非単調命題時間論理とその形式的仕様記述への応用†

佐伯元司††

本論文は，形式的仕様記述に使用するための非単調命題時間論理について述べたものである．時間論理は，並列システムや通信プロトコルといった動的システムを形式的に記述するには有用であるが，実際，人間がシステムの諸性質を検証できるほど完全に記述するのは難しい．というのは，人間は常識に基づいた暗黙的推論を行っているため，システムの細部にまでわたって時間論理で完全に記述することが難しいからである．本論文では，「記述されていない状態遷移は一切起こらない」，「共有変数の値は更新されたことが示されない限り保持される」という2つの暗黙的推論が行えるような非単調な命題時間論理を提案する．この論理は，通常の線形時間論理に既存の temporal オペレータの機能を損なうことなく，McDermott の手法に基づいて，ある時刻における論理式の無矛盾性を表す様相オペレータを追加したものである．その意味論は，証明可能性演算子の不動点を用いることによって与える．さらに，Tableau 法に基づく証明手続きを与え，実際の仕様記述例と仕様の諸性質の推論例についても述べる．ユーザが仕様として記述した論理式の集合に，上記の暗黙的推論を行うための論理式が追加され，推論が行われる．追加される論理式は，あらかじめ用意されたスキーマと仕様として記述された論理式の構文とによって自動的に生成される．

1. まえがき

リアルタイムシステムや通信プロトコルといったプログラムの動作系列が本質であるような動的システムの仕様を記述する手法として，自然言語，状態遷移図(表)，ペトリネット，時間論理などがあるが，各々特有の長所短所を持っている．時間論理は，厳密な数学的基盤を持っているため，システムの各種の性質，例えば safety や liveness などの検証を行う手法が提案されている[1]．しかし，時間論理に意味的基礎をおく仕様記述言語がいまだに実用化されていないのは，効率的な定理証明器の実現が困難なことだけでなく，人間が抱いているシステムのすべての性質を実際に証明できるほど完全に，その仕様を論理式で記述することが難しいことに原因がある．例えば，対象システムとして図1のようなシステムのイメージが人間の頭の中にあったとする．このようなシステムを時間論理式で記述する．ここで atL をプログラム中の L でラベル付けされた statement を実行しているときに真となる命題，$A \Rightarrow B$ は，論理式 A で表される状態から B へ遷移することを表す記号で，temporal operator，論理記号を用いて $A \rightarrow ((A \text{ until } B) \wedge \Diamond B)$ と定義する．temporal operator は第2章で形式的に定義するが，直観的には A until B は B が真になるまで A が真であり続ける，$\Diamond A$ は将来 A が真になる，$\Box A$ は A が将来ずっと真であり続けることを表す．Flag-is-1 は共有変数 Flag の値が1であることを，Flag-is-0 は0であることを表す命題とすると，自然な記述は，

$$\Box((\text{atL} \wedge \text{Flag-is-1}) \Rightarrow \text{atM}),$$
$$\Box(\text{atM} \Rightarrow (\text{atN} \wedge \text{Flag-is-0}))$$
$$\Box(\text{atN} \Rightarrow \text{atK}) \qquad (1.1)$$

となるであろうし，また読者にとってもシステムの概略をつかむのには適当であろう．このような記述は最終状態Kへの到達可能性 ($\text{atL} \wedge \text{Flag-is-1} \rightarrow \Diamond \text{atK}$) を証明するには十分であるが，最終状態で Flag-is-0 であることを証明することは，仕様として記述されていないために，不可能である．しかし，人の思考においては，Flag が0となっていることが常識的である．また，LからMへ遷移するときに，他の状態，例えばKを経由して遷移するようなシステムも仕様を満たすシステムの1つである．というのは，LからMへ遷移する間は ～atK であることが証明されないからである．また，Kから先においても，どの状態へ遷移しても仕様の上では構わない．これらの事実も人の常識的な推論とは異なった結果を与える．このような現象が生じないように時間論理で正確に記述することは可能ではあるが，記述量が多くなり，読者に仕様の本質的な部分が伝わらなくなり，その結果理解しにくくなる．また仕様記述でよく用いられるパターンについて，⇒のようにマクロ的な記法を導入するのにも限界がある．さらに，記述者にとっても，すべてにわたって十分にかつ完全に記述することを強いるのは，誤りを増大させる原因となる．人間がこのような不完全な記述のみからシステムを理解し，諸性質を推論できる

† Non-monotonic Propositional Temporal Logic and Its Application to Formal Specifications by MOTOSHI SAEKI (Department of Computer Science, Faculty of Engineering, Tokyo Institute of Technology).

†† 東京工業大学工学部情報工学科

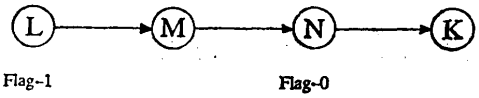

図 1 記述対象システム
Fig. 1 An Example of systems to be specified.

のは，常識に基づいた暗黙的な推論を行っているからである．この例では，暗黙的推論規則として，「記述されていない遷移は一切起こらない」，「共有変数の値は更新されたことが示されない限り保持され続ける」を用いている．したがって，仕様を時間論理式で誤りなく，かつ理解しやすく記述するためには，記述者がこのような暗黙的推論規則から推論される諸性質にまで記述の注意を払うことなく，(1.1)式のようなシステムの本質のみを記述できるような枠組が必要である．

本論文では，時間論理を用いて誤りなく自然な仕様を記述するために，このような暗黙的推論の行える非単調時間論理について述べる．非単調論理については，今までに McDermott や Reiter, Gabbay らがその形式化を行ったが[1)~3)]，これらはいずれも時間論理ではなく，通常の命題論理や述語論理（直観主義論理も含む）をベースとしている．本論文の論理体系では，動的システムの仕様記述に使用するという目的のもとに，従来の◇，□，○，until といった temporal operator の機能を損なうことなく，非単調推論が可能であるように拡張されている．仕様には，記述者によって記述された論理式以外に暗黙的推論を行うための推論規則生成スキーマによって自動的に生成された論理式が追加され，諸性質が検証できる．第2章では，非単調時間論理の構文と意味を形式的に定義する．第3章では，McDermott の不動点を用いる手法によって本論理体系の定理を定義し，Soundness と Completeness を証明する．第4章で Tableau 法に基づく証明手続きを与える．第5章で，本論理体系の有用性を実際の仕様記述例で議論する．

2. 論理体系

2.1 仕様記述のための非単調時間論理

非単調論理の形式化は，これまでにも行われてきたが，McDermott[1)]，Reiter[2)] らの手法は，古典論理の体系に証明可能性を表す様相オペレータ M を導入し，論理式 p の否定が証明されないとき，Mp が証明されるとした．この M は直観的には「矛盾しない」ことを表す．しかし，このようなオペレータの導入は，論理系に特異な性質，{Mp, ～p} は矛盾しない（{Mp, ～p} ⊬ False）を与える．これは，～p が証明できるとき，必ずしも Mp が偽となることが証明されないことによる．～Mp の直観的な意味は，「～p が証明可能」ということであるが，～p は上例では実際に証明可能なため，人間の常識的な直観とは証明可能性が一致していない．人間の直観と一致していない論理を仕様記述に用いるのは，仕様を誤りなく記述させる上で大きな問題となる．Gabbay[3)] の形式化は，直観主義論理体系に同様なオペレータを導入し，上記の問題を解決したが，直観主義論理を形式的仕様記述にどのように用いるかは今後の大きな課題となっている．

