

②⑥ テクニカルパネルディスカッション
「次世代応用システムの方向性を探る」

[パネリスト]

座 長 溝口 文雄(東京理科大学, 日本)
パネリスト Ronald J.Brachman(AT&T Bell Laboratories, 米国)
Catherine Lassez(IBM T.J.Watson Research Center, 米国)
Stephen Muggleton(Turing Institute, 英国)
濂 和男(ICOT, 日本)

[発表要旨]

溝口(司会): 本パネルディスカッションのめざすところは, 1. 知識表現, 2. 論理プログラム, 3. 機械学習, 4. 並列処理の各分野で探求されているパラダイムが, 新世代のアプリケーションにどのような将来性を方向づけるかについて, 異なった視点から議論をつくすところにある。

1. パラダイムシフトは, '70代AIの出発点であったKRパラダイム("Knowledge is Power")から始まったが, R.Brachmanは, 彼のKR言語Classicを中心に, 理論段階から実用に與するまでに取り入れた, さまざまなアプリケーションニーズと, 将来の方向性を論じた。

2. LP, PLPは, 1981FGCSが掲げたパラダイムであったが, 新しいドメインとして, CLPパラダイムが生まれた。C.Lassezは, OR・ロボティクス・定性物理などの分野で重要性を増している制約を, 論理プログラミングに導入したCLPの背景・適用分野・将来について論じた。

3. 知識獲得のボトルネックを解消する背景として, 機械学習と論理プログラミングを統合した新しい研究分野である帰納論理プログラミングが生まれた。S.Muggletonは, 彼の開発したGOLEMによる分子設計への応用などを論じた。

4. K.Takiは, PPパラダイムを, ハードウェア(multi-PSI, PIM)とソフトウェア(KL-1)の両面から論じ, FGCSプロジェクトの成果と, アプリケーション評価を論じた。

今後は, 次世代アプリケーションに向けて, これらのパラダイムを融合した基盤が必要になるであろう。

Dr.Brachman(first speaker): 知識表現は, AIの常に中心的存在であったが, 長い歴史の割に漠然とした煮え切らない部分がある。1980年代のブーム以来, 多くの基礎研究が進んだが, 潜在的にこなせない難しい問題も出ている。

1) 事実による実データか, KRシステムの表現で抽象化するのか 2) どのように真の要件を引き出すのか 3) いままでの"純粋な"KRの研究で十分なのか, について答えなくてはならない。

KR言語CLASSICはLisp, C Versionを経て, 現実のアプリケーションニーズ(実用に向けてのシステムサポート・実装上の考慮・一般ユーザへの考慮・特定アプリケーションニーズ)を取り込むこと

によって大きく進化した。

KRは、さらに今後大規模知識ベース・知識表現と機械学習・異なったアプリケーションで共通に使える上位レベルのモデル化（実在論）・KR標準化などに向かう。

Dr.Lassez(second speaker)：制約は、OR・ロボティクス・スプレッドシート・モデルベース推論を始め、AIの分野で重要な要素となっている。

論理プログラミングに制約を導入すると、自然な形でその意味論が拡張できる。

CLPスキームは、この制約解消とLP両パラダイムを含む枠組みの核となる概念設計である。

プログラミング言語CLP(R)を使った抵当における例では、一つの特徴として暗黙的(implicit)表現が可能のため、各パラメータ間の関連を示す式で記述される。たとえば他のパラメータが全部与えられているときの元金を計算する質問を実行することで制約解が得られるが、もう一つの特徴としてCLP(R)は、抵当の期間(T)と利率(I)だけしか与えられていないゴールでは、残りのパラメータを含んだ、関係を表わす解制約（関係式）が出力される。

このようにCLPによる問い合わせシステムでは、制約による宣言的記述から制約解消により実行時にさらに制約が生成されるため、より高水準な関係記述が可能でありこれを実現するアルゴリズムは唯一つである。

