

TM-1177

デフォルト制約に基づく主観的評価の一貫性
保持について

新谷 虎松 (富士通)

May, 1992

© 1992, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

デフォルト制約に基づく 主観的評価の一貫性保持について

新谷虎松

Toramatsu Shintani

(株)富士通研究所 国際情報社会科学研究所
IIAS-SIS FUJITSU LABORATORIES LTD.

概要 計画や設計などの合成型の問題においては、等式や不等式を用いた関係式として表現される制約のほかに、デフォルトや好みとして特徴付けられる制約（デフォルト制約）も必要とされる。デフォルト制約は、問題解決のために、とりあえず準最適解を得るための選好条件として用いられる。本論文は、ルールプログラミングにおけるデフォルト制約の実現方式とその応用について論じる。デフォルト制約を宣言的に記述し利用するためには、制約を能動的に機能させるための機構が必要である。本論文では、TMS的な機構を拡張することによりデフォルト制約を実現する。応用例として、AHPに基づく主観的評価の一貫性保持について論じる。主観的評価の一貫性保持にはTMS的な機構が利用できるが、TMS的な枠組みでは好みが効果的に反映されないので不十分であった。デフォルト制約を実現したルールプログラミングにより、主観的評価の一貫性は効果的に保持される。

1 まえがき

制約は、解くべき目標を宣言的に記述することにより問題の解決を図る手段であり、知識情報処理において非常に有用なパラダイムである[1][9]。制約の宣言的な性質は、宣言的なプログラミング言語にとり都合の良いものである。近年、論理型プログラミングの宣言的な枠組みに制約機能を取り入れることによる制約論理プログラミングの有用性が議論され、多くの実用的な制約論理型言語が実現されている[9]。これら、制約論理型言語における制約は、等式や不等式を用いた関係式として表現され、能動的に評価

される。計画や設計問題に代表される合成型の問題においては、このような式として表現される制約の他に、デフォルトや好み（もしくは、優先度）として特徴付けられる制約の必要性も議論されている[2][12]。本論文では、特に、これら制約を区別するために、前者の制約を固い制約、後者の制約をデフォルト制約と呼ぶ。固い制約は、問題解決において、必ず満足しなければならない前提条件である。一方、デフォルト制約は、問題解決のために、とりあえず準最適解を得るための選好条件として特徴づけられる。

本論文は、ルールプログラミング[3]におけるデフォルト制約の実現方式とその応用について論じる。筆者等は、論理型言語を用いてルールプログラミングのためのルールコンパイラを構築している[13]。本システムは、KORE/IEと呼ばれ、ルールを論理型言語へコンパイルすることにより、ルールの実行を特別な解釈実行系を介せずに論理型言語の直接的な実行過程として実現する[15]。本システムでは、OPS5[6]的なルールプログラミングとしてルールを宣言的に記述できる一方、論理型プログラミングの利点を取り入れることができる。これにより、制約を実現するための枠組みを得ている。

制約を宣言的に記述し利用するためには、制約を能動的に機能させるための機構が必要である。本研究では、デフォルト制約のこのような機能を実現する枠組みとして、TMS[5]的な機構を利用する。本機構は、KORE/IEにおける非単調推論機能[16]として、仮定に基づく推論の矛盾を解消する。本論文では、この機構をデフォルト制約の実現という視点で拡張する。デフォルト制約は、TMS的な枠組みに好みや優先度を扱う機構を取り入れることにより実現する。デフォルト制約の応用例として、代替案選択支援機構CDSS[14]で用いられるAHP[10]に基づく主観的評価の一貫性保持について論じる。主観的評価の一貫性保持にはTMS的な機構が利用できるが、TMS的な枠組みでは好みは効果的に反映されないので不十分であった。デフォルト制約を実現したルールプログラミングにより、主観的評価の一貫性は効果的に保持される。

本論文の構成は次の通りである。2章では、デフォルト制約とその表現形式について述べる。3章では、ルールプログラミングにおけるデフォルト制約の実現方式について述べる。4章では、デフォルト制約を用いた主観的評価の一貫性保持について述べる。

5章では、ルールプログラミングから見た本研究の特長を述べる。6章ではまとめを述べる。

2 デフォルト制約

デフォルト制約は、問題解決のために、とりあえず準最適解を得るための選好条件として用いられる。固い制約が数式を用いて変数間の関係を宣言的に記述するのと同様に、デフォルト制約は問題を構成する対象の集合 $O(=\{o_1, o_2, \dots, o_n, d_{n+1}, d_{n+2}, \dots, d_m\})$ における不都合な関係を宣言的に記述する。ここで、対象 d_i は対象 o_i とその優先度 $p_i(>0)$ の対として次のように表現される。

$$d_i = \langle o_i, p_i \rangle$$

対象 d_i をデフォルトと呼ぶ。デフォルト制約には一つ以上のデフォルトが含まれる。対象の集合 O は、問題解決に先だって与えられる。デフォルトは、問題解決において仮定的に用いられる情報であり、将来、問題解決過程で不都合が生じれば修正される。たとえば、デフォルト制約は、対象の集合 O の要素 $(=\{o_2, d_3, d_5\})$ を用いて次のようにルールとして記述される。

$$o_2 \wedge d_3 \wedge d_5 \rightarrow \text{false}$$

本デフォルト制約は、対象 o_2, d_3, d_5 が同時には成立しないことを表している。そこで、デフォルト制約解消機構は、デフォルトのひとつを取り消す（もしくは、修正する）ことによりデフォルト制約の解消を図る。具体的には、デフォルト制約解消機構は、デフォルト制約で用いられるデフォルトを見つけ、それらデフォルトの優先度に従ってデフォルトのひとつを修正する。ここで、デフォルト制約で用いられるデフォルトを原因デフォルトと呼び、その集合を $D (\subset O)$ で表す。本例では、 $D = \{d_3, d_5\}$ となる。たとえば、デフォルト d_5 の優先度がデフォルト d_3 の優先度より小さければ（つまり、 $p_5 < p_3$ ）、 d_5 の修正が最初に試みられる（もし、デフォルトの優先度が同じなら、任意のデフォルトが一つ選ばれ、その修正が最初に試みられる）。 d_5 の修正により新たな不都合が生じ