しかし，この問題は記述されたシステムの性質を証明するのに ～Mp の形の式を証明しなければならないかどうかによる．ユーザの記述も，証明すべき論理式も ～Mp の形となっていなければ問題はない．実際の多くの非単調推論においては，Reiter の Normal Default[2)] のように「A が真でかつ p が矛盾しなければ，p である」，つまり Mp の形の式が証明され，「A ∧Mp→p」の形の推論規則によって推論が進んでいく．本論文で対象とする暗黙的推論は，実際の動的システムの仕様記述経験に基づいて，「記述されていない遷移は一切起こらない」，「共有変数の値は更新されたことが示されない限り保持され続ける」と限定した[7)]．これらの推論は，いずれも Normal Default の形をしている．つまり，ある時刻において，「M（遷移 s が起こらない）→遷移 s が起こらない」，「M（var の値が変わっていない）→var の値が変わっていない」の形式となっている．以上のことから，～Mp の形式の論理式が証明されなくても，Mp の形の論理式さえ証明されれば，これらの暗黙的推論が行える．つまり，McDermott らの形式化の手法がそのまま使用できることになる．

仕様記述者には，システムの本質のみに注意を払いながら記述できるように，Mを含む式を使用することは許さないものとし，記述された仕様とあらかじめ用意されたスキーマから生成される Normal Default 形式の論理式と合わせてシステムの諸性質を推論するものとする．このシステムの諸性質（証明されるべき式）は，Mを含まない論理式によって表される．この制約は自然である．これにより，～Mp の形の式を証明しなければ，ある性質が証明できないということが避けられる．

本論文での論理体系は，通常の until を含む一階の

線形時間命題論理に，各 temporal operator の機能を損なうことなく，ある時刻における論理式の無矛盾性を表すオペレータMを McDermott の手法に従って導入したものである．この論理体系を以下 NMT と呼び，通常の単調な命題時間論理の体系を MT と呼ぶ．

2.2 NMT の構文

a）記　　号

論理式を構成する最も原子的な要素としての記号は，命題記号集合 LPROP の元である．この命題記号は，状態（時刻）によってその解釈（真偽値）が変わる．

b）論理式の集合 Formula

Formula（論理式の集合）は，以下の条件を満たす最小集合である．

1) LPROP⊆Formula
2) $p, q \in$Formula ならば
   $\sim p, p \rightarrow q \in$Formula
3) $p, q \in$Formula ならば
   $\Box p, \bigcirc p, p$ until $q \in$Formula
4) $p, q \in$Formula ならば $pMq \in$Formula

便宜上の記法として，$\wedge$，$\vee$，$\longleftrightarrow$，$\Diamond$といったオペレータやオペレータの有効範囲を明確に示すための（ ）も通常の意味で定義し，使用する．さらに，後の章で使用するために2項オペレータ $\underset{\rightarrow}{M}$ を定義する．

$$p\underset{\rightarrow}{M}q=(pM((p \text{ until } q) \wedge (\Diamond q))$$

（ ）による指定がない場合，第1章で定義した⇒も含めて，これらオペレータの優先順位は，

$$\{\sim, \bigcirc, \Diamond, \Box\} > \{M, \text{until}\} > \{\wedge, \vee\} > \{\rightarrow\} > \{\longleftrightarrow\} > \{\Rightarrow, \underset{\rightarrow}{M}\}$$

とする．

2.3 NMT の意味

論理式を解釈する，つまり Formula の元に 0（偽を表す）か 1（真）を割り当てるために，状態の無限列 σ と論理式の集合 $\boldsymbol{A}$ (⊂Formula) を用意する．各 σ_i を状態とし，$\sigma=\sigma_0\ \sigma_1\ \sigma_2, \cdots, \sigma_i, \cdots$とする．各 σ_i は，

$$\sigma_i : \text{LPROP} \cup \{pMq \mid p, q \in \text{Formula}\} \rightarrow \{0, 1\}$$

つまり，LPROP の元と pMq の形をした論理式に 0 か 1 を割り当てる関数である．また，σ_{+i} を σ_i を先頭要素とする σ の部分列，すなわち $\sigma_{+i}=\sigma_i\ \sigma_{i+1}\ \sigma_{i+2}, \cdots$ とする（$\sigma_0=\sigma$）．$\boldsymbol{A}$ は，オペレータ M を含む式を解釈するために使用するもので，直観的には，前提として真であると設定される論理式の集合である．

論理式集合 $\boldsymbol{A}$，状態列 σ における論理式 p の解釈を $V^{\boldsymbol{A}}\sigma(p)$ と記述する．論理式の集合 S に対し $V^{\boldsymbol{A}}\sigma(S)=1$ であるとは，$p \in S$ であるすべての論理式 p について $V^{\boldsymbol{A}}\sigma(p)=1$ であると定義する．$V^{\boldsymbol{A}}\sigma$ は以下のように定義される．

1) $P \in$LPROP のとき $V^{\boldsymbol{A}}\sigma(P)=\sigma_0(P)$
2) $V^{\boldsymbol{A}}\sigma(\sim p)=1$ iff $V^{\boldsymbol{A}}\sigma(p)=0$
   $V^{\boldsymbol{A}}\sigma(p \rightarrow q)=1$ iff
   $V^{\boldsymbol{A}}\sigma(p)=0$ または $V^{\boldsymbol{A}}\sigma(q)=1$
3) $V^{\boldsymbol{A}}\sigma(\Box p)=1$ iff
   $j \geqq 0$ であるすべての j について
   $V^{\boldsymbol{A}}\sigma_{+j}(p)=1$.
   $V^{\boldsymbol{A}}\sigma(\bigcirc p)=1$ iff $V^{\boldsymbol{A}}\sigma_{+1}(p)=1$
   $V^{\boldsymbol{A}}\sigma(p$ until $q)$ iff
   1) $k \geqq 0$ であるすべての k について
      $V^{\boldsymbol{A}}\sigma_{+k}(p)=1$ または，
   2) ある $k \geqq 0$ に対し，
      $V^{\boldsymbol{A}}\sigma_{+k}(q)=1$ であり，
      $0 \leqq j < k$ なるすべての j について
      $V^{\boldsymbol{A}}\sigma_{+j}(p)=1$
4) a）$V^{\boldsymbol{A}}\sigma'(\boldsymbol{A} \cup \{p\})=1$ で
   $V^{\boldsymbol{A}}\sigma'(q)=1$ なる σ' が存在するとき，
   $V^{\boldsymbol{A}}\sigma(pMq)=1$
   b）そうでないとき，
   $V^{\boldsymbol{A}}\sigma(pMq)=\sigma_0(pMq)$

pMq の直観的な意味は p が真であるようなある時刻列 σ のもとで，$\boldsymbol{A}$ と MT の公理系に対して q が矛盾しなければ，pMq は真 ($V^{\boldsymbol{A}}\sigma(pMq)=1$) であるという意味である．

pMq は，McDermott の体系で $M(p \wedge q)$ と本質的に同じであるが，p に時刻を指定するという役割を与え，読解性を強調するために2項オペレータとした．

