

3 L - 7

## 制約概念を導入した知識表現言語

杉本 勉

(NTTデータ通信株式会社)

### 1. はじめに

最近、フレームやオブジェクト指向のモジュール性と、制約による関係記述を融合した知識表現言語の開発が盛んである[1],[2]。この背景には、フレームやオブジェクト指向知識表現はモジュール性を高めることには有効だが、フレームやスロット間の関係記述には弱いという欠点と、その欠点を補うために関係記述に強い制約指向パラダイムがあり、その結果としてこれらを融合したものが注目されるようになったのである。

そこで、我々は知識表現言語として制約概念とフレーム表現に注目し、現在開発を行なっている。本稿ではその中の一部の機能である制約伝播について、フレームに導入したときの考え方や実現方法について述べる。

なお、本システムはSicstus-Prolog vers.0.6で実現している。

### 2. フレームと制約概念

フレームにおけるデーモンの考え方是一種の制約充足(constraint satisfaction)である。ここで、制約充足とは、与えられた制約条件を満たす解を求めるということをいう。例えば、フレームの上位-下位関係を表すスロット(isa, instance\_of)を考える。通常、あるフレームのisaスロットを書き換えるとif\_addedデーモンが起動し対応するフレームのinstance\_ofスロットの記述を書き換える。この動作は、常に全体の知識の整合性をとるという制約条件を満たすために行なわれたものであり、従って制約充足機構の一つの実現方法であると見ることができる。デーモンにはif\_addedデーモンのほかにif\_neededデーモンやif\_removedデーモンがあるが[3]、これらはいずれもあるフレームに対して操作した後の手続きのことである。すなわち、デーモンの考え方はデータの流れが決まっている場合には有効であるが、データがどのように伝播するかが実行時にしかわからない場合には、対応できない。この問題を解決するために、スロット間の関係(すなわち制約)だけを記述し、あとは制約伝播(constraint propagation)や制約充足の技術を利用することが有効である。

次節では、このような場合について例をあげ、制約概念を導入した解決法について述べる。

### 3. 制約伝播による制約充足

図1のような交差点の信号を例に述べる。

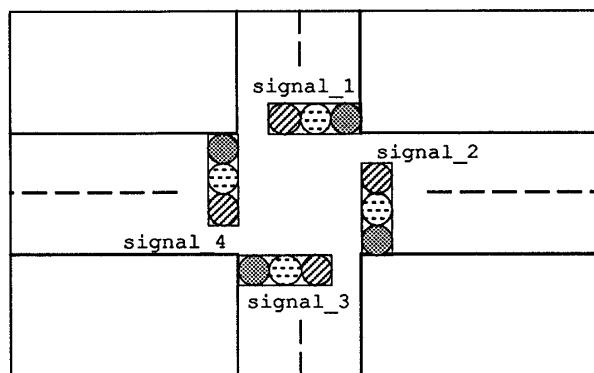


図1 交差点

図1の交差点(intersection\_1)の信号の関係は図2-a,bのように表すことができる。

```
fr signal_1 has
  isa : signal ;
  part_of : [intersection_1] ;
  state : red.
```

図2-a 信号の知識表現の例

```
fr intersection_1 has
  isa : intersection ;
  consist_of : [signal_1, signal_2,
                signal_3, signal_4] ;
  constraints :
    [signal_1@state == signal_3@state, %1
     signal_2@state == signal_4@state, %2
     signal_1@state != signal_2@state]. %3
```

図2-b 交差点の知識表現の例

ここで、交差点(intersection\_1)の信号の間に満たさるべき制約がconstraintsスロットに記述されている。この制約とは、「お互いの正面に位置する信号の表示は等しく(%1,%2)、ななめに位置する表示は異なる(%3)」というものである。これらの記述は従

來のフレーム表現では、各信号フレーム(signal\_1,...)のスロットにデーモンとして記述することになる。そして、どの信号の状態が変化しても全体の整合性がとれるようにするために、手続きをかなり記述しなければならない。また、これらの記述が各フレームに分散しているのでスロット間の関係がわかりにくくなる。これを解決するために、本稿で提案するような、複数のフレームにまたがる関係は上位フレームに記述し、制約として扱う方法が有効となる。