たなら（つまり，他のデフォルト制約に抵触すること）， d_5 の修正を取り消し，次に d_3 の修正が試みられる．以上により，デフォルトをひとつ修正することによるデフォルト制約の解消方式を実現する．

問題解決において対象の集合 O は，推論ルールの実行に伴い拡張もしくは縮小される．ここで，推論ルールにより付加された対象を r_i と表記する．拡張もしくは縮小された対象の集合を $O' (= \{o_1, o_2, \dots, o_m, d_{n+1}, d_{n+2}, \dots, d_m, r_{m+1}, r_{m+2}, \dots, r_l\})$ と表記する．対象の集合 O' は，推論ルールの実行により新たな対象 r_i が付加されることにより拡張される．また，推論ルールの実行に伴い対象の集合 O' の要素である対象 o_i, d_i もしくは r_i が削除されることにより，対象の集合 O' は縮小される．対象の集合 O' の拡張方式は，推論システムに依存する．本研究では，推論システムとしてプロダクションシステムを用いているので，プロダクションシステムで用いられる競合解消戦略により拡張方式が一意に決定される．つまり，デフォルト制約による拡張は必ず一つに決定される．

デフォルト制約は，推論ルールにより付加される対象 r_i を用いて記述することも可能である．対象 r_i を用いたデフォルト制約は，問題解決過程で将来，生じる可能性のある不都合な関係を記述するものである．たとえば，対象の集合 O' の要素 $(= \{o_i, d_4, r_5, r_8\})$ を用いて次のように記述できる．

$$o_i \wedge d_4 \wedge r_5 \wedge r_8 \rightarrow \text{false}$$

本デフォルト制約は，対象 o_i, d_4, r_5 および r_8 が同時には成立しないことを表している．デフォルト制約解消機構は，先に述べた対象の集合 O から構成されるデフォルト制約の解消方式と同様にその解消を図る．ここでは，特に，次のように原因デフォルトの集合 D を求める．最初に，デフォルト制約で直接的に用いられたデフォルトの集合 D' を求める（例では， $\{d_4\}$ ）．次に，デフォルト制約で用いられた対象 r_i （例では， r_5 および r_8 ）を導きだす推論連鎖木に現われるデフォルトの集合 R を求めることにより，

$$D = D' \cup R \quad (\text{ここで, } D' \subseteq O, R \subseteq O)$$

として得る。もし、デフォルト制約の解消過程でデフォルトの集合Rの要素のひとつ d_i が修正されたなら、 d_i に依存した推論連鎖木を消去する。本研究では、推論過程で推論連鎖木を記録することにより、デフォルトの集合Rを求めるための情報や推論連鎖木を消去するための情報を得る。

ここで、デフォルト制約の例を示す。例えば、会議室と時間に関連した次の会合のスケジューリングを考える。

(1) 会合時間は、10:00とする。

(2) 会議室は、できるだけ801号室より813号室が良い。

(1)は、必ず満たさなければならない制約であり、固い制約に相当する。(2)はデフォルト制約であり、会合のスケジューリングを達成するための選好条件として用いられる。

(1),(2)で言及された対象の集合Oの要素は次のように表記される。

$$O = \{\text{meeting}(10:00), \langle \text{room}(801), 10 \rangle, \langle \text{room}(813), 20 \rangle\}$$

ここで、 $\text{meeting}(10:00)$, $\langle \text{room}(801), 10 \rangle$, $\langle \text{room}(813), 20 \rangle$ は、それぞれ、会合時間、デフォルトとしての801号室、デフォルトとしての813号室を表している。(2)より、 $\text{room}(813)$ の優先度（ここでは、20）は $\text{room}(801)$ の優先度（ここでは、10）より大きく設定される（優先度の差の大小はここでは特に意味はない）。本スケジューリングのデフォルト制約は次のように表現される。

$$\text{meeting}(10:00) \wedge \langle \text{room}(801), 10 \rangle \wedge \langle \text{room}(813), 20 \rangle \rightarrow \text{false}$$

制約解消機構は、原因デフォルトの集合D（ここでは、 $\{\langle \text{room}(801), 10 \rangle, \langle \text{room}(813), 20 \rangle\}$ ）を求め、デフォルトの優先度に従い一つのデフォルト（ここでは、 $\langle \text{room}(801), 10 \rangle$ ）を修正し、制約解消の結果として次のような対象の集合O'の要素を導き出す。

$$\{\text{meeting}(10:00), \langle \text{room}(813), 20 \rangle\}$$

もし、制約解消過程において、813号室の利用が都合悪くなれば（つまり、813号室の利用が他のデフォルト制約に抵触すること）、801号室が選択され、次のような対象の集合O'の要素を導きだす。

{meeting(10:00),<room(801),10>}

3 デフォルト制約の実現

本章では、KORE/IEにおけるデフォルト制約の実現方式について論じる。ここでは、2章で論じた問題を構成する対象の集合OおよびO'の要素は、ワーキングメモリ（WM）要素を用いて表現する。デフォルト制約は、プロダクションルールを用いて表現され、その条件部に条件要素として都合の悪いWM要素の組み合わせを記述する。デフォルトはWM要素に優先度を表す情報を付加することにより表現される。デフォルト制約は、デフォルト制約解消機構を呼び出し、デフォルトの優先度（もしくは、好み）に着目して、都合の悪いWM要素の組み合わせを解消する。