3. NMT の公理系と定理

3.1 単調命題時間論理 MT の公理と推論規則

まず，NMT の証明可能性を定義するために，通常の単調な命題時間論理 MT の公理と推論規則を以下に示す．

[公理]

命題論理の公理

$\vdash \bigcirc(p \rightarrow q) \rightarrow (\bigcirc p \rightarrow \bigcirc q)$
$\vdash \bigcirc \sim p \longleftrightarrow \sim \bigcirc p$
$\vdash \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
$\vdash \Box p \rightarrow (p \wedge \bigcirc \Box p)$

$\vdash \Box(p \rightarrow \bigcirc p) \rightarrow (p \rightarrow \Box p)$

$\vdash \Box p \rightarrow p$ until q

$\vdash p$ until q

$\longleftrightarrow q \vee (p \wedge \bigcirc(p \text{ until } q))$

[推論規則]

$\vdash p$, $\vdash p \rightarrow q$ より $\vdash q$

$\vdash p$ より $\vdash \Box p$

この公理系は，文献 6) で示されているように，Sound でありかつ Complete である．ここで，論理式の集合 S について，$\mathrm{Th}(S)=\{p \mid S \vdash p\}$ と定義する．

3.2 非単調時間論理 NMT の定理

NMT において，論理式集合 $\boldsymbol{A}$ から証明可能な論理式の集合 THEOREM ($\boldsymbol{A}$) (以下，定理という) を定義するために，オペレータ TH を定義する．TH は，論理式 p が真となる状態列で矛盾しない論理式 q について，pMq を前提として追加し，3.1 の公理系で証明できる論理式の集合を求める演算である．以下は，その厳密な定義である．

$\mathrm{TH}(S)=\mathrm{Th}(\boldsymbol{A} \cup \mathrm{HYP}(S))$

$\mathrm{HYP}(S)=\{pMq \mid p \rightarrow \sim q \notin S\}$

この TH を用いて，THEOREM($\boldsymbol{A}$) を以下のように定義する．

a) FP=TH (FP) となる最小不動点 FP が存在するとき

THEOREM ($\boldsymbol{A}$)$=\cap\{\mathrm{FP} \mid \mathrm{FP}=\mathrm{TH}(\mathrm{FP})\}$

b) FP=TH (FP) となる最小不動点 FP が存在しないとき

THEOREM ($\boldsymbol{A}$)=Formula

FP が存在しないときは NMT の論理式すべてを，複数個存在するときはそれらの共通集合を NMT における定理の集合と見なす．

$p \in$ THEOREM ($\boldsymbol{A}$) のとき $\boldsymbol{A} \vdash_{\mathrm{n}} p$ と書き，p が NMT で $\boldsymbol{A}$ から証明されるという．特に混乱を生じない限り，$\boldsymbol{A}$ を略すことにする．同様に，「MT で証明される」と混乱を生じない限り，単に「証明される」と略すことにする．以下に THEOREM ($\boldsymbol{A}$) の例を上げる．

[例 1]

$\boldsymbol{A}=$

$\{\Box(\mathrm{atL} \wedge \text{Flag-is-1} \Rightarrow \mathrm{atM}),$ (3.1)

$\Box((\mathrm{atL} \wedge \text{Flag-is-1} \Rightarrow \mathrm{atM})$

$\rightarrow((\mathrm{atL} \wedge \text{Flag-is-1} \underrightarrow{\mathrm{M}} \mathrm{atM} \wedge \text{Flag-is-1})$

$\rightarrow(\mathrm{atL} \wedge \text{Flag-is-1} \Rightarrow \mathrm{atM} \wedge \text{Flag-is-1})))$

$\}$ (3.2)

仕様として見ると，(3.1)式はユーザが記述した式で，statement L を実行し，共有変数 Flag の値が 1 であるなら，次に statement M の実行を行うという意味である．(3.2)式が，「共有変数の値は更新されたことが示されない限り保持され続ける」という暗黙的推論を行うために，第 5 章で述べる推論規則生成スキーマより生成された論理式の一部である．この式は atL∧Flag-is-1 から atM に遷移するとき，statement M において (atM), Flag-is-1 としても矛盾が生じなければ，Mにおいても Flag の値は 1 のまま保持されることを述べている．

(3.1)式のみでは，明らかに

$\boldsymbol{A} \nvdash_{\mathrm{n}} (\mathrm{atL} \wedge \text{Flag-is-1}) \rightarrow$

$\sim((\mathrm{atL} \wedge \text{Flag-is-1}) \text{ until } (\mathrm{atM} \wedge \text{Flag-is-1})$

$\wedge \Diamond(\mathrm{atM} \wedge \text{Flag-is-1}))$

であるから，

$\boldsymbol{A} \vdash_{\mathrm{n}} (\mathrm{atL} \wedge \text{Flag-is-1})\mathrm{M}$

$((\mathrm{atL} \wedge \text{Flag-is-1}) \text{ until } (\mathrm{atM} \wedge \text{Flag-is-1})$

$\wedge \Diamond(\mathrm{atM} \wedge \text{Flag-is-1}))$

である．この結果と (3.2)式，MT の推論規則によって，

$\boldsymbol{A} \vdash_{\mathrm{n}} \mathrm{atL} \wedge \text{Flag-is-1} \Rightarrow \mathrm{atM} \wedge \text{Flag-is-1}$

つまり，statement M に到達するまでに，Flag が更新されたことが証明できないため，M においても Flag は値 1 を保持していることが証明されたことになる．この性質の形式的な証明は 4 章で行う．

3.3 Soundness と Completeness

2.2 節で定義した意味と，前節で定義した証明可能性の関係に関する 2 つの重要な定理 Soundness と Completeness について述べる．これらの定理の証明にあたっては，3.1 節にあげた MT の公理系は，Sound であり Complete であるという事実を用いる．

[定理 3.1 Soundness]

TH の最小不動点が存在し，$\boldsymbol{A} \vdash_{\mathrm{n}} p$ ならば，$\mathrm{V}^{\mathrm{A}}\sigma(\boldsymbol{A})=1$ を満たすどんな状態列 σ についても $\mathrm{V}^{\mathrm{A}}\sigma(p)=1$ である．

[証明]

背理法で証明する．

$\mathrm{V}^{\mathrm{A}}\sigma(\boldsymbol{A})=1$ である σ について，$\mathrm{V}^{\mathrm{A}}\sigma(\mathrm{FP})=0$ となる不動点 FP が存在すると仮定する．

$\mathrm{V}^{\mathrm{A}}\sigma(\mathrm{FP})=\mathrm{V}^{\mathrm{A}}\sigma(\mathrm{Th}(\boldsymbol{A} \cup \mathrm{HYP}(\mathrm{FP}))=0$ (3.3)

MT の定理導出オペレータ Th は，Sound であるから，任意の論理式集合 $\boldsymbol{S}$ について，$\mathrm{V}^{\mathrm{A}}\sigma'(\boldsymbol{S})=1$ を

満たすすべての σ' について

$V^{\wedge}\sigma'(\mathrm{Th}(\boldsymbol{S}))=1$

である.

