

シミュレーション・システムとしてのスプレッド・シートの機能拡張†

金井直樹††

スプレッド・シート・プログラムは非常に便利なツールとして広く世の中に受け入れられている。しかも、多くの場合、単なる集計用ツールとしてではなく、将来を予測するためのシミュレーション・システムとして用いられている。ただし、スプレッド・シートは、本来、集計用ツールとして開発されたため、シミュレーション・システムとしては問題点がある。それは、セル間の依存関係を関数という形式で暗黙的にしか表現できないことである。この結果、現在のスプレッド・シートでは、一方向の再計算機能しか実現できない。我々は、この問題に対処し、スプレッド・シートをより強力なシミュレーション・システムとするための機能拡張として、モデル化手段としてセルとセル間に成り立つ関係を用い、関係中にセル間の依存関係を伝搬制約として明示的に表現すること、および、すべての伝搬制約を満たしながら再計算を実行する双方向再計算機構を提案した。この拡張されたスプレッド・シートが提供するモデル化手段と双方向再計算機能により、スプレッド・シートはより強力なシミュレーション・システムとなると期待される。また、双方向再計算を実現する本手法は、スプレッド・シート特有のものではなく、伝搬制約という制約を考慮した場合の制約伝搬問題に対する一解答を与えていると考えられる。

1. はじめに

Multiplan* や Lotus 1-2-3** 等に代表されるスプレッド・シートは非常に便利なツールとして広く世の中に受け入れられている。しかも、多くの場合、単なる集計用ツールとしてではなく、将来を予測するためのシミュレーション・システムとしてスプレッド・シートは用いられている¹⁾。さらに、スプレッド・シートのこのような利用法に着目し、シミュレーション機能としてスプレッド・シートの考え方を組み込んだアプリケーション・システムの提案もなされている^{6),7)}。しかし、スプレッド・シートの良さの一つは、スプレッド・シートがアプリケーションから独立している点である。この結果、ユーザは自分達の持つ問題に応じたシミュレーションをスプレッド・シート上で容易に実現することができる。

ただし、スプレッド・シートは、本来、集計用ツールとして開発されたため、シミュレーション・システムとしては問題点がある。ゆえに、スプレッド・シートがシミュレーションのためのより強力な機能を提供すれば、ユーザの作業をさらに支援できると考えられる。

我々はこの考え方に基づき、スプレッド・シートの機能拡張を行った^{2)-4),6)}。本論文では、シミュレーション・システムとしてのスプレッド・シートの機能と問題点を整理し、新たに要求される機能を示し、その実現において考慮すべき問題点を明らかにし、一実現法について述べる。

2. スプレッド・シートが持つシミュレーションの機能と問題点

本章では、スプレッド・シートをシミュレーション・システムとして捉えその機能を整理することにより、シミュレーション・システムとしてのスプレッド・シートが持つ問題点について述べる。

2.1 シミュレーション・システム

我々は、シミュレーションとは、「対象世界を抽象化、モデル化し、対象世界で起こるであろうことを、そのモデルの上で実現すること」であると考え、この考え方に従うと、シミュレーションの過程は図1のようにまとめることができる。この過程は大きく分けてモデル化の部分とシミュレーション実行の部分の二つより構成される。このために、シミュレーション・システムは、モデル化するための手段とシミュレーションを実行するための機構の提供が必要になる。

この時、モデル化手段に対しては、「対象世界を的確に、かつ、簡単に表現できる」ことが要求される。また、シミュレーション実行機構に対しては、モデルを構成する要素が変化した時に、それに応じて他の要

† Extended Spreadsheet for a Simulation System by NAOKI KANAI (Tokyo Research Laboratory, IBM Japan, Ltd.).

†† 日本アイ・ピー・エム(株)東京基礎研究所

* Multiplan は米国マイクロソフト社の登録商標である。

** Lotus 1-2-3 はロータス ディベロップメント コーポレーションの登録商標である。

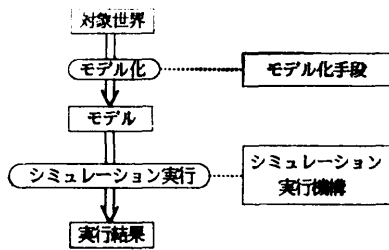


図1 シミュレーションの実行過程
Fig. 1 Simulation process.

表1 シミュレーション・システムとしての
スプレッド・シート

Table 1 Functions of spreadsheets for simulation.

モデル化手段	セル セルの値を決定するための関数
シミュレーション実行機構	再計算機構

素が変化し、モデルを常に無矛盾に保つ機能が要求される。ただし、シミュレーションの目的を考慮すると、単にモデルを無矛盾に保つのではなく、「対象世界で起きることに対応した方法」でモデルを無矛盾に保つことが要求される。

2.2 スプレッド・シートにおけるモデル化手段と
シミュレーション実行機構

シミュレーション・システムとしてスプレッド・シートを捉えると、表1のように、セルとセルの値を決定するための関数がモデルを表現する手段に、再計算機構がシミュレーション実行機構にそれぞれ対応する。

まず、スプレッド・シートにおけるモデル化手段について考えてみよう。スプレッド・シートを用いてモデルを作成する場合、対象世界を「要素と要素間に成立する関係」という局所的かつ宣言的な性質の集まりとして捉える。ただし、ここでいう関係の概念は、単に要素間に成立すべき関係式だけでなく、ある要素の変更が他の特定の要素に対し影響を与えるという要素間の依存関係を含む。

