

## シヨートノート

## 決定表間の演算を用いたルール追加およびルール削除†

齊 藤 剛††

筆者は先に決定表間の演算について報告した。本論文では、この演算の応用として、決定表に対するルール追加および削除なる2つの操作の実現法およびその検査法について述べる。本論文で述べる方法は、追加(削除)するルールもまた決定表により表現し、追加(削除)される決定表との間の演算により各操作を実現する方法である。さらに、演算により、これらの操作の検査に有用な性質が求められることを示す。本論文における議論は、決定表間の演算が決定表に対する各種の操作および検査を行うための「基本操作」の一部と成りうることを示すとともに、決定表の構成・検査の新しい方法の提案である。

## 1. はじめに

本論文では、筆者が先に報告した決定表間の演算の応用として、決定表に対するルールの追加およびルールの削除なる二つの操作が演算により実現できることを示す。さらに、各々の操作の検査に有用な性質もまた演算により求められることを示す。

筆者は、文献1)で決定表間に三つの演算—和、積、否定—を定義し、これらの演算を利用した決定表の構成法および検査法について述べた。さらに、決定表の理論モデルを用いて、これらの演算の性質について形式的に明らかにした。

さて、決定表の作成および更新作業において、ルールの追加およびルールの削除は基本的かつ重要な操作である。本論文で述べるこれらの作業の実現法は、追加(削除)するルールもまた決定表により表現し、追加(削除)される決定表との間の演算により各操作を実現する方法である。

さらに、各々の操作の検査に有用な『実際に』追加(削除)されるルールなる概念を導入し、これらのルールもまた演算により、決定表という利用者にとって見やすい形式で求められることを示す。

本論文における議論は、決定表間の演算が実際的な作業に有用な演算であることを示すものである。また、決定表間の演算が決定表に対する各種の操作および検査を行うための「基本操作」の一部と成りうることを示すものである。さらに、本論文は、演算による決定

表の構成、および、演算を利用した検査等決定表の構成・検査の新しい方法の提案である。

なお、本論文で使用する記号および用語は文献1)に従う。

## 2. ルール追加の実現法とその検査法

ルールの追加とは、追加するルールを元の決定表に書き並べることである<sup>1),2)</sup>。この書き並べることは、両者の決定表に含まれるルールの和集合を求めることに対応する。

筆者は、二つの決定表に含まれるルールの和集合により構成される一つの決定表は、それらの二つの決定表に和なる演算を施した結果と等価であることを導いた(文献1)、定理1)。

例えば図1において、決定表  $T_p$  に、決定表  $T_a$  として表されたルールを追加する場合を考える。このルール追加作業は前述のように、これら二つの決定表の和を求めることである。二つの決定表の和は、各々の決定表におけるアクションの実行条件の論理和を用いて定義されている(文献1)、定義7)。したがって、ルール追加の結果は  $Tr (= T_p \vee T_a)$  となる。詳細は文献1)に述べられている。

次に、追加操作の検査に有用な『実際に』追加されたルールなる用語を導入し、これらが演算により求められることを示す。

追加の結果である決定表  $Tr$  の各ルールは、次の三グループに分けられる。第一のグループは、追加操作以前には含まれていなかったルール、すなわち、 $T_p$  に含まれないが  $T_a$  に含まれているルールである。これらのルールを、このルール追加において実際に追加されたルールと呼ぶことにする。第二のグループは、

† Applications of the Operations on Decision Tables to Rule-Addition and Rule-Deletion by TSUYOSHI SAITOH (Department of Electrical Communication Engineering, Faculty of Engineering, Tokyo Denki University).

†† 東京電機大学工学部電気通信工学科

	R1	R2	
試験 >=80	Y	N	N
試験 >=70	-	Y	-
宿題 >=20	-	-	Y
結果 :=	1	2	

T<sub>p</sub>: A Decision Table

	R1	R2	R3
試験 >=80	Y	N	N
試験 >=70	-	-	N
宿題 >=10	-	Y	N
結果 :=	1	2	3

T<sub>a</sub>: Rules to add to T<sub>p</sub>

	R01	R02	R03
試験 >=80	Y	N	N
試験 >=70	-	Y	-
宿題 >=20	-	-	Y
宿題 >=10	-	-	Y
結果 :=	1	2	3

T<sub>r</sub>: (T<sub>p</sub> ∨ T<sub>a</sub>)  
Result of rule-addition

	R02	R03
試験 >=80	N	N
試験 >=70	N	N
宿題 >=20	N	-
宿題 >=10	Y	N
結果 :=	2	3

T<sub>v</sub>: (¬T<sub>p</sub> ∧ T<sub>a</sub>)  
The rules to be added actually

図1 演算によるルール追加と実際に追加されたルール

Fig. 1 An example of rule-addition.