したがって, (3.3)式が成立するためには,

$V^{\wedge}\sigma(\boldsymbol{A}\cup\mathrm{HYP(FP)})=0$

でなければならない.

$V^{\wedge}\sigma(\boldsymbol{A})=1$ より, $V^{\wedge}\sigma(\mathrm{HYP(FP)})=0$ である. つまり, $p\in\mathrm{HYP(FP)}$ で, $V^{\wedge}\sigma(p)=0$ なる論理式 p が存在することになる. この p は, HYP の定義より $q\mathrm{M}r$ の形をしており,

$$q\rightarrow\sim r\notin\mathrm{FP} \tag{3.4}$$

$V^{\wedge}\sigma(q\mathrm{M}r)=0$ より,

$V^{\wedge}\sigma'(\boldsymbol{A})=1$ を満たすすべての σ' について

$V^{\wedge}\sigma'(q)=0$ または $V^{\wedge}\sigma'(r)=0$.

MT は Complete であるから, $q\rightarrow\sim r\in\mathrm{Th}(\boldsymbol{A})$. Th の単調性より,

$q\rightarrow\sim r\in\mathrm{Th}(\boldsymbol{A})\subseteq\mathrm{Th}(\boldsymbol{A}\cup\mathrm{HYP(FP)})=\mathrm{FP}$.

これは, (3.4)式に反する.

以上により, $V^{\wedge}\sigma(\boldsymbol{A})=1$ である σ について, 不動点 FP が存在すれば $V^{\wedge}\sigma(\mathrm{FP})=1$ となる. よって $V^{\wedge}\sigma(\boldsymbol{A})=1$ であるすべての σ について, $V^{\wedge}\sigma(\cap\mathrm{FP})=1$ となる.

[定理 3.2 Completeness]

TH の不動点が存在し, 論理式集合 $\boldsymbol{A}$ を満たすどんな σ についても,

$V^{\wedge}\sigma(p)=1$ であるならば, $\boldsymbol{A}\vdash_{n}p$ である.

この定理の証明は, MT の Completeness が成立するため, 文献 1) と同じになるので省略する.

4. NMT の Tableau 法に基づく証明手続き

時間命題論理の Tableau 法による証明手続きは, Manna らによって与えられ, その手続きが正しいこともすでに証明されている[5]. また非単調命題論理の証明手続きも, Tableau 法を用いる手法が提案されている[1]. NMT の証明手続きは, これらの手法をもとに, 時間命題論理の証明手続きに非単調オペレータ Mに関する処理を追加したものである.

処理は, 各ノードが論理式の集合からなるグラフを, ノードに含まれる論理式の構文に基づいて, 構成していく. グラフの root ノードは, 論理式集合 $\boldsymbol{A}$ と証明したい式 p の否定 $\sim p$ からなる. グラフが完成すると, ラベル付け規則に基づいて, グラフ中の各ノードに closed か open のラベルを振っていく. closed は, そのノードの含まれている論理式の集合が充足不能であることを, open は充足可能であることを表す. すべてのラベル付けに対して, root ノードに closed が振られた場合が, p が論理式集合 $\boldsymbol{A}$ から証明される, すなわち $\boldsymbol{A}\vdash_{n}p$ であることを示している.

4.1 グラフ (Tableau) の構成法

$\boldsymbol{A}\vdash_{n}p$ かどうかを判定するグラフ G は, $<N, T, R, s>$ の 4 つの組によって表される. ここで N は, ノードの集合で, ノードは論理式の集合である. T, R はノードからノードの集合への関数で, ノード n の子ノードの集合を $T[n]$, $R[n]$ で表す. 非単調オペレータMを扱うために, 子ノードの種類を T で表されるものと, R とに分割した. s は, このグラフの root で, $s\in N$ である. 子ノードを持たないノードを終端ノードと呼ぶ. また, 論理式 p と $\sim p$ を同時に含むノードを拡張不可能なノードと呼び, そうでないノードを拡張可能なノードと呼ぶことにする. T, R, s は, 以下の規則に従って構成される.

[グラフの構成法]

1) $s=\boldsymbol{A}\cup\{\sim p\}$.

2) 異なる論理式の集合を持つノードが作られなくなるまで, 以下の 2-1), 2-2), 2-3) の規則を適用し, グラフを作成, すなわち T, R を作成していく.

2-1) 拡張可能な終端ノード n から 1 つの論理式 q を選び, q の構文に応じて, **表 1** より F_1, F_2 を求め, 以下のように T 型の子ノードを拡張していく.

a) q が α 型の論理式のとき

$T[n]=\{t\}$ とし,

$t=n-\{q\}\cup\{F_1\}$

b) q が β 型の論理式のとき

表 1 判定グラフの子ノードの構成法
Table 1 Construction of descendant nodes in decision graph.

	論理式 q	F_1	F_2
α 型	$\sim\sim r$	r	
	$\sim(r\rightarrow w)$	$r,\ \sim w$	
	$r\wedge w$	$r,\ w$	
	$\sim(r\vee w)$	$\sim r,\ \sim w$	
	$\sim\bigcirc r$	$\bigcirc\sim r$	
	$\Box r$	$r,\ \bigcirc\Box r$	
	$\sim\Diamond r$	$\sim r,\ \bigcirc\sim\Diamond r$	
β 型	$r\rightarrow w$	$\sim r$	w
	$\sim(r\wedge w)$	$\sim r$	$\sim w$
	$r\vee w$	r	w
	$\sim\Box r$	$\sim r$	$\bigcirc\sim\Box r$
	r until w	w	$r,\ \bigcirc(r$ until $w)$
	$\sim(r$ until $w)$	$\sim r,\ \sim w$	$\sim w,\ \bigcirc\sim(r$ until $w)$

$T[n]=\{t,u\}$ とし,
$t=n-\{q\}\cup\{F_1\}$,
$u=n-\{q\}\cup\{F_2\}$

例えば, q が r until w のときは, $T[n]=\{t,u\}$, $t=n-\{q\}\cup\{w\}$, $u=n-\{q\}\cup\{r, \bigcirc (r \text{ until } w)\}$ となる.

上記の規則がどの拡張可能な終端ノードにも適用できなくなれば, 2-2) の適用を行う.

2-2) 規則 2-1) が適用できなくなった拡張可能な終端ノード n について

$T[n]=\{t\}$ とし, $t=\{q|\bigcirc q\in n\}$

この規則 2-2) が適用されるノード n を状態ノードと呼ぶことにする. この規則によって, 状態ノードに T 型の子ノードを作成し, 2-1) へもどる. 上記の規則が適用できない場合は, 2-3) を適用する.

