

TM-0214

知識の制御と構造化
—知的意思決定支援のための環境KORE

新谷虎松(富士通)

August, 1986

©1986, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

知識の制御と構造化

—知的意思決定支援のための環境KORE—

新谷虎松

(富士通(株)国際情報社会科学研究所)

0. 要約

知的意思決定支援システムを構築するための課題を論じ、その技術的要請として汎用の問題解決支援環境KORE(Knowledge Oriented Reasoning Environment)の構成と機能について論ずる。特に、知識の制御と蓄積・構造化の分離、および試作機能について述べる。KOREは、論理型言語上の関係テーブルを共通の内部表現として、オブジェクト指向、データ指向、ルール指向、ネットワーク指向の知識プログラミングを独立のサブシステムで可能とし、協調的に動作することによって問題解決を支援するハイブリッドな環境である。KOREは意思決定過程の各局面における問題解決を支援する環境であり、知的意思決定支援システムはこのKOREを用いて構築される。

1. はじめに

意思決定過程には、①情報活動、②設計活動、③選択活動、④再検討活動の4局面がある。これら局面は、サイモン [Simon 79] が意思決定という言葉を広義に解釈し提唱したものであり、人間の高度な知的活動である意思決定を支援する際に考慮すべき局面である。情報活動（第一局面）とは、意思決定が必要となる条件（もしくは新しい状況）を見極めるために（経済的、技術的、政治的、社会的）環境を探索し機会を見出す活動である。設計活動（第二局面）は、可能な行為を発見、開発し、分析することである。選択活動（第三局面）では、利用可能な行為の代替案から、特定の案が選択される。再検討活動（第四局面）では、過去の選択の再検討が行われる。

既存の意思決定支援研究においては、上に述べた意思決定過程の四局面の一部分を支援するための技法や方法論として、(1)DSS(Decision Support System: 既存の意思決定支援システム)、(2)意思決定分析法、(3)エキスパートシステム等がある。

本論文の目的は、既存の意思決定支援ツール・技法をうまく取り込み、意思決定過程の四局面（広義の意思決定）の支援を有機的に結合する知的意思決定支援システム（知的D

SS) において実現されるべき機能とそのための環境について論じることである(図1参照)。具体的には、問題解決支援環境KOREの提案とその構成、知識の制御と構造化について論じ、試作機能について述べる。

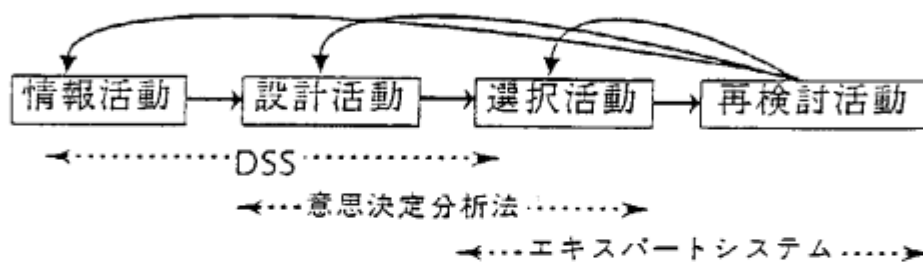


図1. 意思決定過程の4局面と既存意思決定支援ツール・技法

2. 意思決定過程と既存意思決定支援技法・ツール

本節では、意思決定過程の各局面を支援する際に必要となる機能を明らかにするために既存意思決定研究を概観し、3節ではこれら機能を実現するための問題解決支援環境KOREの構成について論じる。

2.1. DSS

DSSと呼ばれる各種のシステムが構築され使用されているが、意思決定支援システムの研究は新しい学際的な研究分野であり(専門誌Decision Support Systemsが1985年に創刊)、現在なお、多くの課題(例えば、非構造的な問題[Keen 78]に対する意思決定支援等)が存在している。

意思決定支援システムという用語はMIS(Management Information System)の反省としてGorryとMorton[Gorry 71]により最初に提案されて以来、様々な意味で使用されている。(既存の)DSSはその機能に着目すると主にデータ指向(データの検索・操作)とモデル指向(最適化モデル、会計モデルの運用)のシステムに分類される[Alter 80]。既存DSSの主流であるデータ指向DSSは主に「情報活動」の支援を目的としている。モデル指向DSSは主に「設計活動」・「選択活動」の支援を目的としているが、現状ではその支援は一部に限られている。

既存DSSは構造が明確な意思決定問題を処理することができる。したがって、このようなDSSを高度に利用するためには、非構造的な問題をDSSが処理できる構造的な問題に分割するDSS専門家(図2参照)が必要となる。DSS専門家は、意思決定者が要

求する概念をDSSが受け入れることのできる特定な質問に変換しDSSを直接に運用する。もし、ある質問に対するデータ、ルーチンそしてモデルが欠けていたなら、それらを補う（DSSを再構築する場合もある）。これには多くの時間が費やされる。

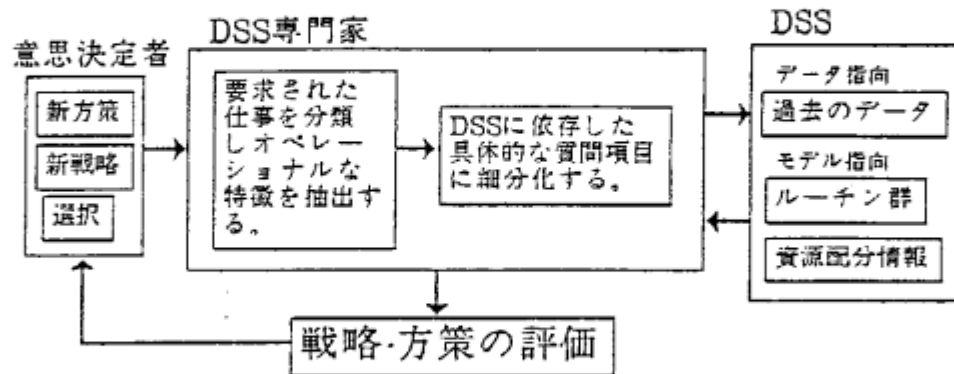


図2. DSS専門家の役割

非構造的な意思決定問題にも対応する支援機能をリアルタイムに実現するための方法論として、知識ベースに基づくDSSの支援機能が有効である。知識とはこの場合、DSSの能力や効率を向上させることを可能にする情報（もしくはデータ）を意味する。知識としては、次のようなタイプの知識を表現する必要がある：①真である命題（もしくは関係）を表現するための事実、②もっともらしい命題を表現するための信念、③一般的な（もしくは有効な）アルゴリズム（モデル）が存在しない場合に用いるためのヒューリスティクス（これは多くの場合ルールとして表現される）