T<sub>p</sub> のルールであり、しかも T<sub>a</sub> のルールでもあったルールである。また第三のグループは、T<sub>p</sub> に含まれるが T<sub>a</sub> には含まれないルールである。これらのうち、第二および第三のグループに属するルールは、実際に追加されたルールではなく、ルール追加以前から既に含まれていたルールである。

ルール追加において、どのようなルールが実際に追加されるルールかを知ることは、そのルール追加の操作の検査に有用である。この実際に追加されたルールは  $\neg T_p \wedge T_a$  により求められる。理由は次のとおり

である。

あるアクション a に関するルールで T<sub>p</sub> に含まれるルールの条件部を X<sub>p</sub> とする。また、そのアクション a に関するルールで T<sub>a</sub> に含まれるルールの条件部を X<sub>a</sub> とする。このとき、前述のように、ルール追加の結果に含まれるアクション a に関するルールの条件部は、各々の論理和すなわち X<sub>p</sub>+X<sub>a</sub> である。ここで X<sub>p</sub>+X<sub>a</sub> のうち X<sub>p</sub> でない部分は、 $\sim X_p \cdot X_a$  である。ゆえに、 $\neg T_p \wedge T_a$  により、実際に追加された部分が求められる。図1の T<sub>v</sub> が  $\neg T_p \wedge T_a$  であり、T<sub>a</sub> の各ルールを T<sub>p</sub> に追加する際に実際に追加されたルールを表す。ここで、アクション「結果:=1」に関するルールが T<sub>v</sub> 中に含まれていないのは、このアクションに関しては、実際になにも追加されていないことを表す。また、T<sub>v</sub> の R02 により、「結果:=2」に関しては「試験が 70 未満かつ宿題が 10 以上 20 未満」の場合が実際に追加されたことが示されている。

また前述の第二のグループは T<sub>p</sub> ∧ T<sub>a</sub> により、第三のグループは T<sub>p</sub> ∧ ¬T<sub>a</sub> により求めることができる。

### 3. ルール削除の実現法とその検査法

決定表からのルール削除において、削除するルールの指定法には何通りかある。例えば、ルール名を指定する方法がある。また、アクションまたは条件を指定し、それらを含むルールを削除する方法もある。これらは、いずれも、元の決定表に含まれている幾つかのルール全体を削除するものである。これに対して、幾つかのルールにより表されている決定条件の一部を削除する場合がある。このような削除の例として、決定表 T<sub>r</sub> (図1) から決定表 T<sub>d</sub> (図2) として表されたルールを削除する場合を挙げる。ここで、アクション「結果:=1」を考える。T<sub>r</sub> におけるこのアクションに関するルールの条件部は「試験 >=80 が真」であり、T<sub>d</sub> により表される削除する条件は「試験 >=90 が偽かつ宿題 >=10 が偽」である。すなわち、「試験が 80 以上である場合に結果を 1 とする」という条件判定から「試験が 90 未満かつ宿題が 10 未満」のときを削除する場合である。この場合、削除後の条件は「試験が 90 以上」または「試験が 90 未満かつ 80 以上かつ宿題が 10 以上」である。このような、条件の一部の削除を人手により行う方法として、各々の条件を論理式で表現し、それらの式変換を手で行う方法が考えられる。しかし、この方法では、条件の数が多い場合や

	R1	R2
試験 >= 90	N	-
試験 >= 80	-	N
宿題 >= 10	N	N
結果 :=	1	2

T d: Rules to delete from Tr

	R01	R02	R03
試験 >= 90	Y	-	-
試験 >= 80	Y	Y	N
試験 >= 70	-	-	-
宿題 >= 10	-	Y	Y
結果 :=	1	2	3

T k: (Tr ∧ ¬ T d)  
Result of rule-deletion

	R01	R02
試験 >= 90	N	-
試験 >= 80	Y	N
試験 >= 70	-	Y
宿題 >= 20	-	Y
宿題 >= 10	N	N
結果 :=	1	2

T c: (Tr ∧ T d)

The rules to be deleted actually

図 2 演算によるルール削除と実際に削除されたルール

Fig. 2 An example of rule-deletion.

