

## 1. はじめに

自律的で、自己組織化していく人工知能を実現するときに、必須となるコアの技術を追及して行きたいと思えます。コアの技術は、自律性と自己組織化能力を保証するだけの基本的な記号オントロジーとプロセスオントロジーを見出すことで得られます。自己組織化能力は学習によってより適切に外界に適応して行くことを示せれば、満足されます。

そこで、本稿では機械学習も含むコアオントロジーと、人工知能としてのアーキテクチャを考えて行きたいと思えます。

## 2. コマンドとイメージ

コマンドは次の形式をしたデータです。

(動詞[、格、名詞/属性詞]・・・)

動詞はコマンド本体で、行動、操作を示します。名詞や属性詞はオブジェクトとその属性です。格は動詞と名詞や属性詞の関係(名詞や属性詞間関係のときには動詞に「be」を用います)を示します。位相情報です。

格は基本的に記号ですが、時としてマップになります。方向は幾らでも細かに刻むことができ、これを表現するにはマップとして格を指定する必要があります。

コマンドは記号の世界の情報ですが、これだけでは人工知能は実現できません。人工知能として完全になるならば、イメージを基盤とした情報処理ができなくてはなりません。例えば、「空を見上げると飛行船が浮かんでいた」という自然言語での情報があつたとします。この文から、「飛行船から下を見ると地面と話者がいる」という情報を推論で得ることができますが、その推論処理はイメージの世界で実現されねば不可能です。

イメージは基本的に自由なオブジェクトの配置関係にありますからグラフで表現されます。配置関係が厳密な場合にはマップを用いて表現します。グラフのノードには必要に応じて動画の断片を配置することになります。このようにすると、時間変化する動画も表現できるからです。もちろんノードに他のイメージを貼り付けることもできるように成っていないとできません。

自然言語はイメージ空間を合成するコマンドとして機能し、イメージ空間でリーゼニングなどの推論を実現し、イメージ空間の断片から言語の発話が生まれるようにする必要があります。

### 3. オントロジー

オントロジーには、次のものがあります。

- (1) 動画像の基本図形とそれを解析して得るためのプロセス
- (2) 基本図形の属性と属性を得るためのプロセス
- (3) 基本図解の配置に関するデータとそれを得るためのプロセス
- (4) 動画像以外のセンサーデータのオントロジーとそれを得るためのプロセス
- (5) アクチュエータのデータのオントロジーとプロセス
- (6) 5W1H1Fのデータとそれを得るプロセス
- (7) 数学能力（代数、順序、位相、集合論、1対1対応）
- (8) 機械学習プロセス（パーセプトロン、ディープラーニング、交差法、強化学習、ネットワーク構築プロセス）
- (9) 自律的な行動を実現する、(1) から (8) 以外のもろもろのプロセスとデータ

オントロジーは唯一ということはありません。あるオントロジーセットを人工知能の基盤として設定しても、更に別のオントロジーセットに変えることができます。前のオントロジーを新しいオントロジーセットで表現すればよいのです。オントロジー上にコンセプトができて行きますが、そのコンセプトの中にはオントロジーのように情報のハブとなるものがあります。「生命」とか、「中心」とかのコンセプトです。オントロジーは情報のハブとなっているものとして定義できます。そこへ、他からの多くの連想があって、そこから他のコンセプトへの多くの連想があるものです。オントロジーセットには自由度があるのです。

ただ、基本オントロジーは身体（センサーとアクチュエータと人工知能機構のコア部分）に接地しているものになります。それが、上記の（1）から（9）なのです。

オントロジー体系は次のような議論がなされ得ます。

オントロジーが推論処理に大きく働きかけるのは評価のプロセスです。オントロジーは情報のハブになるからです。多くのノードからオントロジーに連想のパスが張られます。またオントロジーから行動や、解釈結果ノードへの連想が張られます。

そこで、処理の結果であるコンセプトを評価しようとしたときに、コンセプトと関連する評価項目であるオントロジー（メリットとかデメリットであるとか）が発火することが重要になります。

記号接地には自由度があり、要はセンサーとアクチュエータの値から内部的なイメージとイメージを解析するプロセスのセットへと対応させることなのです。内部的なイメージの形は自由ですし、イメージの体系を自由に設定できます。これらイメージの扱いは後で述べます。