3.1 デフォルトの表現

KORE/IEにおいて、WM要素は、パターン記述として、クラス名および、いくつかの"スロット=値"対であるスロット記述を用いて、次のように表記される。

クラス名 (スロット₁=値₁, ..., スロット_n=値_n)

デフォルトは、このパターン記述に、(a)デフォルト、(b)デフォルトの取り消し、および(c)デフォルトの好み（つまり、優先度）を表すスロットを付加することにより表現する。(a)および(b)は、制約解消機構がデフォルト制約を解消する際の制御情報を保存するためのものである。具体的には、(a)、(b)および(c)を記述するためのスロットとして、それぞれ"default"、"proof"および"preference"がある。defaultおよびproofスロットの値は、"true"もしくは"nil"をとり、preferenceの値は数値である。たとえば、優先度1

0 のデフォルトは、default, proof, preferenceスロットの値がそれぞれ、true, true, 10であるパターン記述として表現される。もし、パターンを記述において、default, proof, preferenceスロットの値を特に指定しなければ、それぞれ仮の値としてtrue, trueおよび1が割り付けられる。これは優先度1のデフォルトを表現したことに相当する。

KORE/IEにおいて、たとえば、デフォルトは、"make_default"コマンドを用いて生成される。make_defaultコマンドは、makeコマンド（新たに生成するWM要素を引数とする）と同様に、次のように用いられる。

```
make_default(room(number=813))
```

この時、default, proof, preferenceスロットの値は、それぞれ、true, true, 1となる。

proofスロットの値がnilの場合は、便宜上、パターンの前に"\\"を付加して表すこともできる。たとえば、次のmakeコマンドにより生成されたパターンのproofスロットの値はnilが設定される。

```
make(\room(number=801))
```

3.2 デフォルト制約の表現

デフォルト制約は、ルールの結論部に手続きsolve_itを記述することにより表現する。手続きsolve_itは、デフォルト制約解消機構に相当し、2章で論じたデフォルト制約を解消するための具体的な枠組みを提供する。例えば、会合のスケジューリングに関連して、"できるだけ10時に813号室は使いたくない"というデフォルト制約は、KORE/IEにおいて次のように表現される。

```
nogood:
    if time(value=1000) & room(number=813)
    then solve_it(preference).
```

ここで、ルール記述において、":"の左に示された"nogood"はルール名である。ルールのシンタックスは、Prologの項を記述するためのシンタックスを取り入れている。"if"

と"then"の間に条件部(LHS)が記述され、LHSはいくつかのLHSパターンと呼ばれるパターン記述がデリミタ"&"をはさんで並べられる。"then"と"."の間に結論部(RHS)が記述され、RHSは、ワーキングメモリ操作コマンドの他、Prologの述語として定義される手続きがデリミタ"&"をはさんで並べられる。ルール of 優先順位は、普通、競合解消戦略により競合解消時に決定される。デフォルト制約は、通常のルールよりも常に優先順位が高くなるように設定される。デフォルト制約間の優先順位は、競合解消戦略で一意に決定される。つまり、優先順位が同一なデフォルト制約は、前もって与えられる競合解消戦略によりユニークに一つ決定される。

ルールnogoodにおいて、2つのLHSパターンは、default, proof, preferenceスロットの値を特に指定していないので、それぞれのスロットには3.1節で述べた仮の値が用いられる。つまり、それぞれのLHSパターンは優先度1のデフォルトを表現したことに相当する。

デフォルト制約におけるLHSパターンのpreferenceスロット値は、照合時に次のように特別に処理される。もしpreferenceスロットの値が仮の値であれば、将来、WM要素との照合時に（WM要素でpreferenceスロットの値が指定されていれば）、具体的な値（つまり、そのWM要素のpreferenceスロットの値）と置き変わる。これは、問題解決過程においてデフォルトの優先度を効果的に使うための工夫である。また、LHSパターンでpreferenceスロットの値を具体的に指定した場合、LHSパターンが表すデフォルト間の優先順位を保存するようにWM要素との照合が行なわれる。つまり、照合時には、preferenceスロットへの値の代入および参照は行なわれずに、デフォルトの優先度で決定される順位のみが着目される。これは、デフォルト制約の解釈を保存するための処理である。

デフォルトの優先度は、普通、デフォルトのpreferenceスロットの値が用いられる。さらに、デフォルトの他のスロット値を処理／比較することにより決定できる。具体的には、デフォルトの優先度は、手続きsolve_itの引数で与える手続きで決定される。例えば、先に例示したルールnogoodでは、手続きsolve_itの引数としてpreferenceを与えて

いる。手続き `preference` は、システムに組み込みの手続き名であり、デフォルトの `preference` スロットの値に着目してデフォルトの優先度を決定する。本システムでは、手続き `solve_it` の引数として `preference` を用いる場合、便宜上、その引数を省略することができる。4 章では、デフォルトの優先度を一対比較値の整合度により決定するために、手続き `solve_it` の引数として `decrease_IR` を与えている。手続き `decrease_IR` は、単にデフォルトのスロットの値を比較するのではなく、デフォルト制約を効果的に解消するための基準に従ってデフォルトの優先度を決定する（手続き `decrease_IR` の詳細は 4.3 節で述べる）。

ここで、KORE/IE におけるデフォルト制約の例を示す。図 1 は、会合のスケージュリングに関連した実行例である。ルール記述は図 2 で示される。ルール `nogood1` および `nogood2` は、デフォルト制約である。これらデフォルト制約は、それぞれ、次のことを主張している。

ルール `nogood1`: "できるだけ10時に813号室は使いたくない。"

ルール `nogood2`: "できるだけ801号室は使いたくない。"

