オブジェクト指向推論方式

1D－8 -オブジェクト指向言語Koolaによる実現について-

濱口 敏英　渡守武 和記　松本 典子　西山 保
松下電器産業(株) 半導体研究センター

1. はじめに

現在、推論方式として、Reteアルゴリズム[1]に基づくインタプリタ型やコンパイル型の推論方式が提案されている。しかし、その実現にはLisp又はインタプリタ部のみC言語が使用されている場合が多い。今回、オブジェクト指向言語Koola[2]によるReteアルゴリズムを用いたコンパイル型推論方式を提案する。また、本方式を我々が開発している論理合成システム[3]に適用したことについても報告する。

2. オブジェクト指向推論方式の特徴

最近では、動的な知識をルールで静的な知識をフレームで表す知識表現が主流となってきている。そこで、オブジェクト指向言語で推論機構を構築することによりルールとフレームとを一体化させた推論システムを容易に実現できる。

本推論方式は、ルール記述からルールオブジェクトを生成し、そのルールオブジェクトにメッセージでデータオブジェクトを送ることによりfireすべきルールを決定して推論を進める。そのため、ルール記述からルールオブジェクトを生成するコンパイル型の推論方式となっている。さらに、高速化のためにReteアルゴリズムによる知識の構造化と途中結果の再利用とを用いている。また、ルールの記述中にはオブジェクトのメソッドを書くことができる等の点で記述性が良い。

3. 推論機構

3.1 クラス構成

クラスの構成を図1に示す。推論クラスは推論方法、競合解消戦略、推論の途中結果や候補の記憶などを記述したクラスである。主なクラスの概要は以下の通りである。

- Candidate(候補クラス)
  推論の照合に成功した情報を記憶するクラス
- InferenceStrategy(推論制御クラス)
  推論方法を記述したクラス
- Candidates(候補集合クラス)
  候補インスタンスを要素に持ち、かつ競合解消戦略を記述したクラス
- RuleSet(ルールセットクラス)
  メッセージで送られてきたデータによって実行可能となる候補を求めるクラス

他に、RuleSetクラスのスーパークラスであるSuperRuleSetクラスや全体の処理手順を記述したMetaRuleクラスなどがある。

データクラスは静的な知識とそれらの知識に関連したメソッドとを持つ。また、階層化により静的な知識の継承とメソッドの継承を有効に使っている。

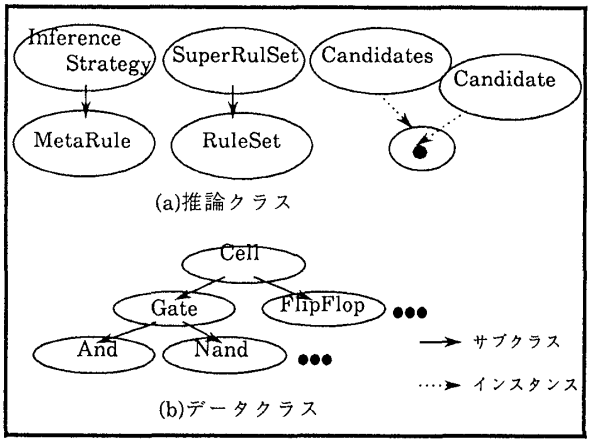


図1 クラス構成

3.2 高速推論

照合処理にはReteアルゴリズムを採用している。Reteネット上のトークンの流れをオブジェクト指向のメッセージセンディングにより実現する。そのため、ネットを4種類のノード(rootノード・1入力ノード・2入力ノード・terminalノード)で構成し、各ノードを部分照合メッセージや次に行うべきノードへ送るメッセージや途中結果修正メッセージやCandidateインスタンスを生成してCandidatesに追加するメッセージなどからなるメソッドとして実現する。

3.3 推論戦略

推論方法や競合解消戦略はメソッド継承と抽象クラスを使って実現している。

- Candidatesクラスの要素追加メソッドを変更したサブクラスを作るだけで違った競合解消戦略を実現する。
- InferenceStrategyクラスでは、すべてのRuleSetクラスのスーパークラスであるSuperRuleSetクラスを用いることによりRuleSetに独立な形で推論方法を実現する。

A Object Oriented Inference Method by Koola
Toshihide HAMAGUCHI, Kazunori TOMOTAKE, Noriko MATSUMOTO, Tamotsu NISHIYAMA
Matsushita Electric Industrial Co.,Ltd.