このように、世界を要素と要素間の関係として整理すると、各要素はスプレッド・シート上のセルに、そして、各関係は関数に対応付けられる。その例を図2に示す。この例は、簡単なローン返済計画のシミュレーションである。この時、毎月返済する分の借入金、ボーナスで返済する分の借入金、総借入金が、それぞれ、セル B2, B3, B4 に対応付けられている。さらに、借入金一万円一回

当りの返済額が C2 と C3 に、そして、一回当たり返済額が D2 と D3 に対応付けられている。

要素間の関係は、セルに定義される値を決定するための関数として表現される。この時、要素間に存在する依存関係を考慮して、関数が定義されるセルが決定され、関数式が要素間に成り立つ関係式から作られる。例えば、図2では以下の三つの関係式が考慮されている。

- (1) $B4 = B2 + B3$
- (2) $D2 = C2 * B2 / 10000$
- (3) $D3 = C3 * B3 / 10000$

この時、毎月の返済額 (D2) を変更した場合のボーナス時返済額 (D3) を求めることを、本シミュレーションの目的とすると、各式中の要素間の依存関係を考慮することにより、セル B2 を求める関数として、

$$D2 / C2 * 10000$$

が関係式(2)より作られ、セル B3 を求める関数として、

$$B4 - B2$$

が関係式(1)より作られ、そして、セル D3 を求める関数として、

$$C3 * B3 / 10000$$

が関係式(3)より作られる。

スプレッド・シートの提供するモデル化手段は、シミュレーション・システムのモデル化手段に要求されている点を以上の方法により実現している。要素や要素間の関係のようなモデルの局所的宣言的な性質はモデル作成者にとって認識しやすく、かつ、表現しやすい。この結果、「対象世界を簡単に表現できる」ことが実現される。また、要素間の依存関係を関数という形式で表現することにより、「対象世界で起きる変化に対応した的確な表現」が実現される。

次に、スプレッド・シートにおけるシミュレーション

	A	B	C	D
1		借入金額	一万円当り返済額	返済額
2	毎月返済分	10,000,000	60	60,000
3	毎ボーナス返済分	20,000,000	500	1,000,000
4	総合計	30,000,000		

関数定義

- (1) セル B3: $B4 - B2$
- (2) セル B2: $D2 / C2 * 10000$
- (3) セル D3: $C3 * B3 / 10000$

図2 シミュレーションのためのスプレッド・シートの使用例
Fig. 2 Example of a spreadsheet for simulation.

ン実行機構について考えてみよう。スプレッド・シートでは、再計算機能を実現する再計算機構が提供されている。この再計算機構により、あるセルの値が変更された時、そのセルに依存する他のセルの値が、定義された関数に従って再計算される。例えば、図 2 においてセル D2 の値が変更されると、定義された関数に従って、セル B2, B3, D3 の値が再計算される。

スプレッド・シートの持つ再計算機能は、シミュレーション実行機構に要求される「対象世界で起きることに対応した方法でモデルを無矛盾に保つこと」を実現している。セルの値が変更されたということは、セルに対応する要素の値が変更されたことを意味し、その結果、その要素を含む関係が満たされなくなる。そして、定義された関数に従ってセルの値を再計算するということは、それらの関数が表している要素間の関係を再び満たすように各要素の値を変更していることに対応する。この時、関数には暗に要素間の依存関係が表現されており、その結果、各関係が「対象世界で起きる方法」で無矛盾に保たれる。スプレッド・シートの再計算機構は、このセルの値の再計算の繰り返し、つまり、要素間の各関係を満たすための動作の繰り返しにより、モデル全体を対象世界で起きることに対応した方法で無矛盾に保つ。

2.3 シミュレーション・システムとしての スプレッド・シートの問題点

スプレッド・シートは本来集計用ツールとして開発された¹⁾。このため、シミュレーション・システムとしてのスプレッド・シートには、モデルの表現能力に限界があり、要素間の依存関係の一部しかモデル上に表現できない。つまり、モデル化手段に要求される「対象世界を的確に表現する」ことが十分に達成できない場合がある。これは、要素間の関係を表現する手段として、関数という制限された方法しか存在しないためである。

モデル上に、要素間の依存関係の一部しか表現できない結果、シミュレーション実行時には、一方向の再計算しか許されない。つまり、関数の出力側であるセルの値を変えても、対応する入力側のセルの値を変えることができない。例えば図 2 において、ボーナス一回当たりの返済金をある額に設定した場合に、毎月の額がいくらになるかシミュレーションしたいとしよう。この場合、セル D3 を変えることにより、関係式を満たすようにセル B3, B2, D2 が変わって欲しい

わけだが、図 2 のように関数が定義されていると、セル D3 の値を変更しても、セル B3, B2, D2 の値は再計算されない。

このように、スプレッド・シートでは、一つのモデル上ではシミュレーションのために変更できるセルが限定される。これは、スプレッド・シートを用いてシミュレーションを行う時に大きな不都合となる。

3. スプレッド・シートの拡張

我々は、スプレッド・シートの拡張として、モデル表現能力を強化し、新たなシミュレーション実行機構を提供することにより、現在のスプレッド・シートが持つ問題に対処できると考えた。本章では、スプレッド・シートの拡張に対する我々の基本方針について述べる。

3.1 拡張されたスプレッド・シートにおけるモデル化手段とシミュレーション実行機構

現在のスプレッド・シートの提供するモデル化手段の問題点は、セル間の依存関係が関数という形式で暗黙にしか表現できない点である。このため、我々は、セル間の依存関係を関係式とは分離して明示的に扱うことを考えた。そして、依存関係を表現する手段として伝搬制約という概念を導入した。伝搬制約とは、あるセルの値が変更された時に、関係式を無矛盾に保つため他のどのセルの値を変更するかを表す制約である。このように考えると、関係は、

