高くできるか) を評価して、最終手を見つける。図形の知識だと、直角三角形が重要とか、円の中心点と半径の関係とかが重要というような基本的な知識が

あって、例題の場合、どれが当てはまるか検証していくことになります。しらみつぶしになると思います。しらみつぶしですが、それは状況によらない、基本的で汎用な知識を調べればいいのです。知識をカテゴリーに分けて、できるだけ少ない範囲の知識をマッチングさせていくということも技術だとも思います。それは、実時間で動く実用的なシステムをつくるという視点で考えて行けば良いことです。

基本は、オンデマンド数式・図形解析と推論ですね・・・センサー機構です、これが重要。更に、将棋ですと、目標がはっきり一つです（王を詰ませる）。ですが、数学の問題の場合、目標は自分で設定しないとイケません。基本的な問いの文から、最終的な目標は設定されますが、途中の部分目標は自分で決定していかねばなりません。これも、連想からえていくことになります。（条件マッチング==>アクション==>評価）というプロダクションシステムで実現していくことになります。重み付け投票法で行くかも知れません。でも、既存技術です。

ということで、自律とオンデマンド解析・推論も既存技術でなんとかなりそうと言えそうです。

おわり