

第五世代コンピュータ・プロジェクトの概観

An Overview of the Fifth Generation Computer Project

古川 康一*
Koichi Furukawa

* (財)新世代コンピュータ技術開発機構
Institute for New Generation Computer Technology

1989年3月7日 受理

Keywords: fifth generation computer, knowledge information processing, parallel architecture, logic programming.

1. はじめに

第五世代コンピュータ・プロジェクトが発足して以来すでに前期3年、中期4年の計7年が経過し、今年度から最終フェーズの後期が始まっている。本稿では、プロジェクトの発足からこれまでの研究活動を振り返って概説したい。また、残された後期3年の研究課題についても述べてみたい。

本プロジェクトは、1982年度に発足したが、その準備として3年をかけて大がかりな調査研究を行っている。その目的は、1990年代の汎用コンピュータがどういったものであるべきかを、機能およびアーキテクチャの両面から探ることであった。その結論は、「知識情報処理を主なアプリケーションとする超並列コンピュータ」であった。「知識情報処理」は、その後起こったAIブームを予見したものであり、「超並列コンピュータ」は、莫大な処理能力を必要とする高度なAIシステムの開発にとって、必要不可欠な道具である。この二つの課題は、同時に達成することが大変困

難であることがよく知られていたが、我々は、ロジック・プログラミングの採用によってその困難性が解決できる、という作業仮説を設定した。それは、ロジック・プログラミングが知識情報処理システムの実現に適しており、しかもその実行過程が容易に並列化できる、という見通しを持っていたからである。この考えを図示すると、図1のように表すことができる。本プロジェクトのこれまでの活動は、この図を精密化することであったといえるであろう。以下に、前期および中期の研究成果を概観し、最後に今後の課題について述べることにしよう。

2. 前期3年の研究成果

プロジェクト開始当初、実用化されていた論理プログラミング言語はPrologだけであったので、我々はまず知識情報処理に対するPrologの妥当性を検証することから始めた。その結果として、プログラミング言語としてのPrologの潜在能力を引き出すことに成功し、メタ・プログラミングや部分計算といった新たなプログラミング技術を開発した。さらに、Prologによる応用システムの開発を続け、VLSI-CADシステム、自然言語処理システム、プログラム検証システムなどの高度なAIシステムの開発を行った。これらのソフトウェアの研究と並行して、我々は論理プログラミングのための専用コンピュータの開発を行った。それがPSI (Personal Sequential Inference Machine) である。

前期研究のもう一つの柱は、言語の並列化である。Prologは、確かにAI応用プログラムの開発に適しているが、並行プログラミング機能の欠如により、オ

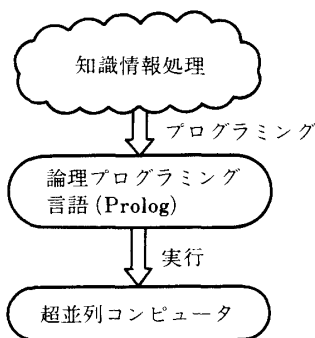


図1 プロジェクト当初の第五世代コンピュータの図式

ペレーティング・システムなどの並列事象を含むシステムの記述が困難である。この欠点を補うために、ESP という言語が ICOT の近山によって開発された。それは、Prolog と Smalltalk を合体させたような言語である。この言語を用いて、PSI のオペレーティング・システムや VLSI-CAD のような応用システムの開発などが行われた。並列事象の記述と共に、プログラムの並列実行を念頭に置いた言語の検討が、これと並行して行われた。我々が検討の対象とした言語は、Keith Clark らによって提案された Relational Language, それを発展させた PARLOG および Ehud Shapiro によって提案された Concurrent Prolog の三つである。これらの言語の詳細な検討の後、上田によって提案された並列論理プログラミング言語が GHC である。これらの言語の特徴は、通信および同期を取りながら並列に実行する複数のプロセスが記述できる点である。GHC は、これらの言語のうち、構文的にもっとも単純であり、実際の並列実行にも適している。この言語の開発は、本プロジェクトのもっとも優れた成果の一つであると言えるであろう。