2-3) 規則 2-1), 2-2) が適用できなくなったとき, 状態ノード n から $\sim(q\mathrm{M}r)$ の形をした過去選択されていない1つの論理式を選び,

$R[n]=\{t\}$, $t=\boldsymbol{A}\cup\{q, r\}$

によって, R 型の子ノード t を作り, 2-1) へ戻る.

n が $\sim(q\mathrm{M}r)$ の形の論理式を含んでいなければ, この規則は n には適用できない.

3) 以上の規則 2-1), 2-2), 2-3) のどれも適用不可能になったとき, グラフの作成は終了する.

4.2 グラフのラベル付け

以上のように作成されたグラフ G に対して, G の各ノードに open, closed の2種類のラベルを以下のラベル付け条件を満たすように振っていく. ここで, $m\in T[n]$ のとき, nTm と書き, T^* でその2項関係 T の推移閉包を表すものとする.

[ラベル付け条件]

1) G のノード n が論理式 p と $\sim p$ を同時に含むときは, n は closed である.

2) G のノード n が以下のような形の論理式を含むとき, 各々以下の条件を満たしていなければ, n は closed である.

2-1) n が $\sim\Box p$ を含んでいるとき, nT^*m かつ $\sim p\in m$ で, open であるノード m が存在する.

2-2) n が $\Diamond p$ を含んでいるとき, nT^*m かつ $p\in m$ で, open であるノード m が存在する.

2-3) n が $\sim(p \text{ until } q)$ を含んでいるとき, nT^*m かつ $\sim p\in m$ で, open であるノード m が存在する.

3) $T[n]$ 中のすべてのノードが closed ならば, G のノード n は closed である.

4) G のノード n が R 型の子ノードを持っていないとき ($R[n]=\phi$), n が closed とラベル付けされないならば, n は open である.

5) G のノード n について, $m\in R[n]$ でノード m が open ならば, n は closed である.

1)〜4) のラベル付け条件は, MT の tableau 法における条件である. 以上のラベル付け規則は, G 中のすべてのノードが open か closed であるかを唯一決定できるほど強力ではない. open に振っても, closed に振ってもどちらでもラベル付け条件を満たすようなノードも存在する. この場合は, 複数個のラベル付けが可能であるとする.

4.3 証明手続き

前節の手法によって得られたグラフ G をもとに, $\boldsymbol{A}\vdash_n p$ かどうか判定する.

[定理 4.1]

$\boldsymbol{A}\vdash_n p$ を判定するグラフ $G=<N, T, R, s>$ のすべての可能なラベル付けにおいて root ノード s が closed にラベル付けされていることと, $\boldsymbol{A}\vdash_n p$ とは同値である.

この判定手続きの正当性を示すために, 以下の2つの補題を証明する. ここで, MT の tableau 法による証明手続きは 5) で示されているように, 正当であるという事実を用いる.

[補題 4.2]

ある最小不動点 FP=TH (FP)=Th($\boldsymbol{A}\cup$HYP(FP)) について $p\in$FP ならば, $\boldsymbol{A}\vdash_n p$ を判定するグラフ $G=<N, T, R, s>$ において root ノード s が closed と振られるラベル付けが存在する.

[証明]

FP より G のノードへのラベル付けを作る. $n=\{p_1, \cdots, p_n, q\}$ なるノード n について,

a) $p_1\wedge, \cdots, \wedge p_n\rightarrow\sim q\in$FP ならば,
n は closed

b) そうでなければ, n は open

のラベル付けを行う. このラベル付けが 4.3 節のラベル付け条件を満たしていることを証明する. 条件 1)〜4) は, MT の証明手続きが正当であることより, 満足されているのは明らかである. したがって, ラベル付け条件の 5) が満たされることを示せばよい.

R 型の子ノード $m=\boldsymbol{A}\cup\{p, q\}$ を持つノード n について考える. m が open に振られているとすると, $p\rightarrow\sim q\notin$FP. これより

$p\mathrm{M}q\in$FP=Th ($\boldsymbol{A}\cup$HYP (FP))

となる.

したがって, n 中の $\sim(pMq)$ 以外のすべての論理式を $p_1, \cdots, p_n$ とすると,

$p_1 \wedge, \cdots, \wedge p_n \rightarrow pMq \in FP$.

$n=\{p_1, \cdots, p_n, \sim(pMq)\}$ は closed とラベル付けされていることになる. これは, ラベル付け条件 5) を満たしている.

[補題 4.3]

$A \vdash_n p$ を判定するグラフ $G=<N, T, R, s>$ において root ノード s に closed と振るラベル付けが存在するならば, $p \in FP$ なる最小不動点 $FP=TH(FP)=Th(A \cup HYP(FP))$ が存在する.

[証明]

与えられたラベル付けより, 不動点を作ることによって証明する.

$H_0=\{(p_1 \wedge \cdots \wedge p_n)\ Mq\}$

状態ノード $n=\{p_1, \cdots, p_n, q\}$ が open}

論理式のペア $s_i=<p, q>$ を1つ選び (ただし, $i \neq j$ のとき $s_i \neq s_j$ である), 以下のようにして論理式の集合列 H_i, S_i $(i=0, 1, 2, \cdots)$ を作る. ただし, ペアの選択の順序は, q の構文の複雑さの順, つまり q が q' の部分式になっているとき q を含むペアのほうが q' を含むものよりも必ず先に選択されるとする.

$p \rightarrow \sim q \in S_i$ のとき $H_{i+1}=H_i$

$p \rightarrow \sim q \notin S_i$ のとき $H_{i+1}=H_i \cup \{pMq\}$

$S_{i+1}=Th(A \cup H_{(i+1)})$

$H=\cup_{i=0}^{\infty} H_i$, $S=\cup_{i=0}^{\infty} S_i$ とすると, $S_i \subseteq S_{i+1}$ より, $S=Th(A \cup H)$ である. よって $H=HYP(S)$ を証明すればよいことになる. これは, 文献 1) と同じ証明であるため, 省略する.

4.4 証明例

実際に証明を行った例を**図2**と**図3**に示す. この例は, 3.2節の例1である. 各図においては, 例えば A と $A \rightarrow B$ を含むノードから直接, A, B を含む T 型の子ノードを作り出すといったようにグラフの拡張過程を一部省略している. 図3は, 図2の～((atL∧Flag-is-1) M (atL∧Flag-is-1) until (atM∧Flag-is-1)∧◇(atM∧Flag-is-1))) を含むノードから R 型の子ノードとして拡張されたグラフである. この図では, R 型の子ノードを含まず, すべてのノードが open と唯一にラベル付けされるパスを示せば, 親ノードが open と唯一にラベル付けされるため, そのようなパスの1つのみを示した.

