Vol.2015-SE-187 No.22
2015/3/12

分解証明法を利用したリアクティブシステム仕様の強充足可能性判定器の提案

中村 風太[†1,a)]　吉浦 紀晃[†1,b)]

概要：時間論理により記述されたリアクティブシステム仕様の満たすべき性質の一つに強充足可能性がある．システム利用者からのいかなる要求列についても，それに対し仕様を満たすようなシステムのある応答列が存在するという性質である．仕様が強充足可能性を満たしているかを判定するためにモデルを利用した判定法が存在するが，二重指数の計算コストがかかり大きな仕様の判定に対してあまり有効ではない．そこで本論文では，モデルを利用しない判定法として，様相論理に対する分解証明法を利用した仕様の強充足可能性判定器の提案を行う．提案法の判定手続きを与え，提案法における健全性及び完全性を示す．

1. はじめに

システム設計段階での解析は，ソフトウェア開発全体を通じてのコスト低減という観点からは，プログラムの検証よりも重要であると言われている [1], [2]．一方モデル検査技法は，ソースコードをモデル化し，その性質を検証することに用いられているが，仕様の段階でシステムが予期しない状態に対して対処可能かどうかを検証することはできない．

日常生活には多種多様なリアクティブシステムが浸透している [3], [4]．リアクティブシステムとは，システム利用者の様々な要求に対して適切なタイミングで応答をするシステムである．例として，エレベータや飛行機の制御システム等が挙げられる．

現在リアクティブシステムの開発現場では，システム仕様を自然言語等で記述している．自然言語で仕様を記述した場合，仕様に曖昧さが生じる，人の手で記述されることで仕様自体に矛盾が生じる，といった問題が起こり，そのような仕様からシステムを構成すると整合性のないシステムが作られてしまう可能性がある．飛行機の制御システム等の高い安全性が求められるシステムの場合，仕様に問題があると人命を含め多大な被害が生じてしまう．

そこで時間や空間，状態の概念を自然に表現可能な時間論理や様相論理を用いて仕様を形式的に記述することにより，仕様から曖昧さを取り除く，仕様の自動検証を行うことが可能である．

リアクティブシステム仕様の満たすべき性質の一つに強充足可能性がある [5]．システム利用者からのいかなる要求列に対しても，仕様を満たすようなシステムの応答列が存在するという性質である．仕様がこの性質を満たしているかを判定するために，オートマトンなどのモデルを利用した強充足可能性判定法が提案されている [5], [6]．

この判定法は式からオートマトンを構成し決定化を行う．決定化されたオートマトンにおいて，システム利用者からの全ての無限要求列に従ったオートマトンの辿り方について充足可能であるかを走査する．オートマトンの構成法や決定化法，走査の手続きがすでに示されておりこの判定法の実装例もあるが，式の長さに対して二重指数という計算コストが伴うため大きな仕様に対してはあまり有効ではない [7]．

一方，モデルを利用しない証明系による仕様性質の判定法が提案されている [8]．推論規則を用いて恒真性等の式の性質を導出し，その結果から判定を行う．証明系を用いた判定法は，式の形によっては，導出される全ての式を導出せずに判定できることがあり，モデルを利用した判定法よりも高速に判定できることがある．しかし，現在提案されている証明系を用いた判定法は証明戦略が与えられていないため，実装に向かないという欠点がある．

そこで本論文ではモデルを利用しない強充足可能性の判定法として，様相論理に対する分解証明法を利用した仕様の強充足可能性判定器の提案を行う．強充足可能性判定にモデルを利用しないため，式の形によっては導出される式すべてを走査することなく判定できることがあり，その場合はモデルを利用する判定法よりも高速に判定できる．ま

†1 現在，埼玉大学大学院理工学研究科
Presently with Saitama University
a) s13mm322@mail.saitama-u.ac.jp
b) yoshiura@fmx.ics.saitama-u.ac.jp

た分解証明法を利用し，様相論理演算子列の統一化手続き，強充足可能性判定手続きを与えることにより，既存の証明系を用いた強充足可能性判定法よりも実装が容易となる．

本論文の構成は，次の通りである．まず第 2 章で，リアクティブシステム，リアクティブシステム仕様の記述言語となる時間論理と様相論理，リアクティブシステム仕様において重要な性質である実現可能性と強充足可能性について述べる．次に第 3 章で，ラベル付き様相演算子，推論規則及び提案する強充足可能性判定器を示す．第 4 章で提案した強充足可能性判定器の健全性，完全性を示す．第 5 章で提案した強充足可能性判定器について考察し，最後に第 6 章で本論文をまとめる．

2. リアクティブシステムと動作仕様

この章では，文献 [5] に基づき，リアクティブシステムの形式的定義を行う．また，仕様記述に用いられる時相論理及び様相論理などを定義し，リアクティブシステム動作仕様の記述方法を述べるとともに，仕様の性質における定義を行う．