3. 中期の研究成果

プロジェクトの中期は、1985 年度から前期に引き続いて開始されたが、中期の研究を一言で言えば、GHC に基づく並列処理の研究とすることができる。より正確には、我々は、GHC のサブセットである FGHC (Flat Guarded Horn Clauses) を、プロジェクトの中核言語として選んだ。そして、FGHC を使いこなすためのソフトウェア研究と、FGHC で書かれたプログラムを実際に並列実行するためのハードウェアの研究開発を始めた。

ソフトウェア研究は、FGHC で種々のプログラムを書くことからスタートした。その結果、FGHC が十分に強力なアルゴリズムック言語であることが確認された。分類や探索などの基本的なリスト処理プログラムは、容易にプログラムをすることができた。ところで、並列論理プログラミング言語は、並行処理の機能を実現した代償として、もともと Prolog が持っていたバックトラックによる全解探索機能を失ってしまった。このため、例えば「条件 $P(x)$ を満たすようなすべての x を求めよ」といった問題を簡単にプログラムで表すことができなくなった。この新たな問題を解決するために、いろいろなアプローチがとられた。その一つは、Prolog プログラムをコンパイルする方式である。与えられたリストのすべての分割を求める

ようなアルゴリズムックな問題は、この方式でうまく取り扱えることがわかった。もう一つは、FGHC の特性を生かした全解探索のための新たなプログラミング・スタイルの開発である。その方法によって、並列構文解析などの複雑な処理がプログラム化できることが示された。また、この新たなプログラミング・スタイルは、問題に内在している並列性を最大限引き出し得ることも示された。

プログラミング技術に関連して、我々は、与えられたプログラムを等価でより効率の良いプログラムに変換する技術の開発も行ってきた。Prolog では、unfold/fold 変換が等価性を保証することはよく知られているが、FGHC の場合は、不用意な変換を行うとデッドロックが生ずるといった問題のあることがわかってきた。この問題を回避して、常に等価なプログラムに変換するような変換規則が ICOT で提案された。

この変換規則を用いて、FGHC プログラムの部分計算器が開発された。そして、実際に並列構文解析のインタプリタを与えられた構文規則に対して(部分計算により)特殊化して、その結果、処理速度を約 5 倍高速化することに成功した。

プログラム変換によって、実際に等価なプログラムに変換されることを示すためには、何をもって二つのプログラムが等価なのか、その基準を示さなければならない。そのための研究として、FGHC のプログラムの意味を形式的に扱う研究が行われた。Prolog プログラムの意味は、それを一階述語論理式と考えて、その式の最小モデルによって与えることができた。しかしながら、FGHC などの並列論理型言語では、オペレーティング・システムのように止まらないプログラムでも意味のあるプログラムが存在するので、最小モデルによる形式化はできない。我々は、外界との情報のやりとりが起るまでの単位をトランザクションとしてとらえ、その単位での振舞いによってプログラムの意味を定め、可能な振舞いの集合を変えない場合には、その変換を等価であると考え、そのような意味で、上記のプログラム変換規則がプログラムの等価性を保つことを示した。

FGHC プログラムの意味に関しては、このほかにも、すべての可能な計算の軌跡をドメインとし、その上での最小モデルによってプログラムの意味を定める、というアプローチを採って研究を進めている。我々は、その考えを応用して、有限時間内に失敗するプログラムの性質を形式化することに成功した。

我々が実用言語として開発しているのは、実は FGHC ではなく、そのスーパーセットで、KL/1 と

呼ばれる言語である。KL/1 は、3層からできている。最上層は、KL1-U と呼ばれ、アプリケーション・プログラムを開発するためのユーザ言語である。中間層は、KL1-C および KL1-P の組である。KL1-C は、KL1 言語族の中核であり、FGHC にメタ機能を付加したものである。メタ機能は、特に重要なものであり、オペレーティング・システムをユーザ・プログラムのデッドロックや失敗から保護する機構を実現するのに用いられている。KL1-P は、プラグマ言語の意味であり、どのゴールをどのプロセッサで実行するかを指定したり、各ゴールにどの程度の計算資源を与えるかの指定を行うための言語である。最下層の言語は、KL1-B と呼ばれ、(KL1-C, KL1-P) の組の処理系を実現するための抽象機械語である。