「情報・設計活動」を支援する③のシーズとしては、Bonczek [Bonczek 81] 等が提案している一階述語論理を用いたモデル（もしくはモジュール）の管理・利用手段がある。彼らが提案する方法は、モデル間結合のメカニズム処理を中心とするモデル管理の手段として次の特徴がある：(a)モデル間結合の追加、変更が柔軟にできる、(b)ユーザが高水準言語（例えば、Prolog等）を用いて必要なデータを要求することにより、それに必要なモデルの構築と実行を、モデル化のための知識を用いて行う、(c)Prologを使ってインプリメントがし易い。そこで、我々は、Prologを用いたデータベースの設計と、このデータベースに推論エンジンを結びつけることのできる機構をKOREの上で実現する。

2.2. 意思決定分析法

意思決定過程における「設計活動」・「選択活動」における支援機能は、我々が最も着目している機能であり、多くの課題が山積している。既存研究においてもこれら機能を積極的に取り入れ知的に支援するシステムはまだ無いようである。我々は、これら局面における支援機能をうまく実現することにより、DSS的な意思決定支援とエキスパートシステムの的な意思決定支援とを橋渡し、図1で示したような広義の意思決定支援を実現することを計画している。

意思決定分析法は主に意思決定局面の「設計活動」や「選択活動」を支援する技法として考えることができる。意思決定分析法には意思決定論[市川 83]やシステム科学等の分野で議論されている技法がある。これら技法は、意思決定者がどのような意思決定問題に直面しているかを明らかにするための手段・技法を提供する。直面する意思決定問題を明らかにしたら、次に重要なことは、様々な意思決定問題に適応する解法を適用するために、意思決定問題を構成する選好関係やある状況に対する意思決定者の満足度を数量化する必要がある。このような満足度が数量化されれば、いくつかの代替案を相互に比較することが可能となり、客観的に数学的手法を利用することができる。満足度のような主観的選好判断に基づいた代替案の比較・評価は、今後、意思決定支援システムの支援機能として重要なものとなろう。

ある決定者の満足度の変化を量的に表現したものが、その決定者の効用と呼ばれる。しかしながら、効用を定義することは、古くから心理学的な分野でさえ様々な議論が行われているが、それを数量的に測定することは困難なことである。そこで、代替案の比較をその(複数の)属性に着目して行う多目的決定問題[樺木 84]として捕らえ、計算機を用いて対話的に決定する手法が有効になる。

この局面における意思決定支援ではシステムはシステム自身の情報を提供するよりもむしろ、ユーザ自身の知識を探索したり構造化することにおいてユーザを支援し、問題解決のための他に必要な行為についてアドバイスを提供する。このような支援をユーザ知識に基づく支援と呼ぶ。

ユーザ知識に基づく支援を具体的に述べることにしよう。この種の支援は、ユーザが自分のバックグラウンド知識と専門知識の大部分を用いることを想定しており、意思決定問題に関連した知識としてユーザが気付いた知識の断片を積極的に支援システムのなかへ取り込む。このユーザ知識に基づく支援においては、システムは知的なノートとして機能す

る。つまり、このモードにおける支援システムは自分自身の情報は提供しないが、ユーザ自身の知識を探索したり、構造化することにおいてユーザを支援し、問題解決のための行為の他に取るべき方向についてのアドバイスを提供する。

ユーザ知識に基づく支援を行うシステムはユーザと会話することにより情報を交換する。この会話は、先ず最初に目標を述べることから始まり、これら目標を実現するさらに詳細な手段を解明して行く。このシステムはユーザの注意を問題に対して決定的である論点へ集中するよう誘導する。つまり、このようなユーザ知識に基づく支援は図3のようなユーザ支援サイクルを実行することにより、ユーザから意思決定支援のための情報を獲得し、構造化することによって、ユーザの意思決定を支援する。

図3は意思決定ネットワークとして考慮されるべきである。従って、目標はいくつかの副目標へ分割され、各々の副目標はその副目標を達成するであろういくつかの可能な行為(action)を有しており、各々の行為はその行為を遂行するいくつかの方法を有しており、また、各々の方法は達成されなければならないいくつかの前提条件を有している。一度、そのような前提条件が明確化されたなら、それら前提条件はさらに新たな副目標を導くことになる。

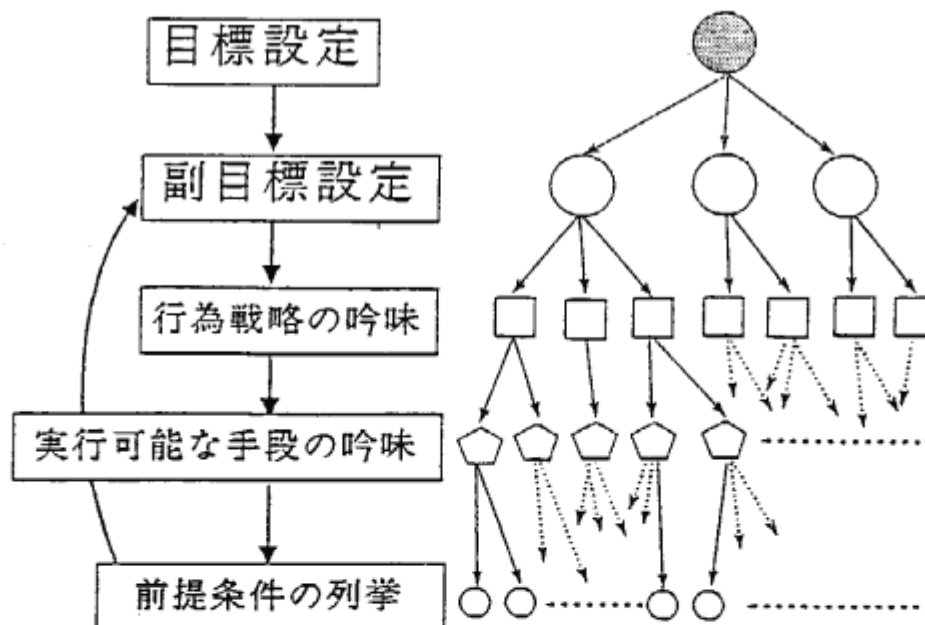


図3. ユーザ知識に基づく意思決定支援サイクル

ユーザは、システムとの会話の間、上でのべたネットワークにおけるノード間の様々な

関係の強さの量を指定することを要請される。これら、指定された量は意思決定ネットワークを通じて伝播される。このことにより、主目標を実現させるために最も決定的である論点へとユーザの会話が集中する。この意思決定支援のためのサイクルは再帰的に繰り返され、ユーザが主目標に対して満足する行為や戦略を見出すことにより、ユーザにより終結させられる。その時ユーザはそこまで行われた意思決定ネットワークの提示とそれより得られる結論とをシステムからレポートの形で得ることができる。

