

論理設計エキスパートシステムLODEXにおける 推論エンジンの開発

2L-8

箕田依子, 及川さおり, 澤田秀穂, 橋本智佳子, 秋元晴雄, 角田多恵子, 丸山文志

富士通株式会社

1 はじめに

論理設計エキスパートシステムLODEXは、設計者の知識を表現した知識ベースを用いてVLSIの設計を支援するシステムである。本稿では、PSI上に開発中の機能設計のための推論エンジンについて述べる。

本推論エンジンは、機能設計支援のために以下の特徴を持つ。

- ① 設計の詳細化のための前向き推論と、前向き推論ルールの各前提条件の検証のために後向き推論を使用する混在型推論方式。
- ② 設計に現れる繰り返し作業をタスクとして格納したアジェンダに基づく実行機構。
- ③ 設計の試行錯誤をサポートするための柔軟な推論制御機構と後戻り機構。

2 知識表現

本システムで扱う知識は、設計対象の知識と設計者のノウハウである。

2.1 論理設計の世界モデルとワーキングメモリ

設計対象の知識とは、ハードウェアの概念を構成する要素についての知識である。要素間には、図1で示すような階層関係が存在するものがある。

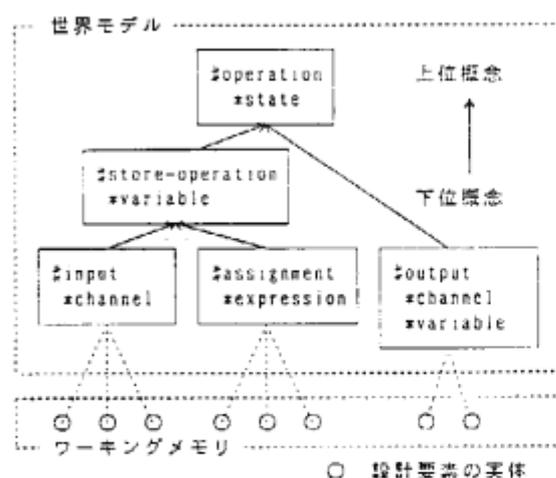


図1 論理設計の世界モデルとワーキングメモリ

論理設計の世界モデルの表現にオブジェクト指向を導入した。オブジェクト指向の継承機構を利用することにより、階層関係を自然に表現できるとともに、要素の記述は要素自身の持つデータとその操作手続きだけになるので管理しやすくなる。

図1は、“最下位の概念 inputが、データchannelの他に、データstate, variableを上位概念であるoperation, store-operationの継承によって持っている”こと、“inputとassignmentは store-operationである”ことを示している。

本システムでは、ESPのプログラムの枠組みであるクラスを用いて設計対象の概念構造を表現し、各クラスを雛形としたインスタンスを用いて設計データの実体を表わす。インスタンスで構成する世界が論理設計エキスパートシステムのワーキングメモリとなる。

2.2 ルール

設計者のノウハウはプロダクションルールで表現する。ルールの例を図2に示す。

```

前向き推論ルール
state-collapse(State1, State2) ::
  can-be-united(State1, State2)
  => :unite(State1, State2);
後向き推論ルール
can-be-united(State1, State2) <:-
  compatible(State1@operation, State2@operation);
  
```

図2 ルールの記述例

前向き推論ルールは、詳細化の知識を“タスクstate-collapseは、前提条件can-be-unitedが成立したとき、副作用を伴う手続きuniteを起動する”という形式で記述する。後向き推論ルールは、前向き推論ルールの前提条件の検証のために“can-be-unitedは、前提条件compatibleが成立するとき成り立つ”という形式で記述する。ルールを設計過程毎にモジュール化したものをKS(Knowledge Source)とする。KSには推論ルールの他にタスク選択、ルール選択、KSの停止のための情報を記述する。制御情報については後述の推論制御の項において詳細を述べる。

Development of Inference Engine of LODEX

Yoriko Minoda, Saori Dikawa, Shuho Sawada, Chikako Kusunoto,
Haruo Akimoto, Taeko Kakuda, Fumihiro Maruyama
FUJITSU Ltd.

