時間の論理の語モデルに対する同値性判定手続きの実現

4Ｊ－１

畑中 秀行(筑波大学 理工学研究科)*　　細野 千春(筑波大学 電子・情報工学系)†
五十嵐 滋(筑波大学 電子・情報工学系)†　水谷 哲也(筑波大学 電子・情報工学系)†

1　序

近年、プログラム検証への期待から時相論理[1]やその拡張をはじめとして、時間の概念を取り入れた論理の研究が盛んに行われている。このような体系での離散時間における真理値の遷移を正則表現[5]で表すことにより、論理式の同値性を正則表現の同値性で判定することができる。この正則表現の同値性判定には決定手続きがあり、自動化が可能である。また、決定手続きの計算は手間がかかることが多く、自動化の意義も大きい。

ここでは決定手続き、論理命題から正則表現への変換規則を示す。

2　公理系

同値性判定のプログラムに用いる語モデルである正則表現の公理系を示す。

本アルゴリズムで使用する公理系はSALOMAAの公理系[4]である。ここでは公理、及び推論規則を示す。

2．1　公理

公理は以下の11個である。

A_1　$\alpha+(\beta+\gamma)=(\alpha+\beta)+\gamma$
A_2　$\alpha(\beta\gamma)=(\alpha\beta)\gamma$
A_3　$\alpha+\beta=\beta+\alpha$
A_4　$\alpha(\beta+\gamma)=\alpha\beta+\alpha\gamma$
A_5　$(\alpha+\beta)\gamma=\alpha\gamma+\beta\gamma$
A_6　$\alpha+\alpha=\alpha$
A_7　$\phi^*\alpha=\alpha$
A_8　$\phi\alpha=\phi$
A_9　$\alpha+\phi=\alpha$
A_{10}　$\alpha^*=\phi^*+\alpha^*\alpha$
A_{11}　$\alpha^*=(\phi^*+\alpha)^*$

2．2　推論規則

推論規則は以下の２個である。

R_1　$\dfrac{\alpha=\beta \quad \gamma=\delta}{\gamma_\alpha[\beta]=\delta}$　　$\dfrac{\alpha=\beta}{\gamma_\alpha[\beta]=\gamma}$

R_2　βがe. w. p. を持たないとき、
$\dfrac{\alpha=\alpha\beta+\gamma}{\alpha=\gamma\beta^*}$

3　正則表現の同値性

同値性判定に用いる正則表現の性質を以下に示す。

3．1　方程式的に特徴づける

全ての正則表現αは以下のように方程式的に特徴づけられることが証明されている。[4]

$$\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_n\end{bmatrix}=\begin{bmatrix}\alpha_{11}&\alpha_{12}&\cdots&\alpha_{1r}\\ \vdots&\vdots&\ddots&\vdots\\ \alpha_{n1}&\alpha_{n2}&\cdots&\alpha_{nr}\end{bmatrix}\begin{bmatrix}\chi_1\\ \vdots\\ \chi_r\end{bmatrix}+\begin{bmatrix}\delta(\alpha_1)\\ \vdots\\ \delta(\alpha_r)\end{bmatrix}$$

αはr個の文字からなる。$\delta(\alpha_i)\equiv\phi^*$ または$\delta(\alpha_i)\equiv\phi$
α_1はαであり、α_{ij}はα_1からα_kのいずれかである。

3．2　正則表現の同値

２つの正則表現$\alpha=\alpha_1$, $\beta=\beta_1$において

$$\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_n\end{bmatrix}=\left[\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_n\end{bmatrix}^T\begin{bmatrix}\gamma_{11}&\gamma_{12}&\cdots&\gamma_{1n}\\ \vdots&\vdots&\ddots&\vdots\\ \gamma_{n1}&\gamma_{n2}&\cdots&\gamma_{nn}\end{bmatrix}\right]^T+\begin{bmatrix}\gamma_1\\ \vdots\\ \gamma_n\end{bmatrix}$$

$$\begin{bmatrix}\beta_1\\ \vdots\\ \beta_n\end{bmatrix}=\left[\begin{bmatrix}\beta_1\\ \vdots\\ \beta_n\end{bmatrix}^T\begin{bmatrix}\gamma_{11}&\gamma_{12}&\cdots&\gamma_{1n}\\ \vdots&\vdots&\ddots&\vdots\\ \gamma_{n1}&\gamma_{n2}&\cdots&\gamma_{nn}\end{bmatrix}\right]^T+\begin{bmatrix}\gamma_1\\ \vdots\\ \gamma_n\end{bmatrix}$$

（γ_{ij}は正則表現でありe. w. p. を持たない。）

の２つの式を満たすとき、

$$\begin{bmatrix}\alpha_1\\ \vdots\\ \alpha_n\end{bmatrix}=\begin{bmatrix}\beta_1\\ \vdots\\ \beta_n\end{bmatrix}$$

