階層的グラフ構造の記述と検証のための様相論理* †

Modal Logics for Description and Verification of Hierarchical Graph Structures

田辺 良則 ‡,§
Yoshinori TANABE
tanabe.yoshinori@aist.go.jp

萩谷 昌己 §,¶
Masami HAGIYA
hagiya@is.s.u-tokyo.ac.jp

‡: 産業技術総合研究所 システム検証研究センター
Research Center for Verification and Semantics, National Institute of Advanced Industrial Science and Technology (AIST)
§: 東京大学大学院情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo
¶: NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

著者らは，様相論理式を用いてグラフ書換系の性質を検証する枠組みを提案しているが，対象のグラフが階層構造を持っているときには，通常の様相論理では系の性質をうまく記述することができない場合がある．このために，様相論理の拡張で，その解釈であるクリプキ構造が階層構造を持つものがのぞまれる．本研究では，このような拡張を定義し，充足可能性判定手続き，書換に対する最弱前条件など，検証に際して必要となる基本的な手続きを構成する．

1 はじめに

我々は，従来研究 [3, 8] において，グラフ構造とみなせる対象に関する検証問題に様相論理を用いる手法を提案してきた．適用してきた対象には，プログラムヒープ (ポインタによって指すという関係をグラフのエッジと見る) やセルオートマトン (隣接関係をグラフのエッジと見る) などがある．これらの対象を，グラフのエッジを遷移関係とするクリプキ構造によってモデル化することを通して，様相論理式によるシステムの性質の記述を行うことで，充足可能性判定を用いたこれらの性質の検証が可能となった．

これらの対象は，単一のグラフとみなすことができた．しかし，より複雑なシステムは，単純にグラフとみなすことが困難である場合がある．システムは，低位の部品の組み合わせで高位の部品が作られ，さらにそれらを組み合わせてより高位の単位となるという構成をとることが多い．これら各段階の組み合わせをグラフとみるのであれば，全体のシステムはグラフが階層をなしていると考えられる．この場合，あるレベルにおける (ある性質を持った) グラフのエッジが，より上のレベルにおけるグラフのエッジとみなされることも多い．

この状況においては，従来の手法を用いたクリプキ構造によるモデル化が単純には遂行できない．一つの解決方法として，すべてのレベルの部品とその構成要素をクリプキ構造の状態と考え，「の構成要素である」という関係を特別な遷移関係と考えるものがある．しかしこの方法には，(直観的なわかりにくさという点を除いても,) 後節で述べるように，充足可能性判定の決定可能性を犠牲にするという問題点がある．そこで本研究では，これら複雑なシステムを自然にモデル化できるように，階層を導入したクリプキ構造を定義する．そして，階層クリプキ構造によって解釈されるように，様相論理を拡張する．

階層的なグラフ構造に関する研究は，数多く存在する．グラフ書換系の分野では，Drewes たちが階層構造の入ったハイパーグラフに対する書換系の定式化 [2] を行っている．ただし，彼らの階層ハイパーグラフには，階層を超えたエッジはない．また，近年 Milner たちは，従来の並行計算を発展させ，bigraph とモバイルプロセスの理論を展開している [4]．bigraph は階層構造 (place graph という) とリンク構造 (link graph という) の 2 つの構造の重層した数学的対象である．

*本研究の一部は，科学技術振興機構戦略的創造研究推進事業 (CREST) 研究領域「情報社会を支える新しい高性能情報処理技術」研究課題「検証における記述量爆発問題の構造変換による解決」の一部として行われた．
†本研究の一部は，文部科学省科学研究費補助金基盤研究 (C)(2)18500003 「グラフからマルチセットへの時相論理を用いた抽象化」の補助を受けた．

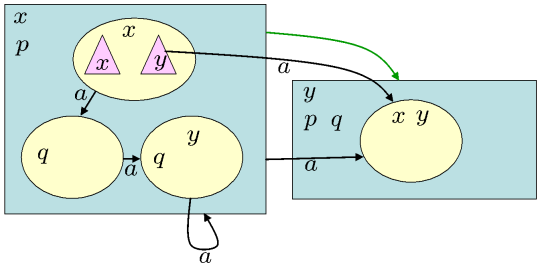

図 1: 階層クリプキ構造のイメージ

階層を超えたリンクも存在している．LMNtal[7] もリンクと階層の 2 種類の構造を持つグラフをデータとして扱うプログラミング言語である．UML[5] の statechart も，リンクと階層の 2 種類の構造を持つ．ただし，statechart は，状態遷移系を記述するために階層グラフ構造を使用しており，階層構造自体が動的に変化することはない．