ルール `revise1` および `revise2` は、デフォルトが取り消された場合に起動され、取り消されたデフォルトの代替案を主張するためのものである。

図 1 の実行例において、デフォルトには、`"*` が目印として付加されている。コマンド `make_default` はデフォルトを生成し、コマンド `run` は前向き推論を起動させるために用いられている。実行例では、コマンド `run` による前向き推論のトレース情報（起動ルールおよび WM）が表示されている。例えば、図 1 左のトレース情報において、(1) は、起動したルール名とその条件要素とマッチした WM 要素のタイムタグを表す。(2) および (3) は、(1) のルールが起動した後の WM 要素を表している。":" の前の数字は、タイムタグを表している。WM 要素の表示には、ユーザ定義のスロットのみが用いられ、デフォルトに関連したスロットは省略されている。(2) の WM 要素は、手続き `solve_it` により生成されたものであり、デフォルトの取り消しを示している（`"\` が付加されている）。

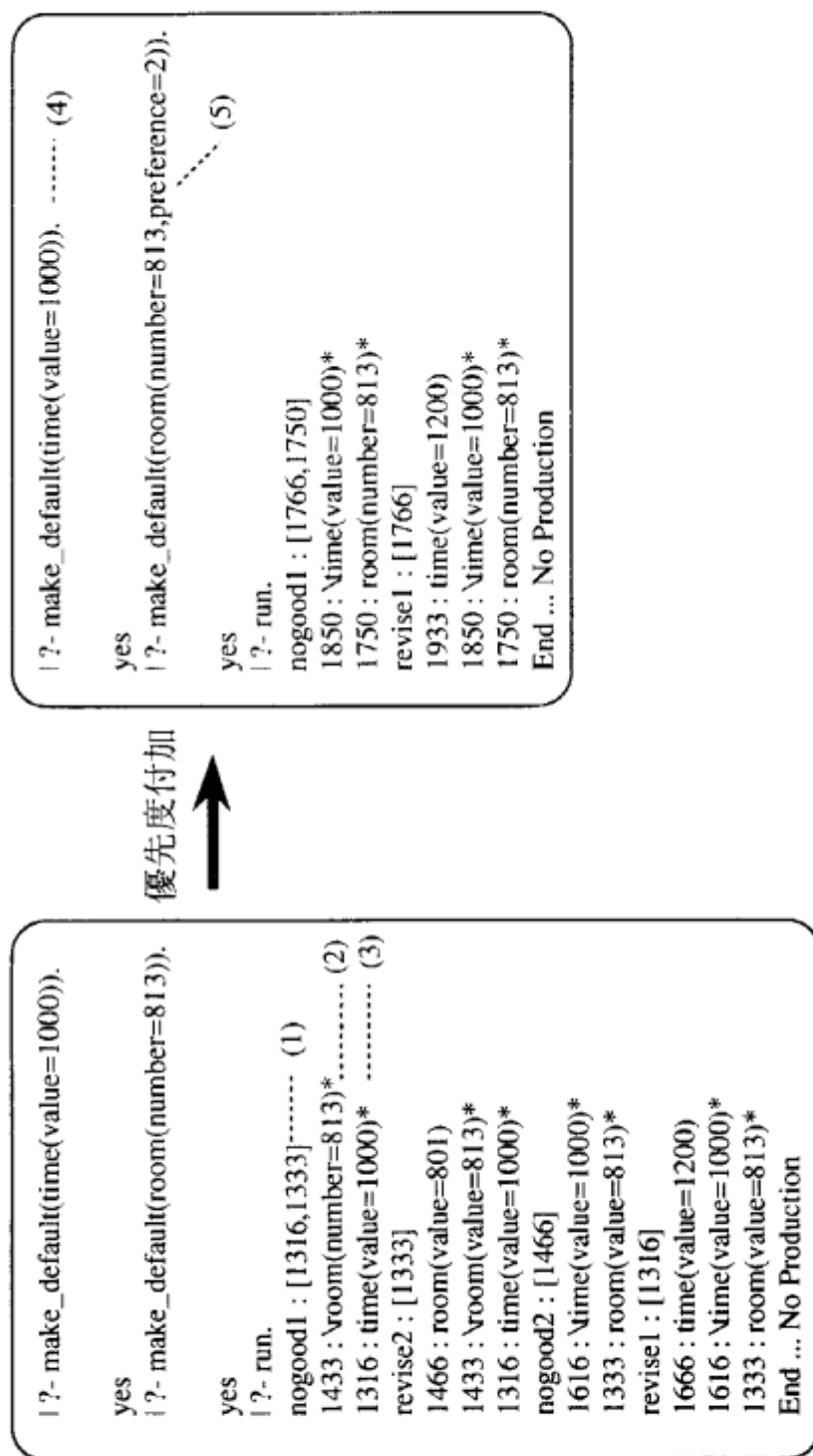


図 1 . 制約ルールの実行例

```

nogood1:
  if time(value=1000) & room(number=813)
  then solve_it(preference).

nogood2:
  if room(number=801)
  then solve_it(preference).

revise1:
  if \time(value=1000)
  then make(time(value=1200)).

revise2:
  if \room(number=813)
  then make(room(number=801)).

```

図2. ルールの記述例

図1左では、あらかじめ与える2つのデフォルトの優先度が仮の値の場合である（本例では、`preference`スロットが省略されているので、それぞれの優先度が仮の値として1になる）。この時、デフォルト制約は上で示したように解釈／実行される。最終的に、デフォルト制約解消機構（つまり、手続き`solve_it`）により制約解消が解消され、会合は、813室で10時以外の時間で開催されることになる。

図1右は、(4)で表すデフォルト（時間10時）よりも(5)が表すデフォルト（部屋813室）の優先度を大きくした場合の実行例を示している。デフォルトの優先度を具体的に与えたことにより、ルール`nogood1`で表すデフォルト制約は照合時に次のように解釈／実行される。

ルール`nogood1`:"できるだけ10時以外に813号室を使いたい"

