正則時相論理の充足可能性判定アルゴリズム†

平石裕実†† 濱口清治†† 矢島脩三††

超大規模論理回路技術の発達により，設計した論理システムが仕様を満たすことを確認するため，形式的検証手法の研究が重要となってきている．形式的検証では対象を形式的に記述する必要があり，命題論理や時相論理など論理体系による手法や抽象データ型などの代数的手法などが研究されてきている．このうち，命題論理を拡張して時間の概念を陽に表現できるようにした時相論理について，組合せ回路を命題論理により記述してきた経緯などから，ハードウェアの仕様記述・設計検証に関連した研究が進められている．設計対象を有限オートマトンと考えると正則集合を表現する能力が必要であるが，従来の時相論理は正則集合を記述する能力がないため，種々の時相論理が提案されてきた．しかし，拡張時相論理は ω-正則集合を表現することができるが多数の論理記号を用いる必要があり，また区間時相論理は正則集合を表現できるが充足可能性判定問題が決定不能になるなどの問題点がある．そこで，われわれは3個の時相論理記号で正則集合を表現することができ，充足可能性判定問題が決定可能である正則時相論理を提案した．形式的検証では仕様と設計の形式的記述に対して，設計が仕様を満足しているかを調べる手法が必要であり，時相論理に基づく場合は充足可能性判定を応用する手法がある．このような観点から本稿では，正則時相論理の充足可能性判定アルゴリズムを示し，その正当性と停止性を示す．

1. まえがき

近年，大規模集積回路技術の発達とともに，ハードウェアによって実現される論理システムが大規模化，複雑化してきている．このため，論理システムが誤りなく設計されていることを保証するため，論理シミュレーションが利用されてきている．しかし，シミュレーションの手法では，設計した回路が仕様を満足しているということを厳密に保証することはできないこと[1]や，順序回路を含むような回路に対しては入力パターンの作成が困難になることなどの問題がある．そこで形式的に回路の正しさを保証する方法，すなわち，形式的検証手法の確立が望まれている．

形式的検証を行う場合，論理体系によって対象の仕様および実現を記述する方法が考えられるが，論理システムの動作は動的であることから，利用する論理体系は，時間の概念を陽に表現できることが望ましい．そこで論理体系として，命題論理に時間の概念を表すための時相演算子を加えた時相論理 (Temporal Logic)[3] をとりあげる．

設計・検証の対象を有限オートマトンと考えると，用いる時相論理は正則集合と等価な表現能力を有していなければならない．しかし，従来の時相論理の体系は正則集合を表現できない[5]．このため拡張時相論理 (Extended Temporal Logic)[5] や区間時相論理 (Interval Temporal Logic)[4] が提唱されたが，拡張時相論理では任意の正則集合を記述しようとすると，多数の演算子の導入が必要であり，また，区間時相論理では，その充足可能性判定問題が決定不能になるなどの問題がある．そこで，我々は3つの時相演算子のみを含み，かつ，充足可能性判定問題が判定可能である正則時相論理 (Regular Temporal Logic)[6] を提案した．

時相論理を論理システムの検証に利用する手法の1つに，充足可能性判定を応用する方法がある[2]．具体的には，次の式 (1.1) の充足可能性を判定し，充足不能ならば，実現した論理システムが仕様を満足していると判定することになる．

A を「実現した論理システムを記述した論理式」，

B を「論理システムに対する仕様を記述した論理式」

としたとき，

$$\sim(A \supset B) \qquad (1.1)$$

(ただし，～は否定 (not) を⊃は含意 (imply) を表す)

充足可能性判定を正則時相論理を用いた形式的検証に応用するには，計算機上で実現可能なように，アルゴリズムを具体的に構築する必要がある．そこで，本稿では正則時相論理の充足可能性判定アルゴリズムを示し，停止性と正当性の証明を与える．また，形式的検証への応用のほかに，このアルゴリズムは，正則時相論理式で与えられた仕様を満足する有限状態機械の生成にも利用することができる．

以下，2章では準備として正則時相論理の定義とそ

† An Algorithm for Satisfiability Problem of Regular Temporal Logic by HIROMI HIRAISHI, KIYOHARU HAMAGUCHI and SHUZO YAJIMA (Department of Information Science, Faculty of Engineering, Kyoto University).

†† 京都大学工学部情報工学教室

の性質を示す．3章で充足可能性判定アルゴリズムを示しその停止性と正当性を証明し，さらに例題を示す．

2. 正則時相論理とその性質

本章では，正則集合と等価な表現能力を持つ正則時相論理 (Regular Temporal Logic 以下，RTL と略記する) のシンタクス，セマンティクスを定義し，さらに RTL の諸性質について述べる．

2.1 RTL のシンタクスとセマンティクス

[定義 2.1] RTL の原始記号

以下の(1)，(2)，(3)を RTL の原始記号と言う．

(1) 原始命題 a, b, c, … …

(2) 論理記号 ～, ∨, ○, ⊡, :

(3) 補助記号 (,) □

ここで，～, ∨, ○, ⊡, : をそれぞれ not, or, next, repeat, concatenation と読む．また，原始命題の集合を AP で表す．

[定義 2.2] RTL の論理式 (以下 RTL 式と略記する)

(1) a が原始命題のとき，a は RTL 式である．

(2) η, ξ が RTL 式のとき，$(\sim\eta)$, $(\eta\vee\xi)$, $(\bigcirc\eta)$, $(\boxdot\eta)$, $(\eta:\xi)$ も RTL 式である．

(3) 以上の(1)(2)を有限回適用して得られるもののみが RTL 式である． □

RTL 式全体の集合を RF で表す．また，∧(and), ⊃(imply), ≡(equivalence), V_T (恒真), V_F (恒偽) も通常の意味で用いる．ただし，単項演算子は二項演算子よりも高い優先順位を，≡は一番低い優先順位を持つものとし，明白な場合は (,) を省略する．

RTL のセマンティクスを線形時間 (linear time) モデル[3] に基づいて以下のように定義する．ただし，RTL では正則集合との関連を論じるため，通常の時相論理の場合と異なり，長さ 0 の状態の空系列 ε に対しても，論理式の真理値を与える．