- (a) セル間に成立すべき関係式
- (b) 伝搬制約の集合

の 2 種類の情報により構成することができる。

伝搬制約を用いてセル間の依存関係を明示的に表現することにより、関数で関係を表現する場合に比べて、より「的確に」モデルを表現することができるようになる。

モデル化手段を拡張すると、新たなシミュレーション実行機構が要求される。我々は、この新しい機構として双方向再計算機構を考えた。双方向再計算機構とは、どのセルの値が変更された場合でも、そのセルを含む関係式が無矛盾になるように、伝搬制約を満たすように変更すべきセルを選択し、そのセルの値を再計算するという動作を繰り返すメカニズムである。

以上をまとめるとシミュレーション・システムとしての拡張されたスプレッド・シートは表 2 のように整理することができる。

拡張されたモデル表現上で、双方向再計算機構が提

表 2 シミュレーション・システムとしての
拡張されたスプレッド・シート

Table 2 Functions of an extended spreadsheet for
simulation.

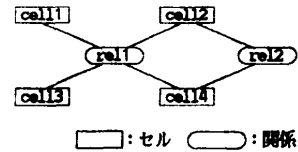
モデル化手段	セル セル間の関係
シミュレーション実行機構	双方向再計算機構

供する再計算機能を、以後、双方向再計算機能と呼ぶことにする。双方向再計算機能は、各関係において、従来のような値変更のための入力用セルと、出力用セルが固定した一方の再計算ではなく、任意のセルを変更しても他のセルの値が再計算されるという双方向の再計算を実現することができる。双方向再計算機能により、シミュレーションという目的に対して今までのスプレッド・シートが持つ不都合に対処できると考えられる。

以上が、スプレッド・シートの拡張に対する、我々の基本方針である。

3.2 双方向再計算機能の実現時に考慮すべき点

双方向再計算機能を実現する際に、伝搬制約の記述能力を適切に設定することが重要である。なぜなら、双方向再計算は、定義された伝搬制約のすべてを満足する方法で各関係における再計算を実行しなければならないが、不用意に伝搬制約の記述を許した場合、伝搬制約を満たすことができず、双方向再計算を実行できない場合があるからである。例えば、図3の例を考えてみよう。関係 *rel 1* は、セル *cell 1*, *cell 2*, *cell 3*, および、*cell 4* の間に、関係 *rel 2* は、セル *cell 2* と *cell 4* の間に成り立つ関係である。この時、伝搬制約として図のように関係 *rel 1* には(1), (2)が、関係 *rel 2* には(3)が定義されていたとする。伝搬制約(1)は、*cell 1* のみが変更された場合、*cell 4* が変更されるべきであることを表している。さらに、伝搬制約(2)は *cell 1* と *cell 2* の双方が変更された場合、*cell 3* が、そして、伝搬制約(3)は *cell 4* が変更された場合、*cell 2* が変更されるべきセルであることを表している。ここで、セル *cell 1* の値が変更されると、関係 *rel 1* の矛盾を解消するために伝搬制約(1)と(2)のどちらかに従って *cell 4* または *cell 3* が変更されなければならない。ここで、もし、*cell 2* が変更されないと仮定すると、関係 *rel 1* においては伝搬制約(1)が満たすべき制約になるため、*cell 4* が変更される。この結果、関係 *rel 2* においては伝搬制約(3)が満たすべき制約となり、*cell 2* が変更され



rel1の伝搬制約

	変更されたセル	変更すべきセル
(1)	cell1 & ~cell2	cell4
(2)	cell1 & cell2	cell3

rel2の伝搬制約

	変更されたセル	変更すべきセル
(3)	cell4	cell2

図 3 伝搬制約の例

Fig. 3 Example of propagation constraints.

る。これは、*cell 2* が変更されないとした仮定に反するため矛盾である。ゆえに、関係 *rel 1* においては *cell 1* の変更に対して伝搬制約(1)は満たすことはできないことがわかる。一方、伝搬制約(2)は、*cell 2* が変更された時に満たすべき制約であるが、この例の場合には、*cell 4* が変更されなかり *cell 2* は変更されることはない。しかし、*cell 4* は伝搬制約(1)を用いないと変更されず、伝搬制約(2)の前提である *cell 2* の変更が実現されることはない。ゆえに、この例のように関係が定義されていると、*cell 1* が変更された場合に、双方向再計算を実行することができない。

このように伝搬制約に全く制限を与えないと、構築されたモデル上で双方向再計算が実行できない場合が生じてしまい、非常に大きな問題である。ゆえに、構築されたモデル上で、いつでも双方向再計算が実行可能であるためには、伝搬制約の記述には何らかの制限が必要である。

4. 双方向再計算機能の一実現法

前章で述べたとおり、双方向再計算機能を実現するためには、適切に伝搬制約の記述能力を制限する必要がある。同時に、伝搬制約の記述能力に応じた双方向再計算機構が実現できなければならない。本章では、我々が開発した一手法の詳細について述べる。

