(4) 新世代コンピュータの応用

① Prolog-Based Expert System for Logic Design

丸山　文宏、真野　民男、林　　一司、角田多苗子
川戸　信明、上原　貴夫（富士通）

本論文は，Prolog をベースにした論理設計のためのエキスパートシステムについて述べる。

設計者は，並列プロセス記述言語 Occam でハードウェアのアルゴリズムを記述する。このレベルは，ハードウェアのアルゴリズムの概念に基づいている必要がない。機能設計フェーズは，この記述からハードウェア記述言語DDLによる機能設計を生成する。DDL記述は，回路設計フェーズ，CMOS設計フェーズによって，最終的にCMOSの回路を生成する。

従来，知識ベースによるアプローチは，医療の診断システム等において成功を収めてきたが，まだ成功例のほとんど無い設計問題を取り上げて知識ベース手法を評価することがひとつの目的である。

もうひとつの目的は，論理型言語の機能の確認である。論理型言語を用いて知識ベース設計システムをどのように構築できるかを評価することが課題である。

我々は，59年度前期に，論理設計支援システムのプロトタイプを VAX 上のC －Prolog で実現した。本論文は，その結果と評価についても述べている。

② Specifying Hardware in Temporal Logic & Efficient Synthesis of State-Diagrams Using Prolog

藤田　昌宏、田中　英彦、元岡　　達（東京大学）

ハードウェアの仕様記述に時間に関する演算子をもったテンポラルロジックを用い，この仕様記述から状態遷移図を自動合成する効率的手法について報告している。一般に，ハードウェアシステムは，演算部と同期部に分けて考えることができるが，システムが大規模で複雑になってくると，特に同期が問題となる。同期部の仕様記述の基礎としてテンポラルロジックを採用することで，タイミングチャートで表わされるような各モジュール間の時間関係を表現することができる。

状態遷移図は，仕様記述をテンポラルロジックの恒等式を用いて，現在と次の時刻の条件に展開することで合成できる。従来の展開手法ではその処理時間はテンポラル演算子の指数オーダを要していたが，ここで提案している段階的合成手法（すでに合成された状態遷移図を知識として蓄え，大域的な状態遷移図を段階的に合成するという手法）では，変数と状態数の多項式オーダに押えられている。この特徴は，Prolog の自動バックトラッキングとパターンマッチング機構により得られ，簡潔に実現されている。

③ A Knowledge Based System for Plant Diagnosis

元田　浩，山田直之，吉田健一（日立製作所）

プラントの異常診断システムにおける知識表現と推論機構について述べる。知識表現はframeとProduction Ruleを用いる。ここでプラント診断における知識は，event-orientedな知識とfunction-orientedな知識の二つが要求される。event-orientedな知識は，例外的な状態を取り扱い，function-orientedな知識は，通常の知識を取り扱う。それぞれの知識は，frame，Production Ruleの中に埋め込まれる。frame構造は，component，causality，criteriality，simulatorの4つのframeから構成されている。component frameは，階層構造内のそれぞれの要素の属性を記述するものであり，causality frameは，Production Systemの効率化を実現するために用いられている。criteriality frameはframeの構造化された知識を推論するために用いられ，simulator frameは，シミュレーションにおいて用いられる関数を記述するためものである。Production Ruleは二つの形式がある。一つは，causalityを表わすためのルールで，もう一つは，一つ以上の中間仮説を含む指示を表わすルールである。プラントの診断は，異常な症状から欠陥要素を調べ，診断結果に基づいて操作活動に必要な忠告を行なうことである。診断は実用に耐えるために高速でなければならない。そこで推論機構に不必要な探索を行なわないメカニズムをもうけた。それはresolutionメカニズムである。Production Systemとしての推論能力は，forward reasoningとbackward reasoningの両方があり，一般的な推論能力としては，上記の二つにresolutionが付加されている。プラントのフィードバック特性は，resolution手続きの履歴をうまく用いることによって，よりよい探索メカニズムを与えることができる。推論プログラムは，リスプで記述されており，シミュレータ，I／Oコントロールは，フォートランで書かれている。