このようなユーザ知識に基づく支援は、問題解決システムABSTRIPS [Sacerdoti 74] 等で用いられている階層的な計画作成技法（問題分割法や手段—目標解析）と多目的計画法（多目的決定問題）等における代替案の評価技法を取り入れ、支援システムとして実現することに特徴がある。

ABSTRIPS等の問題解決システムでは問題解決のための作用素（operators）をあらかじめ与えて置くことにより、問題解決（計画作成過程）が実行される。作用素はある種の規則（もしくは、行動プラン）であり、その適用可能性に対する前提条件を持っている。もしある作用素の前提条件が真になれば、その作用素は実行され、世界の状態は変化させられる。問題解決は、このように世界の状態があらかじめ定めた目標状態に到達することにより達成される。

一方、ユーザ知識に基づく支援においては、主目標を達成するために必要となる作用素の構築を支援（つまり、その前提条件と結果を明らかにすること）することにより、対象としている問題の構造を明らかにし（意思決定過程における「設計活動」に相当する）、ある目的を達成するために取り得る可能な行為（代替案）を多目的計画法等により評価すること（意思決定過程における「選択活動」に相当する）がその主目的となる。このように作成された作用素は、ある問題に対する意思決定者のモデル（信念）として用いることができ、非構造的な意思決定支援のための枠組みを提供する。

そこで、我々は以上のようなユーザ知識に基づく支援機能構築のニーズとして、知識を構造化し利用する機構や知識の一貫性を保持管理するための機能を挙げ、これを実現するためにオブジェクト指向概念に基づく知識表現機構とネットワーク構造を管理するための機構をKOREに実現する。このようなネットワーク管理機構は暗黙推論 [Reiter 80] や truth maintenance system [Doyle 78] を実現する上で重要な dependency-directed back tracking [Stallman 77] のための情報 [McDermott 83] を提供する。

2.3. エキスパートシステム

エキスパートシステムは意思決定過程の局面の「選択活動」・「再検討活動」の一部分を支援するものとして考えられ（主に半ば構造化された(Semi-structured) 意思決定を支援するDSSとは異なり）、非構造的(unstructured)な意思決定問題に対処する手段を提供する。つまり、エキスパートシステムは問題領域に依存したヒューリスティクス（知識）を蓄積した知識ベースを用いることにより、対象とする非構造的な意思決定問題に対処する[Hayes-Roth 83]。

例えば、診断型のエキスパートシステムは、知識ベースに蓄積された知識を運用することにより、先に与えておいた複数の診断項目のなかで妥当な診断項目を列挙し再検討する（選択→再検討）機能を提供することで診断における意思決定を支援する。また、設計型のエキスパートシステム（このアーキテクチャを明らかにしていくことは知識工学における今後の魅力あるテーマであり多くの課題が山積している）は、問題解決過程におけるカレントな状態に適切な知識を運用（検討→選択）していくことにより、抽象度の高い主目標を具体化していくことを支援するものとして考えられる。

知識ベースに基づく支援は、必要とされる知識を知識ベースに前もって蓄積・利用することにより、ユーザの意思決定のための知的な支援を可能にする。

エキスパートシステムにおける以上の支援機能は、知識ベースに基づく意思決定支援機能を実現するために、対象領域に依存した知識が整理・記述されることの必要性を示している。つまり、対象領域ごとに記述し易い知識表現形式を採用する必要がある。我々は、このような要求に対してKOREにハイブリッドな知識表現形式を導入することにより対処する。しかしながら、様々なプログラミングパラダイム（オブジェクト指向・ルール指向・アクセス指向等）[Bobrow 84] を一つの知識表現形式に押し込むことは、プログラマ以外のエキスパートが直接にその表現技法を用いることを困難にし、また、その技法によるアプリケーションの構築も困難にする。そこで、我々は、論理型言語上に関係テーブル概念を用いて様々なプログラミングパラダイムの新たな統合化法をKOREで実現する。

3. 問題解決支援環境KORE

問題解決支援環境KORE(Knowledge Oriented Reasoning Environment)は先に述べた意思決定過程の四局面（情報活動・設計活動・選択活動・再検討活動）を支援するための知識

の制御機構と知識の構造化機能を提供する [新谷 86a] , 知識の制御機構としては, KORE のサブシステムとして, (1)KORE/DB (Data Base subsystem) [平石 86] , (2)KORE/IE (Inference Engine subsystem) [新谷 86b] , (3)KORE/KR (Knowledge Representation subsystem) [片山 86] , (4)KORE/EDEN (Extended Dependency Network subsystem) [新谷 86c] がある。また, 知識を構造化するための表現として我々は関係テーブルとよばれる内部表現形式を用いる。これらサブシステムは, 単独でも独立的に機能し, 全体としてKOREに様々な問題解決支援機能を提供する。これらサブシステムは, 内部表現において共通の関係テーブル表現をもつことにより, KOREとして統合化される (図4参照)。この関係テーブル表現はPrologにおける関係表現として素直に実現される [Kowalski 77] 。

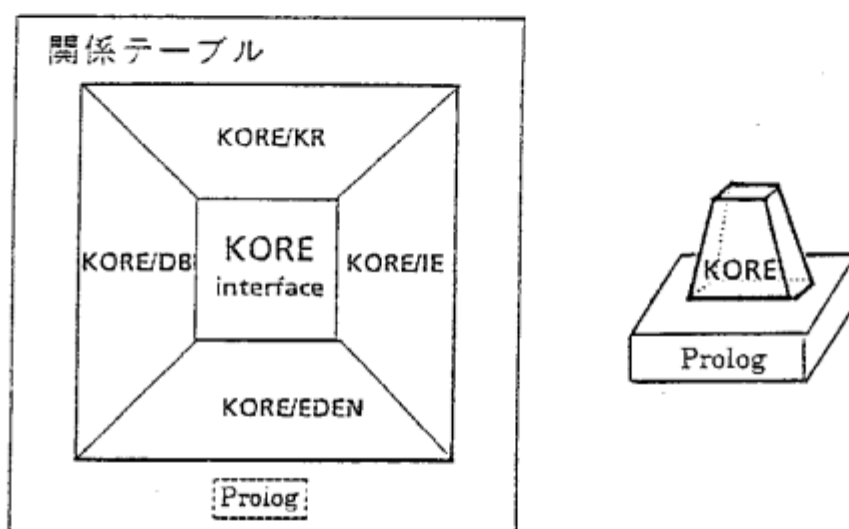


図4. 問題解決支援環境KOREの構成

3.1. 知識の構造化

KOREではKOREサブシステムで用いられる知識の内部表現を統一的に管理するための構造として関係テーブルを用いる。このような内部表現の統一は, 意思決定過程における四局面の支援で用いられる情報を意思決定過程の間, 相互に用いるための手段を提供する。つまり, KOREの各サブシステムはこのような関係テーブルをコミュニケーションの場として用いることにより, 他のサブシステムの問題解決過程の結果を利用したり, 問題解決そのものを支援することが可能になる。

