

プログラム作りということで、すこし考えてみました。オブジェクト指向という枠組みの強力さを頼もしく感じています。

1. データエリア

パターンマッチングして、マッチしていたら、コマンドプロセスを実行するというプロダクションシステムを想定しています。そこで、パターンですが、オブジェクト・・・知識オブジェクト（文章とかイメージとかの）の羅列でいいですね。Javaだとアレイリストにパターンとなる知識オブジェクトを登録していく。実際にマッチングに使う項目にはフラグを立てておくようにする。それで、完全マッチングをしていくだけの処理になります。しかも、通常の知識要素をパターンに持ってこれる。

それと、Prolog コマンドですが、普通のプログラムでは引数に変数があるとユニフィケーションがなされて書き換えられてしまいます。そこで、考える人工知能ではコマンドもオブジェクトですから、引数もオブジェクトなわけで、引数に値が設定されているときに、その旨のフラグを指定しておく必要があります。フラグが設定されていなければユニフィケーションする。それと、Prolog コマンドはワーキングエリアに結果を設定しますが、その結果を変数として返す機構のあった方が良いでしょう。F = f (x , y , z) ; のFのように。

それと、数式処理もオブジェクト指向ベースで行う必要があります。数式を操作したいからです。数式をオブジェクトとして持ち、かつ、その数式の各項とかオペレータもオブジェクトである。たとえば、比例乗数を数式にもって、その乗数を文章から得る・・・前の問題の解答から推論で得るなんてことがあります。数式をオブジェクトとして利用できないと、この問題は解けないことになります。

2. 連想プロセス

思考の根本は連想だと思います。それ以外考えようもないからです。しかし、連想はほっとくと無秩序にランダムに展開して行ってまとまりがつかないものに成っていきます。そこで、文脈（目標）に絞り込んだ連想を行うために、評価プロセスと密接に関連するでしょう。文脈で不要な連想をマスクし、評価でいかに目標に近づくべく作業しているかを保証する。そうして、連想間の相関も密にして、つじつまを合わせていくことも評価プロセスとしては重要な機能になっていくでしょう。

数学の問題も連想が重要と思うのですが、それよりも増して、生物とか地学などの応用

数学の世界でこの連想が重要になるのを感じます。連想による推論がシミュレーションと
いうか、問題を解くための数式とかデータがどんな傾向になるかという当たりをつけるこ
とが求められることもあります。当たりをつけてから方程式を作って解く・・・そんな高
度な連想処理が要求されます。そんな人間臭さも、あくまでも連想関係で解いていかねば
なりません。シミュレーションで得るのは、方程式の傾向評価だから、それもまたプロセ
ス（単調に大きくなる、単調に小さくなる、変曲点を持つ、振動する・・・とうとう）そ
んなことを解析するプロセスを用意することです。有限だろうから、機械学習で得て行け
るものだと思いますが、人手で作っていてもよいプロセスでしょう。

とにかく、連想と連想をマスクすることと、連想を評価すること、そうして目標に向け、
集中していくように制御していくこと・・・そんな機構を作っていく必要があります。

おわり