

知的スプレッドシートにおける

双方向実行について

3C-8

高梨郁子、和田信義、辻秀一

三菱電機(株) 情報電子研究所

1. はじめに

エキスパートシステム構築ツールには、従来汎用ツールが用いられてきたが、一般のユーザが使いこなすのは難しい。そこで、最近ではエキスパート自身の手で知識ベースを作ることができる、応用特化型ツールが有効であると言われている。AceKit<sup>[1]</sup>は、階層的な表の中に、数式だけではなくルールも記述できる、タスク特化型ツールである。しかし、従来Bottom up 的推論機能だけしか提供していなかったため、ユーザの意志決定を支援するツールとしては機能的に不十分であった。そこで、Top down的推論機能である「わりふり機能」を、MELCOM PSI II上のAceKitに実装した。

2. 従来のAceKitとその問題点

AceKitは表形式のユーザインタフェースを持ち、非数値を扱うことができ、オフィスにおける意志決定支援等に有効<sup>[2]</sup>であると思われる。例えば、ある工場の必要経費を見積もる場合には、前年度の状況や、各部門でどの様な制約・条件等を記述できるので場合に応じた多種多様の計算を制御できる。しかし工場予算が企業全体で定められた予算より多かった場合にはどうするか。定められた値に合うようにユーザが試行錯誤を繰り返さなくてはならない。この様な場合人間は各部門での必要経費を求める。Bottom up 的推論と、企業全体で定められた予算に合うように制御、修正するTop down的推論とを交互に繰り返して、制約を満足する解に収束させていく。

そこで今回、AceKitに、従来のBottom up 的推論を支援する機能に加えてTop down的推論を支援する機能を実装した。ここではBottom up 的な従来方式の推論のことを「順方向」と呼び、今回追加したTop down的な推論を「わりふり」と呼ぶ。この機能によって、人間のTop down的な思考も支援できるようになった。但し、Top down方向の推論には確かな知識があるわけではないので、ユーザが用いた過去のデータや、値候補を用いて、ユーザが確認しながらわりふりが行えるように工夫した。

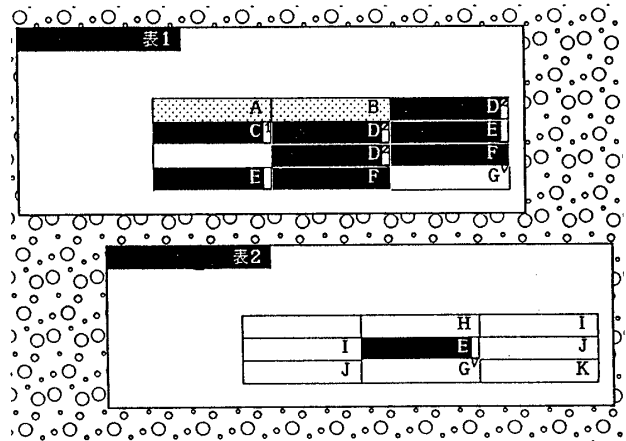


図1 わりふり時のセルのマーク表示

3. 問題の同定

ここで(式-1)のような簡単な式を例として、わりふりについて述べる。

$$C + D = E \quad (\text{式-1})$$

図1において(式-1)をEのセルに定義し、使用変数Cと使用変数Dにそれぞれ値を入力すると、順方向では被使用変数Eの値が求められて、セル上に表示される。これに対してわりふりとは、式が定義されているEに新たな値が入力されたときに、これを満たすような使用変数CとDの値を求めることを意味する。未知変数が二つあるのに対して、これらが満たすべき制約条件の式が一つしかなく、値を決定するには自由度が多すぎるため、無数の解が存在する。その無数の解の中からここで求めたい一つの解を決定するのは大変困難であり、式には書かれていない制約が必要である。これを知っているのは式をつくったユーザ、または使用変数の値の意味を知っているユーザだけである。そこでAceKitでは、わりふりのすべてをシステムが行うのではなく、あくまでユーザの支援をするということの基本方針としてシステムを構成した。

#### 4. わりふり機能の実現

前述したようにわりふりの制約を一番よく知っているのはユーザであるが、すべてのセルについての制約を知っているわけではない。しかし一般に人間は望ましくない解を提示されれば、どこが良くないかを比較的用意に指摘できる<sup>[3]</sup>。そこでまずシステムがヒューリスティクスを用いて場合に応じたわりふりを行ない、わりふった例をユーザに示し、その後ユーザが値を容易に変更・決定できるようにした。ユーザとのやり取りを通じてわりふりを進めるため、以下のような、ユーザにとってわかりやすい枠組みを考えた。

