

PSIにおける推論エンジンKORE/IEの試作

6P-1

新谷虎松¹・〇二神浩道²(¹富士通国際情報科学研究所)(²富士通ソーシャルサイエンスラボラトリ)

1. はじめに

問題解決支援環境KORE (Knowledge Oriented Reasoning Environment) [新谷 86]における推論エンジン・サブシステムであるKORE/IEは、単独で用いることによりOPS5 [Forgy 81]に代表される前向き推論型プロダクションシステム(PS)として機能する。現在、KORE/IEはC-PROLOG及びQuintus-PROLOGを用いて、それぞれVAX-11/780、SUN3/52m上に実現されている[新谷 87]。今回、KORE/IEをPSI上に移植した。本報告ではPSI上のESPを用いた効率的な推論システムの実現方式について論じる。

2. システムの概要

PSI上のKORE/IEのシステムは以下(Fig 1)のように構成される。ルールコンパイラはひとつのルールベースをひとつのクラスとしてESPプログラムへ変換する。このESPプログラムは、推論メカニズムである認識-行動サイクルをESPの計算メカニズムに帰着させるものであり、[新谷 87]で得られたLHSプログラムとRHSプログラムに相当する、LHSメソッドとRHSメソッドを持つ。これらプログラムについては[新谷 87]に詳しいが、簡潔に述べると次のような機能を担う。LHSプログラムは、認識-行動サイクルにおける照合過程を高速化するためのものであり、OPS5のReteアルゴリズムと同様にWMの変化に着目することにより照合過程を効率的に達成する。RHSプログラムは、

競合解消により得られたインスタネーションを用いてルールの結論部を高速に実行するためのものである。ESPプログラムの特長は、これらプログラムをひとつのクラスとして管理可能とすることにより、コンパイル後のルールベースのモジュール性を実現することにある。このようなモジュール性は、PROLOG上のKORE/IEには無く(つまりルールベースは全て統一的な形式でフラットに先に述べたPROLOGプログラムとしてコンパイルされる)、KORE/IEにおけるルールベース間の協調問題解決機能をさらに有効に機能させるための重要なメカニズムとなる。KORE/Tableは、PSのワーキングメモリ(WM)に相当し、WMに対する操作を効率的に実現するための機構である。また、ウィンドウ・インターフェイスは、KORE/IEを容易に操作する環境をユーザに提供する。

2.1 KORE/Table

ESPプログラムに変換されたルールベースはWM要素の追加、削除、更新に伴って参照される。WMは一つ以上のWM要素で構成され、具体的なWM要素はWMクラスのインスタンスとして実現される。ESPにはPROLOGのassert, retract等にあたる機能が存在しない。そこで、WM要素の格納、検索、削除を動的にかつ高速に行う機構の構築が必要とされる。ESPを用いたKORE/IEではKORE/Tableによって、これらの機能を実現する。KORE/TableはKORE/IEのWMを操作する為に特殊化された一種のデータベースシステムでありWMそのものである。具体的には、KORE/Tableはクラス名kore-tableとして参照され、各WM要素クラスはクラス

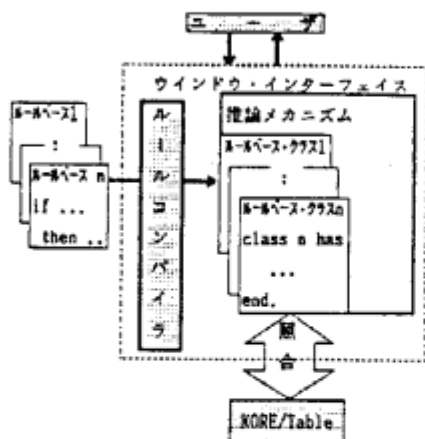


Fig 1. KORE/IEの構成

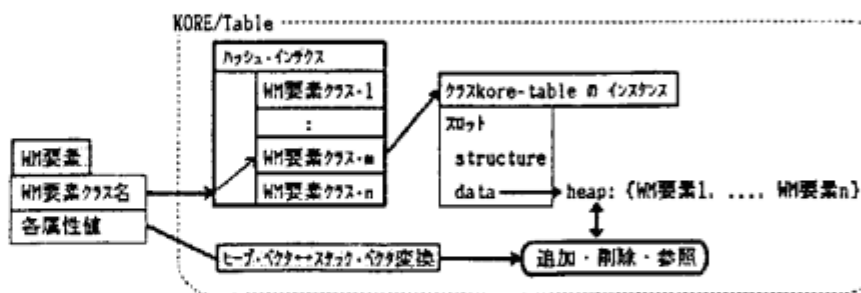


Fig 2. KORE/Tableの内部構成

A Prototype System of Inference Engine KORE/IE using PSI machine

Toramatsu SHINTANI¹, Hiromichi FUTAGAMI²¹IIAS-SIS, FUJITSU LIMITED, ²FUJITSU SSL, Ltd.