オントロジーの詳細について考えて行ってみます。オントロジーコンセプトは次のようなものでしょう。

#### (1) 基本図形のオントロジー

- ・ オブジェクトとして、点、線、対向線、領域（輪郭線、部品）があります。
- ・ オブジェクトの属性として、コーナー/コーナーでない、相対的な大きさ、曲率、角度、対向線の属性（相対的な大きさ、変化率、対向線のなす角度）
- ・ オブジェクトの配置
  
- ・ これらをイメージから解析してコマンドを得るプロセス群があります。
- ・ コマンドからイメージを生成していくプロセスがあります。

#### (2) 5W1H1Fのオントロジー（Fはモダリティ）

- ・ 事象の把握にはこのオントロジーが必要です。事象があったら無条件にこのオントロジーが把握するように、解析プロセスが起動すべきです。原因・結果は時間シーケンスの関係を解析するとか、オブジェクトが誰であるかとか、何を対象にしているとか、基本的にイメージ基盤の解析オントロジーが動くことに成ります。

#### (3) センサー/アクチュエータオントロジー

- ・ 人工知能の外界を定義するのがこのオントロジー群です

#### (4) 学習機構のオントロジー

- ・ パーセプトロンとその拡張であるディープラーニング処理のオントロジーがあります。
- ・ 交差法を実現する、セット処理をするオントロジーがあります。
- ・ ネットワークを強化法で構築するプロセスオントロジーがあります。

#### (5) 認知機構のオントロジー

- ・ パターンマッチングオントロジーがあります。
- ・ ネットワークを走査するオントロジーがあります。
- ・ 部分空間を全件検索するオントロジーがあります。

#### (6) 思考の為のオントロジー

- ・ プランニングのプロセスオントロジーがあります。
- ・ 直観を実現するプロセスオントロジーがあります。
- ・ 目標管理思考を実現するプロセスオントロジーがあります。
- ・ 調停場を実現するプロセスオントロジーがあります。

(7) その他数学的能力などのオントロジー

- ・ 1対1対応
- ・ 開始
- ・ 終了
- ・ 中間
- ・ 変化/静止
- ・ 回転
- ・ 突破/衝突
- ・ 交差/交差なし
- ・ オブジェクトあり/オブジェクト無し
- ・ 含む/含まれる
- ・ 空/密
- ・ not/approve
- ・ 排他
- ・ 不完全/完全
- ・ 一致/異なる
- ・ 新規/既存
- ・ 順序
- ・ カウントアップ/カウントダウン
- ・ 各種位相（重力方向、その他の方向は全て学習によって獲得する。その基盤は3次元マップ）
- ・ 自律行動/他力行動
- ・ メリット/デメリット
- ・ 生/死
- ・ 原因/結果
- ・ 内部/外部
- ・ 閉曲線/開曲線

これらのオントロジーの間にも関係性が見られるが、解析としてはこのような冗長性もメリットがあります。

また、冗長性を持ってオントロジーとしてプロセスを用意しても良いし、より本質的なオントロジープロセスの結果を上位のオントロジーとして学習しても良いです。今回は、本質的なオントロジープロセスは何かということまでは追及していません。

オントロジーの基本は画素解析にあります。冗長性を、この画素解析から議論していきましょう。

(1) 原因/結果

**Concept** の時間シーケンス・・・近い距離に順序を持って生起すること・・・が意味解析プロセスになります。

(2) 不完全/完全

領域内の画素の分布ということで、集合論を実現して、密のとき完全、空白があると不完全とします。画素には **Concept** が対応しています。領域は **Concept** セットです。対応は 1 対 1 対応プロセスで実現します。

(3) 新規/既存

同じパターンの **Concept** が **Concept** セット内にあれば、既存です。一致するものが無い時には新規です。これも画素と領域の集合論で実現できます。

(4) 生/死

時間を追って解析します。画素がなかった領域に画素が生まれれば、生です。存在していた画素が無くなれば、死です。

(5) 開始/中間/終了

時間を追って解析します。線画素の上を端から端まで移動する画素を想定します。始めはその動きの開始点の画素であり、中間は線上の画素であり、移動が終わった端の画素が終了点です。

(6) メリット/デメリット

身体イメージを領域として、画素が増加する部分がメリットで、減少するのがデメリットです。

(7) 自律/他力行動

他から力が働いていないのに動いているイメージが自律で、他から力が働いて動いているのが他力です。

(8) not/approve

ベン図で2つの領域 **a** と **not\_a** を描きます。事象が **a** 画素に対応するならば、事象の **not** は **not\_a** に対応することで、**not** を定義します。**a** 画素の近傍に対応するならば、**approve** です。