5. 仕様記述と仕様の検証への応用

本論文では, 動的システムの内部状態をプログラムの実行進行状態と共有変数の値によって表し, その状態遷移を仕様として記述された時間論理式を真にするモデルの状態列として規定する. プログラム中の各 statement には, 各々異なるラベルが振られている. ある statement のラベルを L とすると, その statement が実行されている状態を表す基本命題を atL∈LPROP とし, この命題によってプログラムの実行進行状態を表すものとする. この基本命題の集合を AT と書く. システムに n 個の共有変数 $var_1, \cdots, var_n$ があるとすると, var_i の値が e_j であることを表す基本命題 var_i is-$e_j \in$LPROP を導入する.

var_i に関するこの基本命題の集合を VAR_i とする. システムの1つの内部状態は,

atL ∧ v_1 ∧ v_2 ∧, ⋯, ∧ v_n

ただし, atL∈AT, $v_i \in VAR_i$ $(1 \leqq i \leqq n)$ の形式の論理式で表す. このような内部状態を表現する論理式の集合を STAT とする.

システムの仕様は, 論理式の集合 ($\subseteq$Formula) の2つ組 $<U, D>$ で表す. システムの仕様において成立する性質とは, 論理式の集合 THEOREM $(U \cup D)$ のことである. U はオペレータ M を含んでいない論理式の集合で, 仕様記述者自身がシステムの仕様として記述した論理式の集合である. この部分が従来の仕様記述に該当する部分である. D は, 暗黙的推論を行うための論理式の集合で, あらかじめ定められた生成スキーマと U の論理式の構文により自動的に生成される. D の生成スキーマは以下のとおりである.

$s_0 \in STAT$, $s_0=atL \wedge v_1 \wedge, \cdots, \wedge v_n$,

$1 \leqq i \leqq n$, $atM \in \overline{AT}_L=AT-\{atL\}$ とする.

1) 状態遷移に関するもの

a) $\Box(s_0 M(\Box \sim atM) \rightarrow (s_0 \rightarrow \Box \sim atM))$

b) $\Box(s_0 M(\Box s_0) \rightarrow (s_0 \rightarrow \Box s_0))$

2) 共有変数に関するもの $(1 \leqq i \leqq n)$

a) $\Box((s_0 \Rightarrow atM) \rightarrow ((s_0 \underrightarrow{M} (atM \wedge v_i)) \rightarrow (s_0 \Rightarrow (atM \wedge v_i))))$

b) $\Box((s_0 \rightarrow (atL\ until\ atM)) \rightarrow (s_0\ M\ ((atL \wedge v_i)\ until\ atM)) \rightarrow (s_0 \rightarrow ((atL \wedge v_i)\ until\ atM))))$

3) $\Box(atL \longleftrightarrow \sim(\bigvee_{atK \in \overline{AT}_L} atK))$

4) $v_{ij} \in VAR_i$, $\overline{VAR}_{ij}=VAR_i-\{v_{ij}\}$

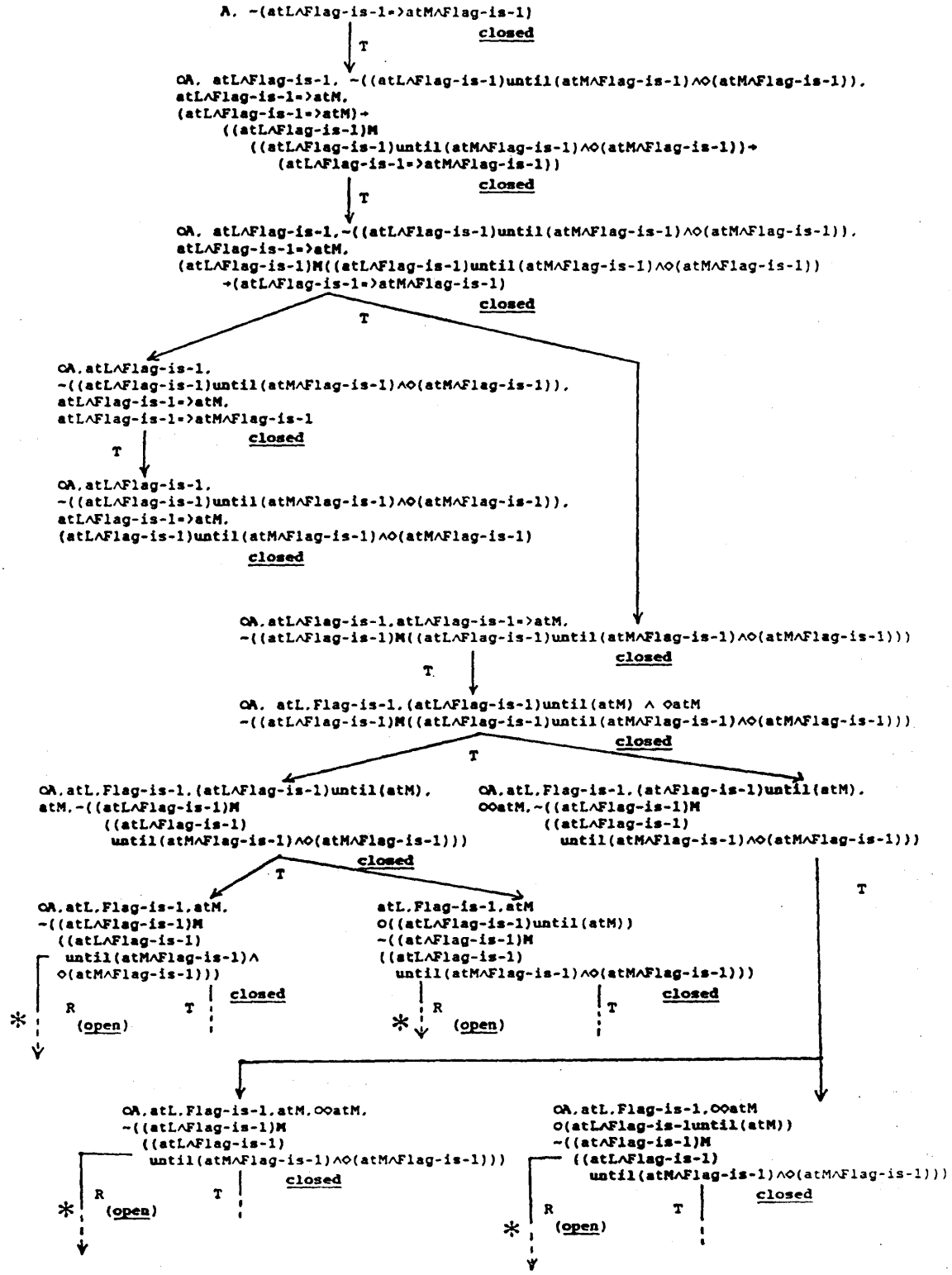

図 2 判定グラフ (1)
Fig. 2 Example No. 1 of tableau construction.