関係テーブルは, Hayes-Roth [1978] 等が提案した黑板モデルにおける黑板として見る

ことが出来る。しかしながら、KOREにおける関係テーブルは異なった知識制御を担うKOREサブシステム間における情報の交換場所であり、KORE/DBにより管理される関係データベースであるという点に特徴がある。

具体的には関係テーブルは図5上で示すようなテーブル構造である。図5下で示すように、関係テーブルはPrologのアサーションによる関係表現として素直に表現ができる。つまり、n項関係表現における引数の位置はテーブルの列に相当し、各々の関係はテーブルの行に相当する。

ここで、我々は、関係テーブルの各行をタプルと呼び、また関係テーブルにおける列要素を属性と呼ぶ。このような呼び方は関係データベースより由来し、関係テーブル自身ある種の関係データベースとなっている。

測定	名前	体重	身長
	太郎	80	172
	次郎	75	180
	花子	55	157

↑↓

測定 (太郎、	80、	172)
測定 (次郎、	75、	180)
測定 (花子、	55、	157)

図5. 関係テーブルの例

3.1.1. 関係表現の拡張

KOREでは、KOREのサブシステムで用いられる知識の内部構造を自然に表現するための工夫として関係表現の機能を拡張している。これら機能はPrologにはない機能であるが、Prologを素直に拡張したものである。拡張機能としては次の4点が挙げられる。

①引数のタイプ宣言機能： 引数で取り得るタイプを宣言することにより知識の処理を効率化する。

この機能は、図5の関係テーブルを例にとると以下のKORE/DBのコマンドにより達成さ

れる。

```
create 測定 with [名前, 体重#number, 身長#number] .
```

createコマンドはテーブルを定義するためのものである。この例では、テーブル名を測定とする関係テーブルを定義しており、その属性として名前、体重、身長があることを示し、特に体重と身長という属性にはnumberというタイプを宣言している。このようなタイプ宣言のもとでの次のような関係は測定（テーブル名）には付加されない。

測定（次郎、重い、高い）

②引数の位置の非固定化機能： 引数の位置の役割を明示することにより、引数の位置の固定化を回避する。これにより、表現の柔軟性を増すことをその目的とする。

例えば、図5の関係テーブルにおける第一行（タプル）を表す関係表現は次のように表すことができ、

```
測定（名前=太郎, 体重=80, 身長=172）
```

これはまた、以下のようにも表すことができる。

```
測定（体重=80, 名前=太郎, 身長=172）
```

③任意引数によるパターンマッチング機能： これはキーワード探索を実現するためのものである。

例えば、図5のような関係テーブルが①の機能により宣言されていれば以下のようなパターンマッチングは成功する。

```
kore_unify( 測定（名前=太郎, 体重=X）,  
            測定（名前=太郎, 身長=172, 体重=80） )
```

このとき、変数Xには属性名体重の値80がバインディングされる。

④タイムタグの付加機能：これは生成された順に単調に増加するユニークな数（タイムタグと呼ぶ）を新たに生成されたタプル（テーブルの行に相当し、具体的には一つの関係（アサーション））へ付加する機能である。このようなタイムタグはKOREにおける推論の制御情報等として用いられる。KOREでは、このようなタイムタグとして、システムが起動されてからのCPUタイムを利用している。

3.1.2. 関係テーブルの管理

KOREでは関係テーブルはKORE/DB を用いて統一的に管理される。関係テーブルはある種の関係データベースであり、KORE/DB は関係データベースを操作するための基本的コマンドを提供する関係データベース管理システムとしてみなすことができる。これらコマンドはPrologにおけるゴールとして記述でき（例えば、節の本体にサブゴールとして記述できる）、KOREサブシステムの共通な関係テーブルアクセス述語として用いられる。以下でそれらの機能と簡単な例を示す。

(1)関係テーブルの定義（create文）

create 測定 with [名前, 体重#number, 身長#number] .

(2)関係テーブルの操作

①挿入（ins 文）：タプルの生成機能

ins [体重=80, 名前=太郎, 身長=172] into 測定.

②更新（update文）：タプルの更新機能

update 測定 set 体重=85 where 名前=太郎.

（解釈：テーブル測定のなかで名前が太郎のタプルの体重を
85に変更せよ。）

③削除（del文）：タプルの削除

del 測定 where 名前=太郎.

（解釈：テーブル測定のなかで名前が太郎であるタプルを削除せよ。）

④トリガ機能：タプルの挿入（insertion）、削除（deletion）、更新（up）により自動的に起動される手続きのこと。

define_trigger 印字1 on insertion of 測定:write('***inserted***').

(解釈: テーブル測定にもしタプルを挿入したら'***inserted***'というメッセージを印字する手続き名 "印字1"を定義しなさい.)

(3)関係テーブルの検索

指定したテーブルの条件式を満たすタプルの指定した属性値を求めるコマンドであり、`select`文と`fetch`文がある。ここで、条件式を含む検索は関係データベースにおける制約に相当し、条件式がなければ射影に相当する。`select`文と`fetch`文との違いは前者は常に検索が成功し(検索すべきものがなければ空リストを返す)、後者は検索条件を満たすものがなければ、失敗することである。ここでいう、成功や失敗はPrologにおけるゴール節の実行を意味する。例えば、`select`文は次のように用いられる。

`select (名前, 体重) from 測定 where 体重<70.`

(解釈: テーブル測定において体重が70より少ないタプルの名前と体重を表示せよ.)