この様に、画素解析アルゴリズムに同じようなことをしているオントロジー記号が複数あります。冗長なのです。

これらのオントロジーは実際に、どう使われるのでしょうか。次にそれを見ます。

文 a:「私はリンゴを食べた」と、文 b:「私はみかんを食べた」という場合、文 b は文 a の否定でしょうか、肯定でしょうか。「りんご」と「みかん」ではオブジェクトが異なりますから否定が言えますが、「私が食べた」では行為が一致しますから、肯定が言えます。

このように、文すなわちコマンドの引数 **Concept** の対応が、**not/approve** のオントロジーの画素と対応してくるのです。この対応を処理するのが、「1対1対応」プロセスです。

このように、オントロジーはセンサー/アクチュエータに記号接地しますが、コマンドもあるプロセスによって、オントロジーに接地されていくのです。この接地の連鎖が発火という処理です。接地は機械的で、そこから意味ある情報を得るのは、直観とか目標管理思考の役割です。そこでは、文脈管理をそていき（シーンイメージの構築をしていく処理）、**not/approve** だと、食べ物の議論下では否定を、なにかをしているという議論では、肯定を選択していくことになります。

オントロジー詳細の働きを次に示します。

「竿で木に成っている栗を取る」というコマンドの実現を考えてみます。

まず、栗を見て、栗が欲しくなります。シミュレーションして、手が届かないことを知り、栗を取ることができないことを認識します。

近くのものを見まわします。石、草、竿があったとします。それぞれ手で持って、栗に届くかみます。人工知能はものを掴むことを知っているとします。手の長さのオブジェクトと竿の画素列を繋げて、栗まで届くことを確認します。ここはオブジェクトの形状の認識は不要で、画素列の解析だけで済みます。届くと、「叩くと栗が落ちる」という知識の下、竿を持って栗を叩き落とします。

動画像解析についての技術について少し述べます。

動画像認識には次の大きなプロセスがあります。いずれもポテンシャル法が有効です。

- (1) オブジェクト画像を構成する部品と枠組みを同定するプロセスがあります。
- (2) コーナーと対向線の検出とそれらの属性の検出プロセスがあります。
- (3) 線の属性の解析プロセスがあります。

ポテンシャル法はニューロンによる重み付投票法とよく似ています。ポテンシャルをニューロコンピュータで実現することも可能でしょう。

イメージの処理について述べます。

コマンドでイメージを生成・合成したり、解析したりします。この時、コマンドは **Concept** を引数に持ちますので、イメージ処理の対象となる画素には **Concept** が対応していません。「イメージを見る」という操作で、特定の画素群に名前を付けて、さらに **Concept** を設定することになります。

イメージが円を持つかどうかをみる処理を考えてみます。イメージを解析して、閉曲線を持つことを得ます。閉曲線を持てばそれが円であるかを解析します。それらの結果は調停場に **Concept** として登録します。その後、調停場からその **Concept** を目標管理思考のプロセスが取り出し、 $x^{**2}+y^{**2}=a^{**2}$  とかの式を知識ベースから得て行くことになります。

#### 4. アーキテクチャ

アーキテクチャとして考えるべきことは次の事柄です。

- (1) 領野分散処理の実現
- (2) ニューロコンピュータとフォンノイマンコンピュータのカップリングの実現
- (3) 直観と目標管理思考と調停場という人工知能のコアとなるプロセスの実現
- (4) 機械学習機構の整備

領野分散処理を実現する基本はデータとかプロセスとかを領野に分散して、かつ、一つに統一したコンセプトとして管理できることではないでしょうか。それには、データに名前を付け、領野に分散していても同名のデータは一つのコンセプトの部分データであると見做せるようにするという事です。それには、名前をデータの格納された位置（アドレス）に領野毎に変換できる機構が必要になります。高速性が必要なので、HashMap によって行うのが良いでしょう。

情報には、キーと値といったデータと共起関係の情報があります。共起関係には基本的にデータ本体を必要としません。部分的なデータが本体から取り出せれば、共起関係は学習できます。そこで、共起関係をデータとは別領野で管理することが有効になってきます。しかも、共起関係は重み付投票とか連想関係という小さくて無数にあるという情報です。ニューロコンピュータで実現するのが効果的です。本体データはプロセスによる加工とか解析とかしますので、フォンノイマンコンピュータであることが有効です。そういうことで、ニューロコンピュータとフォンノイマンコンピュータのカップリングが提唱されます。

