

論理設計エキスパートシステムLODEXにおける 2L-8 推論エンジンの開発

箕田依子, 及川さおり, 澤田秀徳, 楠本智佳子, 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。

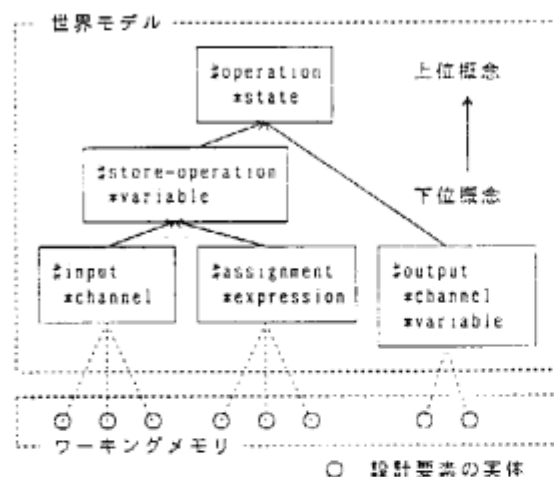


図1 論理設計の世界モデルとワーキングメモリ

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实体を表わす。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, Fumihiko Maruyama
FUJITSU Ltd.

ルールベースは、KS毎に ESPのクラスの認識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、SIMPON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。

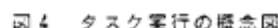
設計の詳細化は、各過程で行うタスクを選択し、選択したタスクにルールのタスク部が照合する前向き推論ルールの実行に伴う副作用がワーキングメモリ上の設計データを変化させることにより実行する。

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。



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。

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法がある。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。



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。

推論過程の説明と復元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。

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。

 $\nabla_{\mathbf{z}} \mathcal{L}(\mathbf{z})$

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第五研究室着下京長（現 NTT）に深く感謝致します。