2.1 リアクティブシステム

A を有限集合とするとき，A 上のすべての有限列の集合 (空列を含まない) を A^+ で表し，すべての無限列の集合を A^ω で表す．また，それらの和集合 $A^+ \cup A^\omega$ を $A^\dagger$ で表す．$A^\dagger$ の要素を，$\hat{a}, \hat{b}$ で表す．$A^\dagger$ の要素は，実際には，A^+ か A^ω のどちらかの要素であるので，それらを区別する場合には，A^+ の要素ならば $\bar{a}$ で，A^ω の要素ならば $\tilde{a}$ で表す．2 つの列 $\hat{a}, \hat{a}'$ の連接を $\hat{a} \cdot \hat{a}'$ で表す．ただし，$\hat{a} \in A^\omega$ のときは $\hat{a} \cdot \hat{a}' = \hat{a}$ とする．$\hat{a}$ の長さを $|\hat{a}|$ と書き， $\hat{a} \in A^+$ ならば，次のように再帰的に定義する．

- $\hat{a}$ が A の要素 1 つからなるときには $|\hat{a}| = 1$
- $\hat{b}$ を A^+，c を A の要素としたとき，$\hat{a}$ が $\hat{b} \cdot c$ ならば，$|\hat{a}| = |\hat{b}| + 1$

$\hat{a} \in A^\omega$ ならば $|\hat{a}| = \omega$ とする．$\hat{a}$ の i 番目 $(0 \leq i < |\hat{a}|)$ の要素を $\hat{a}\,[i]$ で表す．

定義 2.1 (リアクティブシステム) リアクティブシステムとは，3 つ組 $RS = \langle X, Y, r \rangle$ である．ただし，

- X は外部イベントの集合である．外部イベントは，システムに関わる外部環境が生起させるイベントである．
- Y は応答イベントの集合である．応答イベントは，システムが生起するイベントである．
- $r : (2^X)^+ \to 2^Y$ は，リアクション関数で，外部イベントの生起に対してシステムがどのような応答イベントを生起させるかを決定する．

外部イベントはシステムに関わる外部環境が生起させるイベントであるが，例えば，リアクティブシステムの利用者などがこの外部環境の 1 つであると考えられる．外部イベントはシステムの外部環境が生起させるイベントであるため，システム側では外部イベントの操作を行うことはできない．一方，応答イベントは外部イベントに対するシステムの応答である．従ってシステム側は応答イベントを自由に操作することができる．

定義 2.2 (ふるまい) $(2^X)^\dagger$ の要素を外部イベント列と呼び，$(2^Y)^\dagger$ の要素を応答イベント列と呼ぶ．外部イベントと応答イベントをあわせてイベントと呼び，状態の列を状態と呼ぶ．状態の列，つまり，$(2^{X \cup Y})^\dagger$ の要素を，ふるまいと呼び，特に，$(2^{X \cup Y})^+$ の要素を有限のふるまいと呼ぶ．ふるまい σ の i 番目の状態を $\sigma\,[i]$ で表す．有限のふるまい σ とふるまい δ の連接を $\sigma \cdot \delta$ で表す．
外部イベント列 $\hat{a} = a_0 a_1 a_2 \cdots \in (2^X)^\dagger$ に対するリアクティブシステム $RS = \langle X, Y, r \rangle$ のふるまい $behave_{RS}(\hat{a}) \in (2^{X \cup Y})^\dagger$ を以下のように定義する．

$$behave_{RS}(\hat{a}) = (a_0 \cup b_0)(a_1 \cup b_1)(a_2 \cup b_2) \cdots$$

ただし，すべての b_i $(0 \leq i < |\hat{a}|)$ は，リアクション関数 r によって外部イベント列 $a_0 a_1 \cdots a_i$ から定められる．
すなわち，

$$b_i = r(a_0 a_1 \cdots a_i)$$

である．

2.2 動作仕様

動作仕様は，リアクティブシステムの可能なふるまいの集合を定める．リアクティブシステムは，環境からの要求イベント生起と，システムの応答イベント生起の順序やタイミングが重要である．また，リアクティブシステムの仕様を記述する際に曖昧さを取り除くことが重要となる．可能なふるまいを定める手段としては様々なものが考えられるが，本研究では，時間論理により仕様を記述するものとする．但し，用いる時間演算子は $\Box$ 及び $\Diamond$ とする．

時間論理による仕様記述は次のように行われる．まず，記述しようとしているシステムのイベント集合 (外部イベント集合と応答イベント集合) を決定する．次に各イベントに時間論理の命題変数を割り当てる．それぞれの命題変数は，対応するイベントが生起するときに真となるよう意図される．仕様記述者は，これらの命題変数からなる時間論理式を用いてシステムの可能なふるまいの集合を定める．

2.2.1 線形時間論理

定義 2.3 (線形時間論理式) 論理式を次のように定義する．

- 原子命題は式である．
- f が式ならば，$\neg f$ は式である．
- f_1, f_2 が式ならば，$f_1 \wedge f_2$ は式である．
- f_1, f_2 が式ならば，$f_1 U f_2$ は式である．