図1左と図1右の比較で示すように、扱うデフォルトの優先度に差をつけることにより、デフォルト制約の解釈は変更され、これにより、効果的に制約の解消が実現される。

3.3 制約解消機構`solve_it`

手続きsolve_itは、KORE/IEにおける非単調推論のために文献[18]で構築されたTMS的な機構へ、デフォルトの優先順位を考慮するメカニズムを付加することにより実現する。

- ステップ1： 推論連鎖の記録からデフォルト制約を呼び出す原因となったデフォルトのリストDを求める。そして、ステップ2へ行く。
- ステップ2： リストDを指定された手続きProcを用いてソートしその各要素へ制約解消のための情報を付加する。そして、ステップ3へ行く。
- ステップ3： もし、リストDが空リストなら、デフォルト制約の解消は失敗となり、手続きを終了する。さもなければ、ステップ4へ行く。
- ステップ4： リストDの先頭にあるデフォルトDheadの取り消しを宣言する。そして、ステップ5へ行く。
- ステップ5： もし、ステップ4の処理後の新たな推論連鎖の結果、新たなデフォルト制約に抵触するなら、ステップ4の宣言を無効にして、DからDheadを取り除いてステップ3へ行く。そうでなければ、ステップ6へ行く。
- ステップ6： デフォルト制約解消は成功し、手続きを終了する。

図3. 手続きsolve_itの概略

図3は、手続きsolve_itのアルゴリズムの概略を表している。ここでは、ステップ2が本質的である。ステップ2は、制約解消でデフォルトの優先度（好み）を反映させるためのものである。ステップ1で得られるデフォルトのリストDは2章で述べた原因デフォルトの集合に相当する。本手続きのステップ2による特長は、制約解消に関連してデフォルトの優先順位を決定する手続きProc（たとえば、ソート手続き）を指定できることである。ステップ2では、手続きProcを用いてリストDに含まれるデフォルトをチェックする順番およびステップ4でデフォルトを取り消す際に加える付加的情報が得られる。具体的には、指定された手続きProcを用いて、リストDをソートし、リストDの要素に付加的情報を加える。例えば、図2で示すデフォルト制約の手続きsolve_itで用

いられた`preference`は、デフォルトの`preference` スロットの値に着目してリストDを昇順にソートする手続きである（ここでは、特に、リストDに付加的情報は加えられない）。これにより、制約解消時に、`preference`スロットが表す優先度の小さいデフォルトが最初に取り消される。

手続き`solve_it`で指定する手続きは、`preference`以外にも、リストDを処理するための手続きとしてユーザが定義することも可能である。これにより、問題に依存してデフォルト制約を効果的に解消する枠組みを得る。例えば、4章では、リストDを処理する手続きとして`decrease_IR`を導入している。手続き`decrease_IR`は、問題に即してリストDをソートする他に、デフォルトを具体的に更新するための付加的情報をリストDへ加える（詳細は、4章で論じる）。

ステップ4において、デフォルトの取り消しは、デフォルトがある推論ステップまで推論サイクル（つまり、認識-行動サイクル）を戻し、付加的情報を加えてデフォルトの取り消しを宣言する。デフォルトの取り消しの宣言は、3.1節で論じたように、デフォルトを表すWM要素の`proof`スロットの値を`nil`に変更することにより実現する。付加的情報は、デフォルトの指定されたスロットに格納される。ステップ5において、ステップ4の宣言を無効にするには、推論連鎖の記録に基づきステップ4の処理の前の状態に復活させる。

ステップ4および5での推論は、通常の推論とは異なる。むしろ、デフォルト制約を解消するために、特別な推論をおこなうためのメタ推論機構に相当する。ここでは、一つのデフォルトの取り消しによるデフォルト制約の解消をテストするものである。つまり、デフォルトの取り消しが他のデフォルト制約に抵触するかどうかを推論により判定している。

4 主観的評価の一貫性保持

意思決定における設計・選択活動[17]では、意思決定を効果的に支援するために、意

思決定者の主観的な判断や好みを取り入れる必要がある。本章では、デフォルト制約を用いた主観的評価の一貫性保持について論じる。主観的評価は、AHP[10]に基づく一対比較法を用いて定量化される。主観的評価の一貫性は、デフォルト制約を用いてその優先度を反映することにより、効果的に保持される。ここでは、特に、デフォルト制約を効率的に解消するために、デフォルトの優先度を決定するために制約解消機構で指定する手続き `decrease_IR` の実現方式を論じる。

4.1 AHPによる主観的評価

意思決定問題における要素間の優劣は主観的評価から得られる重みの大きさに決定される。本論文では、主観的評価手法として、AHP[10]に基づく一対比較法を用いる。AHPは、システムズ・アプローチと主観的判断を組み合わせることにより、定量分析では扱いきれない決定問題に対処する手法である。この手法では、意思決定者の勘や経験を生かすことがその主眼となっている。AHPに基づく一対比較法には、9点法が用いられる。9点法は、一対比較値として、1（同じくらい重要を示す）から9（絶対的に重要を示す）までの正整数値を与えるものである。一対比較で得られた主観的数値（重み）を用いることにより、各々の要素の重要度が計算される。重要度は一対比較により得られる一対比較行列[12]をもとに固有値や固有ベクトルを求めることにより解析的に得られる。一対比較行列の要素 a_{ij} は項目 I_i と項目 I_j とを一対比較することにより得られる。理論的には、要素 a_{ij} は項目 I_i の重み W_i と項目 I_j の重み W_j を用いて、

$$a_{ij} = W_i / W_j$$

として定義される。具体的に、要素の重要度は、最大固有値の固有ベクトルの値として得られる。一対比較行列は、(1)対角要素が1、(2)要素の値は、

$$a_{ji} = 1 / a_{ij}$$

として定義される。

主観的評価において、AHPを用いた主な理由は、AHPにおいて主観的評価の整合度を

表す指標があることである。この指標は非整合度尺度(I.R.(inconsistency ratio))と呼ばれ、次のように定義される。

$$I.R. = (\lambda_{\max} - n) / (n - 1) \dots (1)$$

ここで、 $\lambda_{\max}$ は先に述べた一対比較行列より得られる最大固有値であり、 n は行列のサイズ（一対比較の要素数）である。理想的な一対比較が行なわれると非整合度尺度 I.R. は 0 になる。もし、この非整合度尺度の値が 0.1 以下であれば、経験的に、一対比較に整合性が有りとは判断できる。非整合度尺度の値が 0.1 より大きくなった場合には、一対比較に整合性がないことを意味しており、一対比較のやり直しが必要となる。一般に、AHP では、一対比較のやり直しは、煩わしい比較作業の繰り返しを再度強いることになり、意思決定者にとり大きな負担になる。