4.1 関係の記述

我々は、双方向再計算機能を実現するために、以下のような関係の記述を用いる。

(a) 関係式：

$$Y = f(X_1, \dots, X_n)$$

という形式で表現する。ただし、 Y , X_1, \dots, X_n はセ

ル名である。

(b) 伝搬制約：以下の三つの制約よりなる。

- (1) X_1, \dots, X_n のうちいずれかが変更され、かつ、 Y が変更されない場合、 Y を変更する。
- (2) X_1, \dots, X_n のうち特定の一つのセルを X_k とする。 Y の値が変更され、かつ、 X_k が変更されない場合、 X_k を変更する。このセルを伝搬制約セルと呼ぶ。ただし、 X_k の値を求める逆関数が存在しなければならない。
- (3) Y と伝搬制約セル X_k の両方が変更された場合、関係式を満たすように再計算を実行するのではなく、その関係が満たされているか否かのチェックを行う。

伝搬制約(b)のうち、個々の関係に固有の情報は、伝搬制約セルだけであり、その他の情報はすべての関係に共通である。この結果、ユーザが定義できる情報は、関係式と伝搬制約セルになる。

(a)と(b)で記述できる情報は、従来のスプレッド・シートで記述できる情報の拡張である。従来のスプレッド・シートではセル間に成り立つ関係を関数として表現していた。これは、(a)で記述する関係式と、伝搬制約のうち(b)の(1)だけを記述することになる。また、従来の再計算機能は、(b)の(1)の伝搬制約を満たすように再計算を実行していることになる。

関係の例を以下に示す。三つのセル *uriage*, *tanka*, *hanbaisu* がそれぞれ売上げ、販売単価、販売数を表す時、関係として、

関係式 : $uriage = tanka * hanbaisu$

伝搬制約セル: *hanbaisu*

が定義されているとする。この時、*tanka* や *hanbaisu* を変更した場合、*uriage* が再計算される。また、*uriage* を変更した場合、*hanbaisu* が再計算される。もし、*uriage* と *hanbaisu* の両者が同時に変更された場合には、関係が満たされているかチェックされる。

以上で表現される関係をグラフ化したものが図4である。五角形は関係ノードを表し、四角はセル・ノードを表す。五角形の先端からのアークは、関係式の左辺 (Y) のセルにリンクされる。また、その他のアークは右辺に現れるセルとリンクされる。そのうち矢印で示されたものは、伝搬制約セルにリンクされる。

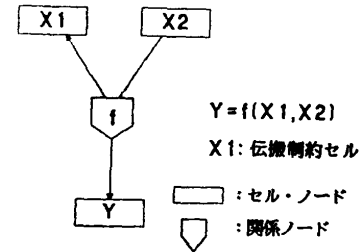


図4 セルと関係のグラフ表現

Fig. 4 Graph representation of cells and a relation.

4.2 双方向再計算機構

双方向再計算を正しく実行するためには、双方向再計算機構は、各関係における再計算を実行する時に、まず第一に、複数存在する伝搬制約のうちどれが満たすべき伝搬制約であるかを認識できなければならない。そして、第二に、選択した伝搬制約を正しく満たすように再計算を実行しなければならない。

そのためには、各関係中のセルのうち再計算実行中に他の関係によって変更される可能性があるセルすべてが、実際に変更されるまで、その関係における再計算を保留しなければならない。

そのため、我々が開発した双方向再計算機構では、マークを導入し、双方向再計算を二つの過程に分けて実行する。一つは、変更される可能性のあるセルを調べるためにアークにマークを付ける過程である。もう一つは、マークを操作することにより、変更される可能性があるセルが実際に変更されたかどうかを識別し、再計算実行可能であれば、満たすべき伝搬制約を決定し、それに従って、再計算を実行する過程である。

以下、双方向再計算機構の詳細について述べる。

4.2.1 マーク

双方向再計算機構で用いるマークには表3に示された *M-cell*, *Input arc*, および、*Output arc* の3種類がある。

マーク *M-cell* は、セルの値が変更されたか否かを判断するために用いられる。マーク *Input arc* は、その関係における再計算以外の原因で値が変更される可能性があるセルを識別するために用いられる。マーク *Output arc* は、その関係における再計算によって値が変更される可能性があるセルを識別するために用いられる。

4.2.2 初期マーク付け過程

初期マーク付け過程では、いずれかのセルの値が変更された場合、再計算実行中に変更される可能性のあ

表 3 双方向再計算機構で用いられるマーク
Table 3 Marks used by a bidirectional recalculation mechanism.

マーク名	マークの内容
<i>M-cell</i>	ユーザによる変更指令の対象となるセル・ノード、または、すでに再計算が実行されたセル・ノードに付けられる。
<i>Input arc</i>	ユーザによる変更指令の対象となるセル・ノード、または、他の関係において <i>Output arc</i> のマークが付けられたアークに接続したセル・ノードと関係ノードをつなぐアークに付けられる。 <i>Input arc</i> には方向があり、その向きは、セル・ノードが始点であり、関係ノードが終点である。
<i>Output arc</i>	<i>Input arc</i> のマークの付いたアークに接続したセル・ノードが変更された場合に再計算すべきセル・ノードと、関係ノードをつなぐアークに付けられる。 <i>Output arc</i> には方向があり、その向きは、関係ノードが始点であり、セル・ノードが終点である。

るセルを調べるために、アークにマーク付けを行う。

この過程では、双方向再計算機構は、まず、すべてのマーク (*M-cell*, *Input arc*, *Output arc*) を消去し、次に、ユーザにより変更されたセルの値をユーザが指定した値に変更するとともに、そのセルに *M-cell* のマークを付ける。また、バックトラック制御機構の初期化を行う。ただし、バックトラック制御機構については後述する。その後、表 3 に示した条件を満たしたアークに *Input arc* と *Output arc* のマーク付けを行う。