略記として，以下を用いる．

- $\top \equiv f \vee \neg f$

- $\bot \equiv f \wedge \neg f$
- $f_1 \vee f_2 \equiv \neg(\neg f_1 \wedge \neg f_2)$
- $f_1 \rightarrow f_2 \equiv \neg f_1 \vee f_2$
- $f_1 \leftrightarrow f_2 \equiv (f_1 \rightarrow f_2) \vee (f_2 \rightarrow f_1)$
- $f_1 \overline{U} f_2 \equiv \neg(\neg f_1 U \neg f_2)$
- $\Box f \equiv f U \bot$
- $\Diamond f \equiv f \overline{U} \top$

定義 2.4 (意味論) P をイベントの集合とし，$\mathcal{P}$ を P に対応する命題変数の集合とする．ふるまい $\sigma \in (2^P)^\omega$ の i 番目の状態が，$\mathcal{P}$ 上の式 f を満たすことを $\sigma, i \models f$ で表し，以下のように再帰的に定義する．

- $\sigma, i \models f$ **iff** $p' \in \sigma[i]$ $(p' \in P$ は $p \in \mathcal{P}$ に対応するイベント$)$
- $\sigma, i \models f_1 \wedge f_2$ **iff** $\sigma, i \models f_1$ かつ $\sigma, i \models f_2$
- $\sigma, i \models f_1 \vee f_2$ **iff** $\sigma, i \models f_1$ または f_2
- $\sigma, i \models \neg f$ **iff** $\sigma, i \not\models f$
- $\sigma, i \models f_1 \rightarrow f_2$ **iff** $\sigma, i \not\models f_1$ または $\sigma, i \models f_2$
- $\sigma, i \models \Box f$ **iff** $\forall k\ (0 \leq k$ ならば $\sigma, i+k \models f)$
- $\sigma, i \models \Diamond f$ **iff** $\exists k\ (0 \leq k$ かつ $\sigma, i+k \models f)$
- $\sigma, i \models f_1 U f_2$ **iff** $\exists j\ (0 \leq j$ かつ $\sigma, i+j \models f_2$ かつ $\forall k\ (0 \leq k < j$ ならば $\sigma, i+k \models f_1))$ または $\forall k\ (0 \leq k$ ならば $\sigma, i+k \models f_1)$
- $\sigma, i \models \circ f$ **iff** $\sigma, i+1 \models f$
- $\sigma, i \models \top$ (任意の i について)
- $\sigma, i \not\models \bot$ (任意の i について)

ここで，$\sigma, 0 \models f$ であるとき，σ は f を満たすといい，$\sigma, 0 \models f$ を単に $\sigma \models f$ と記述する．また，ある σ と i が存在し，$\sigma, i \models f$ となるとき，f を充足可能であるという．

2.2.2 様相論理

定義 2.5 (様相論理式) 古典論理で使用する論理記号の他に，様相論理では必然オペレータ $\Box$ と可能オペレータ $\Diamond$ を使用する．

- P が命題変数であるとき，P は様相論理式である．
- F_1，F_2 が様相論理式であるとき，$\Box F_1$，$\Diamond F_1$，$F_1 \wedge F_2$，$F_1 \vee F_2$，$\neg F_1$ は様相論理式である．

略記として，以下を用いる．

- $\top \equiv F_1 \vee \neg F_1$
- $\bot \equiv F_1 \wedge \neg F_1$
- $F_1 \rightarrow F_2 \equiv \neg F_1 \vee F_2$
- $F_1 \leftrightarrow F_2 \equiv (F_1 \rightarrow F_2) \vee (F_2 \rightarrow F_1)$

定義 2.6 (意味論) モデル M を $\langle W, R, V \rangle$ で表す．W は可能世界の集合であり，R は可能世界間の到達可能関係を与える二項関係である．V は付値関数で各命題変数 p に W の部分集合 $V(p)$ を割り当てる．

モデル M が与えられたとき，世界 $w \in W$ において式 F_1 が真であることを $M, w \models F_1$ と表す．$M, w \models F_1$ は以下のように再帰的に定義される．

- $M, w \models F_1$ **iff** $M, w \not\models F_1$
- $M, w \models F_1 \vee F_2$ **iff** $M, w \models F_1$ または $M, w \models F_2$
- $M, w \models F_1 \wedge F_2$ **iff** $M, w \models F_1$ かつ $M, w \models F_2$
- $M, w_i \models \Box F_1$ **iff** $\forall w_j \in W(w_i R w_j$ ならば $M, w_j \models F_1)$
- $M, w_i \models \Diamond F_1$ **iff** $\exists w_j \in W(w_i R w_j$ かつ $M, w \models F_1)$

ここで，$\forall w \in W\ M, w \models F_1$ のとき，F_1 はモデル M について真であるという．また，$\forall M \forall w \in W\ M, w \models F_1$ のとき F_1 は恒真であるといい，$\models F_1$ と表す．

$\exists w \in W\ M, w \models F_1$ のとき F_1 は充足可能であるという．また $\forall M \forall w \in W\ M, w \not\models F_1$ のとき F_1 は充足不能であるという．

2.2.3 様相論理体系 DS4

様相論理は到達可能関係 R に対する条件により分類される．本論文では，様相論理体系は推移律と継続律を持つ様相論理体系 DS4 を用いる．

定義 2.7 (様相論理体系 DS4) 様相論理の構文規則は通常の定義に従う [9]．様相論理 DS4 の公理系は，様相論理 K の体系に公理として 4，D を加えたものである．これらの公理を以下に示す．

