仕様の自動検証に適したLTLフラグメント

—実現集合を現す決定性オートマトン構成の立場から—

LTL fragments suitable for automatic verification of specification

—From the viewpoint of constructing deterministic automata representing a set of implementations—

島川 昌也 † 萩原 茂樹 † 米崎 直樹 †
Masaya SHIMAKAWA Shigeki HAGIHARA Naoki YONEZAKI
† 東京工業大学 大学院情報理工学研究科 計算工学専攻
Dept. of Computer Science,Graduate School of Information Science and Engineering,
Tokyo Institute of Technology
(masaya,hagihara,yonezaki)@fmx.cs.titech.ac.jp

線形時間論理（LTL）による仕様の各種検証手続き，例えばリアクティブシステム仕様の実現可能性判定などにおいて，その仕様が定める可能な振る舞いをちょうど受理する決定性オートマトンが必要となる．このオートマトンは，仕様を満たすすべての実現を表しており，その構成には Safra's construction 等を用いる極めて複雑で計算コストの高い処理を伴う．本研究では，これらの問題を避けることの可能な 2 種類の LTL のフラグメントを与え，それを用いた仕様記述からの Safra's construction を用いない，よりシンプルな決定性オートマトンの構成法を提案する．

1 はじめに

システム設計段階での解析は，ソフトウェアプロセス全体を通じてのコスト低減という観点からは，プログラムコードの検証より重要であるといわれている [6]．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況に対して対処可能かどうかを検証することはできない．リアクティブシステムとは，環境とのインタラクションの維持を目的とするシステムである．このようなシステムは，環境からの要求に対して適切なタイミングで応答を返すことが強く求めれるため，時間的制約の記述に優れた時間論理は，それらの仕様記述に適している．時間論理という形式言語を用いて仕様記述を行うことで，厳密な記述が可能となるだけでなく，同じく時間論理で別途記述された性質を満たすかどうかを調べることが可能となる．さらに，「任意の要求に対して正しく応答することが可能なプログラムが存在する」（実現可能性 [1]）などのメタレベルの性質により，プログラム作成前に設計に潜んでいた見えにくい欠陥を検出することが可能となる．これにより，より正確かつ堅牢で，その挙動が全て確認された設計を得ることができる．実現可能であることが分かれば，プログラム合成 [9] 可能となる．

しかしがなら，仕様の実現可能性判定やプログラム合成は，一般に，極めて複雑で計算コストの高い処理を伴う．これは，それらが仕様を満たす振る舞いをちょうど受理する決定性 ω オートマトンを基に行われることによる[1]．この決定性 ω オートマトンの構成は，次の二つの意味で困難であるといえる．

- 構成のための処理が極めて複雑である
- その構成が仕様の大きさに対して 2 重指数となる

一般に，決定性 ω オートマトンは，まず，仕様より無限長の語の上の非決定性 Büchi オートマトンを構成し（[5, 4] など），そして，Safra's construction[10] を用い，それを決定化することで構成する．Safra's construction は，もとの ω オートマトンの状態の集合をノードとする木を一つの状態とする複雑な構成である．そのため，その構成のための処理は煩雑となり，各種効率化を施しにくい．また，仕様の非決定性 ω オートマトンは仕様の大きさに対して指数の構成であり，それを決定化するため 2 重指数の構成となる．したがって，大きな計算コストを必要とする．

このような時間論理による検証の難しさは，その部分体系においては避け得る可能性がある．効率よ

[1] このオートマトンは仕様の実現となるすべてのプログラムを表現しており，これを解析することで，実現可能性判定やプログラム合成が可能となる．

く検証が可能で十分な記述力を持った部分体系を見つけ出すことは，実用上の観点から大きな意味がある．しかしながら，効率よく実現可能性判定やプログラム合成を行える部分体系について，ほとんど研究されていない．

本研究では，実現可能性判定やプログラム合成に必要となる決定性 ω オートマトンの構成の観点から LTL の構文を制限した LTL のフラグメントを与え，その式からの決定性 ω オートマトンの構成法を提案する．本稿では，次の二種類のフラグメントを与える．まず，一種類目のフラグメント（LTL^{ep} と LTL^{gp}）は，Safra's construction を用いずに，もとのオートマトンの状態の集合を一つの状態とする単純な構造（powerset construction） により決定性 ω オートマトンを構成することが可能となるフラグメントである．二種類目のフラグメント（LTL^{ep+} と LTL^{gp+}）は，上のフラグメント（LTL^{ep} と LTL^{gp}）のサブフラグメントとなっており，powerset construction により決定性 ω オートマトンを構成できるだけでなく，さらに，その状態数を指数オーダーに抑えられるフラグメントである．

本稿の構成は以下の通りである．次章では，本研究で扱う時間論理やオートマトンについて述べる．3 章では，LTL フラグメント（LTL^{ep} , LTL^{gp}, LTL^{ep+} , 及び LTL^{gp+}）を与える．4 章では，LTL^{ep} , LTL^{gp} 式からの決定性 ω オートマトンの構成について述べ，5 章では，LTL^{ep+} , LTL^{gp+} 式からの決定性 ω オートマトンの構成について述べる．最後に，7 章で本稿のまとめを述べる．

2 準備

2.1 LTL

ここでは，本研究で扱う線形時間論理（LTL）の定義を与える．本研究で扱う LTL は，様相演算子として，強い until 演算子 $\mathcal{U}_s$ と next 演算子 $\bigcirc$ を持つものである．

定義 2.1 (構文). 原子命題の集合 P が与えられたとき，P 上の *LTL* の式は以下のように定義される．

$$f := p \mid \neg f \mid f \wedge f \mid \bigcirc f \mid f \mathcal{U}_s f$$

ここで，p は原子命題集合 P に含まれる原子命題である．

LTL 式は，2^P 上の無限長の列によって評価される．（無限長の列の i 番目の要素は，そこで真となる原子命題の集合であり，そこに含まれない原子命題はそこで偽であることを意味する.）

定義 2.2 (意味論). 無限列 $\sigma \in (2^P)^\omega$ の i 番目が式 f を満たすことを $\langle \sigma, i \rangle \models f$ と表し，以下のように帰納的に定義する．

- $\langle \sigma, i \rangle \models p$ **iff** $p \in \sigma[i]$
- $\langle \sigma, i \rangle \models \neg f$ **iff** $\langle \sigma, i \rangle \not\models f$
- $\langle \sigma, i \rangle \models f_1 \wedge f_2$ **iff** $\langle \sigma, i \rangle \models f_1$ かつ $\langle \sigma, i \rangle \models f_2$
- $\langle \sigma, i \rangle \models f_1 \mathcal{U}_s f_2$ **iff**
  $(\exists j \geq 0)\ \Big(\langle \sigma, i+j \rangle \models f_2$ かつ
  $\forall k (0 \leq k < j)\ \langle \sigma, i+k \rangle \models f_1 \Big)$
- $\langle \sigma, i \rangle \models \bigcirc f$ **iff** $\langle \sigma, i+1 \rangle \models f$

$\langle \sigma, 0 \rangle \models f$ のとき σ を f の**モデル**という．$\langle \sigma, 0 \rangle \models f$ を単に $\sigma \models f$ と書く．