なお，各論理記号の直観的な意味については，例えば文献 6) を参照されたい．

[定義 2.3] RTL のモデル

Σ を状態の集合，$B=\{T, F\}$ を真理値の集合とし，各状態 $s\in\Sigma$ および状態の空系列 ε に対する各原始命題の真理値を与える意味関数 $m:\{\varepsilon\}\cup\Sigma\times AP\to B$ を考える．$\langle\Sigma, m\rangle$ を RTL のモデルと言う． □

[定義 2.4] RTL 式の真理値

Σ 上の有限長系列 $\sigma\in\Sigma^*$ ($\sigma=\varepsilon$ または $\sigma=s_0s_1\cdots s_n$) に対して，RTL 式の真理値を与える関数 $M:\Sigma^*\times RF\to B$ をつぎのように定義する．ただし，$a\in AP$, η, $\xi\in RF$ とし，$|\sigma|$ は系列 σ の長さを表す．また，$|\sigma|\geq 2$ のとき，$\sigma_1=s_1s_2\cdots s_n$, $|\sigma|=1$ のとき，$\sigma_1=\varepsilon$ とする．

(1) $M(\sigma, a)=m(s_0, a)\cdots|\sigma|\geq 1$ のとき
$m(\varepsilon, a)\cdots\sigma=\varepsilon$ のとき

(2) $M(\sigma, \sim\eta)=T$ iff $M(\sigma, \eta)=F$

(3) $M(\sigma, \eta\vee\xi)=T$
iff $M(\sigma, \eta)=T$ または $M(\sigma, \xi)=T$

(4) $M(\sigma, \bigcirc\eta)=T$ iff $\sigma=\varepsilon$ かつ $M(\varepsilon, \eta)=T$
または，$\sigma\neq\varepsilon$ かつ $M(\sigma_1, \eta)=T$

(5) $M(\sigma, \boxdot\eta)=T$
iff $M(\alpha_i, \eta)=T$ なる $\alpha_i\in\Sigma^*$ $(i=1, 2, \cdots, m)$ が存在して，$\sigma=\alpha_1\alpha_2\cdots\alpha_m$ と表せるとき

(6) $M(\sigma, \eta:\xi)=T$ iff $M(\alpha_1, \eta)=T$ かつ $M(\alpha_2, \xi)=T$ なる $\alpha_i\in\Sigma^*$ $(i=1, 2)$ が存在して，$\sigma=\alpha_1\alpha_2$ と表せるとき □

[定義 2.5] RTL 式の充足可能性

$M(\sigma, \eta)=T$ のとき，$\langle\Sigma, m, \sigma\rangle\models\eta$ と表す．RTL 式 η に対して，$\langle\Sigma, m, \sigma\rangle\models\eta$ となる Σ, m, σ が存在するとき，η は充足可能と言う． □

なお，以下では Σ, m が自明な場合，$\langle\Sigma, m, \sigma\rangle\models\eta$ を $\sigma\models\eta$ と書くことにする．

[定義 2.6] ε 充足可能性

次に示す条件 2.1 のもとで充足可能であることを，ε 充足可能と定義する．また ε 充足可能でないことを ε 充足不能であると言う． □

[条件 2.1]

(1) すべての $s\in\Sigma$ に対し，$m(\varepsilon, p_0)=T$, かつ $m(s, p_0)=F$ であるような原始命題 p_0 が存在する．

(2) 各原始命題の空系列に対する真理値 $m(\varepsilon, a)$ (ただし，$a\in AP$) が，あらかじめ与えられている． □

以下本稿では，p_0 を条件 2.1(1) の性質を持つ原始命題として一貫して用いる．

2.2 RTL の諸性質

RTL は拡張正則表現 (通常の正則表現に共通部分と補集合を表す演算子 ∩ と～を導入したものであり，その言語のクラスは正則集合と一致する)[7] と表現等価である．ここで表現等価とは，RTL 式 η に対して，$L\langle\Sigma, m\rangle(\eta)=\{\sigma\mid\sigma\in\Sigma^*, \langle\Sigma, m, \sigma\rangle\models\eta\}$ を考えたとき，任意の m, η に対し，$L\langle\Sigma, m\rangle(\eta)$ が正則集合であり，かつ Σ 上の任意の正則集合 R に対し，L

$\langle\Sigma, m\rangle(\eta)=R$ なる m, η が存在することを言う．

なお，以下では Σ, m が自明なときは $L\langle\Sigma, m\rangle(\eta)$ を $L(\eta)$ と書くことにする．

2つの RTL 式の等価性判定は正則集合の等価性判定問題に帰着して示される（詳細は文献 6) を参照)．例えば $\boxdot\eta$ と $\eta\vee(\eta : \boxdot\eta)$ の等価性を言うためには，$L(\boxdot\eta)$ と $L(\eta\vee(\eta : \boxdot\eta))$ が任意の解釈 $\langle\Sigma, m\rangle$ のもとで等しいことを示せばよい．この場合，$L(\eta\vee(\eta : \boxdot\eta))=L(\eta)\cup L(\eta)L(\eta)^+=L(\eta)^+$ であるから，$\boxdot\eta=\eta\vee(\eta : \boxdot\eta)$ が言える．

次に，3章で用いる初等式を定義する．

[定義 2.7] 初等式

次の(1)-(3)を初等式 (elementary formula)[5] と呼び，その集合を EF と書く．

(1) 原始命題とその否定

(2) ○式…$\bigcirc\eta$ の形をした式．

(3) ε 式…次のように定義される式．

$\alpha^\varepsilon \triangleq (\alpha\wedge p_0) : \mathrm{V_T}$ ただし，$\alpha\in RF$. □

[性質 2.1] ε 式の性質

ε 式に関しては次に示す性質がある．これにより，各原始命題の空系列 ε に対する真理値があらかじめ与えられていれば，任意の ε 式の真偽を判定することができる．

(1) $(\eta\wedge\xi)^\varepsilon\equiv\eta^\varepsilon\wedge\xi^\varepsilon$

(2) $(\eta\vee\xi)^\varepsilon\equiv\eta^\varepsilon\vee\xi^\varepsilon$

(3) $(\sim\eta)^\varepsilon\equiv\sim(\eta^\varepsilon)$

(4) $(\bigcirc\eta)^\varepsilon\equiv\eta^\varepsilon$

(5) $(\eta : \xi)^\varepsilon\equiv\eta^\varepsilon\wedge\xi^\varepsilon$

(6) $(\boxdot\eta)^\varepsilon\equiv\eta^\varepsilon$

(7) $(\eta^\varepsilon)^\varepsilon\equiv\eta^\varepsilon$ □

3. 充足可能性判定アルゴリズム

充足可能性判定を形式的検証に応用する場合，論理回路とその動作を次のように対応させる方法がある．

- 信号線（その値 0, 1)…原始命題（真理値 T, B)
- 時刻（離散時間とする)…RTL のモデルの状態

このとき，ε に対する原始命題の真理値をあらかじめ与えても，記述に支障はないと考えられる．そこで，本章では ε 充足可能性判定について，アルゴリズムとその停止性および正当性を示し，最後に ε 充足可能性判定の例を示す．