図 5 に、関係 $Y=f(X1, X2)$ に付けられたマークの例を示す。ただし、伝搬制約セルは $X1$ であるとする。ここで、 $X2$ がユーザにより変更されたセルであるとする、 $X2$ に *M-cell* のマークが付けられる。また、 $X2$ と関係ノードをつなぐアークに *Input arc* のマークが付けられる。この時、再計算すべきセルは Y であるので、関係ノードとセル Y をつなぐアークに *Output arc* のマークが付けられる (図 5 (A))。さらに、 Y が他の関係における *Output arc* のマークの付いたアークに接続しているとする、 Y と関係ノードの間のアークに *Input arc* のマークが付けられ、そしてこの場合は、 $X1$ が再計算すべきセルになるので、関係ノードと $X1$ の間のアークに *Output arc* のマークが付けられる (図 5 (B))。

なお、以後、*Input arc* の始点であるセルをその関係における *T-cell*、*Output arc* の終点であるセルをその関係における *C-cell* と呼ぶことにする。

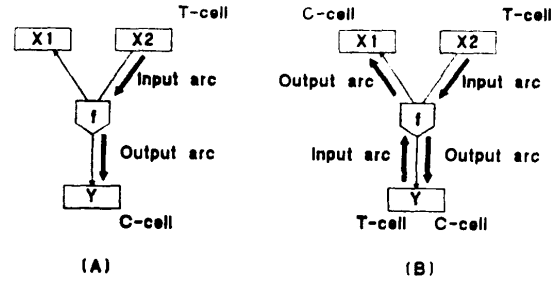


図 5 マーク付けの例
Fig. 5 Example of marking.

4.2.3 再計算実行過程

再計算実行過程では、マークを操作することにより満たすべき伝搬制約を決定し、それによって再計算すべきセルを決定し、その値を再計算する。ただし、4.1 節で述べた関係表現を用いた場合、再計算の過程において、ループが発生する場合がある。また、伝搬制約を満たす複数の再計算方法が存在する場合もある。再計算実行過程は、これらの場合にも正しく対処できるようにアルゴリズムが定式化されている。

この過程は、表 4 に示す七つの規則により実現される。規則 1 から規則 4 までは、再計算を実行するための基本的な規則である。規則 5、規則 6、規則 7 はそれぞれ、ループが発生した場合、複数の再計算方法が存在した場合、ループが発生し、かつ、複数の再計算方法が存在した場合の規則である。

以下、各規則の詳細について優先順位の高い規則から述べる。なお、説明に利用する図中で、ハッチングされているセル・ノードは *M-cell* のマークが付けられたセル・ノードであることを表す。

規則 1 は、すでに値が変更されたセルは、以後の再計算の対象とはなりえないため、*M-cell* のマークの付けられたセル・ノードとリンクしたアークから *Output arc* のマークを削除するために用いられる。

規則 2 は、一つの関係における再計算の対象となる可能性があるセルが一つしかない場合に、そのセルを再計算の対象として確定するために用いられる。

規則 3 は、他の関係によって変更される可能性があるセルすべてが実際に変更されている場合に、その関係における再計算を実行するために用いられる。図 6 に規則 3 の実行例を示す。この例における *T-cell* は $X1$ と $X2$ であり、かつ、それらに *M-cell* のマークが付けられているため、*Output arc* の向かうセル・ノード Y の値を再計算する。そして、 Y に *M-cell* のマークが付けられ、すべてのアークに付けられたマークが削除される。

表 4 双方向再計算実行のための規則
Table 4 Rules for bidirectional recalculation.

- 規則 1**
(条件)
M-cell のマークが付けられたセル・ノードに接続するアークに Output arc のマークが付けられている場合、
(アクション)
それらの Output arc のマークを削除する。
- 規則 2**
(条件)
一つの関係ノードに接続しているアークの中に、Output arc と Input arc の両者のマークが付けられたものがあり、かつ、他に Output arc のマーク付けされたアークが、その関係ノードに接続されていない場合、
(アクション)
Output arc と Input arc 両者のマークの付いたアークから、Input arc のマークを削除する。
- 規則 3**
(条件)
一つの関係におけるすべての T-cell に M-cell のマークが付けられていて、かつ、Output arc のマークの付いたアークがその関係ノードに接続されている場合、
(アクション)
Output arc の向かうセル・ノードの持つ値の再計算を、その関係ノードが持つ関係式を用いて実行するように評価計算部に指令し、その関係ノードに接続されたアークに付けられたすべてのマーク (Input arc と Output arc) を削除し、値が再計算されたセル・ノードに M-cell のマークを付ける。
- 規則 4**
(条件)
一つの関係におけるすべての T-cell に M-cell のマークが付けられていて、かつ、その関係ノードに接続しているアークに Output arc のマーク付けされたものが存在せず、かつ、その関係ノードに接続しているアークに Input arc のマーク付けされたものが存在する場合、
(アクション)
その関係が満たされているか否かの評価を実行するように評価計算部に指令し、その関係ノードに接続されたアークに付けられたすべてのマーク (Input arc) を削除する。
- 規則 5** (ループが発生する場合)
(条件)
一つの関係ノードに接続しているアークに、Output arc のマークが付けられたものがただ一つあり、かつ、その関係における T-cell の中に、M-cell のマークが付けられているセル・ノードが存在する場合、
(アクション)
Output arc の向かうセル・ノードが持つ値の再計算を、その関係ノードが持つ関係式を用いて実行するように評価計算部に指令し、そのセル・ノードに M-cell のマークを付け、その Output arc のマークを削除する。
- 規則 6** (複数の再計算方法が存在する場合)
(条件)
一つの関係ノードに接続しているアークに、Output arc