また、このような検索がPrologにおける節の本体で用いられ、他のサブゴールへその検索結果を引き渡すためには、次のように検索結果を変数にバインディングさせることにより達成する。

..., `select (名前, 体重) to X from 測定 where 体重<70, p(X), ...`

この時、変数Xには名前と体重に対する検索結果のリストがバインドされる。

3.2. 知識の制御

KOREでは、様々な知識プログラミングパラダイム [Bobrow 84] を一つの言語内で実現するのではなく、パラダイム特有の言語/制御を提供する。このような言語/制御を提供しているのがKOREのサブシステムであり、KOREとしてはパラダイム特有の知識制御を提供することに対応する。つまり、各サブシステムはKOREのアプリケーションを構築する際の部品(知識制御機構)となり、様々なアプリケーションを構築するために組み合わせて用いられる。

図6はKOREにおける知識の制御と蓄積(構造化)の区別を説明している。図6で示すように知識の制御とは、例えば、KORE/IE(プロダクションシステムの構造を持っている)におけるルール解釈実行部(PSI)の機能であり、その他の構成要素であるWM(ワーキングメモリー: 宣言的情報)やPM(プロダクションメモリー: 手続き的情報)は関係

テーブルとして構造化され蓄積管理される。つまり、KOREにおける知識の制御とは関係テーブルに蓄積された情報を解釈実行する振舞を言う。KOREでは、知識の制御方式として、KORE/IE によるルール指向的制御、KORE/KR によるオブジェクト指向的制御、KORE/DB によるデータ指向的制御、KORE/EDEN によるネットワーク指向的制御が存在する。

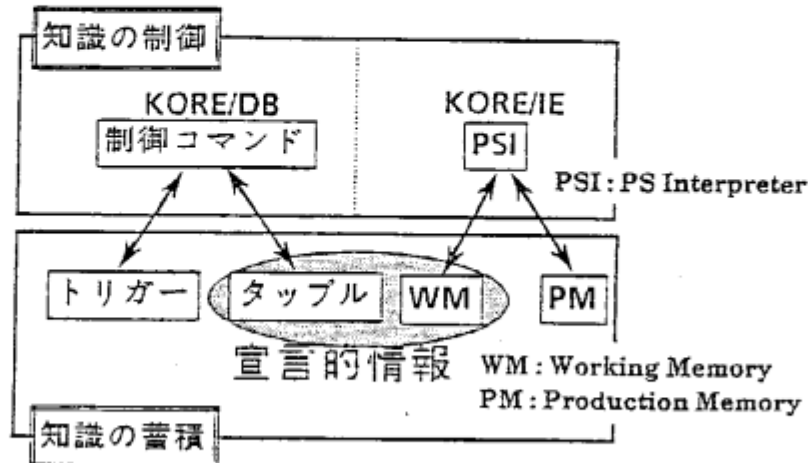


図 6. KOREにおける知識の制御と蓄積

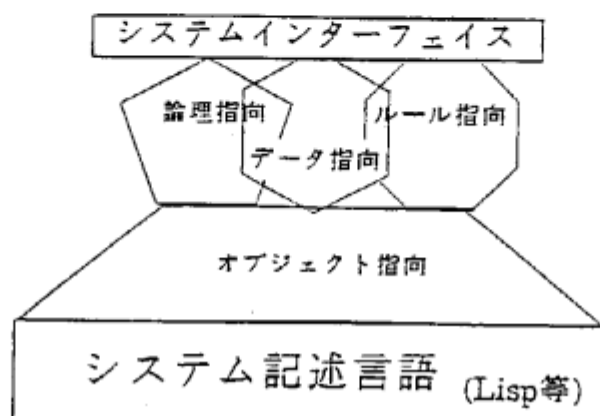
KOREにおけるこのような様々な知識の制御の部品化は、2節で論じた意思決定支援環境を構築する際の強力な手助けとなる。例えば、図6で示した例は、KORE/DB とKORE/IE を部品として使うことにより、関係データベースに推論エンジンを結合したことに相当する。つまり、KORE/DB によるタプルの追加はKORE/IE にとりワーキングメモリーの変化に相当し、これによりルールが起動されるきっかけとなる。また、ルールの起動の結果による、ワーキングメモリーの変化はKORE/DB のトリガー起動のきっかけとなり関係テーブル（データベース）の変更をもたらす。このようにKORE/DB とKORE/IE は協調的に問題解決をはかることが可能になる。

一般的に、多くのプログラミングパラダイムを同じ言語内に取り込んだシステム（例えば、[Bobrow 83]）は、非常に複雑な言語シンタックスになりがちであり、このことはその言語を用いたアプリケーションの開発を困難にすることがある。KOREは有用なプログラミングパラダイムを固有のサブシステムに分散実現することにより、様々なプログラミングパラダイムを容易に用いることを可能にし、かつ、それらサブシステム間の通信（情報交換）は、共通の内部表現を保有することにより容易に達成する。

KOREは、2節で論じた知的意思決定支援システム構築のために必要とされるハイブリッ

ドな知識表現および知識制御を実現するために、既存のハイブリッドな知識プログラミングツール（例えば、KEE [Fikes 85], LOOPS [Bobrow 83], 等）のアプローチ（ひとつのプログラミングパラダイムの上に、様々なプログラミングパラダイムを実現する）とは異なり、固有な知識表現パラダイムを提供するいくつかのサブシステムにより構成される（プログラミングパラダイムを固有のサブシステムに分散し提供する）。前者のアプローチはハイブリッドな知識表現の実現を指向するが、KOREはハイブリッドな知識制御を実現することをその主眼とする。ここに大きな違いがある。（図7参照）。

★既存のハイブリッドな知識プログラミングツール



★KORE

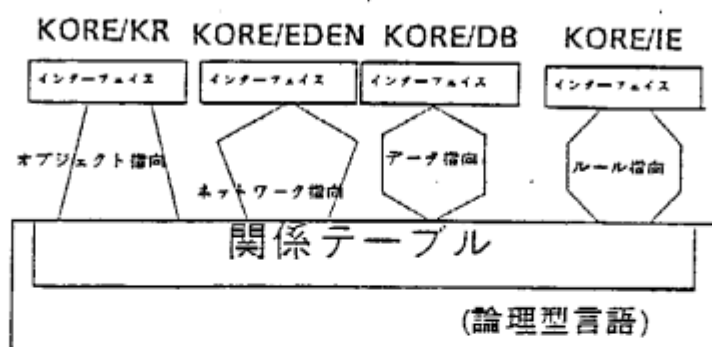


図7. KOREの構成の特徴

KOREサブシステムの基本機能の概要は次のように記述することができる。

3.2.1. KORE/DB

KORE/DB は、KOREにおける関係テーブルを操作管理する機能を担う一方、単独では関係データベースを操作するためのパーソナルな関係データベースシステムとして機能し、意思決定支援の局面においては「情報活動」におけるデータベース指向的支援機能を提供する（KORE/DB の機能については3.1.2 節で論じた）。