論理プログラミングに制約が導入(LP→cLP)されてCLPが生まれたが、今後はさらに制約パラダイムそのものが強力な枠組みになる(CLP→C)であろう。

Prof.Muggleton(third speaker)：ILPは、機械学習と論理プログラミングの2つの側の交点に位置する研究分野といえる。

ILP systemは、实例と背景知識から述語記述を作り出す。

演繹法と帰納法は、科学と哲学の世界で、長い間密接な関係を保っているといえるが、帰納法は、ニュートンの時代から、観察によって理論を構築するために用いている。

ILPはまだ生まれたばかりの新しいパラダイムだが、この帰納法を一般的に表わすと、ILP system Sは次のような仮説を構成する。

$B \wedge H \rightarrow O$

（ここで論理プログラムBは背景知識；Oは観察实例，Hは帰納的仮説。）

ILP systemは、このHを見出すのだが、これにはGeneral-to-specificサーチとSpecific-to-general(Inverse Resolutionに基づく)の2つのアプローチがある。

ILP system GOLEMは、薬の構造に関するルールをつくり出すことにより薬の調合やデザイン分野でその高い精度が実証された。

Dr.Taki(forth speaker)：PPは単一のパラダイムではなく、並列プログラミング・マッピング・OSや言語の設計/実装などの多くのパラダイムセットをまとめたものである。

PPシステムの要件は、1) 並列プログラムの記述力 2) 容易なデバッグ 3) 非決定性プログラム記述能力 4) 柔軟な負荷配分である。

核言語KL1は、並列論理プログラミング言語として、以下の特徴をもつ。

1) データフロー同期 2) 小粒度の並列プロセス 3) 非決定性 4) 実用的な負荷配分とゴールスケジューリング（マッピングを意識しない並列性記述） 5) 荘園構造による効率化。

並列推論マシンPIM上で、多くの実用的なアプリケーション(LSI-CAD: Logic Simulator・

Genome analysis: Protein sequence analysis・Legal reasoning・Go game playing system)が実装され、PPによる高速化が検証された。

[質疑応答]

K.McAloon: まず溝口さんには、どうして最初のスライドに皆さんの話の内容をうまくまとめることができたのかお聞きしたいところです。それはさておき、Brachman博士にいくつかお尋ねしたいと思います。最初はプログラム言語レベルの質問ですが、CLASSICシステムのLispからCへの移行は容易だったか、CはKRシステムにふさわしい言語か。

Brachman: CLASSICはコンパクトなシステムなので、LISP,C言語とAI両方に精通した人が1人で容易に行なえた。

大型の商用システムでは、全体に対するKRの部分が少ないので統合しやすさからC Versionを使った。

McAloon: KRの一般性や汎用性(one size fits all)についてはどうか。

Brachman: 一適格な答はないが、CLASSICの経験から話すと、このようなシンプルでポータブルな言語は、KRシステムをもたないアプリケーションでメインプログラム言語からも、簡単にリンクして呼び出す操作ができる。つまりシンプルさが本来持っている有用性を使っている。多くのやっかいなアプリケーションから集約された問題に有効な機能をCLASSICの基本的なセマンティックを壊さずに拡張していくことによってこの中で人々が一番使っているフィーチャを拡張していくという手法をとり、一般化が進む。

溝口: 次の方向は何かということについて、あまり議論してこなかったが、あえて4名のスピーカーと私の話を一言でまとめるとすれば融合(fusion)という言葉を使いたい。それはさておき、KRの伝統においては、標準化という方向がある。これは、KIF(Knowledge Interchange Format)とEDR(Electric Dictionary Project)に見られる。一方、LPにおいては、次の方向としてパラダイムの融合がある。Parallel CLPではGDCC, PEP sys.KR/Parallel Data Managementでは, Quixote Kappa-P, Object orientationではShapiro-Takeuchi他いくつかの研究, CLPとILPの融合ではPage&Frisch, Mizoguchi&Ohwada, Parallel ILPでは, C.L.LingのKL1を用いたparallel learning systemがある。

これらの統合あるいは融合の流れは、今回のパネルディスカッションのたどりつくところをしめしている。

McAloon: Lassez博士にお尋ねしたい。論理式のもつ特性を、実数演算・幾何・プール演算プログラミングの分野でも使えるようにするLAZY PROGRAMMINGは、あなたにとって役に立っていますか？