公理 K: $\Box(F_1 \rightarrow F_2) \rightarrow (\Box F_1 \rightarrow \Box F_2)$

公理 4: $\Box F_1 \rightarrow \Box\Box F_1$

公理 D: $\Box F_1 \rightarrow \Diamond F_1$

また，体系 DS4 は推移律と継続律を有する．

推移律: $\forall w \forall w' \forall w''(wRw' \wedge w'Rw'' \rightarrow wRw'')$

継続律: $\forall w \exists w'(wRw')$

2.2.4 動作仕様

リアクティブシステムの動作仕様は，3 つ組 $Spec = \langle \mathcal{X}, \mathcal{Y}, \varphi \rangle$ である．ただし，

- $\mathcal{X}$ は外部イベントに対応する命題変数の有限集合である．その命題の成立は対応する外部イベントの生起を表す．$\mathcal{X}$ の要素を，外部イベント命題と呼ぶ．
- $\mathcal{Y}$ は応答イベントに対応する命題変数の有限集合である．その命題の成立は対応する応答イベントの生起を表す．$\mathcal{Y}$ の要素を，応答イベント命題と呼ぶ．
- φ は，$\mathcal{X} \cup \mathcal{Y}$ に含まれる命題変数からなる時間論理式である．但し，用いる時間演算子は $\Box$ 及び $\Diamond$ とする．

簡単のために，動作仕様 $Spec = \langle \mathcal{X}, \mathcal{Y}, \varphi \rangle$ を単に仕様 φ と言うことがある．

2.3 仕様の性質

リアクティブシステムの動作仕様にはいくつかの性質が存在する．この性質は文献 [5] において導入されたものである．ここでは仕様の各性質について定義する．

なお，以下の定義において，$\tilde{a}$ はすべての無限外部イベント列，$\bar{a}$ はすべての有限外部イベント列，$\tilde{b}$ はすべての無限応答イベント列，$\langle \tilde{a}, \tilde{b} \rangle$ は任意の $i \geq 0$ について，i 番目の状態が $(a_i \cup b_i)$ であるようなふるまいである．

2.3.1 実現可能性

リアクティブシステムの仕様が満たすべき重要な性質として実現可能性がある．実現可能性とは，「どのような外部イベントがどのような順序で生起しても，仕様を満たすように応答できるようなリアクティブシステムが存在する」という性質であり，以下のように形式化される．

定義 2.8 (実現可能性) 以下を満たすとき，仕様 φ は実現可能であるという．

$$\exists RS \forall \tilde{a}(behave_{RS}(\tilde{a}) \models \varphi)$$

2.3.2 強充足可能性

強充足可能性とは，実現可能性の必要条件であり，「環境が将来生起する外部イベント列がわかるならば，仕様を満たすように生起させる応答イベント列を決定できる」という性質である．以下のように形式化される．

定義 2.9 (強充足可能性) 以下を満たすとき，仕様 φ は強充足可能であるという．

$$\forall \tilde{a} \exists \tilde{b}(\langle \tilde{a}, \tilde{b} \rangle \models \varphi)$$

3. 強充足可能性判定器

本章では，様相論理に対する分解証明法 [10] を利用した強充足可能性判定器の定義を行う．

3.1 様相論理世界モデル

本論文では，仕様の強充足可能性の判定に様相論理に対する分解証明法を利用する．仕様は時間論理で記述されるため，様相論理の分解証明法を利用しても仕様の強充足可能性を判定できるかは明らかではない．この節では，様相論理の分解証明法を用いることで時間論理で記述された仕様の強充足可能性が判定できることを証明する．時間演算子 $\Box$ と $\Diamond$ を様相論理で解釈する場合には，様相論理での通常の様相演算子として解釈する．ここでは，様相論理のモデルとして，次に定義する DS4 モデルを利用する．

定義 3.1 (DS4 モデル) 可能世界間の到達可能関係が推移的かつ継続的な様相論理世界モデルを DS4 モデルという．

A を時間演算子として $\Box$ と $\Diamond$ のみを利用している式とする．$\Box A$ の様相論理式として DS4 モデルでの充足可能性と時間論理式としてのふるまいにおける充足可能性との等価性を証明する．

定理 3.1 式 $\Box A$ が DS4 モデルが充足するならば，充足するふるまいが存在する．

証明： $\Box A$ がある DS4 モデルで充足可能であるとする．このモデルは継続的であるので，$\Box A$ と A が成り立つ可能世界 w_1 が存在する．その可能世界で $\Diamond B$ の形の式が真になるのであれば，B が真になる可能世界 w_2 が存在し，w_1 から到達可能である．このように，充足する DS4 モデルの可能世界をたどることで，ふるまいを生成することができ，このふるまいでは，$\Box A$ が充足する． ∎

定理 3.2 式 $\Box A$ がふるまいで充足するならば，充足する DS4 モデルが存在する．

証明： 式 $\Box A$ があるふるまいで充足する時，このふるまいの各時点を一つの可能世界とみなした無限列のモデルが存在する．この無限列のモデルは DS4 モデルであり，$\Box A$ を充足する． ∎