④ Control of Heuristic Search in a Prolog-Based Microcode Synthesis Expert System

M.D. Poe (Apple Computer, 米国)

Prolog の組み込みサーチ手法として，topdown ，left-to-right 手法が存在しているが，これに基づいてもっと複雑なヒューリスティックサーチが容易に実現できる。マイクロコード生成エキスパートシステム実現の経験から，このプログラミング技法の確認を報告している。この手法は特別のパターンマッチ用インタープリタを必要としないところに特徴がある。エキスパートシステムUMS （Universal Microcode Synthesizer ）全体はProlog のソースコードで約15,000行であり，ほぼ同程度の大きさの次の3つの部分から構成されている。(1)コンパイラ部，(2)大域データ／コントロールフロー解析部，(3)ルールベースコード変換部。

ターゲット・ハードウェア記述と命令セットの手続き記述はアークとノード表現に変換される。アークはデータと制御の依存関係を，ノードはデータ操作を表わしており，マイクロコードはデータ操作を表わしており，マイクロコード生成の過程は命令ノードからハードウェアノードへの写像としてとらえられる。アーク処理順序はアークリストにより決定されるが，コントロールフローやハードウェアレジスタ・マッピングが情報を用いたヒューリスティックを，このリストの順序を変更することで表現している。また別のヒューリスティックとしては，「現在のコード生成に必要な資源は，直前に完了した生成と類似している」があり，これは以前に利用したハードウェア資源のリストをデータベースにスタックとして保存することで実現している。従来のマイクロコード生成システムでは，特定のマイクロエンジン用にユーザが変換知識を与える必要があったが，UMS は同様の知識をハードウェア記述から自動的に発見できる。

⑤ SIDUR-Structuring Formalism for Knowledge Information Processing Systems

D.D. Kogan, M.J. Freiling (Oregon State Univ., 米国)

近年，データベース技術と人工知能の技術を結びつけなければならない多くの応用分野が表われてきている。例えば，大規模データベースの維持，人工知能システムの推論能力においてこれらのことが要求されている。これらの要求に答えたシステムが開発され，それらは，知識情報処理，知識管理，知識ベースシステムなどと呼ばれている。この論文では，情報の構造化の一方式について述べる。このシステムは，SIDUR と呼ばれ，sigma 表現として知られる宣言的表現を用いて表現要素と処理メカニズムを統合している。sigma 表現の文法は，論理型言語の文法と類似している。その定義は（Name（Role 引数）…（Role 引数））のようになる。

さらに，これらを and ，or ，not ，empty によって結合することができる。リスプの lambda 変数のようなsigma 変数を用いることもできる。一般的に今まで作られた表現言語は，一つの解釈しか行なえなかった。そのような方式では，知識の統合や更新を行なうことが十分に行なえない。そこでSIDURでは，一つの表現形式を用いてモデルの関係に複数の解釈や処理を行なえるようにした。例えば assert ，enquire ，deny のようなsigma 表現に対する操作関数を準備している。これらは知識の宣言や問い合せ，除去を行なうことを容易にする。SIDUR では，これらを拡張し，いくつかの他の操作関数も装備されている。SIDUR の表現構造は，５つの基本的な表現によって行なわれる。それは Data Value Classes ，Object Classes ，Situations ，Computations ，Actions である。これらは frame 構造に類した構造をしている。その処理の制御は，Prolog のメタレベルコントロールと類似したものである。

⑥ LOOKS: Knowledge Representation System for Designing Expert Systems in a Logic Programming Framework

溝口文雄、大和田勇人、片山佳則(東京理科大学)