3.1 充足可能性判定アルゴリズム

従来より，時相論理の充足可能性判定にはテーブル法 (tableau method)[3),5)] が用いられてきており，本節で示すアルゴリズムもこれに基づいている．

アルゴリズムは次の(1)グラフ生成と(2)節点消去の2つの段階からなる．

(1) 与えられた式を現在の状態，次状態以降および空系列にそれぞれに対応する部分，すなわち原始命題またはその否定，○式，ε 式（つまり，初等式）のみからなる式に変形し，これで節点をラベル付けする．次に○式から演算子○を取り除いた式に対して同様の操作を行って得られた式で，新たに生成した子節点をラベル付けする．これを繰り返してグラフを生成する．

(2) (1)で得られたグラフに対し，各節点ごとにラベル付けされている式の充足可能性判定を行い，これを繰り返して，与えられた式に対する判定を行う．

まず，アルゴリズム全体の構成を示す．

[アルゴリズム 3.1] RTLSAT

(入力) RTL 式 h，各原始命題の ε に対する真理値．

(出力) h が ε 充足可能ならば "yes"，それ以外は "no".

(方法) 以下に示す(1)グラフ生成と(2)消去規則による節点の消去の2つの段階からなる．停止した時点で初期節点が残っていれば ε 充足可能であり，そうでなければ ε 充足不能である． □

以下では式の集合を式集合と呼び，式集合 S_i でラベル付けされた節点 v を $v[S_i]$ と記す．また $\mathrm{V_T}$ を原始命題として扱う．

(1) グラフ生成のアルゴリズム

以下で述べるように生成されるグラフの節点は，初等式の集合でラベル付けするものとする．この方針に従い，与えられた式 h に対して，初期節点を {○h} でラベル付けする．次に初等式をリテラルとみなした積和形，すなわち，$\bigvee_i(\bigwedge_j f_{ij})$ の形式（ただし，$f_{ij}\in EF$) に h を変形する．これに対し各積項，すなわち $\bigwedge_j f_{ij}$ に対応する子節点を生成し，この子節点を $S_i \triangleq \{f_{i1}, \cdots, f_{in_i}\}$（積項を集合の形式で表現したもの）によってラベル付けする．次いで $^\forall\bigcirc g_{ij}\in S_i$ から $\bigwedge_j g_{ij}$ を構成し，同じ操作を繰り返してグラフを生成してゆく．したがってこのとき，グラフでは枝の分岐が $\bigvee_i$ に対応することになる．

また，RTL では式が空系列 ε に対しても真になりうるので，各節点の子節点を作る際に，後述の ε 節点という特別な節点を生成しておくことにする．

さらに，同一の式でラベル付けされた節点は生成し

ないものとする.

アルゴリズムを示す前に，非初等式を，初等式からなる積和形に変形するための規則，すなわち分解規則を示す．分解規則は非初等式 η を式 f_{ij} の集合 S_i の集合へ写像する規則であり，次の形で表現される．

$$\eta \to \bigcup_i \{\{f_{i1}, \cdots, f_{in_i}\}\},\quad \eta \to \bigcup_i \{\bigcup_j \{f_{ij}\}\}$$

または，$\eta \to \bigcup_i \{S_i\}$

(ただし η, f_{ij} は RTL 式，$S_i=\{f_{i1}, f_{i2}, \cdots, f_{in_i}\}$, $i=1,2,\cdots,m$, $j=1,2,\cdots,n_i$ であり，m は η によって定まり，n_i は各 i に対して定まる.)

なお，以下本稿では記法の簡略化のため，添字の範囲を省略している.

[手続き 3.1] 分解規則の再帰的適用

分解規則の再帰的適用とは，次に示す Γ から Γ' を得る手続き（式(3.1)）を，再帰的に行うことを言う.

$\Gamma=\bigcup_i \{S_i\}$ が与えられ，$\eta \in S_i$ かつ $\eta \notin EF$ かつ $\eta \to \bigcup_j \{S'_{ij}\}$ のとき，つぎの Γ' を構成する.

$$\Gamma'=(\Gamma-\{S_i\}) \cup \bigcup_j \{(S_i-\{\eta\}) \cup S'_{ij}\} \qquad (3.1)$$

また，この変形を(3.2)のように書き，(3.2)の変形を0回以上行って Γ'' を得たとき(3.3)のように書く.

$$\Gamma \Rightarrow \Gamma' \qquad (3.2)$$

$$\Gamma \overset{*}{\Rightarrow} \Gamma'' \qquad (3.3)$$

□

[定義 3.1] 分解規則

ただし，$\supset$, $\equiv$ は $\wedge$, $\vee$, $\sim$ で表現するものとする.

(1) $f_1 \vee f_2 \to \{\{f_1, f_2\}\}$

(2) $\sim(f_1 \vee f_2) \to \{\{\sim f_1, \sim f_2\}\}$

(3) $\sim\sim f \to \{\{f\}\}$

(4) $\sim\bigcirc f \to \{\{\bigcirc\sim f\}\}$

(5) $f_1 \vee f_2 \to \{\{f_1\}\{f_2\}\}$

(6) $\sim(f_1 \wedge f_2) \to \{\{\sim f_1\}\{\sim f_2\}\}$

(7) $f_1 : f_2$…以下の(i)(ii)(iii)により構成する.