本研究では、このような負担を軽減するために、デフォルト制約を利用する。ここでは、デフォルト制約は、重み付けの整合性を自動的に保持するための枠組みとして利用される。

4.2 デフォルト制約の利用

一対比較は、意思決定者の一対比較に対する信念を表すために、入力時に確実的なものと仮定的なものとの区別される。仮定的な一対比較はデフォルトに相当する。一対比較は、推論の途中で、システムからの質問というかたちで行なわれる。仮定的な一対比較値は、一対比較の整合性に問題が生じた際に制約解消機構により優先的にチェックされ、一貫性の保持のためにその値が変更される。

主観的評価の一貫性の保持に関連したデフォルト制約は、次のように記述できる。

```
consistency:
  if matrix(name=X, inconsistency_ratio > 0.1)
  then
    (solve_it(decrease_IR)
    ;
    otherwise(decrease_IR, new_weight) ).
```

ここで、RHSはPrologのゴールの並びとして解釈され、手続きsolve_itが失敗すると手続きotherwiseが呼び出される。制約の解消は、一対比較に関連したデフォルトを修正することにより、一対比較行列Xの非整合度尺度I.R.の値を0.1よりも小さくすることである。

手続きsolve_itの引数で示したdecrease_IRは、3.3節で述べた一対比較に関連したデフォルト（つまり、意思決定者が指定した”仮定的な一対比較値”）の優先度を決定するための手続きである。手続きdecrease_IRは、非整合度尺度I.R.の値を小さくするために、一対比較値の増減の大きさに着目してデフォルトの優先度を決定する。ここでは、非整合度尺度I.R.の値を小さくするために値の増減が大きいデフォルトほどその優先度は低くなる。

デフォルト制約consistencyは、推論連鎖をたどり、仮定的な一対比較値であるデフォルトを取り消す。具体的に一対比較値を修正するために、次のルールchange_weightが呼ばれる。

```
change_weight:
  if \matrix_element(name=X, row=R, column=C, weight=W, preference=P)
  then
    new_weight(W, W2, P) &
    make(matrix_element(name=X, row=R, column=C, weight=W2).
```

ここで、LHSには、デフォルト制約consistencyにより取り消されるデフォルトのパターンが表現されている。RHSにおいて、手続き”new_weight”は、一対比較行列XのR行C列の値Wを値W2に±2の範囲で微調整するために用いられる。Pは微調整のための増減の方向を示す。この微調整の範囲±2は経験的（および実験的）に得られたものである。これは、ミラーの法則[8]により一度に扱う要素数を最大7つに制限することにより、もとの一対比較法から得られた結果がそれほど矛盾していないことを前提としている。値Wの微調整のための増減の方向Pは、手続きdecrease_IRで決定され、一対比較

行列 X の非整合度尺度 $I.R.$ を小さくする方向に決定される（詳細は4.3節で述べる）。

デフォルト制約consistencyでは、もし手続きsolve_itが失敗したら、手続きotherwiseが呼ばれる。手続きotherwiseは、手続きsolve_itと同様に制約解消のために呼び出され、手続きsolve_itと異なり一度に複数のデフォルトを修正する。述語otherwiseにおける第2引数のnew_weightは、ルールchange_weightのRHSで用いられる一対比較値を具体的に変更するための手続き名である。具体的には、手続きotherwiseは、手続きdecrease_IRを用いてソートされたデフォルトのリスト（つまり、図3におけるリストD）から複数の仮定を取り出し、これら取り出されたデフォルトを同時に修正することにより制約の解消を図る。仮定の取り出し方は、先ず最初は先頭から2つを取り出し（1つの場合は既に手続きsolve_itで検査済みだから）、これらデフォルトの一対比較値を同時に修正する。もしその結果、制約が解消しなければ（非整合度尺度 $I.R.$ 値が0.1より大きければ）、デフォルトのリストの先頭から3つのデフォルトを取り出し、3つのデフォルトを同時に修正する。最終的には、デフォルトのリストの全てのデフォルトを同時に修正する。ここでの修正は、手続きsolve_itとは異なり、単にデフォルトの取り消しを宣言するのではなく、第2引数で指定した手続きを用いて直接にデフォルトの内容を更新する。これは、前向き型プロダクションシステムにおいて、同時にデフォルトを修正するには一回の認識-行動サイクル（すなわち、一回のルールの呼び出し）の間で行なわれる必要があるからである。

4.3 一対比較値の増減

デフォルト制約解消機構solve_itでは、デフォルト（つまり、一対比較）の優先度を決定するために手続きdecrease_IRを用いた。手続きdecrease_IRは、効果的に非整合度尺度 $I.R.$ の値を小さくするようにデフォルトの優先度を決定する。手続きdecrease_IRの実現において、非整合度尺度 $I.R.$ の値を小さくするために一対比較値の増減の大きさを決定することは、一対比較行列の最大固有値を小さくする問題に帰着する。しかしながら、