この他、ルールの記述力を向上させるために、引数の拡張表現を設定した。これは、ある設計概念（クラス）に対して“すべてのインスタンスがある性質を満たす”、“ある条件を満たすインスタンスをすべて見つける”等を簡潔に記述するものである。

ルールベースは、KS毎にESPのクラスの記述に変換したものを実行形式とする。この際に拡張表現のルールは、SIMPOSのマクロ展開機構により処理用手続きに置き換えられる。

3 推論制御機構

設計の詳細化は、各過程で行うタスクを選択し、選択したタスクにルールのタスク部が照合する前向き推論ルールの実行に伴う副作用がワーキングメモリ上の設計データを変化させることにより遂行する。

3.1 タスクの制御

前項で述べたとおり、ルールは設計の一過程に対応するKSに格納されている。スケジューラは、予め決められた順序でKSを呼び出す。

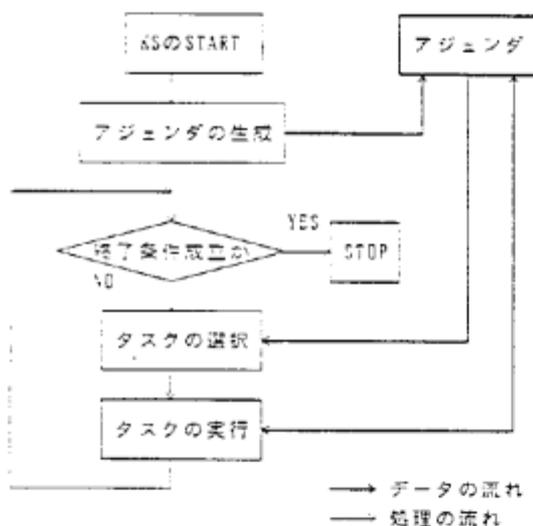


図3 アジェンダ制御の概念図

アジェンダは、KSのアジェンダ生成情報に基づきワーキングメモリを参照して生成したタスクの待行列である。アジェンダ制御情報に基づいたタスクの選択-実行をKSの終了条件が満たされるまで繰り返す。タスクの実行中にルールの実行部、または、アジェンダ制御情報によりタスクの生成、削除の要求が生じた場合は、その時点でアジェンダの内容を書き換える。

KSの終了判定は、KSに記述された終了条件を参照する。終了条件は、“アジェンダが空のとき”、“終了条件を記述したルールの成功”等である。

3.2 タスク実行の制御

タスクの実行は、タスク部が照合した前向き推論ルールを使用して行う。前向き推論の実行には、初めに前提

部が満たされたルールを実行する first hitと、ルールセット内のすべてのルールの前提部を検証した後、戦略に従って1つのルールを選択し実行する方式がある。後向き推論は、ユニフィケーションとバックトラックにより、前向き推論ルールの前提部の検証を行う。

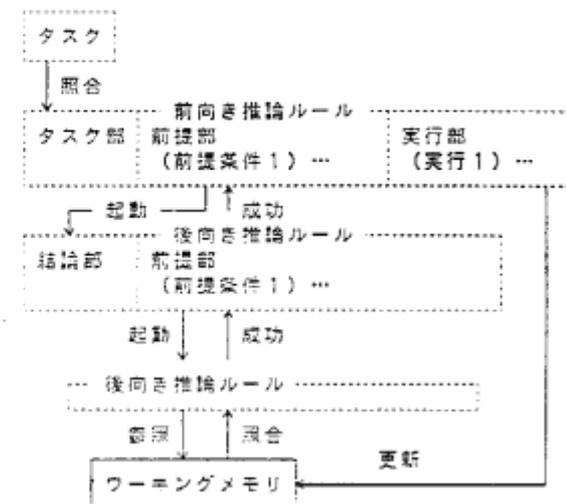


図4 タスク実行の概念図

図4に示すように、前向き推論ルールの前提部は、各前提条件が後向き推論で検証されたとき、すなわち、後向き推論ルールの前提部がワーキングメモリの内容に帰着した場合に成功する。後向き推論は、ワーキングメモリを直接参照する他に、前提部の各条件と結論部が一致する後向き推論ルールを起動し、その前提部の検証に帰着させることがある。これは、関係のあるルールだけを選んで推論することにより検証の効率化を図るためである。

上述の推論制御情報は、ルールの実行部等により動的に変更することができる。

4 その他の機能

推論過程の説明と復戻り時のデータの解放のために、ルールの適用とワーキングメモリ書き換えの情報を設計履歴として保存する機構を備えている。

5 おわりに

論理設計の知識表現にオブジェクト指向を導入したことにより、設計知識のモジュール化、対象表現の階層化を自然な形で実現することができた。

今後の課題として、協調型推論機構のために、やりなおしを効率的に行う機構とそれに伴うデータ変更を扱うワーキングメモリ管理機構の導入がある。

<謝辞>

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り、御支援頂いたICOT第五研究室室長(現 NTT)に深く感謝致します。