(i) $\{\{f_1\}\}$, $\{\{f_2\}\}$ に分解規則を適用可能である限り再帰的に適用し，次のようになったとする.

$$\{\{f_1\}\} \overset{*}{\Rightarrow} \bigcup_i \{\bigcup_k \{s^e_{ik}\} \cup \bigcup_l \{a_{il}\} \cup \bigcup_m \{\bigcirc x_{im}\}\}$$

$$\{\{f_2\}\} \overset{*}{\Rightarrow} \bigcup_j \{\bigcup_k \{t^e_{jk}\} \cup \bigcup_l \{b_{jl}\} \cup \bigcup_m \{\bigcirc y_{jm}\}\}$$

($s_{ik}, t_{jk}, x_{im}, y_{jm} \in RF$, a_{il}, b_{jl} は原始命題またはその否定)

(ii) それぞれの i, j に対し，次の S_i, S_j を構成する.

① すべての l に対し，$a_{il} \neq p_0$ のときア)，そうでなければイ).

ア) $S_i=\bigcup_k \{s^e_{ik}\} \cup \bigcup_l \{a_{il}\} \cup \{\bigcirc((\bigwedge_m x_{im}) : f_2)\}$

ただし x_{im} が存在しないときは V_T を補うものとする.

イ) $S_i=\phi$ (ϕ は空集合を表す)

② $S_j=\{f_1{}^e\} \cup \bigcup_k \{t^e_{jk}\} \cup \bigcup_l \{b_{jl}\} \cup \bigcup_m \{\bigcirc y_{jm}\}$

(iii) $f_1 : f_2 \to \bigcup_i \{S_i\} \cup \bigcup_j \{S_j\}$

(8) $\sim(f_1 : f_2)$

$\{\{f_1 : f_2\}\} \to \bigcup_i \{\bigcup_j \{\xi_{ij}\}\}$

となったとする．$\sim(\bigvee_i(\bigwedge_j \xi_{ij}))$ をド・モルガンの法則，分配則および $\sim\bigcirc f=\bigcirc\sim f$ の関係式を使って，$\bigvee_i(\bigwedge_j \theta_{ij})$ に変形し（ただし ξ_{ij}, $\theta_{ij} \in EF$），分解規則を次のように構成する.

$$\sim(f_1 : f_2) \to \bigcup_i \{\bigcup_j \{\theta_{ij}\}\}$$

(9) $\boxdot f$ 次の(i)～(iii)により分解規則を構成する.

(i) $\{\{f\}\}$ に適用可能な限り分解規則を適用し，次のようになったとする.

$$\{\{f\}\} \overset{*}{\Rightarrow} \bigcup_i \{\bigcup_k \{u^e_{ik}\} \cup \bigcup_l \{c_{il}\} \cup \bigcup_m \{\bigcirc z_{im}\}\}$$

(ただし，$u_{ik}, z_{im} \in RF$, c_{il} は原始命題またはその否定)

(ii) すべての i に対し，次の S_i', S_i'' を構成する.

すべての l に対し，$c_{il} \neq p_0$ のときア)，そうでなければイ).

ア) $S_i'=\bigcup_k \{u^e_{ik}\} \cup \bigcup_l \{c_{il}\} \cup \bigcup_m \{\bigcirc z_{im}\}$

$S_i''=\bigcup_k \{u^e_{ik}\} \cup \bigcup_l \{c_{il}\} \cup \{\bigcirc((\bigwedge_m z_{im}) : \boxdot f)\}$

ただし z_{ij} が存在しないときは V_T を補うものとする.

イ) $S_i'=\{p_0\} \cup \bigcup_k \{u^e_{ik}\} \cup \bigcup_l \{c_{il}\} \cup \bigcup_m \{\bigcirc z_{im}\}$

$S_i''=\phi$ (ϕ は空集合を表す)

(iii) $\boxdot f \to \bigcup_i \{S_i'\} \cup \bigcup_i \{S_i''\}$

(10) $\sim\boxdot f$ (8)と同様 $\{\{\boxdot f\}\}$ に対する分解を行ったのち，$\sim\boxdot f$ に対する分解規則を構成する. □

分解規則に関しては次の補題が成立する.

[補題 3.1] 分解規則の性質

$\eta \to \bigcup_i \{\{f_{i1}, f_{i2}, \cdots, f_{in}\}\}$ に対し次の関係が成立する.

$\eta=\bigvee_i(\bigwedge_j f_{ij})$

(証明) 付録1に示す. □

次にグラフ生成のアルゴリズムを示す.

[アルゴリズム 3.2] グラフ生成

グラフは有向グラフ $G=(V,E)$ (V は節点の集合, E は有向枝の集合) であり, 各節点は式集合によってラベル付けされる.

以下で, $V\leftarrow V\cup\{a\}$ は左辺を右辺で置き換えることを意味する. また, 式集合に p_0 を含む節点を ε 節点, V_T のみを含む節点を V_T 節点と呼ぶ.

① 与えられた式 h に対し, 初期節点を $v_0[\{\bigcirc h\}]$, $V=\{v_0\}$, $E=\phi$, $V_E=\{v_0\}$ とする (V_E は評価中の節点の集合を表す).

② $V_E=\phi$ ならば終了, そうでなければ③へ.

③ V_E から節点を取り出して, これを $v[R]$ とする. $V_E\leftarrow V_E-\{v[R]\}$. このとき, R に○式が含まれていればア) へ, そうでなければイ) へ.

ア) $\bigcirc f_i\in R$ $(i=1,\cdots,n)$ に対して

$$\{\{f_1,\cdots,f_n\}\}\Rightarrow\bigcup_i\{\bigcup_j\{g_{ij}\}\}\triangleq\bigcup_i\{R_i\}\ (g_{ij}\in EF)$$

が得られたとする. 次の 1), 2) を順に行う.

1) ${}^\exists w[\{p_0,f_1,\cdots,f_n\}]\in V$ のとき,

$E\leftarrow E\cup\{(v,w)\}$.

そうでなければ,

$V\leftarrow V\cup\{v'[\{\{p_0,f_1,\cdots,f_n\}\}]\}$,

$E\leftarrow E\cup\{(v,v')\}$.

2) すべての i に対し,

${}^\exists w[R_i]\in V$ のとき a) へ, そうでなければ b) へ.

a) $E\leftarrow E\cup\{(v,w)\}$.

b) $V\leftarrow V\cup\{v_i[R_i]\}$, $E\leftarrow E\cup\{(v,v_i)\}$, $p_0\notin R_i$ のときのみ $V_E\leftarrow V_E\cup\{v_i[R_i]\}$.