直感的には，$f_1 \mathcal{U}_s f_2$ は「f_2 がいつか必ず成立し，それまでの間ずっと f_1 が成立し続ける」ことを意味する．

略記として $\vee, \rightarrow$ を通常のように用いる．また，様相演算子 $\Box$,$\Diamond$,$\mathcal{U}_w$,$\mathcal{R}_w$,$\mathcal{R}_s$ を，$\mathcal{U}_s$ を用いて以下のように導入する．

$$\Diamond f \equiv true\, \mathcal{U}_s\, f \quad \text{[eventually]}$$
$$\Box f \equiv \neg \Diamond \neg f \quad \text{[globally]}$$
$$f_1\, \mathcal{U}_w\, f_2 \equiv \Box f_1 \vee (f_1\, \mathcal{U}_s\, f_2) \quad \text{[弱い until]}$$
$$f_1\, \mathcal{R}_w\, f_2 \equiv \neg(\neg f_1\, \mathcal{U}_s\, \neg f_2) \quad \text{[弱い release]}$$
$$f_1\, \mathcal{R}_s\, f_2 \equiv \neg(\neg f_1\, \mathcal{U}_w\, \neg f_2) \quad \text{[強い release]}$$

ただし，$true$ は恒真な命題とする．

式 f の部分式の集合を $sub(f)$ と表す．式の一番外側の演算子が様相演算子である式を時間式と呼び，f の部分式のうち，時間式であるものの集合を $temp(f)$ と表す．

また，LTL 式は以下の等価性より否定標準形（negation normal form），すなわち，否定が原子命題だけにかかる式へ変形することが可能である．

$$\neg\neg f \equiv f \qquad \neg(f_1 \wedge f_2) \equiv (\neg f_1) \vee (\neg f_2)$$
$$\neg \bigcirc f \equiv \bigcirc \neg f \qquad \neg(f_1 \vee f_2) \equiv (\neg f_1) \wedge (\neg f_2)$$
$$\neg(f_1 \mathcal{U}_s f_2) \equiv \neg f_1 \mathcal{R}_w \neg f_2 \quad \neg(f_1 \mathcal{R}_s f_2) \equiv \neg f_1 \mathcal{U}_w \neg f_2$$
$$\neg(f_1 \mathcal{U}_w f_2) \equiv \neg f_1 \mathcal{R}_s \neg f_2 \quad \neg(f_1 \mathcal{R}_w f_2) \equiv \neg f_1 \mathcal{U}_s \neg f_2$$

本稿の以降では，式は 否定標準形になっているものとする．

2.2 ω オートマトン

ここでは，本研究で扱う無限長の語を扱う有限状態オートマトンについて説明する．

無限長の語を扱うオートマトンの場合，行程も無限長となるため，有限長の語の場合のように行程の最後の状態が受理状態かどうかで受理・不受理を定められない．通常，無限長の語の上のオートマトンは，その行程に無限にしばしば現れる状態や遷移に対して言及し，受理・不受理を定める．本研究では，行程に無限回現れる状態について言及する Büchi オートマトンと co-Büchi オートマトン，無限回現れる遷移について言及する generalized Büchi オートマトンと generalized co-Büchi オートマトンを扱う．

以下でその定義を与える．

定義 2.3 (Büchi（co-Büchi）オートマトン). 非決定性 *Büchi*（*co-Büchi*）オートマトン $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, F \rangle$ は，Σ：アルファベット，Q：有限の状態の集合，$q_0 \in Q$：初期状態，$\delta \subseteq Q \times \Sigma \times Q$：遷移，$F \subseteq Q$：受理条件より構成される．$\Sigma$ 上の語 α に対する行程は，状態の列 ϱ で，$\varrho[0] = q_0$，かつすべての $i \geq 0$ において $(\varrho[k], \alpha[k], \varrho[k+1]) \in \delta$ となるものである．Σ 上の無限長の語 α に対して，以下を満たす行程が存在するとき，$\mathcal{A}$ は α を受理するという．

$$inf(\varrho) \cap F \neq \emptyset \quad (Büchi)$$
$$inf(\varrho) \cap F = \emptyset \quad (co\text{-}Büchi)$$

ここで，A 上の無限列 $\tilde{a}$ に対して無限回出現する要素の集合を $inf(\tilde{a}) = \{a \in A | \forall i \exists j > i. \tilde{a}[j] = a\}$ と定める．

定義 2.4 (遷移ベース generalized Büchi（generalized co-Büchi）オートマトン). 非決定性 *generalized Büchi*（*generalized co-Büchi*）オートマトン $\mathcal{A} = \langle \Sigma, Q, q_0, \delta, \mathcal{F} \rangle$ は，Σ:アルファベット，Q:有限の状態の集合，$q_0 \in Q$:初期状態，$\delta \subseteq Q \times \Sigma \times Q$:遷移，$\mathcal{F} \subseteq 2^{\delta}$:受理条件より構成される．$\Sigma$ 上の語 α に対する行程は，遷移 δ 上の列 $\varrho = (s_0, a_0, t_0)(s_1, a_1, t_1)\ldots$ で，$s_0 = q_0$，かつすべての $i \geq 0$ において $a_i = \alpha[i]$，$t_i = s_{i+1}$ となるものである．Σ 上の無限長の語 α に対して，以下を満たす行程が存在するとき，$\mathcal{A}$ は α を受理するという．

$$\forall F \in \mathcal{F}. inf(\varrho) \cap F \neq \emptyset \quad (generalized\ Büchi)$$
$$\exists F \in \mathcal{F}. inf(\varrho) \cap F = \emptyset \quad (generalized\ co\text{-}Büchi)$$

オートマトン $\mathcal{A}$ が受理する無限長の語の集合を $\mathcal{A}$ の受理言語といい，$L(\mathcal{A})$ と表す．状態 q から q' へ Σ 上の有限列 $\bar{a} \in \Sigma^*$（長さ n とする）で遷移可能である，すなわち，ある状態列 $\bar{q}$ が存在し（$|\bar{q}| = n+1$），$\bar{q}[0] = q, \forall i < n.(\bar{q}[i], \bar{a}[i], \bar{q}[i+1]) \in \delta, \bar{q}[n] = q'$ であるとき，$q \underset{\bar{a}}{\rightarrow} q'$ と書く．また，q から q' へ遷移可能である，つまり，$q \underset{\bar{a}}{\rightarrow} q'$ である $\bar{a}$ が存在するとき，$q \underset{*}{\rightarrow} q'$ と書く．

オートマトンがすべての $q \in Q$,$a \in \Sigma$ に対して $(q, a, q') \in \delta$ となる q' が高々 1 つしか存在しないとき，そのオートマトンは**決定的であるといい**，そのオートマトンを**決定性オートマトン**と呼ぶ．

なお，以降では，非決定性（Non-deterministic）と決定性（Determinisitic）をそれぞれ “N”， “D” と記す．また， Büchi，Co-büchi , 遷移ベース（Transition based）Generalized Büchi と Generalized Co-büchi をそれぞれ “B”，“C”，“TGB”，“TGC” と記し，オートマトンを “A” と記す．例えば，非決定性 co-Büchi オートマトンは “NCA” と記す．

3 LTL フラグメント

本研究で扱う LTL のフラグメントについて述べる．

まず，次のように 2 つのフラグメント（LTL^{ep} と LTL^{gp}）を定義する．

定義 3.1 (LTL^{ep} と LTL^{gp} の構文). 原子命題の集合 P が与えられたとき，P 上の LTL^{ep} の式は以下のように構成される．

$$f := g \mid f \wedge f \mid f \vee f \mid \bigcirc f \mid f\,\mathcal{U}_w f \mid f\,\mathcal{R}_w f \mid f\,\mathcal{U}_s g \mid g\,\mathcal{R}_s f$$