KL1 の並列実行のためのハードウェアは、いくつかの段階を設定して開発が進められた。始めは、前期に開発した PSI マシンの上に KL1 の処理系を作り、それを6台つなげて動かした。これは、Multi-PSI V1 と呼ばれている。次に、マイクロコードによって KL1-B の処理系を記述し、そのようにして作られたプロセッサ (KL1 マシン) を 64 台つなげて、より本格的なマルチプロセッサ Multi-PSI V2 を開発した。プロセッサ間の結合トポロジーは、メッシュ結合を採用した。また、パケットの自動転送を高速に行うために、専用の LSI の開発も行った。現在、Multi-PSI は、オペレーティング・システムの核部分の開発が終了し、デモンストレーション・プログラムが動いている段階である。昨年 11 月 28 日から 12 月 2 日まで行われた国際会議 FGCS '88 で、4 種のデモンストレーション・プログラムが公開された。

Multi-PSI のオペレーティング・システム (PIMOS) は、ユーザ・プログラム・コードの記憶管理、計算時間や入出力デバイスなどの資源管理、ユーザ・プログラムとの通信機能のサポートなどを行う。PIMOS 自身 KL/1 で記述されており、それは今まで開発された KL1 のプログラムの中で最大のものである。KL1 によって PIMOS が効率良く実現できたことは、KL1 の表現力の豊かさを示すものである。

Multi-PSI の開発と並行して、より高速でかつ大規模な並列推論マシン PIM の開発が進められている。PIM は、Multi-PSI と異なり、構成要素のプロセッシング・エレメント自身の VLSI 化を図っている。PIM は、開発要素が強く、特にプロセッサの結合トポロジーに関しては、問題に依存して最適なものが異なると予想されるので、数種類の結合方式の実現を考えている。それらは、大きく分けて、階層構造と、

単層のメッシュ構造の二つがある。階層構造は、クラスタと呼ばれる最下層と、それらを結合する上位層とからなる。クラスタは、メモリを共有する 8 台のプロセッシング・エレメントからなり、強力な一貫性キャッシュ機構による高速並列処理を目指している。上位層をなすクラスタ間の結合は、ハイパキューブ結合、およびクロスバ結合の 2 方式を予定している。最終的には 1000 台のプロセッサを結合した並列プロセッサの開発を目指しているが、当初は 128 台規模のマシンの開発を 1990 年の頃までに完成させることを予定している。

並列推論マシンと共にハードウェアのコンポーネントを成す知識ベースマシンについては、前期は関係データベース・マシンの開発を行ってきたが、辞書項目や数式処理における公式集などのより複雑な知識を格納するためには表現力を強化することが必要になり、構造データを扱えるようなシステムの開発を行った。構造データとしては、Prolog に現れる項をもその対象とし、関係演算を同一化を基にした演算 (unification join など) へ拡張した。さらに、複数のプロセッサからの共有メモリへのアクセス競合を回避することを目的として、マルチポート・ページメモリの開発を行った。

知識ベースに関連する研究としては、このほか、KAPPA と呼ばれるデータベース管理システムの開発を PSI 上で行ってきた。KAPPA の特徴は、変数を含まない構造データを非正規関係として形式化して、その上で必要となる演算を実装した点である。理論的な研究としては、否定を伴う再帰的な質問を正しく、しかも効率良く処理するアルゴリズムなどの研究も行われた。

知識情報処理システムを開発するための高水準言語として、我々は、制約論理プログラミング言語に注目し、CIL および CAL と呼ばれる二つの言語を開発した。CIL は、特に自然言語処理の研究のために開発され、談話理解システム DUALS の開発に用いられた。CIL の特徴は、引数の数の定まらない複合不確定項と呼ばれるデータ構造を導入したこと、および二つの複合不確定項に現れる特定の引数どうしが満たすべき条件を制約として記述できること、の二つである。DUALS で用いられている用語辞書は、CIL を用いて記述されている。特に、複合不確定項は、フレームのような構造を表現するのに適しており、その機能を利用して辞書が開発された。