共起の学習にはパーセプトロンを要素とするディープラーニングの他に、交差法による、ノードセット管理というネットワークと単語や行動要素などの本体データの出現順序を学習して行くためのネットワークとがあります。それぞれ、別の領野に創っていくと良いでしょう。

共起関係により、知識ベースには位相が埋め込まれます。位相を表現するだけのためにネットワーク（グラフ）が創られてもよいでしょう。その位相関係を走査することによって、直観が実現できます。直観はノードをたどり、時には知識ベースを解析してデータを得ます。そのノードを評価してメリットがあれば、調停場にノードのアドレスを埋め込みます。直観は基本的にビックデータから有効なノードだけを取り出し、スモールデータとして調停場に格納し、意志による思考である目標管理思考プロセスに渡すことにあるのです。

なお直観の実現に部分空間の全件検索という手法もあります。



調停場は、またコンセプトからのコマンド発行に対応します。コマンド表現をコンセプトとして設立し（既存のコマンドであればそのコンセプトを利用する）、スケジューラプロセスに渡します。そして、思考として直観が展開していきます。

## 5. オントロロジーとアーキテクチャの保証する人工知能の完全性

自律性、自己組織化性の完全性を保証しなくてはなりません。その議論をしていきます。自律性と自己組織化性は次の3つの事象に現れます。

- (1) 認知とリーズニングの展開
- (2) 思考の展開
- (3) 学習の展開

人工知能は、外界に適用するために、外界の構成要素であるオブジェクトとか事象とオブジェクトの属性それとオブジェクト間の関係（格）を把握することと、オブジェクトとか事象間の共起関係を把握して行くこと、そしてそれらの情報から外界の状況を推論していくことを行います。共起関係は連想関係として、ニューロコンピュータで実現するとなりました。また、共起以外のオブジェクトとか事象に関するデータは解析をしたり、推論に利用したりするためにイメージを基本としてフォンノイマンコンピュータで実現するとなりました。

そのことを前提として、人工知能の全ての基本行動・操作・解析プロセスとデータを網羅しておけば、汎用人工知能として完全でしょう。基本プロセスと基本データはオントロロジーとしてあらかじめ用意することになります。そのような網羅性であることは可能か？という問題があります。永い数学の歴史で、数学能力の基本は代数構造、順序構造、位相構造と集合論と1対1対応であることが言われています。その数学を基盤とする要素オントロロジーは既に述べたとおり幾つかのもので網羅的です。あとは、人工知能のアーキテクチャが持つ要素コマンド群を網羅することです。人工知能のセンサーとアクチュエータの要素値は人工知能の設計によるわけで、可変ですが有限であることは論を待ちません。その他の人工知能のアーキテクチャは人工知能に共通であり、認知とリーズニング、思考、学習の各機能を実現するためのデータであり、プロセスです。それも大体のところは既に述べたところです。全てはイメージであるとなりました。だから、イメージの画素の間の対応と画素の動き、それと画素の間の関係を解析する・・・またパターンマッチング機構が実現できればオントロロジー群として網羅的であるといえて、そのオントロロジーから導出する上位のオントロロジー記号はすべて支持され、人工知能として万能であると保証できるこ

とが言えます。

纏めますと、人工知能はイメージを処理することとイメージからオブジェクトを得て、そのオブジェクトの関係を共起と数学的能力で把握していくことだと言えるということです。全ての基盤はイメージの画素の処理なのです。その処理が網羅的であることが万能性の保証です。

行動も重要な学習対象です。行動の要素は断片的なモデル予測制御の流れです。その組み合わせと、組み合わせの接合部の補てんする行動要素からなります。モデル予測制御部分群はスパースコーディングの基となっていて、学習対象となります。接合部の補てんは、「躍度最小モデル」とか「分散最小モデル」とかのアルゴリズムで生成することも、簡単にベジェ曲線近似しても良いでしょう。

## 6. 自動プログラミングへの展開

プログラムの構造は次のようなものです。

- (1) データエリアコード
- (2) データエリア操作コード
- (3) 目標の処理・操作コード
- (4) インターフェースコード
- (5) インターフェース操作コード

プログラムを創るときには、人工知能の思考処理が動きます。その基本は物事の動作イメージとその上のコマンド群の生成です。知識から有効な処理を構築するとき、自然言語に近い表現で目標を設定しなくてはなりません。だからコマンドが先ず創られるのです。このコマンドを数式や「if then else」とか「do while」といったコードに落とし込む作業があるのです。文脈や目標にたいする「考え方」は人工知能の思考処理でコマンドの操作として実現します。コマンドはコンセプトです。このコンセプトに対応して数式やコンピュータのコードが定義されているのです。この2つのコンセプト間のコードの対応を持って、コンピュータの言語に則った処理コードを吐き出していきます。