kore-tableのインスタンスとして実現される (Fig 2)。このインスタンスは、WM要素クラス名をキーにして、ルールのインスタンスに格納される。スロットstructureはWM要素クラスの構造 (パターン) の定義を格納する。スロットdataにはヒープに変換された各WM要素の属性値をバツの要素として持つヒープ・バツを格納する。WM要素を参照する際には、このヒープ・バツの各要素を順次検索して、ユニファイを行う。このように、KORE/Tableは単純な機構を用いて高速なWM要素の検索を行う。ここで、KORE/Tableの主な機能を紹介する。まず、KORE/IEにおいてWM要素クラスの構造定義は、literalizeコマンドを用いて次のように記述され、WMで用いて次のように記述され、WMで用いられるパターンのテンプレートとして宣言される。

literalize(WMクラス名, 属性名定義リスト),

ここで、属性名定義リストはパターンで用いられる属性とそのタイプ及びデフォルト値を規定するものである。このliteralizeコマンドは次のようなKORE/Tableのメソッドとして解釈実行される。

```
:add-structure(#kore-table, WMクラス名,
               属性名総数, 属性名バツ,
               属性値タイプバツ, 属性デフォルト値バツ),
```

また、WM要素クラスのインスタンスの追加、検索、削除のうち、例えば、検索機能はWM要素クラス名を指定して以下のメソッドで実現される。

```
:get-data(#kore-table, WMクラス名,
           (属性値1, ..., 属性値n)).
```

WM要素の検索・削除時には、それぞれのメソッドの第3引数のスタック・バツの要素に適宜に変数を組み入れることによって、PROLOG同様に柔軟なデータ検索を実行する。

2.2 ESPプログラムの構造

ルールコンパイラによって変換されたルールベースは下図 (Fig 3) のような、ルールベース名をクラス名とし

```
class <R-ベース名> has
  attribute
    watch-mode,
    strategy,
    cs;
  <LHSメソッド>;
  <RHSメソッド>;
end.
```

Fig 3. ルールベースの構造

た ESPプログラムである。ここで、スロットwatch-modeはデバッグ時に必要なトレース情報表示モードの情報を格納する。スロットstrategyはルールベースに定義されている競合解消戦略を格納する。また、スロットcsはルールベースが持つ競合集合を格納する。ESPプログラムに変換されたルールベースは全て、クラスie-rulebaseが直

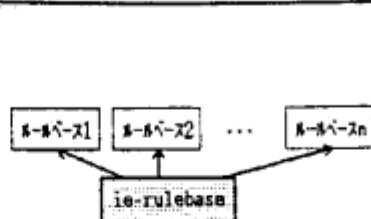


Fig 4.1

接継承している (Fig 4.1)。各ルールベースのLHSメソッドとRHSメソッドはクラスie-rulebaseを通じて呼び出される。また、協調問題解決を行う為に、ESPプログラム中で他のルールベースを直接継承することにより (Fig 4.2)、ルールベース間に階層性を持たせることができる。

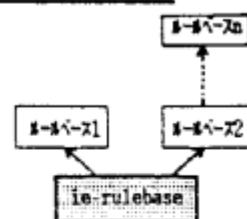


Fig 4.2

2.3 ユーザ・インターフェイス

KORE/IEでは SIMPOSのウィンドウ・システムを利用して、ユーザフレンドリなインターフェイスを提供する。インターフェイスは初期モード、デバッグモード、及び実行モードの3つのモードを用意している。初期モードではKORE/IEシステム全体に対する操作を行う。デバッグモードと、実行モードは初期モードから選択される。デバッグモード時には、ルールベースのデバッグを容易に行うためのツール群が用意されている。デバッグの済んだルールベースの実行は実行モードで行う。デバッグモードにおいてもルールベースの実行は可能であるが、トレースを行うための処理が多く含まれるので、実行効率は充分に発揮されない。そこで、初期モードにおいてデバッグ済みのルールベースをエキスパート・プログラムとして、実行モードに登録する。

3. おわりに

本稿では、PROLOG上で実現されたKORE/IEに基づきESPを用いた効果的な推論システムの実現方法について論じた。PROLOGで作成されたKORE/IEに比べると実行速度は向上している。これは、KORE/Tableが効率の良いWMの参照を行ったためである。また、ウィンドウを用いることにより、かなりシステムの操作性が向上した。

なお、本研究は第五世代コンピュータ・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

- [新谷 86] 新谷, 他: 論理型言語に3Dグラフィック問題解決支援環境KORE, The Logic Programming Conference '86 予稿集, pp43-50(1986)
- [Forgy 81] C.L.Forgy: OPS5 User's Manual CMU-CS-81-135, July., (1981)
- [新谷 87] 新谷: 推論エンジンKORE/IE-反駁能力に基づいた高速推論エンジンの実現-, The Logic Programming Conference '87, 予稿集, pp233-241(1987)