LTL^{gp} の式は以下のように構成される．

$$f := g \mid f \vee f \mid f \wedge f \mid \bigcirc f \mid f\,\mathcal{R}_s f \mid f\,\mathcal{U}_s f \mid g\,\mathcal{R}_w f \mid f\,\mathcal{U}_w g$$

ここで，g は，P 上の命題式（$\neg$，$\wedge$，$\vee$ より構成される式）とする．

LTL^{ep} は，「いつか必ず満たさなければならない制約は命題式によって記述しなければならない」というように LTL の構文を制限したフラグメントである．否定標準形では，そのような制約は，強い until 演算子 $\mathcal{U}_s$ と強い release 演算子 $\mathcal{R}_s$ を用い直接記述可能である．すなわち，式 $f_1 \mathcal{U}_s f_2$,$f_2 \mathcal{R}_s f_1$ においては，それぞれ f_2 が将来必ず満たされなけければならない制約となる．したがって，このフラグメントでは，そのような f_2 が命題式となるように構文を制限をしている．

また，LTL^{gp} は，LTL^{ep} と双対な関係にあり，「常に満たされなければならない制約は命題式によって記述しなければならない」というように制限したフラグメントである．そのような制約は，弱い until 演算子 $\mathcal{U}_w$ と弱い release 演算子 $\mathcal{R}_w$ を用いて直接記述可能であり，$f_1\mathcal{U}_w f_2$,$f_2\mathcal{R}_w f_1$ において，それぞれ f_1 が常に満たされなければならない制約となる．したがって，このフラグメントでは，このような f_1 が命題式となるように構文を制限をしている．

LTL^{ep} では，例えば，以下のような仕様記述が可能である．

$$\Box(Button_{open} \rightarrow Open) \tag{1}$$

意味：「常に開ボタンが押されたならば，ドアを開く.」

$$\Box(Request \rightarrow \Diamond Service) \tag{2}$$

意味：「リクエストされたならば，いつかサービスされる.」

$$\Box((a_1 \wedge \neg a_2 \wedge \Diamond a_2) \rightarrow (b\, \mathcal{U}_s a_2)) \tag{3}$$

意味：「イベント a_1 が生起し，その後イベント a_2 が生起するならば，その間ずっとイベント b が生起し続けなければならない.」

記述できない代表的なものとしては，

$$\Box\Diamond a \rightarrow \psi \quad (\equiv \Diamond\Box\neg a \vee \psi) \tag{4}$$

意味：「イベント a が断続的に生起されるならば，ψ を満たさなければならない.」
といった仕様記述が挙げられる．

これに対して，LTL^{gp} では，(1) 式などを記述することが可能であるが，(2) 式や (3) 式はこのフラグメントでは記述できない．

さらに，本研究では，上で与えた制限をより強めたフラグメント（LTL^{ep+} と LTL^{gp+}）も扱う．これらを以下のように定義する．

定義 3.2 (LTL^{ep+} と LTL^{gp+} の構文). 原子命題の集合 P が与えられたとき，P 上の LTL^{ep+} の式は以下のように構成される．

$$f := g \mid f \wedge f \mid g \vee f \mid \bigcirc f \mid f\,\mathcal{U}_* g \mid g\,\mathcal{R}_* f$$

また，LTL^{gp+} の式は以下のように構成される．

$$f := g \mid f \vee f \mid g \wedge f \mid \bigcirc f \mid f\,\mathcal{R}_* g \mid g\,\mathcal{U}_* f$$

ここで，g は，P 上の命題式（$\neg$，$\wedge$，$\vee$ より構成される式）とする．

LTL^{ep+} は，「$\vee$ 演算子で結合される式の片方，$\mathcal{U}$ 演算子の右側，$\mathcal{R}$ 演算子の左側が，それぞれ命題式でなければならない」というように制限したフラグメントである．また，LTL^{gp+} は，LTL^{ep+} と双対な関係にあるフラグメントである．

例えば，LTL^{ep+} では，(1) 式，(2) 式は記述できるが，(3) 式は記述できない．

より実際的な LTL 仕様の記述をみてみると，[11] で与えられている「m 台リフトを持つ n 階建てエレベータシステム」の仕様は，LTL^{ep+} によって（当然，LTL^{ep} においても）記述可能なものとなっていた．

4 LTL^{ep} 式，LTL^{gp} 式からの決定性 ω オートマトンの構成

本章では，前章で与えた LTL^{ep} 式，LTL^{gp} 式からの決定性 ω オートマトンの構成法を与える．

LTL^{ep} 式の決定性 ω オートマトンは，LTL^{gp} 式からの決定性 ω オートマトンの構成手続きを利用して，得ることが可能である（LTL^{ep} 式の否定をとった式が LTL^{gp} 式となり，また，決定性オートマトンにおいては受理条件を変更することでその補集合を受理する決定性オートマトンを得ることができるので）．そこで，本章では，まず LTL^{gp} 式からの決定性 ω オートマトンの構成法について述べ，その後 LTL^{ep} 式からの決定性 ω オートマトンの構成法について述べる．

4.1 LTL^{gp} 式からの NCA の構成

LTL^{gp} 式からの決定性 ω オートマトンの構成は，まず，そのモデルをちょうど受理する NCA を構成し，それを決定化することで行う．通常，LTL 式から決定性 ω オートマトンを構成する場合は，NBA を構成する．それに対して，構文に制限を与えた LTL^{gp} 式では，NBA よりも表現力が弱く，単純な手続きで決定化が行える NCA を構成することが可能となる．

以下でその構成法を与える．ここで与える NCA の構成は，時間論理の決定手続きであるタブロー法に基づくものである．

手続き 4.1 (分解手続き).
入力：式集合 S
出力：$(\emptyset$，式集合，式集合$)$ の集合 Σ

1. （初期化）$\Sigma := \{(S, \emptyset, \emptyset)\}$ とする．
2. （分解）任意の $(Proc_i, Cur_i, Next_i) \in \Sigma$，式 $f_{ij} \in Proc_i$ について，以下の操作 *(a)* から *(g)* の

いずれかを，f_{ij} の形に応じて適用する．これを任意の要素の $Proc_i$ が空となるまで繰り返す．ここで，$Proc'_i := Proc_i - \{f_{ij}\}, Cur'_i := Cur_i \cup \{f_{ij}\}$ とする．

(a) $f_{ij} = p$ *or* $\neg p$ のとき，$(Proc_i, Cur_i, Next_i)$ を以下で置き換える．
$(Proc'_i, Cur'_i, Next_i)$.

(b) $f_{ij} = f_1^{\bigcirc}$ のとき，$(Proc_i, Cur_i, Next_i)$ を以下で置き換える．
$(Proc'_i \cup \{f_1\}, Cur'_i, Next_i)$.

(c) $f_{ij} = f_1 \wedge f_2$ のとき，$(Proc_i, Cur_i, Next_i)$ を以下で置き換える．
$(Proc'_i \cup \{f_1, f_2\}, Cur'_i, Next_i)$.

(d) $f_{ij} = f_1 \vee f_2$ のとき，$(Proc_i, Cur_i, Next_i)$ を以下で置き換える．
$(Proc'_i \cup \{f_1\}, Cur'_i, Next_i)$,
$(Proc'_i \cup \{f_2\}, Cur'_i, Next_i)$.