これら 2 つの定理から次のことがいえる．

定理 3.3 $\Box A$ が DS4 モデルが強充足不能ならばふるまいにおいても強充足不能であり，また，その逆も成り立つ．

3.2 ラベル付き様相演算子，節形式

仕様 φ に出現する全ての $\neg$ 演算子を命題変数の直前に移動する．意味的に等価なその式の形に次の規則を用いて任意の式を変形できるため，このような操作を行なっても一般性を失わない．

- $f \to g \Rightarrow \neg f \vee g$
- $\neg(f \wedge g) \Rightarrow \neg f \vee \neg g$
- $\neg(f \vee g) \Rightarrow \neg f \wedge \neg g$
- $\neg\neg f \Rightarrow f$
- $\neg\Box f \Rightarrow \Diamond\neg f$
- $\neg\Diamond f \Rightarrow \Box\neg f$

φ 中の様相演算子 $\Box$ の出現にそれぞれ異なった変数ラベルを，様相演算子 $\Diamond$ の出現にそれぞれ異なった定数ラベルをつける．これを φ にラベルをつけるといい，ラベルのついた φ を φ^* で表す．変数ラベル p をつけた様相演算子 $\Box$ を $\Box_p$ で表し，定数ラベル a をつけた様相演算子 $\Diamond$ を $\Diamond_a$ で表す．φ^* に対して，以下の規則に従い様相演算子の分配を行い，その後和積標準形に変形した式 φ の節形式といい，φ^c で表す．

- $\Box_p(f \wedge g) \Rightarrow \Box_p f \wedge \Box_p g$
- $\Box_p(f \vee g) \Rightarrow \Box_p f \vee \Box_p g$
- $\Diamond_a(f \wedge g) \Rightarrow \Diamond_a f \wedge \Diamond_a g$
- $\Diamond_a(f \vee g) \Rightarrow \Diamond_a f \vee \Diamond_a g$

節形式は $\Gamma_1 \wedge \ldots \Gamma_n$ の形の式となる．ここで，Γ_i は $\alpha_{i1}L_{i1} \vee \ldots \vee \alpha_{im}L_{im}$ の式の形であり，L_{ij} はリテラル，α_{ij} はラベル付きの様相演算子の列 (空列を含む) である．Γ_i を節と呼ぶ．

3.3 推論規則

定義 3.2 (結合) L と $\bar{L}$ を相補的なリテラルとし，L と $\bar{L}$ がそれぞれ別の節に含まれるとする．この時，L と $\bar{L}$ の対を結合といい，L 及び $\bar{L}$ をそれぞれ結合要素という．

定義 3.3 (位置) ラベル付き様相演算子列に対して，リテラルに近いラベル付き様相演算子から 1 から昇順に自然数の番号を振ったとき，この番号を位置といい，ラベル付き様相演算子に番号を振ることを位置をつけるという．

定義 3.4 (代入) 変数ラベルがついた様相演算子 □ を長さ 1 以上のラベル付き様相演算子の列 $\{\Diamond, \Box\}^+$ への置換を代入という．

定義 3.5 (統一化) α，β をラベル付き様相演算子の列とする．α と β を代入により一致させることを統一化といい，α と β を統一化する代入が存在するとき，α と β は統一化可能であるという．

定義 3.6 (統一化手続き) α 及び β をそれぞれラベル付き様相演算子列とする．以下の手続きに従い，ラベル付き様相演算子列の統一化を行う．以下の手続き上で，演算子，演算子列，□ 及び ◇ は，それぞれラベル付き様相演算子，ラベル付き様相演算子列，ラベル付き □ 及びラベル付き ◇ を表す．

(1) α 及び β に位置をつける．α と β の中で最も大きな位置を位置 M とする．
(2) α，β において，位置 1 から順に演算子を見たとき，二つの列中で最初に □ が出現する位置を ι とする．ι がなければ手続きを終了する．
(3) 位置 1 から ι の間全ての位置で α，β の演算子が一致していれば手続き 4 へ，そうでなければ手続きを終了する．
(4) ι から昇順に見て，二つの列中で最初に □ が最初に出現する位置を i とする．但し，位置を i とする □ は，その □ がある位置にはもう一方の列に他の演算子が存在するような □ である．位置 i とする □ が存在しなければ，i を $M+1$ とする．
(5) 位置 ι にある □ に，この □ がある演算子列とは別の演算子列の ι から $i-1$ の間にある演算子列を代入する．
(6) 位置 ι から $i-1$ の間全ての位置で演算子が一致していれば手続き 8 を，一致していなければ代入前の演算子列に戻し，手続き 7 を行う．
(7) 手続き 5 における □ と別の演算子列の同じ位置にある演算子が □ である場合，手続き 5 に戻る．但し，手続き 5 の □ を同じ位置の別演算子列の □ に替える．手続き 5 における □ と別の演算子列の同じ位置にある演算子が □ でない，もしくは，手続き 5 を二回行った場合は，手続きを終了する．
(8) ι を i とする．$\iota = M+1$ であれば手続きを終了する．そうでなければ，位置を付け直し，その結果最も大きな位置が変わったら位置 M を付け直した後に手続き 4 へ戻る．