談話理解実験システム DUALS 第 3 版は、これまでの成果を踏まえて、談話理解能力の一般化を目指して研究開発が進められた。そのため、対象文に依存し

た発見的な処理手順はできるだけ排除され、一般的な問題解決能力のみを手掛りに談話文を理解するシステムが開発された。その一般問題解決戦略として、制約解消が採られた。

談話理解実験システムと共に、LTB (Language Tool Box) と呼ばれる自然言語処理のための共通ツールの開発が並行して進められた。それらは、形態素解析プログラム、構文解析プログラム、意味解析プログラムなどからなる。これらのうちの幾つかは、並列アルゴリズムの開発も行われた。

知識情報処理システムの一つとして重要なものに、エキスパート・システムがある。我々は、本プロジェクトのアプローチの正当性を示すために、幾つかのエキスパート・システムを開発してきた。それらは、電子装置の故障診断システム、VLSI-CAD システム、計算機室レイアウト・システムなどである。これらのエキスパート・システムは、具体的な問題を対象としたものであるが、それと並行して、より一般的なツールの開発も進めている。その一つが、設計支援システム構築ツール MECHANICOT である。MECHA-

NICOT も談話理解システムと同様制約解消を問題解決戦略として用いている。

本プロジェクトでは、このほかにも、いろいろな研究を手掛けている。その一つは、知的プログラミング・システムである。プログラムの検証、デバッグ、変換・合成などを行うシステムが開発されている。また、与えられた仕様を定理と見なしてそれを数学的に証明し、その証明の中からプログラムを抽出するシステム CAP の研究開発も進められている。

もう一つの特筆すべき研究は、人工知能の基礎研究である。知識表現、常識推論、定性推論、帰納推論、類推、学習などの研究が行われている。また、人間の思考活動を支援する論証支援システムの開発も行われている。

4. おわりに

本論文では、我々のプロジェクトのこれまでの研究開発の概観を与えたが、それは図1のフレームワークを図2に示すようにより精密化したことになっている。今後の課題は、これまでに開発してきた各コンポーネントを統合化することである。より具体的には、知識情報処理に適した制約論理プログラミングやメタプログラミングなどの強力なプログラミング技法と、並列実行に適した並列論理プログラミングの間のギャップをいかに埋めるかということと、並列推論マシンと知識ベースマシンをどう統合化して、最終的な第五世代コンピュータのプロトタイプとするか、の二点をもっとも重要な課題である。この統合化が成功すれば、高度な知識情報処理プログラムをそのプロトタイプ上で高速に実行することが可能となるであろう。また、本プロジェクトの究極的な目標を達成するために、人工知能あるいは知識情報処理の基礎を固める研究もこれから一層大切になる。学習システムや仮説推論システムなどの高度な人工知能システムには膨大な計算が必要になることが予想され、並列処理の応用としても興味深い、

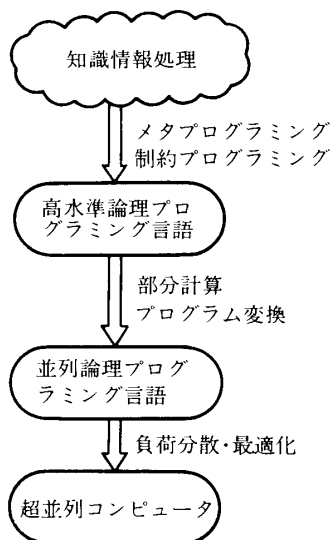


図2 精密化された第五世代コンピュータの図式

著者紹介



古川 康一

1965年東京大学工学部計数工学科卒業。1967年同大学院修士課程修了。同年、電気試験所(現、電子技術総合研究所)入所。1982年より(財)新世代コンピュータ技術開発機構へ出向。現在、同研究所研究担当次長。工学博士。第五世代コンピュータの基礎ソフトウェアの研究に従事。編著書に「知識情報シリーズ」(共立出版)、「Prolog 入門」(オーム社)などがある。人工知能学会、ソフトウェア科学会、情報処理学会各会員。