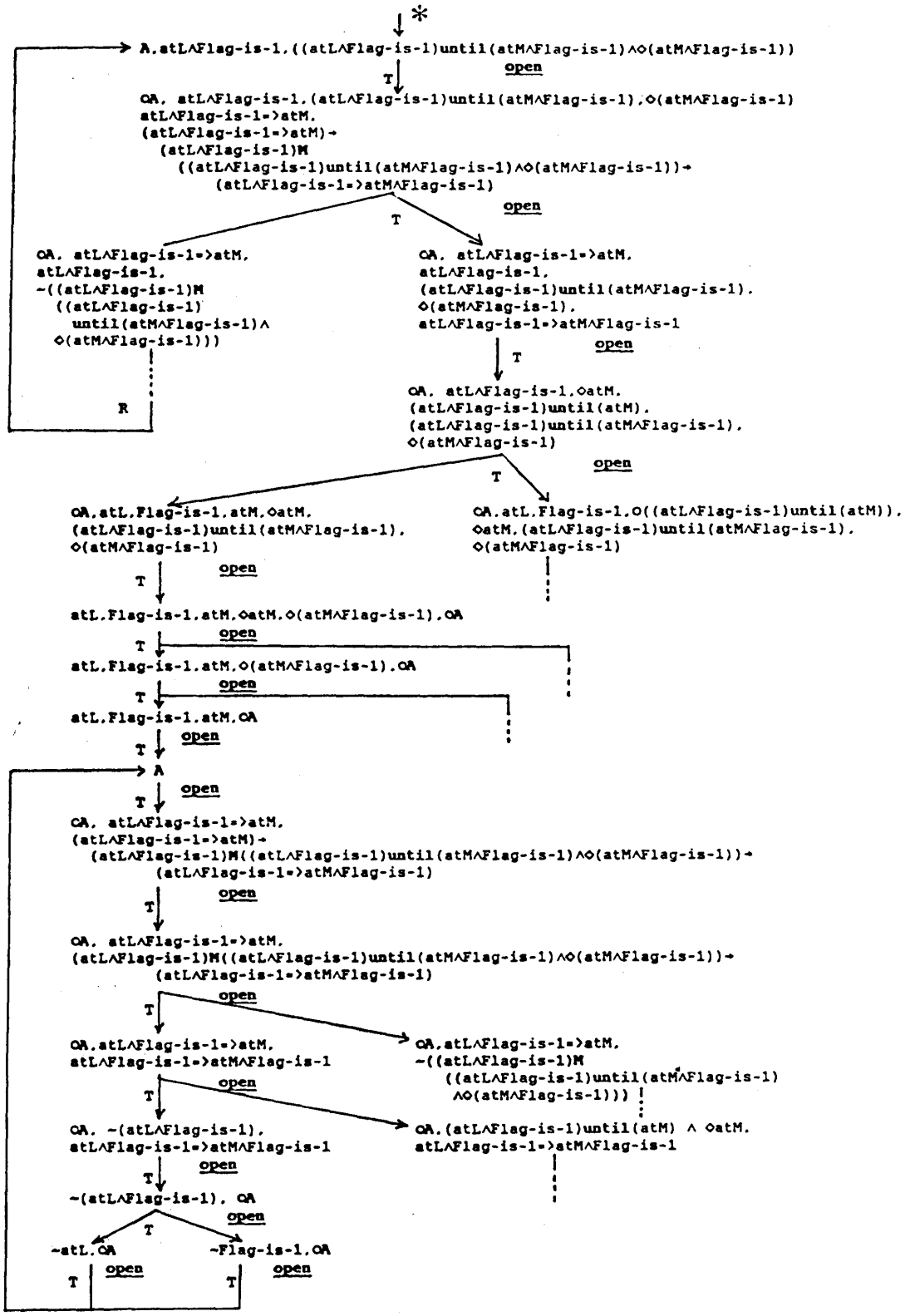

図 3 判定グラフ (2)
Fig. 3 Example No. 2 of tableau construction.

$$\Box\,(v_{ij} \longleftrightarrow \sim(\bigvee_{v_{ik}\in VAR_{ij}} v_{ik}))$$

1-a) は到達可能性が証明されない statement には到達できない，b) は，3)，4) より $s_0 \longleftrightarrow (\bigvee_{s_i\in STAT-\{s_0\}} s_i)$ であることが証明されるため，他のどの状態にも到達することができなければ，その状態に最終状態として停留し続けるという意味である．2) は，変数の値の保存に関する暗黙的推論を行うためのもので，a) は statement M において var_i の値が状態 s_0 のときの値と等しいとしても矛盾がなければ，そのまま変化せずに保持されると推論できることを述べている．b) は statement L を実行中は，var_i の値が保持されることを述べている．これらのスキーマは，値の保持される区間の終端状態の情報，つまりどの状態まで値が保持されるかという情報を含ませる必要があるため，until オペレータを使用したものとなっている．3) は必ずどれか唯一の statement が実行されているということを，4) は変数の値は唯一であることを表した式である．これらの式は，STAT の元の任意の組み合わせによって生成され，そのため実際にシステムの諸性質を推論する際には不要なものもある．

これらのスキーマから生成される論理式は，TH について複数個の最小不動点を作り出す可能性がある．例えば，

$U = \{s_0 \rightarrow \Diamond\ (\mathrm{atL} \vee \mathrm{atM})\}$

とし，1-a) のスキーマから生成される論理式のみについて考えると，最小不動点は $s_0 \rightarrow \Box\sim$atL を含むが $s_0 \rightarrow \Box\sim$atM を含まないものと，その逆に $s_0 \rightarrow \Box\sim$atM を含むが $s_0 \rightarrow \Box\sim$atL を含まないものの2つ存在する．したがって，$s_0 \rightarrow (\Box\sim \mathrm{atL} \vee \Box\sim \mathrm{atM})$ が THEOREM $(U \cup D)$ の元となる．この意味は，s_0 から非決定的に statement L かMを実行することのみが示されていれば，矛盾がない限り実際にはどちらか片方のみを実行し，残りの片方は s_0 からは全く実行制御が移らないことを表している．

以下で実際の記述例について述べる．

[例2]

AT = {atL, atM, atN, atK},

VAR = {Flag-is-0, Flag-is-1},

U = {□(atL ∧ Flag-is-1 ⇒ atM),
□ (atM ⇒ atN ∧ Flag-is-0),
□ (atN ⇒ atK)}

を記述者が与えたシステムの仕様とする（この例は第1章の図1の例である）．このシステムの性質として，L, M, N, K の順で statement が実行され，その間 Flag は L, M で 1, N, K で 0 である，つまり

$U \cup D \vdash_n$ atL∧Flag-is-1 ⇒ atM∧Flag-is-1

$U \cup D \vdash_n$ atM∧Flag-is-1 ⇒ atN∧Flag-is-0

$U \cup D \vdash_n$ atN∧Flag-is-0 ⇒ atK∧Flag-is-0

$U \cup D \vdash_n$ atK∧Flag-is-0 ⇒ □(atK∧Flag-is-0)

であることを示す．以下，$U \cup D$ を省略し，$\vdash_n p$, $\vdash p$ を各々 $U \cup D \vdash_n p$, $U \cup D \vdash p$ の意味として用いる．