手続き 8 で手続きを終了した場合，α 及び β は統一化可能とする．統一化する代入は手続き 5 の結果である．手続き 8 以外で手続きを終了した場合は統一化不可能とする．

定理 3.4 ラベル付き様相演算子の列が統一化可能ならば，統一化手続きにより統一化できる．

証明： 対偶，すなわち統一化手続きにより統一化できなければ，ラベル付き様相演算子の列が統一化不可能であることを示す．以下の証明では，演算子，演算子列，□ 及び ◇ は，それぞれラベル付き様相演算子，ラベル付き様相演算子列，ラベル付き □ 及びラベル付き ◇ を表す．

演算子列 α 及び β が統一化不可能であるとき，次のいずれかを満たす時である．なお以下で演算子列 α の位置 p の演算子を α_p と表す．演算子列の最大の位置を M とする．

(1) α_1，β_1 が共に ◇
(2) α_M，β_M が共に ◇
(3) ◇ のみの演算子列があり，この演算子列はもう一方の演算子列より短い
(4) 一方の演算子列に ◇ が含まれており，もう一方の演算子列に □ がない

(1) 及び (2) の場合，いくら代入を行なっても位置 1 もしくは M における ◇ は一致しないために統一化不可能である．(3) の場合，代入は ◇ に対しては行えず，また演算子列は減少させることはできない．故に，2 つの演算子列の最も大きな位置が揃うことがなく統一化不可能である．(4) の場合，一方の演算子列に出現する ◇ はもう一方の列には出現しないため，もう一方の演算子列の □ に代入を行うことで統一化するが，□ が存在しなければ代入を行えないため統一化不可能である．

統一化手続きにより統一化できない時，統一化手続きは手続き 8 以外で終了する．統一化手続き 2 及び 3 で終了する場合演算子列は (1) を，手続き 7 で終了する場合演算子列は (2)，(3)，(4) のいずれかを満たすため，統一化不可能である．よって，統一化手続きにより統一化できなければ，演算子列は統一化不可能である． ■

定義 3.7 (推論規則)

$$\frac{\alpha L \vee \Gamma \quad \beta \bar{L} \vee \Gamma'}{(\Gamma \vee \Gamma')^{\sigma(\alpha,\beta)}}$$

α，β はラベル付き様相演算子の列である．$\sigma(\alpha, \beta)$ は，α と β を統一化する代入である．$(\Gamma \vee \Gamma')^{\sigma(\alpha,\beta)}$ は $(\Gamma \vee \Gamma')$ に含まれるラベル付き様相演算子の列に代入 $\sigma(\alpha, \beta)$ を適用した結果得られる式である．異なる節に同じ変数ラベルが出現しても，それらを異なる変数ラベルとして扱う．

3.4 強充足可能性判定器

以下の操作を仕様の強充足可能性が判定されるまで行う．

(1) 空である式集合 U に φ^c の節を全て入れる．以降，式集合 U の要素を式と呼ぶ．
(2) U に恒真式が含まれる場合，その式を U から取り除く．
(3) φ が強充足不能と判定される，もしくは推論規則を適用できるような式の組み合わせがなくなるまで以下を繰り返し行う．
  (a) U に空式もしくは外部イベント命題のみの式が存在する場合，φ を強充足不能と判定する．
  (b) U に存在する結合を持つ任意の二つの式に対して推論規則を適用する．外部イベント命題の結合と

応答イベント命題の結合の両方を二つの式が持つ場合は，先に応答イベント命題の結合から推論規則を適用する．

(c) 推論規則から導出された式が U に存在しない式であり，かつその式が恒真でなければその式を U に追加する．

(4) φ を強充足可能と判定する．

定理 3.5 U に恒真式を追加しなくても，導出される結果は変わらない．

証明： 恒真式を T，任意の式を T'，T 及び T' から推論規則を用いて導出される式を τ とする．この時 τ は $\tau = T'$ となるか $T' \to \tau$ の関係を持つ．従って，推論規則を適用する式に τ を用いる場合は T' を用いても結果は変わらない．故に U に T を追加する必要はない． ■

4. 健全性と完全性

本章では，提案した判定法の健全性と完全性の証明を行う．最初に式集合 U が有限集合であることを示し，本判定法が停止することを確かめる．次にリアクティブシステムの外部イベント命題及び応答イベント命題における真偽値の決定の仕方と U 中の式の形に着目して本判定法の健全性及び完全性を証明する．

4.1 式集合 U の有限性

補題 4.1 及び 4.2 を示し，それらを用いて定理 4.1 を示す．定理 4.1 より，式集合 U は有限であるため本判定法は停止する．

補題 4.1 式集合 U の要素数は有限である．

証明： 推論規則は結合に対して適用する．従って，U に結合が存在する限り適用することができるが，結合の数は，φ^c に存在するリテラル数未満である．ここで，φ^c に存在するリテラルは有限であるので，U に対する推論規則の適用回数は有限回である．