3.2.2. KORE/IE

KORE/IE は、KOREにおけるルール指向的問題解決・知識表現機能を担う一方、単独では OPS5 [Forgy 81] 等で代表される強力な前向き推論型プロダクションシステムとして機能し（KORE/IE の基本的機能や言語シンタックスはOPS5をベースにしている）、意思決定支援の局面において知識を運用する際の推論エンジンとして機能する。

KORE/IE のルール定義を例で示すと次のように記述できる。

```
on_floor: if
    goal(status=active, type=on,
          object__name=X) &
    monkey(on\= floor)
then
    modify(2, on=floor) &
    modify(1, status=satisfied).
```

KORE/IE の RHS（結論部）では、OPS5で用いられるRHS アクション（make, modify, remove）とユーザ定義のProlog述語を用いることができる。特に、make, modify, remove などの RHS アクションはワーキングメモリーを操作するものであり、KORE/IE では、これら機能はそれぞれ、関係テーブル操作述語の ins文、update文、del文（3.1.2 節参照）により素直に実現されている。また、ルールはRete [Forgy 82] 風アルゴリズムによりコンパイルされ、高速なパターンマッチングが行われる。

KORE/IE のトップレベルコマンドとしては、先に挙げた make, modify, remove 以外に (1) literalize と (2) run コマンド等がある。(1)のliteralizeはOPS5同様、ルール中に使用されるクラス名、属性名を宣言するものであるが、これはKORE/DB における関係テーブルを定義するcreate文により素直に実現される。(2)の runコマンドはKORE/IE の認知行動サイクルを実行させるものであり、次のように引数として認知行動サイクルの回数（正負の整数を指定する）と融合解消戦略（現バージョンではOPS5と同様にmea とlex がある）を

指定することに用いられる。

```
run (20, mea) .
```

ここで、もし、認知行動サイクルの回数として負の数を指定すると指定した数だけ認知行動サイクルの実行が戻される。

3.2.3. KORE/KR

KORE/KR は、知識を階層的に構造化し運用するためにオブジェクト指向概念 [Goldberg 83] に基づく知識表現・利用機能を提供する。

KORE/KR におけるクラス定義の記述例を示すと次のようになる。

```
defclass scroll__window ::  
  supers window ;  
  metas meta__object ;  
  value__set  
    wp=[2,6,5],  
    partition=hbm,  
    attribute=an,  
    scroll=jump__scroll;  
  method  
    move:[Top,Bottom]: 'move window':  
      Line is Bottom-Top+1,  
      putvalue(self,wp,[Top,Bottom,Line]),  
      (^ refresh);  
  ....
```

この例で、本文字はKORE/KR のキーワードを示している。例で示したように、KORE/KR でのクラス定義は(1)関係記述部 (supers, metas, partof), (2)スロット記述部(value__set), (3)メソッド記述部(method)により構成される。

関係記述部はKORE/EDEN により管理され、自由度の高いメッセージパッシング機能を実現している。KORE/KR では次のようなメッセージパッシング方式がある。

①通常のメッセージパッシング法

```
scroll__window ← new, window1.
```

(解釈: クラスscroll__windowに new というメッセージを送り新たなインスタンス window1を生成する)

②ある階層以下のオブジェクトを対象とするメッセージパッシング法

③あるクラスオブジェクトの全てのインスタンスを対象としたメッセージパッシング法

```
scroll_window <~ refresh: @ ( wp, [_,8,5] ).
```

(解釈: クラスscroll_windowの全てのインスタンスの内スロットwpの値が
[_,8,5] とマッチするものは全てリフレッシュしなさい.)

またスロット記述部は関係テーブルとして他のKOREサブシステムと同様な形式で内部表現される. このようなKOREサブシステム間における関係テーブルによる内部表現の統一化は, KOREサブシステム間の協調した問題解決支援のための基本的な機能を提供する.

例えば, KORE/KR のインスタンスの生成やそのスロット値の変更をKORE/IE におけるルールのトリガー条件として用いることができ, また, KORE/IE におけるルールの実行過程 (ワーキングメモリー) を構造化された知識としてKORE/KR を用いて管理することができる (図8 参照). これは, KORE/IE を推論エンジンとして, KORE/KR を知識ベースとして用いたことに相当する.

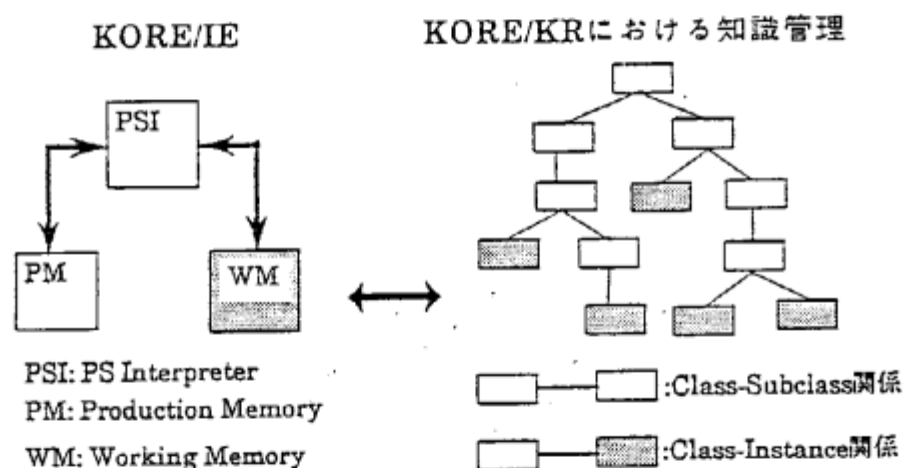
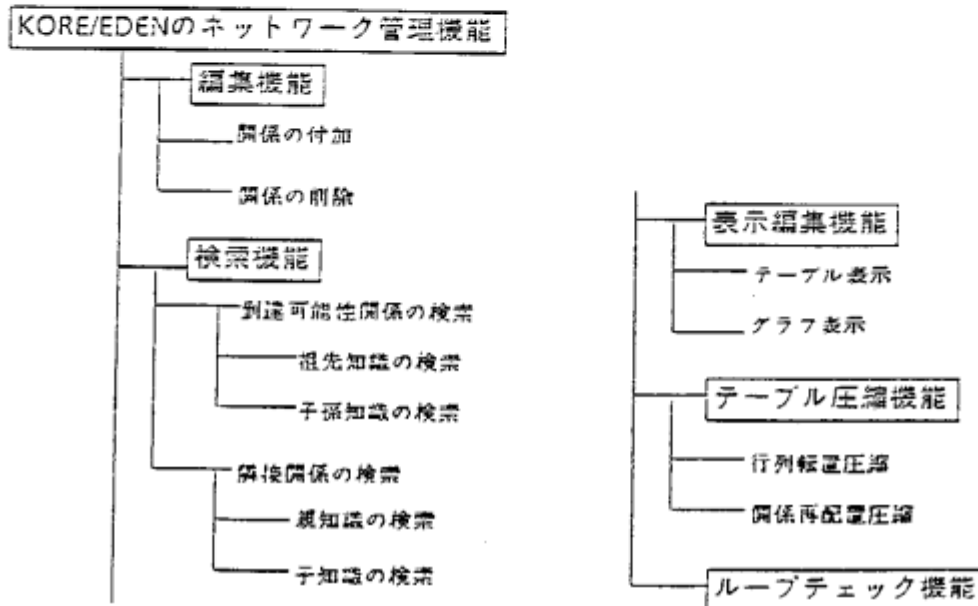