イ) ${}^\exists w[\{V_T\}]\in V$ のとき, $E\leftarrow E\cup\{(v,w)\}$.

そうでなければ, $V\leftarrow V\cup\{v'[\{V_T\}]\}$, $E\leftarrow E\cup\{(v,v')\}$.

④ ②へ戻る. □

(2) 消去規則による節点消去のアルゴリズム

節点にラベル付けされている式の充足不能性を以下の消去規則により判定し, 該当節点を消去してゆく.

[アルゴリズム 3.3] 節点消去

グラフを生成したのち, 以下の定義 3.2 の消去規則を該当する節点がなくなるまで繰り返し適用する. □

[定義 3.2] 消去規則

節点 $v[R]\in V$ が次のいずれかを満足すれば消去する.

(1) ある a が存在して $a\in R$ かつ $\sim a\in R(a\in AP)$.

(2) ε式 $\alpha^\varepsilon\in R$ が存在して, $M(\varepsilon,\alpha)=\mathrm{F}$.

(3) $p_0\in R$ かつある $f\in R$ について $M(\varepsilon,f)=\mathrm{F}$.

(4) $\sim V_T\in R$.

(5) $(v,v')\in E$ なるすべての v' が消去されている.

(6) v から V_T 節点または ε 節点に至る経路がない. □

なお, ここではアルゴリズムを簡明にするため, グラフ生成ののち節点消去を行っているが, 節点生成と同時に消去規則を適用することにより, 効率の改善が可能である (3.3 節を参照).

また, ここではε充足可能性判定, すなわち空系列 ε に対する原始命題の真理値をあらかじめ与えた場合を対象にしているが, 本節のアルゴリズムを拡張して ε に対する原始命題の真理値を与えない場合のアルゴリズムも容易に構成することができる[8].

さらに, 消去規則適用後のグラフからは, 適当な変形によって状態遷移図を得ることができる. 詳細は本論文の範囲を越えるので省略するが, この状態遷移図は与えられた RTL 式 h を仕様とする有限状態機械とみなすことができる. つまり, アルゴリズム RTLSAT は, RTL 式で記述された仕様から, 有限状態機械を生成することにも利用できる.

3.2 アルゴリズムの停止性と正当性

本節では, 3.1 節で示したアルゴリズム RTLSAT の停止性と正当性について述べる.

[定理 3.1] RTLSAT は停止する.

(証明) 付録2に掲げる. □

[定理 3.2] RTLSAT が yes と出力すればε充足可能, no と出力すればε充足不能である.

(証明) h を与えられた式とする. $L(\bigcirc h)=\sum L(h)$ であるから, $L(h)=\phi$ (h がε充足不能) と $L(\bigcirc h)=\phi$ ($\bigcirc h$ がε充足不能) は同値である. 以下では, $\bigcirc h$ のε充足可能性判定が可能であることを示す.

「初期節点が消去されている」→「$\bigcirc h$ はε充足不能」

節点 $v[R]\in V$ が消去規則によって消去されるとき, $f_i\in R$ $(i=1,\cdots,n)$ に対し $\bigwedge_i f_i$ がε充足不能であることを示す. v が(1)-(6)の各々の消去規則によって消去された場合について調べる.

(1)-(4)の場合は自明

(5) ${}^\forall\bigcirc g_j\in R$ に対し $\bigwedge_j g_j$ がε充足可能でないとすれば, $\bigwedge_i f_i$ もε充足可能でない.

(6) 消去規則(6)より, v から始まる任意の有限の経路 $vv_1\cdots v_l$ について, $v_l[R_{v_l}]$ に対し R_{v_l} は○式

を含む．すなわち，v_i の子節点に $\bigwedge_i f_i$ のε充足可能性判定は依存している．このことは $\bigwedge_i f_i$ を真にする有限の状態系列が存在しないこと，すなわち $\bigwedge_i f_i$ がε充足不能であることを意味する．

「初期節点が消去されていない」→「$\bigcirc h$ がε充足可能」

消去規則(5)と(6)より，初期節点が残っていれば初期節点からε節点または V_T 節点へ至る経路が存在する．そのような経路の中で長さが有限のものを取り出し，初期節点から順に番号 i ($i=0,1,\cdots,n$) をつけ，v_i と表すことにする（つまり，初期節点は v_0）．

このとき，$\langle\Sigma,m,\sigma\rangle\models\bigcirc h$ なる $\langle\Sigma,m,\sigma\rangle$ を次の(1)-(3)により構成する．

(1) $\Sigma=\phi$, $\sigma=\varepsilon$ とする．

(2) $i=0$ から $n-1$ まで以下の①-③を繰り返す．

① $\Sigma\leftarrow\Sigma\cup\{s_i\}$.

② $v_i[R_i]$ に対して，$a\in AP$ として，

$m(s_i,a)=\mathrm{T}$ if $a\in R_i$

$m(s_i,a)=\mathrm{F}$ if $\sim a\in R_i$

$m(s_i,a)$ の値は任意 if $a,\sim a\notin R_i$.

③ $\sigma\leftarrow\sigma s_i$.

(3) v_n が V_T 節点ならば， $\sigma\leftarrow\sigma\tau$ ($\tau\in\Sigma^*$ は任意)．

v_n がε節点ならば， $\sigma\leftarrow\sigma$.

したがって，一般に $\sigma=s_0\cdots s_{n-1}\tau$ ($\tau\in\Sigma^*$) と書くことができる．また，$\sigma_i=s_i\cdots s_{n-1}\tau$ ($i=0,\cdots,n-1$)，$\sigma_n=\tau$ であるとする．

$\langle\Sigma,m,\sigma\rangle\models\bigcirc h$ を i に関する帰納法により示す．

(基礎) $i=n$ のとき $v_n[R_n]$ に対し $f_{nj}\in R_n$ とする．

v_n が V_T 節点ならば，$\sigma_n=\tau\models V_T$.

v_n がε節点ならば，$\sigma_n=\varepsilon\models\bigwedge_j f_{nj}$.