(e) $f_{ij} = \bigcirc f$ のとき，$(Proc_i, Cur_i, Next_i)$ を以下で置き換える．
$(Proc'_i, Cur'_i, Next_i \cup \{f^{\bigcirc}\})$.

(f) $f_{ij} = f_1 \mathcal{U}_* f_2$ のとき，$(Proc_i, Cur_i, Next_i)$ を以下で置き換える．
$(Proc'_i \cup \{f_2\}, Cur'_i, Next_i)$,
$(Proc'_i \cup \{f_1\}, Cur'_i, Next_i \cup \{f_{ij}\})$.

(g) $f_{ij} = f_1 \mathcal{R}_* f_2$ のとき，$(Proc_i, Cur_i, Next_i)$ を以下で置き換える．
$(Proc'_i \cup \{f_2, f_1\}, Cur'_i, Next_i)$,
$(Proc'_i \cup \{f_1\}, Cur'_i, Next_i \cup \{f_{ij}\})$.

Cur_i に p, $\neg p$ とが含まれる場合，その $(Proc_i, Cur_i, Next_i)$ を削除する．

手続き 4.2 (LTL^{gp} から NCA の構成).
入力：*LTL*^{gp} 式 ψ
出力：*NCA* $\mathcal{N}_\psi = \langle 2^P, Q, q_0, \delta, F \rangle$

1. （初期化）$Q := \{q_0\}$，$q_0 := \{\psi\}$，$\delta := \emptyset$ とする．

2. （状態遷移）状態 $q \in Q$ に対して，分解手続きを行った結果を Σ とする．このとき，以下のように Q,δ を更新する．
$Q := Q \cup \{N \mid (\emptyset, C, N) \in \Sigma\}$,
$\delta := \delta \cup \{(q, s, N) \mid (\emptyset, C, N) \in \Sigma, p \in C \Rightarrow p \in s, \neg p \in C \Rightarrow p \notin s\}$.
これを Q，δ が変化しなくなるまで繰り返す．

3. （受理条件）受理集合 F を次のように定める．
$F := \{q \in Q \mid \exists f \in q.$
f が $f_1 \mathcal{U}_w f_2, f_1 \mathcal{R}_w f_2$ の形をした式でない $\}$.

定理 4.3 (正当性). ψ を *LTL*^{gp} 式とし，$\mathcal{N}_\psi$ を手続き *4.2* による *NCA* とする．このとき，$\mathcal{N}_\psi$ は ψ のモデルをちょうど受理する．

証明 補題 A.2 と補題 A.5 より示す．■

定理 4.4. *LTL*^{gp} 式 ψ から手続き *4.2* によって構成される *NCA* の状態数は，高々$2^{|temp(\psi)|}+1$ である．

証明 初期状態以外の状態の式集合に含まれうる式は，$f_1 \mathcal{U}_* f_2$, $f_1 \mathcal{R}_* f_2 \in sub(\psi)$，もしくは，$\bigcirc f_1 \in sub(\psi)$ である $f_1^{\bigcirc}$ という式である．したがって，そのような式の種類は $|temp(\psi)|$ と一致する．よって，初期状態以外の状態の式集合の種類は $2^{|temp(\psi)|}$ であり，構成される NCA の状態数は高々$2^{|temp(\psi)|}+1$ である．■

4.2 NCA に現れる構造的な特徴

前節にて LTL^{gp} から NCA の変換手続きを与え，LTL^{gp} 式のモデル集合を NBA よりも表現力の弱い NCA で表現可能であることを示した．しかし，LTL^{gp} の持つ特性は，それだけではない．LTL^{gp} 式より前節で与えた手続きによって構成される NCA には次のような特徴が現れる．

定理 4.5. ψ を *LTL*^{gp} 式とし，$\mathcal{N}_\psi$ を ψ の手続き *4.2* により構成される *NCA* とする．このとき，$\mathcal{N}_\psi$ は以下の特徴をもつ．

- 任意の $q \in Q \setminus F$ において，$q \to q'$ かつ $q' \overset{*}{\to} q$ である $q' \in Q$ が存在するならば，$q = q'$ である．

証明 $q \in Q \setminus F$ としたとき，$\mathcal{N}_\psi$ の任意の q から q への遷移 $q, q_1, q_2, \ldots, q_{n-1}, q$ において，$q_i = q (1 \leq i \leq n-1)$ であること，すなわち，その状態列に対応する式集合 $N, N_1, N_2, \ldots, N_{n-1}, N$ のすべてが同じ $f_1 \mathcal{U}_w f_2$，$f_2 \mathcal{R}_w f_1$ の形の式によってのみ構成されていることを示す．

まず，N に含まれる式のうち，その式を真部分式としてもつ式が N に含まれない式 f を考える（$f = f_1 \mathcal{U}_w f_2$ or $f_2 \mathcal{R}_w f_1$）．このような f は，一度 $f \notin N_i$ となると再び $f \in N_j (j > i)$ となることはない．したがって，$\forall i. f \in N_i$ である．また，このように f が制

約として残り続ける場合，この式より，他の次へと残る制約は生じない．なぜなら，f がある N_i に含まれ，その f が N_{i+1} へとそのまま残る場合，f_2 が処理されることはなく，f_1 のみが処理される．LTL^{gp} の定義より f_1 は命題式であり，この f_1 より次へと残る制約は生じないからである．

また，N に含まれる式のうち，その式 f を真部分式としてもつ式 f' が N に含まれる場合も，$\forall i.f' \in N_i$ であり，f' より他の次へと残る制約は生じないと仮定すると，上と同様の理由により，$\forall i.f \in N_i$ であり，f より他の次へと残る制約は生じない．

よって，任意の $f \in N$ において $\forall i.f \in N_i$ であり，任意の $f \notin N$ で $\forall i.f \notin N_i$ である．■

この特徴は，直感的には，NCA において「受理状態でない状態を含む強連結成分は，その状態のみからなる」ということである．このような特徴を持つ NCA では，受理行程にはただ一つの状態のみが無限回現れる（その状態は自己ループを持つことになる）．この特徴が，単純な決定化を可能とする．

4.3 構造的な特徴を持つ NCA から DTGCA の構成

前節で与えた NCA に現れる「受理状態でない状態を含む強連結成分は，その状態のみからなる」という特徴を利用することで，powerset construction（NCA の状態の集合を決定化後のオートマトンの状態とする構成）を基にした決定化が可能となる[2]．以下で，その決定化手続きを与える．

手続き 4.6 (特徴を持つ NCA から DTGCA の構成).
入力：定理 *4.5* の特徴を持つ *NCA* $\mathcal{N} = \langle \Sigma, Q, q_0, \delta, F \rangle$
出力：*DTGCA* $\mathcal{D} = \langle \Sigma, Q', q'_0, \delta', \mathcal{F}' \rangle$

1. （初期化）$Q' := \{\{q_0\}\}$，$q'_0 := \{q_0\}$，$\delta' := \emptyset$ とする．

2. （状態遷移）任意の $q' \in Q'$，$a \in \Sigma$ に対して，Q'，δ' を次のように更新する．
   $Q' := Q' \cup \{q'_{succ}\}$
   $\delta' := \delta' \cup \{(q', a, q'_{succ})\}$
   q'_{succ} は以下のように定める．
   $q'_{succ} = \{q_{succ} \in Q \mid q \in q', (q, a, q_{succ}) \in \delta\}$
   これを Q'，δ' が変化しなくなるまで繰り返す．