図8. KORE/IE とKORE/KR の協調問題解決

3.2.4. KORE/EDEN

KORE/EDEN は、KOREにおいて知識を管理・利用するために必要となる知識間の関係情報を保存し、知識ベース管理機能の一機能を担う一方、単独ではネットワーク [Brachman 83] を操作するための強力なシステムとして機能する。

KORE/EDEN の機能はネットワーク化された知識間の関係を知識テーブルとして効率的に保存し利用する次のようなネットワーク管理機構を提供することである。



知識テーブルは特殊な関係テーブルであり、ネットワークにおけるノード間の隣接関係や到達可能性関係に関する情報を表現する行列である。知識テーブルの内部表現はこの行列の行要素を2進数に圧縮し正整数として保存する関係テーブルで表現される（これにより知識テーブルの記憶容量の削減と操作の効率化を達成する）。知識テーブルへの操作（関係の付加、削除、知識テーブルからの検索）は、これら正整数に対してビット論理演算操作（整数ビット論理積・和、ビットシフト等）を適用する。したがって、行要素を整数として一度に扱うことができ、効率的な効率的な操作を達成する [新谷 85d]。

以上のようなKORE/EDEN の関係情報の管理機能はKOREにtruth maintenance のための基本的な情報等を提供する。例えば、KORE/IE により新たに推論されたデータは、そのデータを支持するデータをjustification としてKORE/EDEN を用いて関係づけられることができる。このようなjustification はtruth maintenance のために必要であり、また、KORE/IE の説明機能を達成するための情報になる。さらに、このようなjustification を保存

することにより、あるデータがデータベースから削除されたならそのデータに依存するデータを自動的に消去することも可能になる

4. 知的意思決定支援の要請とKOREの関係

第2節で論じた知的意思決定支援システムを構築するための要請はKOREのサブシステムを組み合わせて用いることにより達成される。KOREでは異なった知識制御を関係テーブルを介することにより自由に組み合わせ統合化できる（図9参照）。

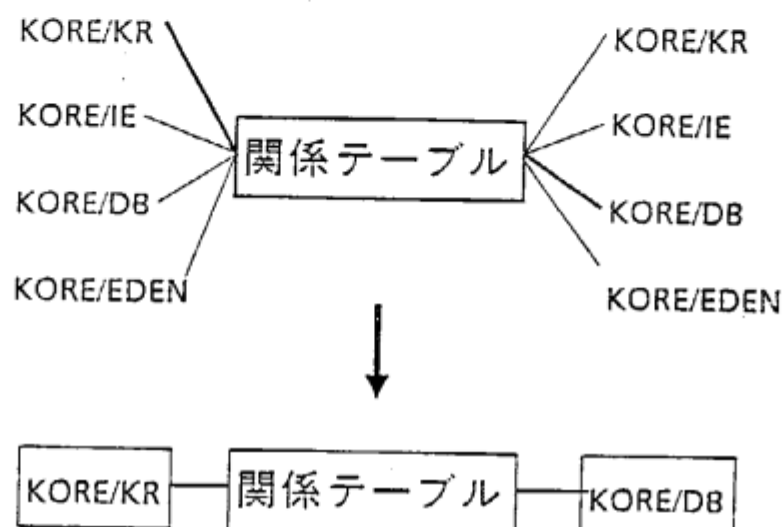


図9. KOREサブシステムの組み合わせ統合化

図9では、KORE/KR とKORE/DB の統合化例が示されている。このように統合したシステムでは、KORE/DB により作成された関係データベースをKORE/KR を用いて階層性を有した関係構造として管理ができる。さらに、KORE/KR におけるメソッド定義を用いることにより、より複雑なデータ処理のためのプログラムを記述することが可能になる。また、逆に、KORE/KR により構築された情報（特に、スロット情報）はKORE/DB の宣言的で簡潔な探索コマンドにより容易に引き出すことが可能になる。

KOREの以上のような機能は、意思決定過程の四局面の各局面を支援するための問題解決機構を構築するために必要となる基本的で統一的な環境を提供していることに相当する。

例えば、2.1 節の「情報・設計活動」で要請されたデータベースへの推論エンジンの組み込みは、KORE/DB とKORE/IE を統合化することにより達成できる。2.2 節の「設計・選択活動」で要請された知識を構造化し利用する機構や知識の一貫性を保持管理するための

機能は、KORE/KR とKORE/EDEN を統合化することにより実現される。2.3節の「選択・再検討活動」で要請された簡潔なハイブリッドな知識表現形式は、KOREサブシステムを知識制御のための部品として用いることにより実現される。

関係テーブルの統一的な内部表現形式は、意思決定過程の四局面の全体を通してKOREサブシステム共通の情報であり、KOREサブシステムによるアプリケーションの橋渡しとして機能する。このことは、既存の意思決定支援ツール・技法（DSS、エキスパートシステム、意思決定分析法等）をKOREを用いて構築することにより、素直に結合させることに相当し、我々が目的とする知的な意思決定支援システムを構築するための技術的要請に応えるものである。