自動プログラミングは数学や物理学の問題を解くのと同じ処理です。コンセプトはコマ

ンド群ですが、人とのインターフェースは自然言語でしょう。数式もあります。図形もあります。自然言語や数式、図を解析し、コマンド群で課題を捉えます。それと同時に、内部イメージ（グラフとマップ）でも問題を表現します。コマンドとイメージを操作しながら問題を解いていくのです。プログラムを創っていくのです。イメージを解析して得られるのはコマンド表現ですから、これを数式に置き換える知識が必要に成ります。数式を操作する知識も必要になります。

知識では関連をもったコンセプトは近く、あるいはグループ（部分空間）を創っているでしょう。遠いコンセプト間はアクセス関連が疎になっていきます。知識ベースには位相が埋め込まれているのです。この位相をグラフとマップで表現します。パス数が小さいほど近いとします。マップ内で近いほど近い知識（関連が深い知識）とします。そのパスを手繰って、直観が働き、次々と解決候補を調停場に保存していくことが、思考の前処理で、知識ベースというビックデータから効率よく少数の効果的解答を見出す準備をします。目標管理思考が意志の下で、結果を評価しながら深いリーズニングをしていくのです。

## 7. 自律と自己組織化の実現

自律は、目標を持って行動することで、手詰まりの時には、知識ベースを全検索するか、手足をとにかく動かしてみることです。数学の問題や物理学の問題では、図形で問題点を表現するように試みて行く、という風に、デッドロックしないことが肝要です。ただし、なんでも闇雲に行動してはいけない。知識の近傍を維持することです。そういう制御が必要です。

自己組織化は、そもそも本稿で論じている人工知能はイメージを基盤にしていますので、外界をそっくり内面化しているものです。だから、認識能力（解析能力とパターンマッチング能力）が十分なようにプロセスを充実すれば、そのまま学習によって、外界を内面化していくことができます。すなわち、自己組織化はイメージベースの知識ベースとしたことで、保証されているわけです。

なお、イメージの解析プロセスの実現にあたっては、ポテンシャル法が有効であることは、前に述べました。

## 8. おわりに

自律と自己組織化を中心として人工知能を考えてきました。認識と行動と評価と学習というものがキーとなる技術です。キーとなる技術のコアとして、人間が作り込むものは何か、コアが学習して行き、外界に適応して行く仕組みはどうあるべきか、それがオントロジーであり、自律と自己組織化を保証するアーキテクチャです。

問題に特化して、プログラムを設計することも有効ですが、より一般的な人工知能の基盤技術を追及していくことは意義があることだと思われまます。それは、人工知能はどうユウ方向に向かって進化していくかを積極的に考えて行くことで、技術を見通しよくすることだからです。

今回はその目的にどれだけ答えられたでしょうか。更に精進していく所存です。

汎用人工知能は、目標に特化していないために、より一般的な行動要素の組み合わせとして問題解決にあたりまます。それは、産業ロボットのようにプログラムさえ変えればどんなことでもこなすという形をとるものだと言えまます。それぞれのプログラムは自律的にロボット自身が学習によって組み立てるのです。そのときに重要なことは、行動要素というものは作り付けとしてロボットに与えられるものであり、ロボットは、そこへには関与できないということです。それでも尚且つ汎用で何でもできるということは要素が網羅的で、どんな環境にも組み合わせだけで対応できるものであるべきということです。そのため、オントロジーとしての行動要素、アーキテクチャ要素というものの設定は重要な局面といえまます。そのとき役に立つのは人間自身の内省によるオントロジーの抽出です。人間ができることは汎用的な行動です。だから人間から抽出したオントロジーは完全であるといえまます。

オントロジー体系に完全性というものを持ち込むには理論化ということが有効に成りまます。数学的に厳密化する意義があるのです。本論でどこまで前進できたか、問われるところだす。

コーディングにあたっての留意事項を述べまます。

イメージ処理はメモリを多量に消費し、機械学習は超並行処理で・・・プログラムは小さいものからなる・・・GPUを多用すると思われまます。あとは、領野間の通信をどうするか・・・効率よいネットワークが要望されまます。

イメージ処理というかデータ処理は基本的に並行処理への要求は少ないと思われまます。直観については超並行処理になりますが、これは機械学習と同じネットワーク処理だすので、GPU領野で実現すると良いと思われまます。

おわり