となり、それぞれが等しくなる。

3．3　対応した方程式的特徴づけ

２つの正則表現(α, β)を方程式的に特徴づけしたとき、
$\alpha_{ij}=\alpha_k$, $\beta_{ij}=\beta_k$　$(1\leq k\leq n)$
となれば、対応した方程式的特徴づけという。

対応した方程式的特徴づけができれば$|\alpha|=|\beta|$、できなければ$|\alpha|\neq|\beta|$である。

4　決定手続きアルゴリズム

２つの正則表現の同値性を判定するプログラムは、対応した方程式的特徴づけを利用する。等式として得た式の右辺、左辺の正則表現をそれぞれ方程式的に特徴づけし、方程式の数が最小になるようにする。その結果、対応した方程式的特徴づけになれば真、そうでなければ偽とする。

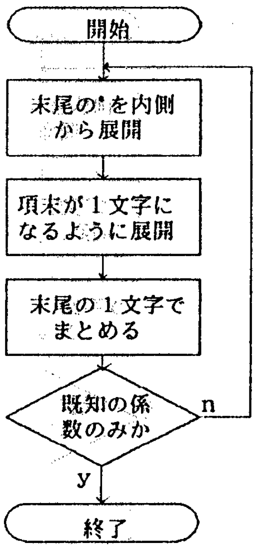

図4．1　方程式化の流れ図

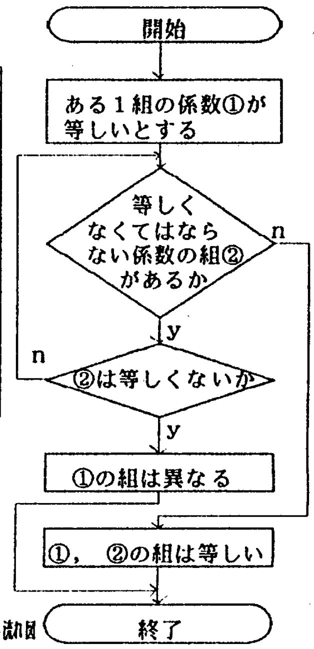

図4．2　方程式化の最小化の流れ図

Implementation of a Decision Procedure of the Equivalence in Word Models for Logic for Time
*Hideyuki Hatanaka, Master's Program of Engenieering, University of Tsukuba
†Chiharu Hosono, Shigeru Igarashi, Tetsuya Mizutani, Institute of Information Science, University of Tsukuba

4．1　方程式化アルゴリズム

方程式的特徴づけは図4．1の手順に従って行われる。このとき、末尾の1文字でまとめた部分(α_{ij})を係数と呼ぶ。また、各方程式の左辺も同様に係数と呼ぶ。

4．2　方程式の最小化アルゴリズム

4．1のアルゴリズムで出現した係数には冗長なものが含まれている。同値性の判定を行うにはこの冗長性は邪魔なものとなるので、図4．2の手順に従い冗長な係数を取り除き、最小の数の方程式による特徴づけにする。

5　時間の論理からの変換

アルファベット1文字がある時刻における全命題に対する真理値の1つの割り当てを表すとする。すると文字列は命題の真理値の時間による変遷を表していると考えられる。よって、正則表現は自然数時間（離散的な時間）における時間の論理の語モデルとできる。

5．1　拡張

時間の論理から正則表現への変換を行うため、以下の拡張をする。

5．1．1　正則表現の拡張

拡張した正則表現の定義に以下に示す。

$\Sigma=\{\chi_1, \chi_2, \cdots, \chi_r\}$はアルファベット全体の集合とする。

(1)　Σの元、およびϕは正則表現である。

(2-a) α, βが正則表現であるとき、α^*, $\alpha\cup\beta$, $\alpha\beta$は正則表現である。

(2-b) α, βが正則表現であるとき、α°, $\alpha\cap\beta$は正則表現である。

Sは入力されたアルファベットの集合を表し、S^*は入力されたアルファベットによって生成される有限の文字列全体の集合を表す。S^*を全体集合として補集合を定義する。