推論規則に用いる統一化においては，統一化するラベル付き様相演算子の列は有限であり，また様相演算子につくラベルの種類も有限である．手続きでラベルの付け直しは行わないため，統一化のパターンは有限種類である．

U に対する推論規則適用回数が有限回であり，統一化のパターンが有限であるため，式集合 U の要素数は有限である． ■

補題 4.2 一度の推論規則適用時に導出される式の長さは有限である．

証明： 推論規則では，結合要素以外の節を論理和で結ぶ．従って推論規則から導出される式は，推論規則を適用する式に含まれるリテラルの数の合計以下となる．節に含まれるリテラルの数は有限であるため，推論規則から導出される式のリテラルの数は有限である．

統一化は，代入を行うことによりなされる．統一化手続きより，一つのラベル付き □ に代入するラベル付き様相演算子列の長さは，代入されるラベル付き □ を含むラベル付き様相演算子列とは別のラベル付き様相演算子列の長さ以下となる．ラベル付き様相演算子列の長さは有限長であるため，統一化後のラベル付き様相演算子列の長さは有限長である．

推論規則から導出される式のリテラル数は有限であり，統一化後のラベル付き様相演算子列の長さが有限長であるため，一度の推論規則適用時に導出される式の長さは有限である． ■

定理 4.1 式集合 U は有限集合である．

証明： 補題 4.1 より U の要素数が有限である．φ^c の各節は有限長の式であり，かつ補題 4.2 より推論規則から導出される式の長さは有限長である．

U の要素数は有限であり，かつ全ての要素は有限長であるため，U は有限集合である． ■

4.2 健全性

定理 4.2 式集合 U 中の式に外部イベント命題のみの式が存在するならば，仕様 φ は強充足不能である．

証明： U 中に外部イベント命題のみの式が存在する時，その式は外部イベント命題の真偽値によって偽となることがある．この式が偽とならない時は式が恒真式である場合であるが，そのような式は U 中に存在しない．外部イベント命題のみの式が偽となるような外部イベント命題の真偽値において，U に存在する応答イベント命題の真偽値をどのように割り当てても外部イベント命題のみの式が真となることはなく，φ^c も充足しない．この時 φ も充足しないため，ある外部イベント命題の真偽値割り当てに対して全ての応答イベント命題の真偽値割り当てで φ を充足しない．よって U 中の式に外部イベント命題のみの式が存在するならば，φ は強充足不能である． ■

4.3 完全性

補題 4.3 推論規則を適用できるような式の組み合わせがない U 中の式全てに応答イベント命題が含まれるならば，外部イベント命題の真偽値によらず U 中の式全てを真にするような各応答イベント命題の真偽値割り当てが存在する．

証明： 式に存在する応答イベント命題が結合要素でない場合，この式を真にすることができる応答イベント命題の真偽値割り当てが存在するのは自明である．式に存在する応答イベント命題が結合要素である場合について以下で述べる．

式 f に存在する応答イベント命題 y が結合要素である場合，y の結合要素である $\neg y$ を含む式 f' には y の結合要素でない，すなわち y の真偽値に関係なく真偽値を割り当てられる応答イベント命題 y' が含まれる．含まれない場合，

f と f' から推論規則により外部イベント命題のみの式が導出され仮定に反する．また，y' が結合要素である場合，結合要素 $\neg y'$ を含む式 f'' には y' の結合要素でない応答イベント命題 y'' が含まれる．

y を真とすることで，f を外部イベント命題の真偽値によらず真とすることができる．この時，f' に含まれる $\neg y$ は偽となるが同時に含まれる y' を真とすることで，f' は真とすることができ，同様に y'' を真とすれば f'' を真とするできる．

f'' を外部イベント命題の真偽値によらず真とできない場合，f'' 中に y'' が含まれないか，y'' が真と出来ない場合が考えられる．y'' が含まれない場合は f' と f'' から推論規則により，外部イベント命題のみの式が導出されるため仮定に反する．y'' が真と出来ない場合，y'' の結合要素 $\neg y''$ を含む式 F において，$\neg y''$ を真にしなければ式を真にすることができないような F が存在する．この時 f，f'，f'' から推論規則により導出される式と F に推論規則を適用すると外部イベント命題のみの式が導出されるために仮定に反する．

よって，推論規則を適用できるような式の組み合わせがない U 中の式全てに応答イベント命題が含まれるならば，外部イベント命題の真偽値によらず U 中の式全てを真にするような各応答イベント命題の真偽値割り当てが存在する． ■

定理 4.3 推論規則を適用できるような式の組み合わせがない式集合 U 中の式に空式もしくは外部イベント命題のみの式が存在しないならば，仕様 φ は強充足可能である．

証明： 推論規則を適用できるような式の組み合わせがない U 中の式に空節もしくは外部イベント命題のみの式が存在しない時，U 中の式全てに応答イベント命題が含まれる．この時補題 4.3 より，U 中の全ての式は外部イベント命題の真偽値によらず真とできるような各応答イベント命題の真偽値割り当てが存在する．