3. （受理条件）受理条件は次のように定める．
   $\mathcal{F}' = \{F'_q | q \in F\}$
   $F'_q := \{(q', a, q'_{suc}) \in \delta' | q \in q' \Longrightarrow (q, a, q) \notin \delta\}$

[2] このような特徴のない一般的な NCA の決定化は，[8] によって与えられている交代 Büchi オートマトンから NBA への変換のための構成（subset construction）を利用すれば可能となる．

手続き 4.6 の正当性の証明

$[\alpha \in L(\mathcal{N}) \Rightarrow \alpha \in L(\mathcal{D})]$ $\alpha \in L(\mathcal{N})$ と仮定すると，$inf(\varrho_n) \subseteq Q \setminus F$ となる受理行程 ϱ_n が存在する．ここで，$\mathcal{N}$ の特徴より その行程に $Q \setminus F$ に含まれる状態が複数無限にしばしば現れることはないため，$|inf(\varrho_n)| = 1$ である．よって，ある状態 $q_{loop} \in Q \setminus F$ が存在し，

$$\exists i. \forall j \geq i : \varrho_n[j] = q_{loop} \qquad (a)$$

である．次に α に対する $\mathcal{D}$ の行程について考える．$\mathcal{D}$ の構成より，α に対して唯一の行程 ϱ_d が存在して，$\varrho_d[i] = (S_i, \alpha[i], T_i)$ とおくと，

$$S_i = \{\varrho[i] \in Q | \varrho \text{ は } \alpha \text{ に対する } \mathcal{N} \text{ の行程 } \}$$

である．(a) より，あるポイント i_{loop} 以降で，$q_{loop} \in S_j$ かつ $(q_{loop}, \alpha[j], q_{loop}) \in \delta$ である．したがって，$\forall j \geq i_{loop}.(S_j, \alpha[j], T_j) \notin F'_{q_{loop}}$ であり，$inf(\varrho_d) \cap F'_{q_{loop}} = \emptyset$ を満たす．よって，ϱ_d は $\mathcal{D}$ の受理行程であり，$\alpha \in L(\mathcal{D})$ である．ゆえに，$\alpha \in L(\mathcal{N}) \Rightarrow \alpha \in L(\mathcal{D})$ である．

$[\alpha \in L(\mathcal{D}) \Rightarrow \alpha \in L(\mathcal{N})]$ $\alpha \in L(\mathcal{D})$ と仮定すると，α に対する受理行程 ϱ_d が存在する．受理条件より，ある $q_{loop} \in Q \setminus F$ が存在し，$inf(\varrho_d) \cap F'_{q_{loop}} = \emptyset$ である．つまり，ある i_{loop} 以降で $\varrho_d[j] \notin F'_{q_{loop}}$ であり，

$$\forall j \geq i_{loop} : q_{loop} \in S_j \ and \ (q_{loop}, \alpha[j], q_{loop}) \in \delta \qquad (b)$$

を満たす．ここで，$q_{loop} \in S_{i_{loop}}$ であるので，$\mathcal{N}$ において，

$$q_0 \underset{\alpha[0 \ldots i_{loop}-1]}{\rightarrow} q_{loop} \qquad (c)$$

である．また，(b) より，

$$q_{loop} \underset{\alpha[j]}{\rightarrow} q_{loop} \quad (j \geq i_{loop} \text{ において}) \qquad (d)$$

である．よって，(c) と (d) より，次の条件を満たす $\mathcal{N}$ の α に対する行程 ϱ_n が存在する．

$$\exists i. \forall j \geq i : \varrho_n[j] = q_{loop}$$

この行程は $inf(\varrho_n) = \{q_{loop}\} \subseteq Q \setminus F$ であることから，受理条件を満たす．よって，$\alpha \in L(\mathcal{N})$ である．ゆえに，$\alpha \in L(\mathcal{D}) \Rightarrow \alpha \in L(\mathcal{N})$ である．■

定理 4.7. *$\mathcal{N}$ を定理 4.5 の特徴を持つ状態数 n の NCA とする．$\mathcal{N}$ に対して手続き 4.6 によって構成される DTGCA の状態数は高々2^n である．*

これまでに与えた LTL^{gp} 式から NCA の変換，NCA から DTGCA の変換により，LTL^{gp} 式からの決定性 ω オートマトンの構成が可能となる．

定理 4.8. *ψ を LTL^{gp} 式とする．このとき，ψ のモデルをちょうど受理する DTGCA を状態数高々$2^{2^{|temp(\psi)|}+1}$ で構成可能である．*

証明 定理 4.4 と定理 4.7 より成立する．■

4.4 LTL^{ep} 式からの決定性 ω オートマトンの構成

続いて，LTL^{gp} 式からの DTGCA の構成を用い，LTL^{ep} 式から決定性 ω オートマトンを構成可能であることを示す．

LTL^{ep} と LTL^{gp} は互いに双対な関係にあり，また，命題にかかる否定の出現に関しての制約はない．したがって，次の定理が成立する．

定理 4.9. *f を LTL^{ep} 式とする．このとき，$\neg f$（を否定標準形に変形した式）は LTL^{gp} 式である．*

証明 2.1 節で与えた等価性を利用して $\neg f$ を否定標準形に変換することを考える（f は否定標準形になっていると仮定する）．この変換によって，f に現れるすべての until 式と release 式は，それぞれ双対な演算子へと置き換わる．したがって，f が LTL^{ep} であれば，この式は LTL^{gp} である．■

また，決定性オートマトンおいては受理条件を双対なものへと変更することで，容易にその受理言語の補集合を受理する決定性オートマトンを得ることができる．

定理 4.10. *$\mathcal{D}_{GC} = \langle \Sigma, Q, q_0, \delta, \mathcal{F} \rangle$ を DTGCA とし，$\mathcal{D}'_{GB} = \langle \Sigma, Q \cup \{q_*\}, q_0, \delta', \mathcal{F}' \rangle$ を DTGBA とする．ここで，$T_* := \{(q_*, a, q_*) | a \in \Sigma\}$ とおき，*

$$\delta' := \delta \cup \{(q, a, q_*) | \neg \exists q'.(q, a, q') \in \delta\} \cup T_* ,$$

$$\mathcal{F}' := \{F \cup T_* | F \in \mathcal{F}\} .$$

とする．このとき，$L(\mathcal{D}'_{GB}) = \Sigma^\omega \setminus L(\mathcal{D}_{GC})$ である．

証明 DTGCA $\mathcal{D}'_{GC} = \langle \Sigma, Q \cup \{q_*\}, q_0, \delta', \mathcal{F}' \rangle$ とすると，$L(\mathcal{D}'_{GC}) = L(\mathcal{D}_{GC})$ であり，また，$\mathcal{D}'_{GC}$ と $\mathcal{D}'_{GB}$ はすべての $\alpha \in \Sigma^\omega$ に対して必ずある同一の行程 ϱ が存在する．また，

$$\begin{aligned} \alpha \in L(\mathcal{D}'_{GB}) &\Longleftrightarrow \forall F \in \mathcal{F}'.inf(\varrho) \cap F \neq \emptyset \\ &\Longleftrightarrow \neg\exists F \in \mathcal{F}'.inf(\varrho) \cap F = \emptyset \\ &\Longleftrightarrow \alpha \notin L(\mathcal{D}'_{GC}) \end{aligned}$$