(帰納的段階) $v_i[R_i]$ ($i\geq 1$) に対し $f_{ij}\in R_i$ として，$\sigma_i\models\bigwedge f_{ij}$ と仮定する．$v_{i-1}[R_{i-1}]$ に対し，

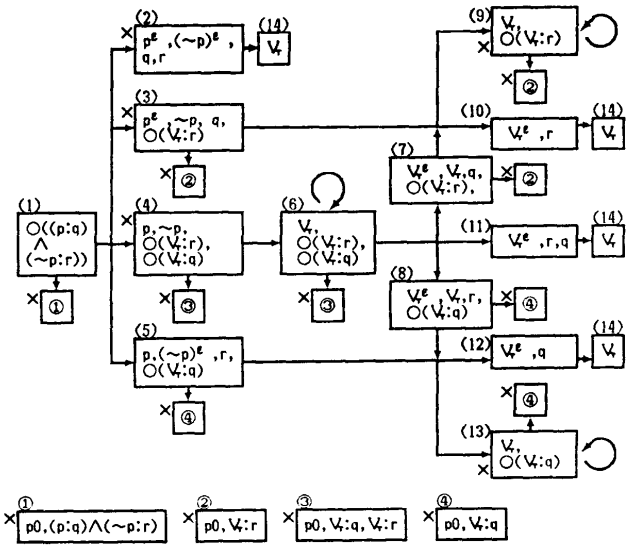
$R_{i-1}=\{s_1{}^{\varepsilon},\cdots,s_k{}^{\varepsilon},a_1,\cdots a_l,\bigcirc x_1,\cdots,\bigcirc x_m\}$

とする．ここで，$s_j,x_j\in RF$, a_j は原始命題またはその否定とする．

$\sigma_{i-1}\models s_1{}^{\varepsilon}\wedge\cdots\wedge s_k{}^{\varepsilon}\wedge a_1\wedge\cdots\wedge a_l$

は消去規則(1)(2)と m の構成より明らか．$\sigma_i\models\bigwedge f_{ij}$, 補題3.1および消去規則(3)より，$\sigma_i\models x_1\wedge\cdots\wedge x_m$. したがって，

$\sigma_{i-1}\models\bigcirc x_1\wedge\cdots\wedge\bigcirc x_m$.



ただし，p,q,r はεに対して偽となる原始命題とする．

図1 ε充足可能性判定の例
Fig. 1 Example of deciding ε-satisfiability.

ゆえに，$\sigma_0\models\bigcirc h$ すなわち $\langle\Sigma,m,\sigma\rangle\models\bigcirc h$ となる．

□

3.3 例

本節ではアルゴリズム RTLSAT 1 による RTL 式のε充足可能性判定の例を示す．

(入力) $h=(p:q)\wedge(\sim p:r)$

ただし，p, q, r はεに対し偽となる原始命題とする．

(グラフ生成) **図1**に式 h に対するグラフを示す．以下では番号は図中の節点に付けられた番号に対応するものとする．

図中，同一番号の節点は同一の節点とみなす．①-④はε節点である．(1)が子節点を生成する過程を見ると，まず，ε節点①を生成する．次に，$\bigcirc h$ から h の部分を取り出し分解規則を適用すると，式(3.4)の右辺の式集合を得る．

$$h\Rightarrow\{\{p^{\varepsilon},(\sim p)^{\varepsilon},q,r\},\{p,(\sim p)^{\varepsilon},r,\bigcirc(V_T:q)\},\{p^{\varepsilon},\sim p,q,\bigcirc(V_T:r)\},\{p,\sim p,\bigcirc(V_T:r),\bigcirc(V_T:q)\}\}\qquad(3.4)$$

これらと同一の式集合でラベル付けされた節点は存在していないので，(2)-(5)が生成される．

このようにグラフ生成を続けてゆくと，図1を得る．

(節点消去) 図中，×印を付けた節点には消去規則が適用される．(4)は消去規則(1)により，(2)と(3)は消去規則(2)により，①-④は消去規則(3)により，

また(9)と(13)は消去規則(6)により消去される.

(ε充足可能性判定) 図1では初期節点が残っている. したがって, h はε充足可能である. 定理3.2で示した方法により経路(1)(5)(12)(14)を選び, $\langle\Sigma, m, \sigma\rangle \models h$ なる Σ, m, σ を構成すると次のようになる.

$\Sigma = \{s_1, s_2\}$.

$m(s_1, p)=\mathrm{T}$, $m(s_1, r)=\mathrm{T}$, $m(s_2, q)=\mathrm{T}$.

$\sigma = s_1 s_2 \tau$ ($\tau \in \Sigma^*$ は任意).

3.1節で述べたように節点生成と同時に節点消去を行うと効率を改善することができ, 例えば図1では, 節点(2)-(4)が消去されることから節点(6)-(11)の生成を省略することが可能である.

4. むすび

時相論理は時間に関する概念を表現することができるという性質を持っているため, これまで, ソフトウェアやハードウェアの記述に利用されてきており, その充足可能性判定は形式的検証に利用することができる. そこで本稿では, 正則集合と等価な表現能力を持つことが示されている正則時相論理の充足可能性判定アルゴリズムを提案し, その停止性と正当性を示した. また, このアルゴリズムは与えられた論理式を仕様とする有限状態機械の生成にも応用することができる.

今後の課題としては, まずアルゴリズムの改良が挙げられる. 命題論理の充足可能性判定を真に含む問題であるため, 少なくとも指数時間の計算量を必要とすると考えられ, 大きな困難が予想されるが一層の工夫を必要とする点でもある.

また, 充足可能性判定のほかにモデル・チェッカの手法, すなわち実現をモデルとみなし, 仕様を表現する論理式の真偽判定により検証を行うことも考えられる.

さらに, 正則時相論理による仕様および実現の一般的な記述手法の確立は, ハードウェアをいかに記述するかという問題とともに, 解決してゆくべき興味深い課題であろう.

謝辞 種々御議論いただいた高木直史博士, 石浦菜岐佐氏をはじめ本学矢島研究室の皆様に深謝いたします.