- のマークが付けられたものが二つあり、かつ、そのうちの一つに Input arc のマークが付けられており、かつ、その関係における T-cell の中に M-cell のマークが付けられているセル・ノードがあり、かつ、バックトラック制御部が、規則 6 をその関係に適用することを許可した場合、
(アクション)
バックトラック制御部に対し、規則 6 をその関係に適用したことを通知し、Output arc のみがマーク付けされたアークと接続したセル・ノードに M-cell のマークを付け、そのアークから Output arc のマークを削除する。
- 規則 7** (ループが発生し、かつ、複数の再計算方法が存在する場合)
(条件)
一つの関係ノードに接続しているアークに、Output arc と Input arc の両者のマークが付けられたものが二つあり、かつ、その関係における T-cell の中に M-cell のマークが付けられているセル・ノードがあり、かつ、バックトラック制御機構が、規則 7 をその関係に適用することを許可した場合、
(アクション)
バックトラック制御機構に対し、規則 7 をその関係に適用したことを通知し、伝搬制約セルを表すセル・ノードと関係ノードを結ぶアークから Output arc のマークを削除する。

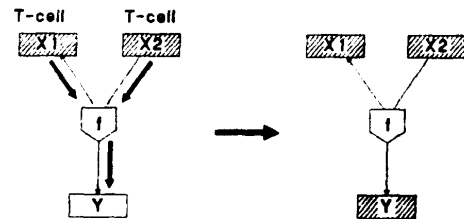


図 6 規則 3 の実行例
Fig. 6 Example of Rule 3.

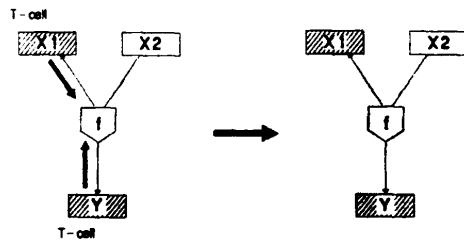


図 7 規則 4 の実行例
Fig. 7 Example of Rule 4.

規則 4 は、他の関係によって変更される可能性があるセルすべてが実際に変更され、かつ、再計算の対象となるセルが存在しない場合に、その関係が満たされているかをチェックするために用いられる。もし、関

係式が満たされていない場合、その事実をユーザに警告する。ただし、関係の成否にかかわらず、再計算は続行される。図7に規則4の実行例を示す。この例における T-cell は Y と X1 であり、かつ、それらに M-cell のマークが付けられている。また、関係ノードに接続しているアークに、Output arc のマーク付けがされたアークが存在せず、Input arc のマーク付けがされたアークが存在するため、規則4が適用されチェックが行われる。そしてアークに付けられたすべてのマークが削除される。

規則5は、ループに対処するために用いられる。図8に規則5を用いてループに対処する例を示す。この例では、ループが発生しており、セル Y の値を再計算しないかぎり、セル X2 の値が再計算されることはない。この時、関係 R1 にリンクしたアークに付けられた Output arc はただ一つであり、かつ、セル X1 に M-cell のマークが付けられているため、規則5の条件が満たされる。そして、関係 R1 に規則5を適用することにより、セル・ノード Y の値が再計算され、Output arc のマークが削除される。この結果、関係 R2 に規則3が適用され X2 の値が再計算

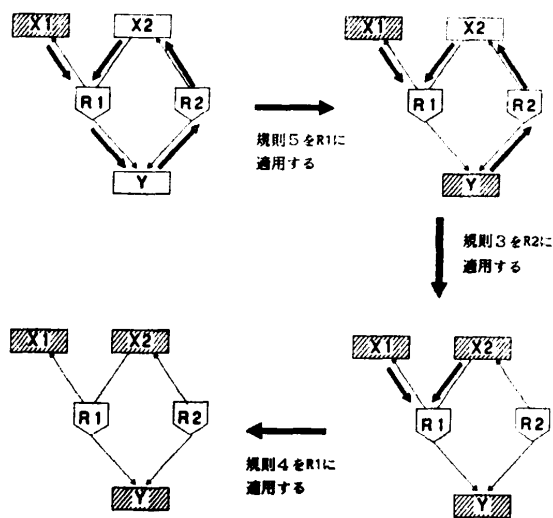


図8 規則5を用いてループに対処する例
Fig. 8 Example of the solution of a loop problem using Rule 5.

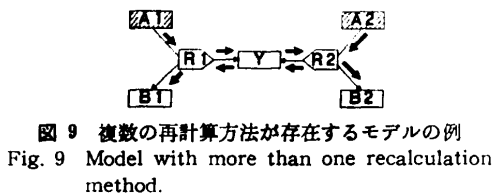


図9 複数の再計算方法が存在するモデルの例
Fig. 9 Model with more than one recalculation method.