である．よって，$L(\mathcal{D}'_{GB}) = \Sigma^\omega \setminus L(\mathcal{D}'_{GC})$ であり，$L(\mathcal{D}'_{GB}) = \Sigma^\omega \setminus L(\mathcal{D}_{GC})$ である．■

したがって，以下のような方法で，LTL^{ep} 式 ψ より DTGBA を構成することが可能となる．

1. $\neg\psi$（この式は LTL^{gp} 式となる）から手続き 4.2 により NCA $\mathcal{N}_{\neg\psi}$ を構成する．
2. $\mathcal{N}_{\neg\psi}$ から手続き 4.6 により DTGCA $\mathcal{D}^{GC}_{\neg\psi}$ を構成する．
3. DTGCA $\mathcal{D}^{GC}_{\neg\psi}$ から，定理 4.10 で与えたような DTGBA $\mathcal{D}^{GB}$ を構成する．

ここで，$L(\mathcal{D}^{GC}_{\neg\psi}) = L(\mathcal{N}_{\neg\psi}) = \{\sigma | \sigma \not\models \psi\}$ であり，$L(\mathcal{D}^{GB}) = 2^P \setminus L(\mathcal{D}^{GC}_{\neg\psi})$ であることから，$L(\mathcal{D}^{GB}) = \{\sigma | \sigma \models \psi\}$ である．

定理 4.11. *ψ を LTL^{ep} 式とする．このとき，ψ のモデルをちょうど受理する $DTGBA$ を高々$2^{2^{|temp(\psi)|}+1}+1$ の状態数で構成可能である．*

5 LTL^{ep+}, LTL^{gp+} 式からの決定性 ω オートマトンの構成

前章では，LTL^{ep} と LTL^{gp} は，複雑な処理を伴う Safra's construction による決定化を用いずに，powerset construction を基にした単純な手続きで決定性 ω オートマトンの構成が可能となることを示した．しかし，複雑な処理を避けることはできても，本手続きにおいて，その状態数は式の長さに対して 2 重指数オーダーであり，その構成にかかる計算量は大きい．本章では，そのサブフラグメントである LTL^{ep+} と LTL^{gp+} においては，複雑な処理を避けられるだけでなく，その状態数を指数オーダーに抑えらることを示す．

まず，LTL^{gp+} の決定性 ω オートマトンの構成について述べる．LTL^{gp+} は，LTL^{gp} のサブフラグメントである，すなわち，LTL^{gp+} 式は，LTL^{gp} 式でもあるということから，前章で与えた手続きによって

DGCA を構成することができる．LTL^{gp+} より構成される NCA に次に与えるような特徴が現れることにより，構成される DGCA の状態数が指数オーダーに抑えられる．

定理 5.1. *LTL^{gp+} 式 ψ から手続き 4.2 によって構成される NCA の状態数は，高々$|temp(\psi)|+2$ である．*

証明

まず，φ を LTL^{gp+} 式とし，$\{\varphi\}$ に対して分解手続きを適用させるとき，計算過程での Σ の任意の要素 $(Proc, Cur, Next)$ が以下の条件を満たすことを示す．

- $|Proc \cap temp'(\varphi)| \leq 1$ ， $|Next| \leq 1$ and,
- $|Proc \cap temp'(\varphi)| = 1 \Rightarrow |Next| = 0$.

$temp'(\varphi)$ は $\{f \in sub(\varphi) | temp(f) \neq \emptyset\}$ とする．

ステップ 1 の初期化における $(\{\varphi\}, \emptyset, \emptyset)$ は，明らかに条件の満たす．ステップ 2 において，$(Proc_i, Cur_i, Next_i)$ を $f_{ij} \in Proc_i$ の形に応じて置き換える際に，置き換えられる $(Proc_i, Cur_i, Next_i)$ が上の条件を満たす場合，置き換える要素も上の条件を満たすことを示す．（ここでは，$f_{ij} = f_1 \wedge f_2$, $f_1 \mathcal{U}_* f_2$ 場合のみ示す．）

$f_{ij} = f_1 \wedge f_2$ の場合．置き換える要素の $Next$ は，もとと変わらない．また，$Proc_i$ も LTL^{gp+} の特徴より $f_1 \wedge f_2$ の片方の式が必ず命題式のみであるため，$|Proc_i \cap temp'(\psi)| = |(Proc'_i \cup \{f_1, f_2\}) \cap temp'(\varphi)|$ である．よって，置き換える要素も上の条件を満たす．

$f_{ij} = f_1 \mathcal{U}_* f_2$ の場合．$Proc_i$ に時間式である f_{ij} が含まれているため，$|Next_i| = 0$ であり，また，$|Proc'_i \cap temp'(\varphi)| = 0$ である．これより，1 つ目の置き換える要素は，$|(Proc'_i \cup \{f_2\}) \cap temp'(\varphi)| \leq 1$ かつ $|Next_i| = 0$ であり，条件を満たす．また，LTL^{gp+} の特徴より f_1 が必ず命題式のみであるため，2 つ目の置き換える要素は，$|(Proc'_i \cup \{f_1\}) \cap temp'(\varphi)| = 0$ かつ $|Next_i \cup \{f_{ij}\}| = 1$ であり，条件を満たす．

よって，分解手続きの計算過程での Σ の任意の要素で上の条件を満たす．

これより，LTL^{gp+} 式 φ に対して，$\{\varphi\}$ に分解手続きを適用して得られる結果を Σ_{out} とするとき，任意の $(\emptyset, C, N) \in \Sigma_{out}$ において $|N| \leq 1$ である．

次に，LTL^{gp+} 式 ψ より構成される NCA の構造について考える．上の結果より，初期状態の遷移先となる状態の式集合は，いずれも要素が一つの式集合，もしくは，空集合となる．また，その要素が一つの式集合を持つ状態の遷移先の状態も，要素が一つの式集合，もしくは，空集合となる．なぜなら，φ の任意の部分式も LTL^{gp+} 式となるからである．したがって，構成されるオートマトンの任意の状態の式集合は，要素が一つの式集合，もしくは，空集合となる．また，初期状態以外の状態に含まれうる式は，until，release 式，もしくは，$f^{\bigcirc}$ という形の式であり，その種類の数は $|temp(\psi)|$ と一致する．よって，オートマトンの状態となりうるのは，その他に初期状態，式集合が空の状態のみであり，構成される NCA の状態数は，高々$|temp(\varphi)|+2$ である．■

定理 5.2. *LTL^{gp+} 式 ψ から高々$2^{|temp(\psi)|+2}$ の状態数の DTGCA を構成することが可能である．*

証明 定理 4.7 と定理 5.1 より成立する．■

続いて，LTL^{ep+} 式からの決定性 ω オートマトンの構成について述べる．LTL^{ep+} と LTL^{gp+} は双対な関係にあるため，定理 4.9 と同様な性質が成立する．

定理 5.3. *f を LTL^{ep+} 式とする．このとき，$\neg f$（を否定標準形に変形した式）は LTL^{gp+} 式である．*

したがって，LTL^{ep+} においても，LTL^{gp+} からの構成を用い，DTGBA を構成できる．その際に構成する NCA が線形オーダーの状態数で抑えられるため，それを決定化し，受理条件を変更し，得られる DTGBA の状態数も指数オーダーに抑えられる．

定理 5.4. *LTL^{ep+} 式 ψ から高々$2^{|temp(\psi)|+2}+1$ の状態数の DTGBA を構成することが可能である．*

証明 定理 4.7，定理 4.10，及び定理 5.1 より成立する．■

6 他のフラグメントとの関係

[7] で与えられている時間的性質（ふるまい集合）の分類と本研究で与えたフラグメントの関係について述べる．そこでは，時間的性質を reactivity，recurrence，persistence，obligation，safety，guarantee という 6 つのクラスに分類している（recurrence と persistence，safety と guarantee は，それぞれ双対な関係にある）．これまでに，それらと LTL の構文との対応関係や，オートマトンに現れる特徴との対応関係について研究が行われている [3, 2].