Lassez: 一線型代数という限られた分野の例では、このパラダイムは大変有用と思われるが、他の多くの分野についても多分同様に思う。

McAloon: つづいて、Muggleton博士にお尋ねします。機械学習とILPとの関係について、さらにコメントをいただけますか？

Muggleton: 機械学習では、概念を説明したとおり、仮説をつくり、信念に付加され正当化される

が、これに多くの事実を使っている。新薬設計の例では、今まで設計者は多くの薬の副作用があるかを、ねずみで実験するしかなかった。

McAloon：瀧さん、並列性のコントロールはユーザに委ねたらいいのかプログラマにまかせたらいいのかについてどう考え。

瀧：プログラマは、並列性全体をコントロールする必要がある。現在はアプリケーションのために、ユーザとシステムプログラマ両者が共同で見なければならないが、将来はライブラリやPIMOSの負荷配分システムのように、システムがサポートするようになる。

Motoda：溝口さんのスライドを見るまで、スピーカーの皆さんの話がどう関連してくるのか分からなかったけれども、たぶん私自身の考えはBrachman博士の提起されたもの特に最後のスライドに示されたものに一番近いと思う。そこで、そのことについて質問をしたいと思う。

現在のシステムは知識ベースが貧弱であるためにシステムが貧弱といえる。巨大な知識を入れ込む作業が問題となり、そこで機械でつくることを考えるわけだが、現状では、単調で地味な作業である。この点とさらに人間は、例えば、絵で他人に説明するように、通常は同時にビジュアルな推論・記号推論などを行なっている。この同時に異なった推論を行なう重要性について各人のコメントをいただきたい。

溝口：人々はともすると理論を忘れて現実のアプリケーションばかり考えるが、これらの理論がしっかりしていなければシステムはうまく動かないだろう。今回のパネリストの多くのパラダイムを使えば、次世代のアプリケーションを健全な状態にもっていける。これが今回のパネルディスカッションの我々のゴールである。

Brachman：同感である。常識を知識ベース化するのには、10年のスパンが必要だが、ポイントは、日常生活という拡大な範囲を考えずに、まず工学の立場というふうにしぼって生産性を上げたことがある。しかし現実には巨大な知識を操作しなくてはならず、これは今後のテーマである。

Lassez：数学の分野は現実のアプリケーションからすると限られた狭い範囲なので、あまりコメントできない。

Muggleton：これは機械学習にとっては、多大な責任を課せられる問題提起であり、他のパネリストと同様に感じている。特に人間が巨大な知識を直接扱うには、分子構造を3Dで表現するような、視覚にうったえる方法が有効であった。

瀧：PPparadigmは大規模データベースをハンドリングするパワーそのものを与えてくれる点とさらにPPを使う側に意識させないようにすることでこの問題に貢献できると思う。

会場からの質問(P.Wegner)：それぞれのアプリケーションの中で、Prologの役割は必須なのか、多くのパラダイムの1つなのか。

Brachman：特にPrologに制限されないが、ホーン節はパワフルだと思う。第5世代プロジェクトのQuixoteは確かPrologではないと思う。

Lassez：Prologのクリーンな使いやすい特性を計算モデルとしてCLPに使ったが、単一パラダイムですべて問題解決できるとは思わない。

CLPスキームは拡張されて、その先Prologは消えてなくなるわけではない。他のパラダイムも使えると思う。

Muggleton：機械学習の中心は、ニューロのように必ずしもロジックを必要としないが、ロジックを使った場合、コンセプトがよく見えて理解できる点が重要である。

瀧：並列ロジックプログラミングは並列性に必須かという質問だが、KL-1は並列性などが加わって、より一般的な目的の言語になってきた経緯がある。並列性やマッピングの記述はGHC文法からは分離している。また、細粒度なオブジェクト指向言語は非決定的なものをハンドリングできる。

溝口：PrologはFGCSにおいて知識情報処理、並列処理、知識に対する論理的な見方の間のギャップを橋渡しするという大きな役割を果たしたと思う。全ての研究者もおそらくそう考えていると思いますが、ここで、次世代応用の将来方向を定めるような単一のパラダイムはないということをパネルの結論として挙げておきたいと思います。パネルに参加して下さった皆さんに感謝します。