参考文献

1) 丸山文宏, 上原貴夫: 方式・機能・論理設計の検証, 情報処理, Vol. 25, No. 10, pp. 1062-1070 (1984).
2) 中村 宏ほか: Tokio に基づく論理回路の検証, 情報処理学会研究報告, 86-DA-34-2 (1986).
3) Rescher, N. and Urquhart, A.: *Temporal Logic*, Springer-Verlag, Wien, New York (1971).
4) Hailpern, B.: Verifying Concurrent Processes Using Temporal Logic, Ph. D thesis, Stanford University, New York (1980).
5) Wolper, P.: Temporal Logic can be More Expressive, *Proc. 22nd Annual Symp. on Foundation of Computer Science*, pp. 340-348 (1981).
6) 平石裕実, 矢島脩三: 正則集合と表現等価な正則時相論理 RTL, 情報処理, Vol. 28, No. 2, pp. 117-123 (1986).
7) Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: *The Design and Analysis of Computer Algorithms* Addison-Wesley, Menlo Park, Massachusetts (1974).
8) 平石裕実, 瀧口清治, 矢島脩三: 正則時相論理の充足可能性判定アルゴリズム, 電子情報通信学会技術研究報告資料, COMP 87-2, pp. 11-20 (1987).

付録1 補題3.1の証明

$S(a)=\{s \mid s\in\Sigma, m(s,a)=\mathrm{T}\}$.

$L(\eta)=\{\sigma \mid \sigma\in\Sigma^*, \sigma\models\eta\}$.

と定める. ここで $a\in AP$, $\eta\in RF$ とする.

定義3.1の分解規則において,

(1)-(6)については文献[6]を参照.

(7)について, a_{il}, b_{jl} を原始命題またはその否定, s_{ik}, t_{jk}, x_{im}, y_{jm} を RTL 式として,

$f_1=\bigvee_i f_{1i}$, ここで,

$$f_{1i}=\bigwedge_k s_{ik}^{\varepsilon}\wedge\bigwedge_l a_{il}\wedge\bigwedge_m\bigcirc x_{im}.$$

$f_2=\bigvee_j f_{2j}$, ここで,

$$f_{2j}=\bigwedge_k t_{jk}^{\varepsilon}\wedge\bigwedge_l b_{jl}\wedge\bigwedge_m\bigcirc y_{jm}.$$

$$g=\bigvee_i g_i\vee\bigvee_j g_j \qquad (1)$$

ここで g_i と g_j はそれぞれ f_{1i} と f_{2j} に対応させて

$g_i=\bigwedge_k s_{ik}^{\varepsilon}\wedge\bigwedge_l a_{il}\wedge\bigcirc((\bigwedge_m x_{im}) : f_2)$

……すべての l に対し $a_{il}\neq p_0$ のとき

$\mathrm{V_F}$……それ以外のとき

$g_j=f_1^{\varepsilon}\wedge\bigwedge_k t_{jk}^{\varepsilon}\wedge\bigwedge_l b_{jl}\wedge\bigwedge_m\bigcirc y_{jm}$

であるとする. g_i と g_j はそれぞれ分解規則(7)で得られる S_i, S_j に対応する RTL 式であるので,

$L(f_1 : f_2)=L(g)$

を示せばよい.

$$L(f_1:f_2)=L(f_1)L(f_2)$$
$$=(\{\varepsilon\}\cap L(f_1))L(\bigvee_j f_{2j})$$
$$\cup(\Sigma^+\cap L(\bigvee_i f_{1i}))L(f_2)$$
$$=\bigcup_j((\{\varepsilon\}\cap L(f_1))L(f_{2j}))$$
$$\cup\bigcup_i((\Sigma^+\cap L(f_{1i}))L(f_2))$$

である．そこで任意の i, j について，

$$(\{\varepsilon\}\cap L(f_1))L(f_{2j})\cup(\Sigma^+\cap L(f_{1i}))L(f_2)$$
$$=L(g_j)\cup L(g_i) \qquad (2)$$

を証明すればよい．ただし，式(1)で $\varepsilon\in L(g_i)$ ならば $\varepsilon\in L(g)$ であり，g_i の定義から $\varepsilon\in L(f_{1i})\subseteq L(f_1)$, $\varepsilon\in L(f_2)$ である．f_2 の定義より，$\varepsilon\in L(g_j)$ となる g_j が必ず存在するので，以下では $\varepsilon\notin L(g_i)$ を仮定して，式(2)を証明してよい．

(1) $(\{\varepsilon\}\cap L(f_1))L(f_{2j})=L(g_j)$ を示す．

① $\varepsilon\in L(f_1)$ のとき，

$$(\{\varepsilon\}\cap L(f_1))L(f_{2j})=L(f_{2j})$$
$$=\Sigma^*\cap L(f_{2j})=L(g_j)$$

② $\varepsilon\notin L(f_1)$ のとき，$L(f_1)=\phi$ であるから，

$$(\{\varepsilon\}\cap L(f_1))L(f_{2j})=L(g_j)=\phi$$

(2) $(\Sigma^+\cap L(f_{1i}))L(f_2)=L(g_i)$ を示す．

① ある a_{il} が p_0 のとき，

$L(g_i)=L(V_E)=\phi$ であり，$L(f_{1i})=\varepsilon$ または ϕ であることより $(\Sigma^+\cap L(f_{1i}))L(f_2)=\phi$.

② それ以外のとき，

$\varepsilon\notin L(g_i)$ より，

$$L(g_i)$$
$$=\bigcap_k L(s_{ik})\cap\bigcap_l S(a_{il})\Sigma^*\cap\Sigma(\bigcap_m L(x_{im}))L(f_2)$$

ここで，$L(s_{ik})=\Sigma^*$ または ϕ であることより，

$$=(\bigcap_k L(s_{ik})\cap\bigcap_l S(a_{il})\Sigma^*\cap\bigcap_m\Sigma L(x_{im}))L(f_2)$$
$$=(\Sigma^+\cap L(f_{1i}))L(f_2)$$

(8), (10) $f_1:f_2$ および $\boxdot f$ に対する分解は(7)および(9)より自明．また，ド・モルガン則，分配則および $\sim\bigcirc f\equiv\bigcirc\sim f$ が成立することは(1)-(6)より明らかであるから(8), (10)についても補題が言える．

(9)について

$\boxdot f\equiv f\vee(f:\boxdot f)$ の関係[6]に基づいて分解規則は構成されている．(7)と同様に証明される． □

付録 2 定理 3.1 の証明

RTLSAT の停止性を示す．まず次の(*1)を言う．

「$\{\bigcup_i\{R_i\}\}$ (R_1 は式集合) に対し，分解規則を有限回，再帰的に適用すれば，初等式のみを含む式集合に変形できる」 (*1)