そこで与えられている safety や guarantee は，すべての状態を受理状態とする非決定性（全称性）オート

マトンとの対応関係があるため，この safety や guarantee に対応するフラグメントでも，powerset construction を用いた決定性 ω オートマトンの構成が可能である．本研究で与えた LTL^{ep} と LTL^{gp} は，それらのフラグメントよりも弱い制限を与えたフラグメントとなっており，より広い範囲の LTL 仕様において，本稿で与えた決定化手続きは適用可能である．

また，recurrence と persistence は，weak と呼ばれる構造的特徴を持つ非決定性（全称性）オートマトンとの対応関係があり，Safra's construction よりはシンプルな subset construction[8] を用いて決定性 ω オートマトンを構成することが可能となる．それに対し，本研究で与えた LTL^{ep} と LTL^{gp} は，recurrence と persistence に対応する LTL のフラグメントよりも強い構文的な制限を与えたものではあるが，weak よりも強い特徴「受理状態でない状態を含む強連結成分は，その状態のみからなる」を持つ NCA との対応関係がある．これにより，subset construction よりも単純な powerset construction を用いて決定性 ω オートマトンを構成することができる．

7 おわりに

本稿では，実現可能性判定やプログラム合成において必要となる決定性 ω オートマトンの構成の観点から，リアクティブシステムの仕様記述言語である LTL を構文的に制限した次のような LTL フラグメントを与えた．

- powerset construction を用いた決定化が可能であるフラグメント（LTL^{ep} と LTL^{gp}）
- さらに，状態数を指数オーダーで抑えられるフラグメント（LTL^{ep+} と LTL^{gp+}）

LTL^{ep}（LTL^{gp}）は，LTL に対して，「いつか必ず満たさなければならない制約（常に成り立ち続けなければならない制約）は命題式によって記述しなければいけない」という制限を与えたフラグメントである．LTL^{gp} 式からは「受理状態でない状態を含む強連結成分は，その状態のみからなる」という特徴を持つ NCA を構成可能であり，この特性を利用することで，LTL^{ep} と LTL^{gp} は powerset construction を基にした単純な手続きで決定性 ω オートマトンを構成することが可能となる．LTL^{ep+}（LTL^{gp+}）は，LTL^{ep}（LTL^{gp}）のサブフラグメントであり，LTL^{gp+} からは状態数が線形オーダーの NCA を構成できる．これにより，LTL^{ep+} と LTL^{gp+} から構成される決定性 ω オートマトンの状態数は指数オーダーに抑えられる．

本研究で与えた決定化手続きは，Safra's construction による手法のような複雑な処理を伴わず，単純な集合演算により構成される powerset construction を基にしたものである．そのため，状態数の削減だけでなく，BDD などを利用した実装を行いやすいため，実装上の効率化も期待できる．

参考文献

[1] Martín Abadi, Leslie Lamport, and Pierre Wolper. Realizable and unrealizable specifications of reactive systems. In *Proceedings of the 16th International Colloquium on Automata, Languages and Programming*, pp. 1–17, 1989.

[2] Ivana Cerná and Radek Pelánek. Relating hierarchy of temporal properties to model checking. In *Mathematical Foundations of Computer Science*, Vol. 2747, 2003.

[3] Edward Chang, Zohar Manna, and Amir Pnueli. Characterization of temporal property classes. In Werner Kuich, editor, *Automata, Languages and Programming, 19th International Colloquium*, Vol. 623 of *Lecture Notes in Computer Science*, 13–17 July 1992.

[4] Paul Gastin and Denis Oddoux. Fast LTL to Büchi automata translation. In *Proceedings of the 13th International Conference on Computer Aided Verification*, 2001.

[5] Rob Gerth, Doron Peled, Moshe Y. Vardi, and Pierre Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *Protocol Specification Testing and Verification*, pp. 3–18, 1995.

[6] Daniel Jackson. Automating first-order relational logic. In *Proceedings of the 8th ACM SIGSOFT international symposium on Foundations of software engineering*, pp. 130–139, 2000.

[7] Z. Manna and A. Pnueli. A hierarchy of temporal properties. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pp. 205–205, 1987.

[8] Satoru Miyano and Takeshi Hayashi. Alternating finite automata on omega-words. *Theor. Comput. Sci.*, Vol. 32, pp. 321–330, 1984.

[9] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proceedings of the 16th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 179–190, 1989.

[10] Shmuel Safra. On the complexity of omega-automata. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pp. 319–327, 1988.

[11] Vaithinathan Vanitha, Kenji Yamashita, Kimiyuki Fukuzawa, and Naoki Yonezaki. A method for structuralisation of evolutional specifications of reactive systems. In *ICSE 2000, The Third International Workshop on Intelligent Software Engineering (WISE3)*, pp. 30–38, 2000.

A 手続き 4.2 の正当性

補題 A.1. $\mathcal{N}$ を手続き *4.2* による *NCA* とし，$\mathcal{N}$ の状態 $q \in Q$ から $\sigma \in (2^P)^\omega$ の受理行程 $qq_1q_2\ldots$ が存在すると仮定する．この行程の状態と遷移に対応する式集合を $N_0, C_1, N_1, C_2, N_2, \ldots$ とすると，$\sigma \models \bigwedge C_1$ である．ここで，$(\emptyset, C_{i+1}, N_{i+1})$ は N_i に対して分解手続きを適用して得られる Σ の 1 つのある要素であり，N_i は q_i の式集合を表す．

証明 $f \in C_1$ ならば $\sigma \models f$ であることを式の構造に関する帰納法で示す（f がリテラル，$f_1 \mathcal{U}_s f_2$ の場合を示し，それ以外は省略する．）．

$f = p, \neg p$ である場合．$(q, s, q_1) \in \delta$ である s は C_1 に含まれるリテラルと無矛盾なものとして定められる．$\sigma[0]$ によって q_1 に遷移可能であることから，$\sigma \models f$ である．

$f = f_1 \mathcal{U}_s f_2$ の場合．式の分解の仕方より，$f \in C_i$ ならば，$f_1 \in C_i$ かつ $f \in N_i$，もしくは $f_2 \in C_i$ である．また，$f \in N_i$ ならば，$f \in C_{i+1}$ である．よって，$\forall j. f_1 \in C_j$ もしくは $\exists j. f_2 \in C_j \wedge \forall k \leq j. f_1 \in C_k$ のいずれかである．受理行程であることから，あるポイント以降 f が N_i に現れない．したがって，前者を満たすことはなく，$N_0, C_1, N_1, C_2, N_2, \ldots$ は後者を満たす．よって，帰納法の仮定より，$\exists j. \sigma[j\ldots] \models f_2 \wedge \forall k \leq j. \sigma[k\ldots] \models f_1$ であり，$\sigma \models f$ となる．■