一般にはこの種の問題を効果的に解くことが困難なことから、本システムではAHPで得られる一対比較行列の特徴から、修正のための増減の方向を決定する。

例えば、増減の方向の決定は図4を用いて説明できる。図4では2つ（上と下）の“車の選定”に関連した一対比較行列が示されている。一対比較行列（上）の行列要素で下線のある一対比較値は仮の重み付けを表しており、矢印の方向は修正のための増減の方向、そして矢印の太さは増減の量を表している。例えば1行4列の要素の一対比較値は仮の値として7が与えられ、一対比較の整合性を保持するために7より小さくする必要があることを示している。増減の量は、太さが2番目の矢印で示されているので中程度の減を行なう必要がある。増減の量は3つの矢印（大、中、小）を用いることにより表現される。



図4. 一対比較値修正のための増減方向の決定

図4の一対比較行列（下）は、以上のような増減を決定するために用いられる情報で

あり、一対比較行列（上）の各項目に対する重み W_i を用いて求める。一対比較行列（上）において i 行 j 列の一対比較値 a_{ij} は、項目 I_i の重みと項目 I_j の重みを一対比較して得られた相対的重み付けであり、理想的には W_i/W_j を求めたことに相当する。一対比較行列（下）の a_{ij} 要素は、この W_i/W_j より定義される。このように構成される一対比較行列（下）の最大固有値は項目数 n と等しくなり、3.1節の式(1)から非整合度尺度 $I.R.$ は0となる。つまり、一対比較行列（下）は理想的な一対比較を行なった場合として見なせる。そこで、この一対比較行列（下）に近付けるように一対比較行列（上）の各要素を修正することにより、非整合度尺度 $I.R.$ の値を効果的に小さくできる。先に述べた、一対比較値を修正するための具体的な増減の数値は以上ようにして決定される。

4.4 評価

一対比較法は、意思決定者の主観的評価を簡便に取り入れることができる利点を有している。一方、その重み付けの整合性（すなわち、一対比較の推移律）を成立させるためには、要素間で煩雑な一対比較のやり直しを強いられるのが一般的である。本論文では、重み付けの整合性を保存するために、デフォルト制約を用いた。これにより、矛盾を含む一対比較は自動的に調整され、一対比較の優先度を反映することにより重み付けの整合性は効果的に保持された。

デフォルト制約を用いた重みの自動調整は、4.3節で述べた理想的な一対比較に近付ける方向に行なわれる。ここでの自動調整の妥当性は、基本的には、意思決定者の責任においてチェックする必要がある。しかしながら、経験的に次の3つの理由により、意思決定者にとりかなり妥当な結果が自動的に得られている。第一の理由は、一対比較の一度に比較すべき要素数（最大7つ）を制限したことや要素間の独立性のチェックをすることにより、一対比較から得られる最初の結果がそれほど矛盾していないことにある。第二の理由は、一対比較に意思決定者の信念を反映することにより、自信のない一対比

較である仮の値を修正したことにある。第三の理由は、重みの自動修正を ± 2 の範囲に制限したとにより、意思決定者の許容範囲と思われる自動修正結果を得られることにある。

5 本研究の特長

KORE/IEは、Prolog上に構築された前向き推論型プロダクションシステムであり、OPS5風のルールプログラミング環境を提供する。プロダクションシステムでの推論は、認識-行動サイクルと呼ばれる内部メカニズムが順方向（つまり、照合から競合解消そして実行へと順に処理される）に回転することにより実現する。認識-行動サイクルは、WMに変化（つまり、WM要素の付加および削除）が生じれば必ず順方向に向かって処理される。このことは、プロダクションシステムの実行過程では、バックトラッキングや推論の取り消しが行なわれないことを意味している。さらにプロダクションシステムでは、タイムタグの最新性等に着目して推論の方向を一意になるように制御している。OPS5で代表されるルールプログラミングは、以上のような認識-行動サイクルの一方向性に基づいたプログラミング技法である。

本研究の特長は、認識-行動サイクルを逆方向へ戻すための制御機構を構築することにより、プロダクションシステムへ問題解決に即してバックトラッキングや推論の取り消し機能を実現するものである。これにより、ルールプログラミングの宣言的な記述力や問題解決能力を向上させることができる。例えば、本論文で述べたデフォルト制約における制約解消機構solve_itは、認識-行動サイクルを逆方向へ戻すための制御機構である。ここでの特別な制御は、単に認識-行動サイクルを戻すのではなく、デフォルトの優先度に着目して認識-行動サイクルを逆方向へ戻し、そのデフォルトの取り消しを主張したことにある。デフォルトの取り消しは、WMに変化に相当し、これにより認識-行動サイクルが順方向に処理される。最終的には、デフォルト制約が解消されるまで、認識-行動サイクルが両方向にわたって処理される。

デフォルト制約を用いた推論は、制約充足のための探索問題を解決していることに相当する。ここでは、探索を効率化するために（つまり、ヒューリスティクスを導入する手段として）、手続きsolve_itの中でデフォルトの優先度を決定するための手続き（`preference`および`decrease_IR`）が用いられた。さらに、手続きsolve_itは、制約充足に関連して自動的に制約解消の判定をすることにより、制約を評価するための機構となっている。デフォルト制約を用いることにより、人間が行なう試行錯誤的問題解決を効果的にシステムへ記述できる。3章で述べた主観的評価の一貫性保持は、実質的には人間が行なう試行錯誤的な戦略をシステムへ効果的に置き換えたことに相当する。ここでは、与えられた一対比較行列の非整合度尺度L.R.の値を小さくするために、デフォルトの優先度に従い感度分析的に次々と仮定的（つまり、不確実）な一対比較値の値を変更した。