1)，2) の生成スキーマより生成される D の元は，STAT の元が8つあるため，1-a) によって 24 個，1-b) で8個，2-a) で 24 個，2-b) で24個，3) で4個，4) で2個ある．

1) atL∧Flag-is-1 ⇒ atM∧Flag-is-1 の証明及び atK∧Flag-is-0 ⇒ □(atK∧Flag-is-0) の証明

これらの推論は，各々 2-a)，1-b) より生成される D の論理式を用いて，例1と同様で行える．

2) atN∧Flag-is-0 ⇒ atK∧Flag-is-0 の証明

U 中の論理式より，

$\vdash_n$ atN → ◇atK (5.1)

$\vdash_n$ atN∧Flag-is-0 → atN until atK (5.2)

D の生成スキーマ 2-b) より，

□((atN∧Flag-is-0 → atN until atK) →
(((atN∧Flag-is-0)M
((atN∧Flag-is-0) until atK))
→ (atN∧Flag-is-0
→ ((atN∧Flag-is-0) until atK))))
∈D (5.3)

$\nvdash_n$ atN∧Flag-is-0 →
∼((atN∧Flag-is-0) until atK) であるから

$\vdash_n$ (atN∧Flag-is-0)M
((atN∧Flag-is-0) until atK) (5.4)

(5.2)，(5.4)式と MT の公理，推論規則より

$\vdash_n$ atN∧Flag-is-0
→ (atN∧Flag-is-0) until atK (5.5)

(5.1)，(5.5)式より

$\vdash_n$ atN∧Flag-is-0 ⇒ atK (5.6)

((atN∧Flag-is-0 $\underrightarrow{M}$ (atK∧Flag-is-0))
→ (atN∧Flag-is-0
⇒ (atK∧Flag-is-0)))∈D (5.7)

(5.6)，(5.7)式より，例1の推論と全く同様にして，

$\vdash_n$ atN∧Flag-is-0 ⇒ atK∧Flag-is-0

が得られる．

3) atM∧Flag-is-1 ⇒ atN∧Flag-is-0

statement M から statement N への遷移に関して

は，1)，2) の場合とは全く異なる．(5.6)式を証明したのと同様にして，

$$\vdash_n \mathrm{atM} \wedge \mathrm{Flag\text{-}is\text{-}1} \Rightarrow \mathrm{atN} \qquad (5.8)$$

が得られるが，

$$\vdash \Box\ (\mathrm{atM} \Rightarrow \mathrm{atN} \wedge \mathrm{Flag\text{-}is\text{-}0}) \qquad (5.9)$$

$$\vdash \Box\ \sim(\mathrm{atM} \wedge \mathrm{atN})$$

（生成スキーマ 3) によって得られる）

$$\vdash \Box\ (\mathrm{Flag\text{-}is\text{-}0} \longleftrightarrow \sim \mathrm{Flag\text{-}is\text{-}1})$$

（生成スキーマ 4) によって得られる）

と，MT の公理，推論規則によって，

$$\vdash \mathrm{atM} \rightarrow \sim((\mathrm{atM}\ \mathrm{until}\ (\mathrm{atN} \wedge \mathrm{Flag\text{-}is\text{-}1})) \wedge \Diamond(\mathrm{atN} \wedge \mathrm{Flag\text{-}is\text{-}1})) \qquad (5.10)$$

となり，このため，例1と同様な推論が行えず，$\mathrm{atM} \wedge \mathrm{Flag\text{-}is\text{-}1} \Rightarrow \mathrm{atN} \wedge \mathrm{Flag\text{-}is\text{-}1}$ が結論できない．したがって，結果として (5.8)，(5.9)式より

$$\vdash_n \mathrm{atM} \wedge \mathrm{Flag\text{-}is\text{-}1} \Rightarrow \mathrm{atN} \wedge \mathrm{Flag\text{-}is\text{-}0}$$

となるのみである．

6. むすび

本論文では，動的システムの仕様を記述するための時間論理において暗黙的推論が行えるように，従来の各種の temporal operator の性質を損なうことなく，非単調論理に拡張する方法について述べた．本手法では，動的システムをプログラムと共有変数とに分割して記述するという手法において，暗黙的推論を仕様記述経験より「記述されていない遷移は，一切起こらない」，「共有変数の値は更新されたことが示されない限り保持され続ける」の2つとし，これらの推論が行えるように，McDermott 流の解釈に基づいて無矛盾性オペレータを導入した．このような手法の実用化には，prover の入力となる論理式が増大するため，効率の良い prover が必要であるが，我々は，形式的仕様記述の本質的な問題点を prover のような自動化ツールの実現ではなく，記述者にいかに仕様を意図どおりに書かせるかということであると考えている．本手法は，この問題に対する1つのアプローチである．

今後の課題としては，暗黙的推論規則の種類に応じてより精練された非単調論理の形式化，さらには述語論理への拡張を行う必要があると思われる．また生成スキーマが上記の2つの推論を正しく，かつ十分に行えるかどうかは，記述者が意図したシステムを形式的仕様記述言語によって正しく十分に記述できているかどうかの問題と同じである．仕様記述経験を通して，その正当性を検証することとともに，それを行うための prover などのツールの開発も今後の課題である．

謝辞 日頃御指導頂く，本学榎本肇名誉教授，片山卓也教授，志村正道教授，米崎直樹助教授に感謝致します．また，本稿を注意深く読んでくださった蓬萊尚幸君に感謝します．

参考文献

1) McDermott, D. and Doyle, J.: Non-Monotonic Logic I, *Artif. Intell.*, Vol. 13, pp. 41-72 (1980).
2) Reiter, R.: A Logic for Default Reasoning, *Artif. Intell.*, Vol. 13, pp. 81-132 (1980).
3) Gabbay, D. M.: Intuitionistic Basis for Non-Monotonic Logic, *LNCS*, Vol. 138, pp. 260-273 (1982).
4) Manna, Z. and Pnueli, A.: *Verifications of Concurrent Programs: The Temporal Framework, Correctness Problem in Computer Science*, pp. 215-273, Academic Press, London (1981).
5) Manna, Z. and Wolper, P.: Synthesis of Communicating Processes from Temporal Logic Specifications, *ACM Trans. Prog. Lang. Syst.*, Vol. 6, No. 1, pp. 68-93 (1984).
6) Gabbay, D., Pnueli, A., Shelah, S. and Stavi, J.: On the Temporal Analysis of Fairness, *Proc. of 7th ACM Symp. on Principles of Programming Languages*, pp. 163-173 (1980).
7) 米崎，佐伯，荒俣，西：TELL/NSL における動的記述の検証，第 30 回情報処理学会全国大会講演論文集，pp. 497-498 (1985).

（昭和61年1月30日受付）
（昭和62年4月15日採録）

佐伯 元司（正会員）

昭和31年生．昭和53年東京工業大学工学部電気電子工学科卒業．昭和58年同大学院情報工学専攻博士課程修了．工学博士．同年より東京工業大学工学部情報工学科助手．パターン認識，マンマシン・システム，ソフトウェア工学などの研究に従事．