補題1より分解規則の再帰的適用は，初等式をリテラルとみなしてブール式の場合と同様に積和形に変形することと等価である．したがって，分解の際に出現する式すべてが，：と $\boxdot$ を含まなければ(*1)は明らか．

また，f_1, f_2, f を積和形に変形することができたとすると，分解規則(7), (9)より $f_1:f_2$, $\boxdot f$ もまた積和形に変形することができる．ゆえに (*1) が言える．

「グラフ生成の手続きは停止する」 (*2)

RTL 式 g が与えられたとする．

(基礎) g が原始命題またはその否定とする．このとき (*2) は明らか．

(帰納的段階) $f_1, f_2, f, \sim f_1, \sim f_2, \sim f$ に対し，(*2)が成立すると仮定する．

(1) $g=f_1\wedge f_2$ のとき，$f_1, f_2, f_1\wedge f_2$ に対するグラフをそれぞれ，$G_1=(V_1, E_1)$, $G_2=(V_2, E_2)$, $G=(V, E)$ とする．仮に，$\{f_1, f_2\}=\{f_1\wedge f_2\}$, $\{V_T, w_1, w_2, \cdots w_n\}=\{w_1, w_2, \cdots, w_n\}$ とみなせば，$^\forall v_1[R_1]\in V_1$, $^\forall v_2[R_2]\in V_2$ に対して，$v'[R_1\cup R_2]\in V'$ なる V' を構成すると $V\subseteq V'$．ゆえに，$g=f_1\wedge f_2$ に対してはグラフ生成は停止する．

(2) $g=f_1\vee f_2$ のとき，f_1, f_2 いずれに対するグラフ生成に帰着され，停止する．

(3) $g=f_1:f_2$ のとき，f_1 に対するグラフ G_1 を次の①，②により変形する．f_2 に対するグラフを G_2 とする．

① ε 節点 $v[R]\in G_1$ に対し $R=\{p_0\}\cup\bigcup_k\{s_k^e\}\cup\bigcup_l\{a_l\}\cup\bigcup_m\{\bigcirc x_m\}$ とする．S_i を G_2 の初期節点の子をラベル付けする式集合として，

$$v_i[\bigcup_k\{s_k^e\}\cup\bigcup_l\{a_l^e\}\cup\bigcup_m\{x_m^e\}\cup S_i],$$
$$v[\{p_0\}\cup\bigcup_k\{s_k^e\}\cup\bigcup_l\{a_l\}\cup\{\bigcirc((\bigwedge_m x_m):f_2)\}]$$

を $v[R]$ の親の節点の子として作り $v[R]$ を取り除く．

② $v[R]$ (v は ε 節点ではない) に対し，

$$R=\bigcup_k\{s_k^e\}\cup\bigcup_l\{a_l\}\cup\bigcup_m\{\bigcirc x_m\}$$

(ただし，s_k, $x_m\in RF$, a_l は原始命題またはその否定) としたとき，v を $R'=\bigcup_k\{s_k^e\}\cup\bigcup_l\{a_l\}\cup\{\bigcirc((\bigwedge_m x_m):f_2)\}$ でラベルしなおす．ただし x_m がないときは V_T を補う．

①と②で生成された節点のうち，ε 節点以外を V_E としてアルゴリズム 3.1 の③の手続きを適用すると，

f_2 に対するグラフに帰着し停止するが，このグラフはまた，$f_1 : f_2$ のグラフになっている．

(4) $g=\sim(f_1 : f_2)$ のとき，$f_1 : f_2$ に対するグラフ生成において出現するすべての式に対し，その否定を取り，$\sim\bigcirc f \equiv \bigcirc\sim f$ と $\sim(\eta^e)\equiv(\sim\eta)^e$ の関係式により得られる初等式に変形する．こうして得られた初等式の集合を F とする．$\sim(f_1 : f_2)$ に対する分解規則の構成の仕方を考えると，$\sim(f_1 : f_2)$ に対するグラフ生成の際に出現する式集合は F の部分集合である．F が有限であることより，グラフ生成の停止性が言える．

(5) $g=\boxdot f$ のとき，$\boxdot f$ に対する分解規則は，$\boxdot f = f \vee (f : \boxdot f)$ とみなして構成されている．したがって，f に対するグラフ生成が停止すること，(2)および(3)より，$f : \boxdot f$ に対するグラフ生成は有限個の節点を生成したのち，$\boxdot f$ の分解に帰着され停止する．

(6) $g=\sim\boxdot f$ のとき，(4)と同様の議論によりグラフ生成は停止する．

(7) $g=\sim(f_1 \vee f_2)$，$g=\sim(f_1 \wedge f_2)$ のとき，それぞれ(1)と(2)と同様にして，グラフ生成は停止する．

(8) $g=\bigcirc f$，$g=\sim\sim f$，$g=\sim\bigcirc f$ のとき，それぞれ f，f，$\sim f$ のグラフに帰着され停止する．

以上より任意の式に対して，(*2)が成立する．

また，節点消去に関してはたかだかグラフの節点の個数回，消去規則が適用されて停止する． □

(昭和63年4月 7 日受付)
(平成 元 年1月17日採録)

平石 裕実（正会員）

昭和26年生．昭和48年京都大学工学部電子工学科卒業．昭和50年同大学院修士課程（電気工学第二）修了．同年京都大学工学部情報工学教室助手．昭和59年同教室講師．昭和62年同教室助教授．工学博士．論理設計用 CAD，計算機グラフィクス等の研究に従事．電子情報通信学会会員．

濱口 清治（正会員）

昭和39年生．昭和62年京都大学工学部情報工学科卒業．同大学院修士課程存学中．論理設計検証，時相論理の研究に従事．電子情報通信学会会員．

矢島 脩三（正会員）

昭和8年生．昭和31年京都大学工学部電気工学科卒業．同大学院博士課程修了．工学博士．昭和36年より京都大学工学部に勤務，昭和46年情報工学科教授．昭和35年京都大学第一号計算機 KDC-1 を設計稼動，以来，計算機，論理設計，オートマトン等の研究教育に従事．著書は「電子計算機の機能と構造」（岩波，57年）等．本学会元常務理事，元会誌編集委員（地方），元 JIP 編集委員．電子情報通信学会元評議員およびオートマトンと言語研専元委員長，North-Holland 出版 IPL 編集委員，IEEE Senior Member.