- ①「順方向」と「わりふり」の2つの推論モードを設ける。
- ②わりふりの及ぼす範囲をtree内に制限する(図2)。
- ③あるセルの値を変えないように設定できる(値の固定)。
- ④誤差範囲の設定可能。
- ⑤わりふりの状態をセルにマーク表示する。
- ⑥わりふりの途中結果を用いての順方向への発火が可能。

##### (1) 数式のわりふり方法

数式の場合は、「比例配分」、「均等配分」、「比率配分」、の3つの配分方法を用いる。デフォルトの設定があるのでユーザは特に意識する必要はないが、問題の意向に添った値が得られなかった場合はユーザがわりふりし直す。

##### (2) ルールのわりふり方法

ルールのわりふりにおいては、数式のような配分方法を適用することができず、また逆向きにルールを実行することはできない。そこでユーザが既に入力している値候補(各セルが取り得る値の候補)を利用する。使用変数の値候補すべてについてルールを発火させ、その結果が被使用変数の値に矛盾しないものを解とする。例えば図2において、わりふりモードでセル1に"ff"を入力したとすると、使用変数であるセル2、セル3の値候補のすべての組み合わせ(全6組)を発火させ、セル1の値が"ff"になるものが解となる。すべての解を表示する事はできないので1つ目に見つかったものをセルに表示するが、その値ではつぎのわりふりに進めないことがある。そこで次のわりふりに進むことができず中断しているセルや、ユーザの意図に添わない値のセルがあった場合には、解となり得る値候補の一覧表を表示してユーザが値を選び直す(generate & test)。

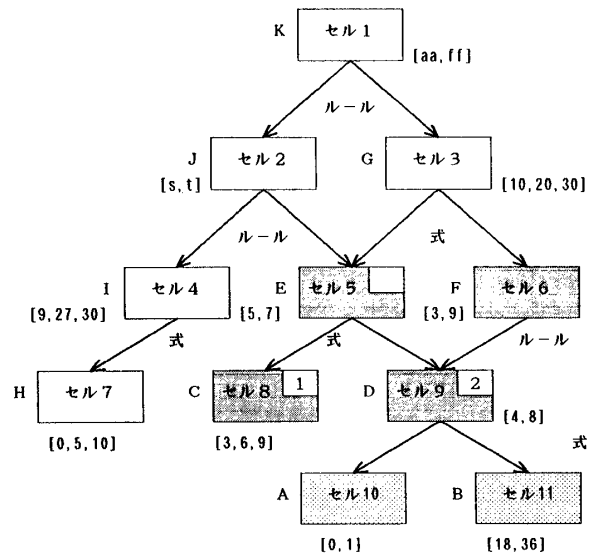


図2 セルのtree構造 ( []内は値候補)

#### 5. おわりに

AceKit上の「わりふり機能」について述べた。ここで述べたわりふりは、単なる逆計算・逆実行ではなく、個々の式、ルールの意味に合ったわりふりを行なうことができる。ヒューリスティクスを埋め込み、それによってまずシステムがわりふりを行なう。そしてその値を見ながらユーザが修正する、という流れでわりふりの値を定める。このため、ユーザがわりふりの動きを理解しやすいように、またわりふりがうまく収束するように、わりふりの状態表示、ダイレクトマニピュレーション、わりふりの範囲の制限等の手法を用いた。従来からAceKitはユーザの意志決定の支援を行なっていたが、この枠組みを用いることによって、より人間の思考過程に近い形態での支援が可能となった。

このような「わりふり機能」はエキスパートシステム構築ツールに限らず、スプレッド・シートの拡張版としても有効に用いることができる。今回示した「わりふり機能」では、制約論理プログラミング<sup>[4]</sup>のように制約式から確実な解を求めることはできないが、反対に、必要な制約が書かれていない場合にもユーザとのやり取りを通じて解を求めることができる。従って、制約を式やルールで表現できない場合や、曖昧な制約を解く場合に有効である。

##### 〔参考文献〕

- [1] "AceKit: 知的スプレッドシート構築ツール"、和田、増山、近藤、川村、太細、情処第34回全国大会
- [2] "知識情報処理技術のOAへの応用"、和田、島田、近藤、太細、川村、三菱技報、1986.9
- [3] "インタフェースの科学"、淵一博監修、共立出版
- [4] "制約ロジック・プログラミング - 知識処理への新しいパラダイム -"、横井、相場、情処学会、vol. 30, NO. 1, Jan. 1989