され、その後、関係 R1 に規則4が適用され R1 が満たされているかどうかのチェックが行われる。

規則6は、図9のように、伝搬制約を満たす複数の再計算方法が存在する時に用いられる。図9では、関係 R1 における再計算と関係 R2 における再計算のどちらを先に実行しても、伝搬制約を満たす再計算方法が存在するが、その二つの再計算方法は異なっている。このような状況では、規則6を用いて、先に再計算を実行する関係を選び、再計算を続行する。ただし、この場合、複数の再計算方法があるため、この規則の適用時にはバックトラック制御機構により、チョイス・ポイントが設定される。バックトラック制御機構、および、チョイス・ポイントについては 4.2.4 項で述べる。規則6の実行の様子を図10を用いて説明しよう。図10(A)では、関係 R1 に規則6を適用している。この例では関係ノード R1 に接続しているアークのうち、セル Y と、セル B1 に接続するアークに Output arc が付けられており、かつ、Y に接続するアークに Input arc が付けられている。さらに、セル A1 に M-cell のマークが付けられている。バックトラック制御機構は、現在のチョイス・ポ

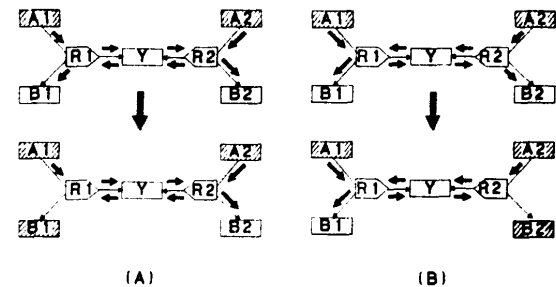


図10 規則6を用いて複数の再計算方法に対処する例
Fig. 10 Examples of the choice of a recalculation method using Rule 6.

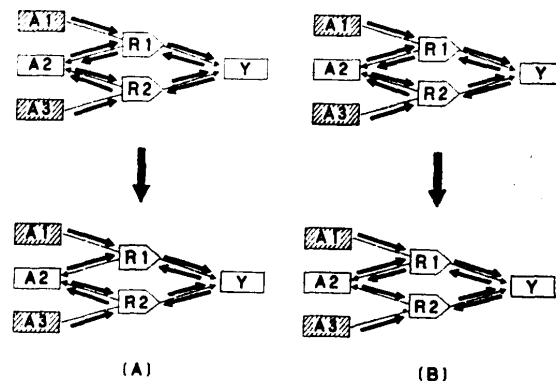


図11 規則7の例
Fig. 11 Examples of Rule 7.

イントにおいて、今までに關係 $R1$ に規則 6 が適用されたことがないと確認すると、規則 6 を關係 $R1$ に適用することを許可する。この結果、セル $B1$ に M -cell のマークが付けられ、セル $B1$ に接続するアークから *Output arc* が削除される。このアクションの持つ意味は、この関係における再計算すべきセルを Y に設定することである。ただし、セル $B1$ が変更されることによって他の関係における再計算が実行される可能性があるため、セル $B1$ の値は変更せずに M -cell のマークを $B1$ に付ける。このように、關係 $R1$ に規則 6 が適用された結果、關係 $R1$ に規則 2、規則 3 が適用され Y の値が再計算され、 $R2$ に規則 1、規則 3 が適用され $B2$ の値が再計算される。その後、バックトラックにより再計算がチョイス・ポイントまで戻されると、図 10 (B) のように $R2$ に規則 6 を適用することにより、 $R2$ により Y の値が再計算され、 $R1$ により $B1$ の値が再計算される。

規則 7 は、ループが発生し、かつ、複数の再計算方法が存在する場合に用いられる。図 11 に規則 7 の実行例を示す。この時、図 11 (A) のように、 $R1$ に規則 7 を適用することにより、 $R1$ と $A2$ の間のアークから *Output arc* が削除され、チョイス・ポイントが設定される。 $R1$ に規則 7 を適用した結果、規則 2、規則 5 が $R1$ に適用され、 $R1$ により Y が再計算される。その後、規則 1、規則 2、規則 3 が關係 $R2$ に適用され、 $A2$ が $R2$ により再計算される。そして、關係 $R1$ に規則 4 が適用され、 $R1$ の成否がチェックされる。バックトラックにより再計算がチョイス・ポイントまで戻されると、図 11 (B) のように $R2$ に規則 7 を適用することにより、 $R2$ により Y の値が再計算され、 $R1$ により $A2$ の値が再計算され、 $R2$ の成否がチェックされる。

4.2.4 バックトラック制御機構

バックトラック制御機構は、伝搬制約を満たす複数の再計算方法が存在する場合に、バックトラックを制御する。要求される機能は、以下の 4 点である。

- (1) 規則 6 および規則 7 が用いられた時に、バックトラックのためにチョイス・ポイントを設定する機能。チョイス・ポイントを設定する場合、その時のモデルの状態（各セルの値とマーク付けの状態）を記録する。また、同一のチョイス・ポイントにおいて同じ規則が同じ関係に複数回適用されることを避けるため、そのチョイス・ポイントにおいて規則

6、規則 7 がどの関係に適用されたかを記録する。

- (2) 規則 6、規則 7 をある関係に適用してもよいかを判断する機能。これは、そのチョイス・ポイントにおける規則の適用記録を調べることにより実現できる。ただし、規則 6 と規則 7 では、適用許可の判断基準が若干異なる。ある関係に対する規則 6 の適用は、その関係に対する規則 6 の適用記録が、そのチョイス・ポイントにない場合に、許可される。しかし、ある関係に対する規則 7 の適用は、そのチョイス・ポイントにおいて、その関係に対する規則 7 の適用記録がなく、かつ、規則 6 の適用記録が全く存在しない場合に、許可される。規則 7 の適用判断のために、規則 6 の適用記録を調べる理由は、規則 7 が規則 6 より優先順位が低いため、あるチョイス・ポイントにおいて、規則 6 の適用可能である場合に、バックトラック時に、規則 7 を用いることを禁止するためである。