ここで，U 中に存在する任意の外部イベント命題について考える．任意の外部イベント命題を含む式には必ず応答イベント命題が存在し，その応答イベント命題は，任意の外部イベント命題の真偽に関わらず式を真にすることができる．このことが U に含まれる外部イベント命題全てにおいて述べることができ，かつ U には仕様 φ の外部イベント命題が全て含まれている．よって，全ての外部イベント命題に対して φ を真とできるような応答イベント命題が存在するため，式集合 U 中の式に外部イベント命題のみの式が存在しないならば，仕様 φ は強充足可能である． ■

5. 考察

本研究では，様相論理に対する分解証明法を用いることにより，証明系による仕様の強充足可能性判定器の提案を行い，本判定器の停止性，健全性及び完全性を示した．

定理 4.1 より，本判定器は停止し，定理 4.2 及び 4.3 より，本判定器は仕様の強充足可能性判定に対して健全かつ完全である．判定手続きでは，強充足不能と判定できた場合その時点で判定を終了するため，既存のモデルを利用した強充足可能性判定器よりも高速に判定できる場合がある．また本研究では，判定器の判定手続き及びラベル付き様相演算子列の統一化手続きも示しており，本判定器は既存の証明系を用いた強充足可能性判定器よりも実装に向く．

しかし本判定器で判定に用いる仕様は時間演算子 □ 及び ◇ のみを利用して記述されるために，時間演算子全てを利用して記述される仕様よりも記述力が劣る場合がある．

例として空調制御システムの仕様を示す．仕様に $\Box(too\_hot \rightarrow coolUcomfortable)$ という記述があるとする．この記述は「室内が暑いとき，快適な室温になるまで冷やす」を表す．但し too_hot は室内が暑い，$cool$ は冷やす，$comfortable$ は室内が快適な温度であることを表す．時間演算子 U を用いているために「快適な室温になるまで冷やす」という表現の記述ができるが，時間演算子 □ 及び ◇ のみではこの記述を行うことは難しく，記述できたとしても仕様が複雑かつ長くなってしまう．

従って本研究の判定器が時間演算子全てを用いて記述された仕様に対しては，既存の手続きよりも改善されているとは一概には言い切れない．しかし，仕様が時間演算子 □ 及び ◇ を用いて記述されている，もしくは様相論理を用いて記述されている場合では，本判定器は既存の判定器より高速に強充足可能性が判定される可能性があることや実装に向いているといった点で有用である．

6. おわりに

6.1 まとめ

本研究では，モデルを利用しない強充足可能性判定法として，様相論理に対する分解証明法を利用した強充足可能性判定器の提案を行い，提案した判定器の強充足可能性判定に対する停止性，健全性，完全性を示した．仕様が時間演算子 □ 及び ◇ を用いて記述されたもの，もしくは，様相論理で記述されたものについては，既存の判定法よりも高速に判定できる可能性がある，実装に向くといった点が有用であると考えられる．しかし，□ 及び ◇ 以外の時間演算子も用いて記述された仕様については，本判定器では，強充足可能性の判定は行えない．

6.2 今後の課題

本論文にて用いた仕様記述方法では仕様の記述力が劣る場合が考えられる．従って，本判定器を □ 及び ◇ 以外の時間演算子も用いて記述した仕様に対しても判定を行えるよう拡張することが課題として挙げられる．それには，分解証明法を時間論理にも適用できるよう拡張する必要がある．

参考文献

[1] Daniel Jackson: “Automating first-order relational logic”, Proceedings of the 8th ACM SIGSOFT international symposium on Foundations of softwareenigineering, pp.130-139, 2000.
[2] 島川 昌也，萩原 茂樹, 米崎 直樹: “ 仕様の自動検証に適した LTL フラグメント”, 日本ソフトウェア科学学会第 23 回大会, 2006.
[3] David Harel, Amir Pnueli: “On the development of reactive systems”, K.R.Apt,editor,Logics and Models of Concurrent Systems, Volume F13 of NATO ASI Series, pp.477-498.springer-Verlag, 1985.
[4] Zohar Manna, Amir Pnueli: “the temporal logic of reactive and concurrent systems Specification”, Springer-Verlag,New York, 1992.
[5] 森 亮靖，米崎 直樹: “時相論理によるリアクティブシステム仕様の実現可能性に関する分類”，コンピュータソフトウェア，Vol. 15, No.3 pp. 17-24，1998.
[6] 荻原 茂樹，米崎 直樹: “Muller オートマトンを用いたリアクティブシステムの仕様検証法とその完全性”，第二回システム検証の科学技術シンポジウム，pp. 52-63，2005.
[7] 島川昌也, 萩原茂樹, 米崎直樹: “リアクティブシステム仕様の強充足可能性判定問題の計算量について”, 第 4 回システム検証の科学技術シンポジウム予稿集, pp.21-28, 2007.
[8] 木村 友洋: “時間論理によるリアクティブシステム仕様の実現不能性判定のための証明系の構築”，埼玉大学工学部情報システム工学科卒業論文，2008.
[9] Hughes G.E, Cresswell M.J: “A New Introduction to Modal Logic”, Routledge, 1996.
[10] 荻原 茂樹，大石 正彦，米崎 直樹:“有限フレームを意味的基礎として持つ様相論理に対する分解証明法”，ソフトウェア発展，岩波書店，pp.78-91，2000.