図2の定義だけで、どこの信号の状態(stateスロットの値)を変化させても制約伝播によって信号の状態は変化し全体の整合性を保つことができる。これは制約記述から得られる制約ネットワークを利用することによって、制約伝播のパスを導くことができるからである。これは以下のように行なわれる。

まず、図2-bのconstraintsスロットの記述から図3-aにある制約ネットワークが得られる。この状態ではまだ伝播するためのパスは決まっていない。そして例えばsignal\_1のstateが変更されると、これを頂点とした制約フローグラフが生成され(図3-b)、これが伝播のパスとなる。

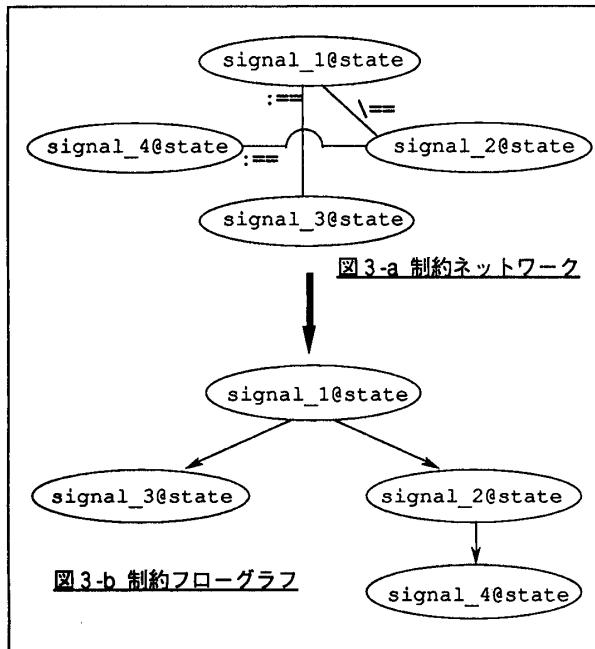


図3 制約ネットワークから制約フローグラフへ

制約フローの伝播は探索と同様、縦型と横型があるが、現在は横型を採用している。すなわち、signal\_1@stateが変わるとsignal\_3@stateとsignal\_2@stateへ伝播しこれらの制約はそれぞれ":=="と"\=="なのでこれらを満たすようにそれぞれのスロットが更新される。さらにsignal\_2@stateからsignal\_4@stateに伝播する。

この構造変換によって、ループ状になっている制約ネットワークにおける制約違反の検出も行なうことができる。例えば、信号機の状態が図4のように定義されているとする。

```
constraints :
[ signal_1@state \== signal_2@state,
  signal_2@state \== signal_3@state,
  signal_3@state \== signal_4@state,
  signal_4@state \== signal_1@state ]
```

図4 交差点の知識表現の例(2)

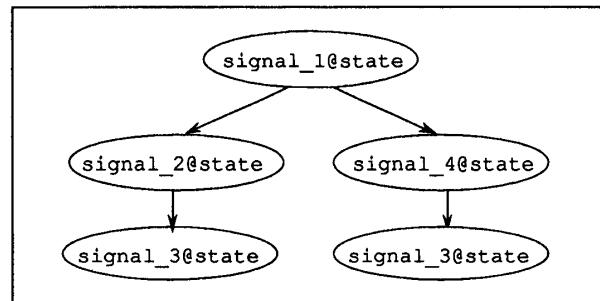


図5 図4のデータフローグラフ

先と同様にsignal\_1のstateが変更されると、図5のような制約フロー構造が生成される。ここで、制約フロー構造のリーフにsignal\_3@stateが2回出てくる。それぞれ伝播してくるパスが違うので、二重に値が書き換えられると、制約違反になりこの値の書き換えは無効になる。

#### 4. おわりに

本稿では現在開発を進めている制約概念とフレームを融合した知識表現言語について、制約伝播機能を中心に述べた。本システムは定性推論のための知識表現言語を目指しており、今後はそのために必要な制約充足機能(弛緩法、併合法)や不等式処理などを組み込むことを考えている。

#### 参考文献

- [1] 中島, 制約伝播機構を内蔵するオブジェクト指向言語:COOL, 情報処理学会論文誌 Vol.30, No.1, 1989.
- [2] Yokoyama,T., An Object-Oriented and Constraint-Based Knowledge Representation System for Design Object Modeling, Proc.of The Sixth Conf. on AI Applications, IEEE, 1990.
- [3] P.H. ウィンストン他, LISP, 培風館.