論理型言語を用いた知識表現システムである LOOKS の特徴と，その応用について述べたのが本論文である。LOOKS は Prolog を用いたオブジェクト指向型の知識表現システムであるが，次の特徴を持っている。すなわち，メタプログラミングを採用することによって，オブジェクト指向とルール指向を統合し，エキスパートシステムの開発を容易にした。このために，Bowen & Kowalski の提案したオブジェクトレベルとメタレベルの融合の概念をメタプログラミングに導入している。

実際の応用は，緑内障の医療診断システム LOOKS / Glaucoma である。個々の疾患とルールがオブジェクトの階層構造で表現され，どのオブジェクトがどのように使われるかをメタプログラムにより作成されている。基本的には，疾患の世界，ルールの世界，患者のデータ世界の三つの世界モデルをオブジェクト指向概念で実現している。推論の方式は，逆向き推論を採用している。

⑦ Mandala: A Logic Based Knowledge Programming System

古川康一，竹内彰一，国藤進，安川秀樹，大木優（ICOT）
上田和紀（NEC）

Mandala は，論理型プログラミングの枠組の中で，知識情報処理システムを構築するための知識プログラミング・システムであると同時に知識ベース管理システム構築のための基盤でもある。この二面性は論理型プログラミングの二面性，すなわち，論理型プログラムの手続き型及び宣言的解釈から直接導き出される。また，Mandala は並列実行環境をユーザに提供しており，これがMandala におけるオブジェクト指向プログラミングスタイルの支援の基礎となっている。Mandala における個々の オブジェクトは知識ベースと推論エンジンを有し，並列に動作する問題解決機とみなすことができる。これは，複数の問題解決機がメッセージを交換しながら協調して1つの問題の解決に当たる協調型問題解決システムの構築のときに非常に有用となる枠組を与えている。

本論文には，Mandala の基本思想，知識プログラミング及び知識ベース管 理についての記述がある。知識プログラミングに関しては，Mandala の基本構成要素である「実体」の記述を中心に，オブジェクト指向プログラミングやルール・ベース推論，プログラミング環境について述べられ，知識ベース管理に関しては，主に知識の同化を中心にその実現方法などが述べられている。

⑧ An Object Oriented Approach to Knowledge Systems

所　真理雄、石川　裕（慶応義塾大学）

知識システムの構築に対して，様々なプログラミング言語（システム）が提案されているが，ここではDKOM（Distributed Knowledge Object Modeling）という方法を提案し，DKOMを基本に設計したオブジェクト指向言語ORIENT84/Kについて述べている。DKOMにおいて，知識システムはKnowledge Objectの記述から成る。このObjectは次の3つの部分から構成されている。

monitor 部 ：オブジェクトのコントロールモニタをする。

behavior 部 ：Smalltalkのようなmethodを持っている。

knowledge部：Prologのようなルールと事実を持っている。

主な対象として，人間の知能的な活動を考え，それらをシミュレートできるものを目的としている。この知識システムは，宣言的（ある領域の知識システムとして）と手続き的（objectの動作を記述）の両方の表現と，それらの実行メカニズムを利用する。また，rule/factレベルとobjectレベルのモジュール化の両方を採用するものである。つまり，オブジェクトの集合として，知識システムを記述し，各オブジェクト内の知識は，ルールかfactのレベルで表現される。

ORIENT84/Kは，metaclass−class−instance階層を持ち，スーパークラスからの多重継承がある。Knowledge Objectは，並列に実行し，オブジェクト間はメッセージ・パッシングにより交信する。Knowledge Objectを構成する3部分には，それぞれ次のものがある。monitor部には，add−permission，delete−permission命令があり，behavior部には，add−method，delete−method命令がある。Knowledge−base部には，addKB，append KB（Prologのassertaとassertzに対する），deleteKBがある。

ORIENT84/Kのプロットタイプは Franz Lispで実装されている。最終バージョンは，オブジェクト指向アーキテクチャに実装する。

他に，ORIENT84／Kで行なったExpert Systemと他のプログラミング言語でとの比較も行なっている。

⑨ Intelligent Information Retrieval: An Interesting Application Area for the New Generation Computer Systems