補題 A.2 (健全性). ψ を LTL^{gp} とし，$\mathcal{N}_\psi$ を手続き *4.2* による *NCA* とする．このとき，$\sigma \in L(\mathcal{N}_\psi)$ ならば，$\sigma \models \psi$ である．

証明 $\sigma \in L(\mathcal{N}_\psi)$ ならば，σ による受理行程 ϱ が存在する．この行程の状態と遷移に対応する式集合を $N_0, C_1, N_1, C_2, N_2, \ldots$ とすると，$\psi \in C_1$ である．よって，補題 A.1 より，$\sigma \models \psi$ である．■

補題 A.3. S を式の集合として，Σ を S に対する分解手続きによって得れる集合とする．このとき，以下が成立する．

$$\bigwedge S \equiv \bigvee_{(\emptyset, C, N) \in \Sigma} (\bigwedge C \wedge \bigcirc \bigwedge N)$$

ここで，$f^\bigcirc$ の形の式は f として解釈する．

証明 ステップ 2 にて，置き換えられる $(Proc_i, , Cur_i, Next_i)$ と置き換える $(Proc'_i, Cur'_i, Next'_i)$，$(Proc''_i, Cur''_i, Next''_i)$（場合によっては 1 つ）が，

$$\begin{aligned}&(\bigwedge Proc_i \wedge \bigwedge Cur_i \wedge \bigcirc \bigwedge Next_i)\\ &\equiv (\bigwedge Proc'_i \wedge \bigwedge Cur'_i \wedge \bigcirc \bigwedge Next'_i)\\ &\quad \vee (\bigwedge Proc''_i \wedge \bigwedge Cur''_i \wedge \bigcirc \bigwedge Next''_i)\end{aligned}$$

を満たす（時間論理式の意味より）．よって，題意は示される．■

補題 A.4. $\mathcal{N}$ を手続き *4.2* による *NCA* とし，S を $\mathcal{N}$ のある状態 q の式集合，σ を 2^P 上の列とする．このとき，$\sigma \models \bigwedge S$ ならば，$\sigma[1\ldots] \models \bigwedge N$ となる式集合 N の状態 q' へ $\sigma[0]$ で遷移可能となる．さらに，$f = f_1 \mathcal{U}_* f_2$，$f_2 \mathcal{R}_* f_1 \in S$ であり，$\sigma \models f_2$ であるならば，$f \notin N$ の状態へ遷移可能である．

証明 S に対する分解手続きによる結果 Σ において，上の条件を満たす $(\emptyset, C, N)$ が存在すればよい．補題 A.3 より，$\bigwedge S \equiv \bigvee_{(C,N)\in\Sigma}(\bigwedge C \wedge \bigcirc \bigwedge N)$ であり，$\sigma \models \bigwedge C \wedge \bigcirc \bigwedge N$ である C, N が存在する．任意の C に含まれるリテラル f において $\sigma \models f$ であるため，N を持つ状態へ遷移可能である．また，$\sigma \models \bigcirc \bigwedge N$ であることから $\sigma[1\ldots] \models \bigwedge N$ である．

式の分解の仕方より，N に $f = f_1 \mathcal{U}_* f_2$，$f_2 \mathcal{R}_* f_1$ の形の式が入るのは，$f \in C$ ときのみである．その分解を行うとき，N に f に追加せず，C に f_2 を追加するものが存在する．そのようにして得られた (C, N) も，$\sigma \models f_2$ であれば，$\sigma \models \bigwedge C \wedge \bigcirc \bigwedge N$ である．よって，題意は示された．■

補題 A.5 (完全性). ψ を LTL^{gp} 式とし，$\mathcal{N}_\psi$ を手続き *4.2* による *NCA* とする．このとき，$\sigma \models \psi$ ならば，$\sigma \in L(\mathcal{N}_\psi)$ である．

証明 $\sigma \models \psi$ であるときに，$\mathcal{N}_\psi$ に σ の受理行程が存在することを示す．

補題 A.4 より，$\sigma \models \psi$ であるならば，$\sigma \models (\bigwedge C_1 \wedge \bigcirc \bigwedge N_1)$ となる C_1 と N_1 が存在する（ここで，C_1, N_1 は分解手続きにより $\{\psi\}$ に対して得られる式集合のペアの 1 つである）．また，これを繰り返し適用し，$\sigma[i\ldots] \models (\bigwedge C_{i+1} \wedge \bigcirc \bigwedge N_{i+1})$ となる C_i と N_i が存在する．このようにして定まる $\{\psi\}, N_1, N_2, \ldots$ は，$\mathcal{N}_\psi$ での σ に対する行程となる．ここで，$f = f_1 \mathcal{U}_* f_2, f_2 \mathcal{R}_* f_1 \in N_i$ かつ $\sigma[i\ldots] \models f_2$

であるとき，$f \notin N_{i+1}$ である N_{i+1} へ遷移するような行程を考える（補題 A.4 より，このような行程は存在する）．このような行程が, 状態に現れる任意の式 f において，以下を満たすことを示す.

$\exists i.(\forall j \geq i.f \in N_j \ or \ \forall j \geq i.f \notin N_j)$
（$f = f_1 \mathcal{U}_w f_2 or f_2 \mathcal{R}_w f_1$ のとき）
$\exists i \forall j \geq i.\ f \notin N_j$ （その他のとき）

まず，$f = \psi$（入力の式）において，条件を満たすことを示す. 分解手続きの特徴より，N_{i+1} に現れる式は N_i に現れる式の部分式のみである．よって，状態に常に現れつづける，もしくは，ある i 以降現れないかのいずれかである．f が until，release 式以外の場合，明らかに後者であり，条件を満たす．$f = f_1 \mathcal{U}_s f_2, f_2 \mathcal{R}_s f_1$ である場合を考える．$\sigma \models f$ であるとき，その意味の定義より $\sigma[i \ldots] \models f_2$ である i が存在し，このとき，$f_2 \in C_i \ f \notin N_i$ である遷移を選ぶ．この i 以降，f を部分式として持つ式が他に存在せず，f は行程に現れないので，条件を満たす．$f = f_1 \mathcal{U}_w f_2, f_2 \mathcal{R}_w f_1$ である場合は，N に常に現れつづける，もしくは，ある i 以降現れない，いずれの場合でも条件を満たす.

次に，式 f を真部分式として持つ任意の式において条件を満たすと仮定して，f もその条件を満たすことを示す．仮定より，以降の状態で f を真部分式として持つ式が $f_1 \mathcal{U}_w f_2, f_2 \mathcal{R}_w f_1$ の形をした式のみあり，以降ずっとそれらの式が現れるポイント i が存在する．ここで，行程の選び方より，$\forall j \geq i.\sigma[j \ldots] \not\models f_2$ であり，$\forall j \geq i.f_2 \notin C_j$ である．したがって，この $f_1 \mathcal{U}_w f_2, f_2 \mathcal{R}_w f_1$ の形の式より，次へと残る制約は生じない（LTL^{gp} の特徴より f_1 は命題式であるから）．よって，$j \geq i$ において f が現れるとすれば，それは f が制約として残り続ける場合のみである．したがって，ψ が条件を満たすのと同様の理由により，f は条件を満たす.

以上より，任意の式において条件を満たすことがいえる．この条件を満たす行程であれば，受理条件を満たすことから，$\mathcal{N}_\psi$ に σ の受理行程が存在する．よって，題意は示された．■