5.1 関連研究

本研究と特に関係ある研究として、TMS[5]や制約階層に関連したBorning[2]や佐藤[12]の“柔らかい制約”の研究がある。デフォルト制約解消機構の実現に関連して、本研究では、TMSの仮定の一貫性保持の枠組みを拡張した。TMSでは、仮定による矛盾を解消するために、仮定を取り消すための枠組みを与えている。しかし、デフォルト制約の実現において、仮定の優先度や、推論システムに対して推論を制御するための枠組みが不十分であった。本研究では、TMSの矛盾解消の枠組みに仮定（つまり、デフォルト）の優先度を扱うメカニズムを導入し、デフォルトに関連したルール（つまり、デフォルト制約）の解釈／実行を制御するための枠組みを導入した。Borningの制約階層は、ホーン節の本体に固い制約だけでなく、優先度を持った制約を表現できるようになっている。しかしながら、優先度を解釈するための制約評価メカニズムが組み込みであるという制限があった。一方、本研究の枠組みでは、デフォルトの優先度の解釈を自由に変更したり、問題特有のヒューリスティクスを導入する手段を提供した。4章で述べた手続き`decrease_IR`は、この一つの例である。佐藤の“柔らかい制約”の研究は、モデル論に

基づいてBorningの制約階層のひとつの定式化を与えている。ここでは、二階述語論理を導入しており、実際のインプリメントではシステムの高速化が課題となっている。デフォルト制約の実行メカニズムは、プロダクションシステムKORE/IEをベースにしておりその推論の高速性を取り入れ、実用に耐える性能を得ている。しかしながら、TMSの枠組みを利用したことにより、推論の履歴を記録管理する必要性があり、KORE/IEの推論の高速性を十分に利用できていない。デフォルト制約の実行メカニズムの高速化は今後の課題である。

6 おわりに

本論文は、ルールプログラミングにおけるデフォルト制約の実現方式とその応用について論じた。デフォルト制約は、TMS的な機構を拡張することにより実現し、ルールの記述力を向上させた。応用例として、AHPに基づく主観的評価の一貫性保持について論じた。ここでは、デフォルトの優先度を決定するための手続きdecrease_IRについて論じた。デフォルト制約により、主観的評価の一貫性は効果的に保持された。

今後の課題として、(1)ATMSを利用したデフォルト制約解消機構の構築、および(2)KORE/IEの制約論理型言語上への実装がある。(1)では、ATMSの非単調推論への枠組み[18]を利用することによりデフォルト制約を解消するための機構を実現する。これにより制約解消において複数の解を得ることが期待できる。(2)のKORE/IEの制約論理型言語上への実装は、ルールプログラミングにデフォルト制約と(式として表現される)固い制約との融合を指向するものである。KORE/IEでは、ルールを論理型言語へコンパイルすることにより、推論の内部メカニズム(例えば、照合過程やルールの実行等)を特別な解釈実行系を介せずに論理型言語の直接的な実行過程として実現しており、論理型言語の利点を取り入れることができる。このような特長は、特別な機構を構築することなしに、KORE/IEを制約論理型言語への実装を可能とする。これにより、デフォルト制約と数式を用いた固い制約を組み合わせた制約プログラミングを実現するための枠組み

を得る。

尚、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なわれたものである。

参考文献

- [1]相場,古川：制約プログラミングについて-制約ロジックプログラミングを中心として
-, 人工知能学会誌, Vol.6, No.1(1991), pp.47-59.
- [2] Borning,A. and Maher,M. : Constraint Hierarchies and Logic Programming, Proc. of ICLP89
(1989), pp.149-164.
- [3]Cooper,T. and Wogrin,N. : Rule-based Programming with OPS5, Morgan Kaufman
Publishers, 1988.
- [4]DeTurck,D.M.: The Approach to Consistency in the Analytic Hierarchy Process, Mathematical
Modelling, Vol.9 (1987), pp.345-352.
- [5]Doyle,J. : Truth Maintenance System, Artificial Intelligence Vol.12 (1979), pp.231-272.
- [6]Forgy, C.L. : OPS5 User's Manual, CMU-CS-81-135, July,1981.
- [7] Harker, P.T.: Incomplete Pairwise Comparisons in the Analytic Hierarchy Process, Math.
Modelling, Vol.9 (1987), pp.838-848.
- [8]Miller,G.A.: The Magical Number, Seven, Plus or Minus Two: Some Limitations on our
Capacity for Processing Information, Psychology Review, Vol.63 (1956), pp.81-97.
- [9] 溝口, 古川, Lassez,J-L. 編：制約論理プログラミング、共立出版、1989.
- [10]Saaty,T.L.: The Analytic Hierarchy Process, McGraw Hill, 1980.
- [11]Saaty,T.L. and Takizawa,M.: Dependence and Independence: From Linear Hierarchies to
Non-linear Networks, European Journal of Operational Research, Vol.26 (1986),
pp.229-237.
- [12]佐藤：解釈の順序による柔らかい制約の定式化, 情報処理学会論文誌, Vol 31, No.6
(1990), pp.772-782.

- [13] Shintani,T.: A Fast Prolog-Based Production System KORE/IE, Logic Programming: Proc. of the Fifth International Conference and Symposium(edited by R.A.Kowalski and K.A.Bowen), MIT Press (1988), pp.26-41.
- [14]新谷：問題解決支援機構KORE/CDSSにおける知識の構造化,情報37回全国大会, (1988), 2J-5.
- [15]新谷：prologにおけるプロダクション照合フィルタの高速化, 情報処理学会論文誌, Vol 32, No.1 (1991), pp.20-31.
- [16]Shintani,T.: An Approach to Nonmonotonic Inference Mechanism in Production System KORE/IE, LNAI 383, Logic Programming'88, Springer-Verlag (1989), pp.38-52.
- [17] Simon,H.A. (稲葉, 他訳): 意思決定の科学, 産業能率大学出版部,1979.
- [18]Zetzche,F. : Non-monotonic Reasoning with ATMS, Lecture Notes in Artificial Intelligence 390 (1989), pp.119-128.