3.4 コンパイル方式

2段階のコンパイル方式を採用する。

① ルールコンパイラによりルール記述からRuleSetクラスを自動生成する。

② RuleSetクラスのコンパイルを行い、推論クラスおよびデータクラスとリンクして実行イメージを作る。

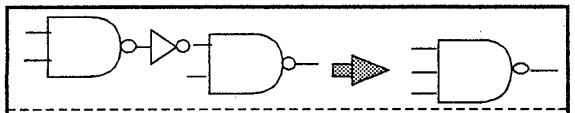
実行時には、この実行イメージにデータを送ることにより推論を進める。

4. Koolaによるシステムの実現

本方式の適用例としてオブジェクト指向言語Koolaを用いて論理合成システムを開発した。

4.1 ルール記述

回路最適化ルールの記述例を図2に示す。#priorityには自分より優先度の低いルール名を記述する。#connectionには回路パターンをフレーム形式で記述する。#conditionには回路パターン以外の制約条件を式で記述する。#conclusionには最適化後の回路をフレーム形式と式とを混在して記述できる。



```
RuleSet ReduceRule
|   nandinverternand |
    #priority
    #connection
        gate1   a_kind_of   :   nand
                input       :   [X1  |  L1]
                output      :   Y1  ;
        gate2   a_kind_of   :   inverter
                input       :   Y1
                output      :   Y2  ;
        gate3   a_kind_of   :   nand
                input       :   [Y2  |  L3]
                output      :   Y3  ;
    #condition
                [   gate1   fanout  ]   ==  1   ;
                [   gate2   fanout  ]   ==  1   ;
    #conclusion
        [|X4]   =   (   [X1  |  L1] +   [|L3]   );
        gate4   a_kind_of   :   nand
                input       :   [|X4]
                output      :   Y3  ;
```

図2 回路最適化ルールの記述例

(注)[X1 | L1] とは要素X1と残りリストL1のリスト
[| L1] とはリストL1

4.2 RuleSetクラスの構造

図3は図2のルール記述における#connectionからデータの接続関係(回路パターン)を解析して生成されるクラス定義のReteネット部分を図式化したものである。例えば、2入力ノードのnand_inverterの場合は、左のトークンと右のトークンとが接続している組を見つける部分照合メッセージと、照合に成功したトークンを次のノードに送るセルフメッセージ[self nandinverter: token to_nand: nandCells] と途中結果メモリを修正するメッセージとに変換される。

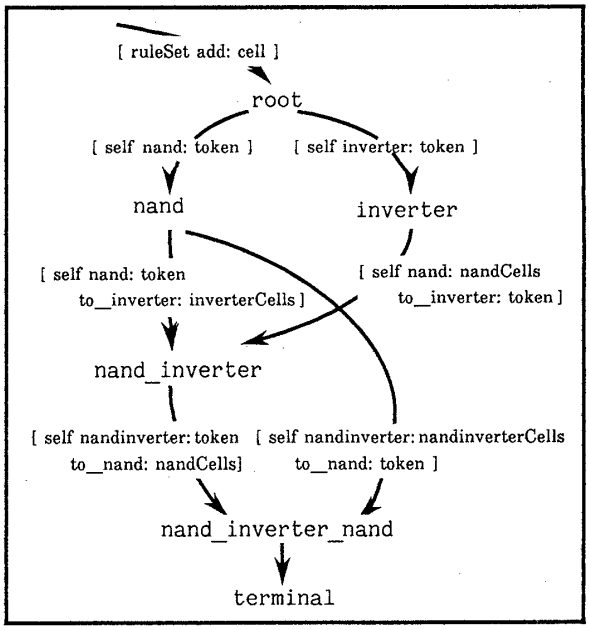


図3 ネットワークとメッセージ

4.3 推論処理

具体的な推論処理はInferenceStrategyクラスに記述した戦略に従って行われる。例えば、図3のRuleSetクラスに、変化したセルとしてinverterを送ると、まずrootノードはnandとinverterのノードにメッセージを送る。inverterノードは照合に成功し、途中結果メモリinverterCellsの修正を行なって、セルフメッセージ[self nand: nandCells to_inverter: token] を送る。このメッセージをnand_inverterノードが受け取り同様の処理を行う。terminalノードに到達すると、このノードはトークンからCandidateを生成してCandidatesの修正を行う。修正がすべて終わると競合解消戦略に基づいてCandidatesから選択されたルールを起動することによって推論が進む。

なお、MetaRuleクラスにはどのRuleSetクラスをどの推論方法で処理するか、およびその戦略順序等が記述されている。

5. おわりに

ルール記述からReteアルゴリズムに基づいたネットワークを作り、それに従ったメッセージセンディングによって候補を見つけるオブジェクト指向の推論機構を開発した。

今後、各クラス、メソッドの改良を行うことでさらに高速化を図る。また、今回適用した論理合成システムの実用化を行うとともに、汎用エキスパートシステムへ拡張して行きたい。

参考文献

1) Forgy,C.L.:" Rete : A first Algorithm for the Many Pattern / Many object Pattern Match Problem ", Artificial Intelligence 19,pp.17-39,1982

2) 渡守武 , 他 :" オブジェクト指向言語Koola -新しい概念と機能 - ", 情報処理学会第40回全国大会

3) 西山 , 他 :" 論理合成エキスパートシステムLODES", 計測と制御 Vol.27No.10,pp.923-924,19