G.P. Zarre (Centre National de la Recherche Scientifique,フランス)

本論文は知的情報検索システムにおける推論の手続き（RESEDAと呼ぶ）を述べたものである。この検索システムは，Advanced Interface Unit（AIU），Knowledge Processing Unit（KPU）およびKnowledge Base Unit（KBU）の三つのユニットから構成されている。AIUは問い合わせ部であり，自然言語ではないが，一定の問い合せパターンを用い検索を行なう。この問い合せに対して，内部の知識ベースから必要な情報を検索するために使われるのがKPUである。KPUではその問い合せパターンが知識ベースのどの部分であるかどうかを調べ，次に問い合せパターンの構造を解析する。この解析した結果を使って，検索の情報を推論していく。この方式には，問い合わせに直接応答するルールによって推論とそれに関連した仮説による推論がある。その結果，KPUは検索出力を作り出す。検索の対象は，フランスの歴史的事実である。

以上のRESEDAは APLを用いてインタープリタとして作成されている。APLを用いた理由は,主として効率の点から考えて選ばれた。

⑩ Knowledge Representation and Inference Environment: KRINE, An Approach to Integration of Frame, Prolog and Graphics

小川　裕、島　健一、菅原俊治、高木　茂（電電公社武蔵野通信研究所）

設計の問題を扱うエキスパートシステムでは，設計対象の階層的表現，パターンマッチング機能を含む設計知識の表現，設計対象の表示及びインタラクティブな編集機能などが重要であるが，これらをオブジェクト指向の知識表現やフレームによる知識表現だけで記述するのは難しい。Knowledge Representation and Inference Environment（KRINE）は，上記の問題を解決するために開発されたフレームをベースにした知識表現システムで以下の特徴を持つ。(1)フレームのデータをPROLOGのユニフィケーションの機能を用いて扱うことで，設計対象に対するパターンマッチングの手続を容易に記述できる。(2) try and error といった設計知識をPROLOGのバックトラック機能を用いて記述できる。(3)設計対象をフレームのデータ構造を直接解釈することによって表示する為に，フレームによるグラフィック環境を持っている。

本論文では，KRINEの設計思想，システムの構成，フレームによる知識表現のメカニズム等が述べられている。

⑪ Stalking "Coherence" in the Topical Jungle

B. Grau (Univ. Paris VI, フランス)

この論文では，textをそれぞれの文のtopicに注目して解析するシステムを示す。文は構文意味解析パーザによってtopicが抽出され，それらのtopicからcontextが作られる（bottom−up process）。また，contextはすでに解析されたtextのtopicの関数として求められ，次の文のtopicとのcoherenceが調べられる（top−down process）。ここで，coherenceとは，「文とcontextとの間に意味的又は語用論的なリンクが存在する事」と定義する。

各topicに関する知識は，フレーム・ネットワークでシステムに与えられる。それは二種類のリンクを持つ。一つは，「乗物を修理する」のは「物を修理する」事であるというようなtopicの階層リンクであり，もう一つは，「乗物を修理する」時には，「故障部品を修理する」といったdescriptiveなリンクである。

システムはtextの解析結果をtopicの木として表現する。また，その中から適当なtopicを選び出した集合をcontextとする。新しい文のtopicは，context中のtopicのいずれかにリンクかあるかどうかが調べられ，それによって，topic deviation，topic shift，topic level shiftとしてtopicの木が更新される。

このようにして，システムはcontextとのcoherenceに注目しながらtextを解析し，topicの展開を把握する。それによって，textの参照領域を限定する事ができる。