さらに詳しく述べると、以下のような。まず、あるチョイス・ポイントにおいて、規則 6 を適用可能な関係が N 個あり、規則 7 を適用可能な関係が存在したとしよう。この時、優先順位を考慮すると先に規則 6 が適用される。そして、規則 6 を適用可能な関係が N 個あることから、バックトラックが N 回あった後には、規則 7 が適用可能になってしまう。ここで規則 7 を適用すると、そのチョイス・ポイントにおいて規則 6 が適用可能であったにもかかわらず、優先順位の低い規則 7 を適用することになる。ゆえに、この場合には、規則 7 の適用を禁止しなければならない。

- (3) バックトラックを実行する機能。バックトラックの実行を命じられた時に、バックトラック制御部は、最も新しいチョイス・ポイントに戻り、モデルの状態を復元する。そして、再計算を再実行する。もし、適用可能な規則がない場合には、そのチョイス・ポイントを削除し、さらにバックトラックを続ける。チョイス・ポイントがない場合にはバックトラックを終了する。
- (4) 初期化の機能。すべてのチョイス・ポイント

を削除する。

5. 双方向再計算機の実行例

本章では、前章で述べた手法を用いて開発された双方向再計算機能の実行例について図 12 を用いて述べる。例は、前述のローンの返済計画のシミュレーションである。ローンを返済する計画を立てる時、考慮しなければならないことに、毎月分返済する額とボーナス分で返済する額をどのように分配するかという問題がある。この問題は、双方向再計算機能を用いることにより容易に対処できる。セルの間に、図 12 (A) の三つの関係（関係式と伝搬制約セル）が定義されているとする。

ここで、借入金として 3000 万円が必要であるとしよう。そのため、B4 にその数値を入力すると、図 12 (B) のように、まず関係(1)に従って B2 の値が再計算され、次に、関係(2)に従って D2 の値が再計算される。この結果、毎月分ですべて返済する場合の

	A	B	C	D
1		借入金額	一万円当り返済額	返済額
2	毎月返済分	0	60	0
3	毎ボーナス返済分	0	500	0
4	総合計	0		

関係定義

- (1) $B2 = B4 - B3$ 伝搬制約セル: B3
 (2) $D2 = C2 * B2 / 10000$ 伝搬制約セル: B2
 (3) $D3 = C3 * B3 / 10000$ 伝搬制約セル: B3
 (A)

返済額が求められる。

ここで、毎月の返済分を 12 万円に抑えたいとしよう。そのためには、D2 に 120000 を入力すればよい。この結果、図 12 (C) のように、まず、関係(2)に従って B2 の値が再計算され、次に関係(1)に従って B3 の値が再計算され、最後に、関係(3)に従って D3 の値が再計算される。そして、この場合の毎ボーナスでの返済額が求められる。

さらに、毎ボーナスでの返済額や、ボーナス分の借入金がある値に設定した時の各返済額も、D3 や B3 の値を変更することにより求められる。

以上のように、関係式のほかに伝搬制約を明示的に書き、双方向再計算機能を用いることにより、すべてのセルが直接変更できるようになる。この結果、さまざまなシミュレーション結果を容易に得ることができ

6. おわりに

スプレッド・シートをシミュレーション・システムとして捉えた場合、スプレッド・シートは現状でも強力なシミュレーション・システムである。しかし、スプレッド・シートが持つ問題点は、セル間の依存関係を関数という形式で暗黙的にしか表現できないことであり、その結果、一方の再計算機能しか実行できない。我々は、この問題に対処し、スプレッド・シートをより強力なシミュレーション・システムとするための機能拡張として、モデル化手段としてセルとセル間に成り

しては、双方向再計算機能が実現可能な範囲で、伝搬制約の記述能力をより高めていくことである。

謝辞 この論文を書くにあたって、さまざまなアドバイスをいただいた戸沢義夫氏に、深く感謝します。

参 考 文 献

- 1) Kay, A.: コンピュータ・ソフトウェア, サイエンス, Vol. 14, No. 11, pp. 15-23 (1984).
- 2) 金井, 福永, 横井: インテリジェント・スプレッド・シートとオブジェクト指向パラダイム, *WOOC '87* (1987).
- 3) 金井, 福永, 横井: 計画作成支援のためのインテリジェント・スプレッド・シート, 第 35 回情報処理学会全国大会論文集, 4 N-4, pp. 1405-1406 (1987).
- 4) 金井, 横井, 戸沢, 福永: 双方向自動再計算機能を持ったスプレッド・シートの実現, 第 36 回情報処理学会全国大会論文集, 5 N-4, pp. 1357-1358 (1988).
- 5) Kanai, N., Yokoi, S., Fukunaga, K. and Tozawa, Y.: An Expert System to Assist Production Planning, *Proc. of the Int. W/S on*

AI for Industrial Applications, pp. 219-224 (1988).

- 6) 金井: スプレッド・シートにおける双方向自動再計算機能実現のための一手法, 第 37 回情報処理学会全国大会論文集, 4 J-4, pp. 1405-1406 (1988).
- 7) Fischer, G. and Rathke, C.: Knowledge-Based Spreadsheets, *Proc. AAAI-88*, pp. 802-807 (1988).

(昭和 63 年 12 月 23 日受付)

(平成 元年 7 月 18 日採録)



金井 直樹 (正会員)

昭和 35 年生。昭和 58 年東京大学工学部電子工学科卒業。昭和 60 年同大学院修士課程修了。同年日本アイ・ビー・エム(株)入社。現在、同社東京基礎研究所に勤務。知識工学、特にプランニング、エキスパート・システムなどの研究を行っている。ソフトウェア科学会、人工知能学会各会員。