5．1．2　公理の拡張

2．1で示した公理を以下のように変更する。

A_1　$\alpha\cup(\beta\cup\gamma)=(\alpha\cup\beta)\cup\gamma$
A_2　$\alpha(\beta\gamma)=(\alpha\beta)\gamma$
A_3　$\alpha\cup\beta=\beta\cup\alpha$
A_4　$\alpha(\beta\cup\gamma)=\alpha\beta\cup\alpha\gamma$
A_5　$(\alpha\cup\beta)\gamma=\alpha\gamma\cup\beta\gamma$
A_6　$\alpha\cup\alpha=\alpha$
A_7　$\phi^*\alpha=\alpha$
A_8　$\phi\alpha=\phi$
A_9　$\alpha\cup\phi=\alpha$
A_{10}　$\alpha^*=\phi^*\cup\alpha^*\alpha$
A_{11}　$\alpha^*=(\phi^*\cup\alpha)^*$
A_{12}　$(\alpha\cup\beta)\cap\alpha=\alpha$
A_{13}　$(\alpha\cap\beta)\cup\alpha=\alpha$
A_{14}　$\alpha\cap(\beta\cup\gamma)=(\alpha\cap\beta)\cup(\alpha\cap\gamma)$
A_{15}　$\alpha\cup(\beta\cap\gamma)=(\alpha\cup\beta)\cap(\alpha\cup\gamma)$
A_{16}　$\alpha\cup\alpha^\circ=x^*$
A_{17}　$\alpha\cap\alpha^\circ=\phi$
A_{18}　$(\alpha\chi_i)^\circ$
$=x^*\chi_1\cup\ldots\cup\alpha^\circ\chi_i\cup\ldots\cup x^*\chi_r\cup\phi^*$

xは$\chi_1\cup\ldots\cup\chi_r$、$\alpha-\beta$は$\alpha\cap\beta^\circ$の略記である。

5．2　変換規則

5．2．1　論理命題から正則表現への変換

ある時刻における命題変数(P_1～P_r)の真理値を次の表のように1文字に割り当てる。

	P_1	P_2	・・・	P_r
χ_1	T	T	・・・	T
⋮	⋮	⋮	⋱	⋮
χ_2	F	F	・・・	F

表5．2　論理命題から正則表現への変換

5．2．3　変換規則M

時相論理の論理式から正則表現への変換規則Mを示す。

Pは命題、pはPに対応するアルファベットの和集合、Fは任意の論理式とする。

$$M(P)=px^*$$
$$M(\neg P)=x^+-M(P)$$
$$M(F_1\vee F_2)=M(F_1)\cup M(F_2)$$
$$M(F_1\wedge F_2)=M(F_1)\cap M(F_2)$$
$$M(\bigcirc F)=xM(F)$$
$$M(\Diamond F)=x^*M(F)$$
$$M(\Box F)=(x^+-x^*(x^+-M(F)))$$

ここでx^+は$x^*-\phi^*$の略記である。

5．2．3　$x^+-\alpha$となる正則表現を求める

引き算(−)が含まれたままの正則表現では、時間による論理式の真理値の遷移が直観でとらえにくくなる。そこで、引き算を含む正則表現をこれと等価な引き算を含まない正則表現にする。

$\alpha_1\equiv\alpha$とすると$\alpha_1=\alpha_{11}c_1+\cdots+\alpha_{1r}c_r+\delta_1$
と方程式化できる。
$x^+=x^*c_1+\cdots+x^*c_r+\phi$　であるので
$x^+-\alpha=(x^*-\alpha_{11})c_1+\cdots+(x^*-\alpha_{1r})c_r+\phi$
となる。ここで
$x^*-\alpha_k$
$=(x^*-\alpha_{k1})c_1+\cdots+(x^*-\alpha_{kr})c_r+(\phi^*-\delta_k)$
であるので
$x^+-\alpha$
は方程式化することができる。さらにα_1はn個の方程式で表されるとすると$x^+-\alpha$は高々n＋1個の方程式で表せる。

得られた方程式を解くことにより、$x^+-\alpha$となる正則表現を得ることができる。

6　まとめ

正則表現の同値性判定プログラムを製作した。時間の論理から正則表現に変換を行うプログラは現在製作中であり、完成させることが課題である。

正則表現は有限な文字列であるため、現在のところ有限時間での語モデルにしかなり得ない。時間を扱う論理は無限の時間を対象としているため、表現力の違いが生じてくる。この違いを補うために正則時相論理(RTL)[6]などが考案されているが、解決には至っていない。今後も正則表現の拡張および変換規則の改善などが必要である。

参考文献

[1]Hopcroft, J. E. Ullman, J. D. 野崎昭弘 高橋正子 町田元 山崎秀記 共訳 :オートマトン言語理論 計算論I, サイエンス社, 1984.
[2]Kröger, F. : Temporal Logic of Programs, Springer Verlag, 1987.
[3]Moszkowski, b. c. : Executing Temporal Logic Programs, Cambride Univ. press, 1986.
[4]Salomaa, A. : Two Complete Axiom systems for the Algebra of Regular Events, Journal of the Association for Computing Mchinery Vol.13, No.1, 1966.
[5]Salomaa, A. 北川敏男 佐藤慶子 共訳 : オートマトン論, 共立出版, 1974.