なお，このシステムでは，bottom−up processを導入している事がSchank等のシステムと大いに違う所である。

⑫ Steps Toward an Actor-Oriented Integrated Parser

上原邦昭、落合　亮、三上　理、豊田順一（大阪大学）

'Integrated Parser' (IP)は従来分離して行なわれてきた構文解析，文脈処理を同時に行い，かつ内部構造が統一されているため処理手続きの記述が容易で，しかも効率の良いパーザである。ＩＰでは文の内部表現と，この内部表現を決定するための文法記述はLexical Functional Grammar（ＬＦＧ）に基づいている。ＬＦＧでの構文解析は，部分機能構造の受け渡しによって行なわれる。この部分機能構造の受け渡しをACTOR理論におけるactor間でのメッセージパッシングと考えることで，ＬＦＧによる構文解析過程をうまく体系化することができる。さらにactor間のメッセージパッシングによってLFGパーザを構成することにより，actorの長所である，モジュラリティがあり，文法規則間での相互干渉を伴わず，明晰性に富むパーザが構成できる。

⑬ More on Gapping Grammars

V. Dahl (Simon Fraser Univ., カナダ)

GAPPING GRAMMAR（GG）は，構成要素間にあるgap（未解析のまま新しい位置に移され，その位置で後に解析される構成要素別）を陽に参照できる論理文法であり，EXTRAPOSITION GRAMMARを一般化したものとして設計された。

この論文では，自然言語および形式言語のパージングにどのようにGGを用いることができるか，またGGによりどのようにして強力かつ効率的に，また簡潔な形式でいくつかの言語現象を扱うことができるかについて論じる。その中で特に，free word order（ほとんどすべての可能なordering が受理可能である）を記述する規則を，可能なordering それぞれについて別々の規則を書かないで表現するGGの形式を提案し，また左外置，右外置を含む小さな英語のGGを示す。

⑭ Definite Clause Translation Grammars and the Logical Specification of Data Types as Unambiguous Context Free Grammars

H. Abramson (Univ. of British Columbia, カナダ)

データ型は，あいまい性のない文脈自由文法だと考えられる。その要素は，文脈自由文法の生成するderivation tree になる。更に，その文法記述からは，そのデータ型についての関数や述語を定義するのに使うデータ型の組み立てや文解のためのオペレータを得ることができる。我々の開発したデータ型をあいまい性のない文脈自由文法で記述できるDefinite Clause Translation Grammars（DCTG）（Abramson 1984）に我々は変更を加えた。例えば，二分木の文法記述は以下のようになる。

leaf : tree :: = string

branch : tree :: = " (", left :tree , ", " , right :tree , ") " .

分解のためのオペレータ，left , right（直観的にはstringも）は，文法をprolog節に変換するコンパイラーによって生成されるsemantic な属性である。これらのオペレータは，tree のパーザと述語leaf, branch と共にデータ型tree についてのより複雑な述語をつくるのに使われる。tree の枝をあらゆる段階で反転する述語を定義する場合などleft, right は重要な役割をはたす。

データ型の文脈自由文法による記述を使えば，さまざまなデータ型を使用できるシステムを論理型プログラムで記述できる。関数型プログラムkaviar は文脈自由文法としてのデータ型が基礎になっている。

⑮ Parallel Interpretation of Natural Language

J.B. Pollack (Univ. of Illinois, 米国)
D.L. Waltz (Thinking Machines, 米国)

モジュール化された知識の相互作用を特徴とする自然言語処理システムに関する研究である。

自然言語の解釈では，言語に依存した単語や句構造に関する知識や，典型的な状況，事件，役割，文脈などの現実世界に関する知識などの多くの知識を必要とするが，これらを入力，文脈，長期的な知識から動的に決定されるアクティベーション・ネットワークとして表現し，この上で解釈が行なわれる。意味的にあいまいな文の解釈における，文脈の影響を示す例と，意味的なガーデンパス文の解析を示す例が説明される。このシステムは，多くの言語現象とする仮説を提起し，またその仮説を容易に検証できるようになっている。

また，概念は単なるノードではなく，マイクロフューチャーの集合で関連付けられるべきである，という提案を行ない，これを例を使って説明している。

さらに現在はリスプマシン上で研究が進められているが，VLSI インプリメンテーションについて意識しており，アーキテクチャについてもコメントが加えられる。