削除する(される)条件の組合せが複雑になると困難である。

本論文で示す演算による削除は、これら条件の一部分の削除を主とした削除である。前述の、アクションまたは条件の指定による削除も演算により可能であるが本報告では省略する。ここで、あるアクション a に関するルールで、Tr に含まれるルールの条件部を Xr、また、Td に含まれるルールの条件部を Xd とする。このとき、Tr から Td を削除した結果の決定表におけるアクション a の条件部は、 $Xr \cdot (\sim Xd)$  により表現される。したがって、ルール削除の結果は Tk ( $= Tr \wedge \neg Td$ ) である。

次に、ルール追加と同様に『実際に』削除されたルールなる用語を導入する。一般に、削除しようとしたルールと実際に削除されるルールは異なる。例えば、上記例と同様にアクション「結果 := 1」に関して考える。ここで、Td により表された削除すべき条件

の組合せは「試験 >= 90 が偽かつ宿題 >= 10 が偽」である。しかし、実際に削除される条件は、削除される条件と削除すべき条件の論理積、すなわち、「試験 >= 90 が偽かつ試験 >= 80 が真かつ宿題 >= 10 が偽」である。前述の記号 Xr, Xd を用いれば、 $Xr \cdot Xd$  が実際に削除される部分である。したがって、 $Tr \wedge Td$  により求められる。図 2 の Tc ( $= Tr \wedge Td$ ) が、この例における実際に削除されたルールである。Tc において、アクション「結果 := 3」に関するルールが含まれていない。これは実際に削除されたルールが存在しないことを意味している。

また、Tc において「宿題 >= 20 が真かつ宿題 >= 10 が偽」なる条件は、決して満たされることのない条件であるが、これらが含まれている。これは、筆者のモデルでは、条件を単に文字列として扱い、条件の意味および条件間の関係が考慮されていないことに起因する。これら条件間の論理的関係の考慮については文献 1) で述べた。

#### 4. おわりに

本報告では、先に報告した決定表間の演算により、決定表に対する二つの操作が実現できること、また、それらの操作の検査に有用な性質が演算により求められることを示した。文献 1) で示したように、筆者の演算は決定表の集合上でブール代数を形成する。したがって、それらの演算で構成される写像は自明のものも含め 16 通り存在する。それらのうち、本報告では 4 個の写像の実際の意味を示したことになる。しかし、本報告における  $Tx \wedge \neg Ty$  のように、同じ写像であっても、それをどのように見るかによってその意味が異なる場合もある。したがって、演算の組合せが、決定表に対する種々の操作および各種性質の抽出に利用できると思われる。筆者らは演算および決定表の入出力を中心とし本報告および文献 1) で示した操作が利用できる決定表の利用支援システムを開発した<sup>3)</sup>。これらについては別に報告する。

謝辞 本研究の遂行および本論文の作成にあたり、ご指導いただいた東京電機大学 平松啓二教授、守屋慎次助教授に深謝いたします。また、スタンレー電気株式会社 久志本琢也氏、および情報システム第一研究室の諸氏に感謝します。また、貴重なご意見ご指示を賜りました査読者に感謝いたします。

## 参 考 文 献

- 1) 齊藤: 決定表間の演算とその応用, 情報処理学会論文誌, Vol. 27, No. 3, pp. 281-288 (1986).
- 2) Cheng, C.W. and Rabin, J.: Synthesis of Decision Rules, *Comm. ACM*, Vol. 19, No. 7, pp. 404-406 (1976).
- 3) 齊藤, 平松, 守屋: デジショナルルールベースシステムの開発とその手書き文字認識への応用, 第33回情報処理学会全国大会論文集, 5P-1 (1986).

(昭和61年11月27日受付)

(昭和62年7月9日採録)



齊藤 剛 (正会員)

昭和25年生. 昭和48年東京電機大学工学部電気通信工学科卒業. 昭和51年同大学大学院工学研究科修士課程修了. 昭和54年4月より, 東京電機大学工学部電気通信工学科助手. 言語処理, 決定表, ユーザインタフェース等の研究に従事. 電子情報通信学会, 日本ME学会, 日本LST学会等会員.