5. おわりに

本研究の目的は人間の意思決定を支援するための知的な意思決定支援環境を構築することである。本論文では、意思決定の定義を明らかにするためにサイモンの意思決定過程の四局面の定義を導入し、知的意思決定支援環境を構築するための課題とアプローチについて論じた。特に、このような知的意思決定支援環境を具体的に構築するための技術的要請として、既存意思決定支援ツール・技法をうまく取り込み統合化するための環境であるKOREの有効性を論じた。

KOREにおける基本機能は知識の制御と知識の蓄積・利用に大別される。つまり、知識の制御はKOREの各サブシステムに分散され、サブシステム固有の知識プログラミング技法を提供する。知識の蓄積・利用は関係テーブルとして統一化される。このことにより、各サブシステムは、KOREのアプリケーションを構築する際の部品（知識制御機構）となり、様々なアプリケーションを構築するために組み合わせて用いられる。

KOREのサブシステムは論理型言語上に構築された関係テーブルを共有することにより統合される。このような統合化は、いたづらにシステムの言語シンタックスを複雑にすることを避けるために、そして、知識表現パラダイム特有の強力な知識プログラミング技法を分散化（部品化）してKOREに実現するために導入された。このようなKOREの環境においては、サブシステムのユーザは、そのシステムを運用する上で他のサブシステムの技法や、共通的な内部表現を知る必要はない。例えば、KORE/DB を利用するユーザはそのシステムの使い方（制御の概念）と関係テーブルスキーマ（3種類）を把握することにより、KORE

/DB を通じて他のシステムにより構築された知識を検索したり、利用することが可能になる。

関係テーブルによるサブシステムの統合化の実現は、論理型言語の強力な関係記述能力をKORE/DB として集約し、積極的に利用したことによる。KOREの能力の拡張（例えば、知識ベースの2次記憶の利用等）は、このように能力を集約したKORE/DB を拡張することにより達成できる。

謝辞

日頃より御指導頂く当研究所北川敏男会長ならびに榎本肇所長に感謝致します。本研究を進めるに当たり、貴重なご意見を頂いた当研究所戸田光彦主任研究員に深謝いたします。また、この発表をするに当たり、当研究所片山佳則、平石邦彦の各研究員と多くの討論を致しました。彼等の貢献は大きく、かつ重要な役割を果たされた事に感謝致します。

尚、本研究は、第5世代コンピュータ・プロジェクトの一環として行われたものである。

参考文献

- [Alter 80] S.L.Alter : Decision Support System: Current Practice and Continuing Challenges, (Addison-Wesley, Reading, Mass., 1980).
- [Bobrow 83] D.G.Bobrow and M.J.Stefik : The LOOPS Manual, KB-VLISI-81-13, 1983.
- [Bobrow 84] D.G.Bobrow : IF PROLOG IS ANSWER WHAT IS THE QUESTION, The International Conference on Fifth Generation Computer System, pp.138-145 (1984).
- [Bonczek 81] R.H.Bonczek, C.W.Holsapple and A.B.Winston, A Generalized Decision Support System Using Predicate Calculus and Network Data Base Management, Operations Research, Vol.29, No.2, pp.263-281 (1981).
- [Brachman 83] R.J.Brachman : What IS-A is and isn't : An Analysis of Taxonomic Links in Semantic Networks, COMPUTER (1983) October.
- [Doyle 78] J.Doyle : Truth Maintenance System for Problem Solving, MIT, AI-TR-419, (1978).
- [Fikes 85] R.Fikes and T.Kehler: The Role of Frame-Based Representation in Reasoning, Communication of the ACM, vol.28, No.9, (1985).
- [Forgy 81] C.L.Forgy, OPS5 User's Manual, CMU-CS-81-135, July (1981).
- [Forgy 82] C.L.Forgy : Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence, 19, pp.17-37, (1982).
- [Gorry 71] G.A.Gorry and M.S.Scott Morton : A Framework for Management Information Systems, Sloan Management Review, vol.13, no.1, pp.55-70, (1971).
- [Hayes-Roth 83] F.Hayes-Roth, D.A.Watterman, and D.B.Lenat (Ed.): Building Expert System, Addison Wesley, (1983).
- [平石 86] 平石邦彦, 新谷虎松, 片山佳則: 問題解決支援環境KORE (その4) - 関係データベース・サブシステムKORE/DBとその概要 -, 情報処理学会第32回全国大会 (1986).
- [市川 83] 市川惇信編: 多目的決定の理論と方法, 計測自動制御学会 (1983).
- [片山 86] 片山, 新谷, 平石: 問題解決支援環境KORE (その3) - 知識表現サブシステムKORE/KRとその概要 -, 情報処理学会第32回全国大会, 5L-10, 1986.
- [Keen 78] P.G.W.Keen and M.S.S.Morton : Decision Support Systems: An Organizational Perspective, (Addison Wesley, Readings, Mass., 1978), 264.
- [Kowalski 77] R.Kowalski, Logic for Problem Solving (Elsevier North Holland, 1977), pp.31-44

- [McDermott 83] D.McDermott : Contexts and Data Dependencies : A Synthesis, IEEE , Vol. PAMI-5, No.3, pp.237-246 (1983) [Reiter 80] R.Reiter: A Logic for Default Reasoning, Artificial Intelligence, 13, pp.81-132, (1980).
- [Simon 79] H.A.Simon (稲葉他訳) : 意思決定の科学, 産業能率大学出版部 (1979).
- [Sacerdoti 74] E.D.Sacerdoti : Planning in a hierarchy of abstraction spaces, Artificial Intelligence, 5, pp.115-135, (1974).
- [榎木 84] 榎木, 中山: 多目的決定の動向と展望, システムと制御, 28-11, PP.619-627 (1984).
- [新谷 86a] 新谷, 片山, 平石, 戸田 : 論理型言語によるハイブリッドな問題解決支援環境KOREの設計—知的意思決定支援システムへの接近—, The Logic Programming Conference 86 (予稿集), pp.43-50, (1986).
- [新谷 86b] 新谷, 平石, 片山, 戸田: 関係テーブルに基づく推論エンジンKORE/IE , 情報33回全国大会, 7M-1, PP.1411-1412 (1986). (発表予定)
- [新谷 86c] 新谷, 片山, 平石: 問題解決環境KORE (その2) —知識記憶利用機構 KORE/EDENとその応用—, 情報32回全国大会, 5L-9, (1986).
- [新谷 85d] 新谷, 片山: 知識テーブル (その実現) —知識ベースにおける知識記憶管理方式とその応用—, 情報31回全国大会, 1P-8, (1985).
- [Stallman 77] R.M.Stallman, G.J.Sussman : Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis, Artificial Intelligence , 9, pp.135-196 (1977).