本研究では，最終的には bigraph のような階層グラフの動的な変化や書換に関する検証を行うことを目的としている．本稿では，その基本となるべき枠組みを提案する．

我々の提案してきた枠組みにおいて，検証を実行する際の基本的な道具は論理式の充足可能性判定手続きである．階層化された論理においてもこの手続きが構成できることを示す．また，対象に対する操作についての性質の検証を行う場合には，これに加えて，論理式の操作に関する最弱事前条件を求めることが必要になる．従来研究において考察した操作のいくつかについて，階層化された論理式についても最弱事前条件が求められることについても述べる．

2 階層様相論理

2.1 概要

階層クリプキ構造は，通常のクリプキ構造に階層を導入したものであり，階層論理式は，階層クリプキ構造によって解釈される論理式である．これらの正確な定義は 2.2 節以降に記述するが，本節では概要を述べる．

階層クリプキ構造のイメージを図 1 に示す．このクリプキ構造の状態は，青い四角形，黄色い楕円，桃色の三角形の 3 種類がある．楕円は，四角形を親として持ち，三角形は楕円を親として持つ．子供はかならずあるとは限らない．

状態の性質を表すものに，命題記号とノミナルの 2 種類がある．この図では，p および q が命題記号，x と y がノミナルである．どちらも，どの状態で成立するかが指定されている．図では，成立している命題を図形の内側に示した．命題記号は自由に成立/不成立を決められるが，ノミナルは，同じ親を持つ子供の中では，1 箇所でしか成り立つことができない．

クリプキ構造の遷移関係は，階層関係とは独立に定められており，必ずしも兄弟でなくても遷移する場合がある．おのおのの遷移には様相と呼ばれるラベルがつけられる．様相には，基本様相およびそれを組み合わせた複合様相がある．図では，基本様相による遷移を黒い矢印で表した．ここでは，基本様相として a だけが存在するとしている．基本様相による遷移があると，その遷移を上位の階層から見た時の遷移が定義できる．緑色の矢印がその例であり，この遷移に対応する様相は (いくつもあるが，たとえば) $(x\ y\ a\ x)$ である．これは，左から順に「矢印の元の状態の子供のうち x が成り立つものの子供のうち y が成り立つものから，a による遷移を行うことで，矢印の先の状態の子供のうち x が成り立つものに遷移する」と読む．

このようなクリプキ構造を記述する論理として，階層様相論理を定義する．通常の様相論理式の構成に加え，$\lceil\varphi\rceil$ という構成が使える．これは，「子供で φ が成り立つ (ものがある)」という意味である．たとえば，図では，左側の四角で，x の隣に q が成り立つものがあるので，左側の四角で $\lceil@_x\langle a\rangle q\rceil$ が成立しているし，同じところで $\langle x\ y\ a\ y\rangle(p\wedge q)$ も成立している．

2.2 様相と論理式

ノミナル (nominal) の有限集合 Nom と基本様相の有限集合 BMod をパラメタとする階層様相言語を定義する．

PS を命題記号の集合，PV を命題変数の集合 とする．様相の集合 Mod と論理式の集合 Form を次のように定義する．ただし，$p\in$ PS, $X\in$ PV, $n\in$ Nom, $a\in$ BMod とする．

$$\mathrm{Mod}\ni m ::= a \mid \mathbf{o} \mid \varphi m \mid m\varphi \mid m\vee m \mid m\wedge m \mid m^{-1}$$

$$\mathrm{Form}\ni \varphi ::= p \mid n \mid X \mid \lceil\varphi\rceil \mid \neg\varphi \mid \varphi\vee\varphi \mid \langle m\rangle\varphi \mid \mu X\varphi$$

ただし，$\lceil\varphi\rceil$ において，φ に自由な命題変数は存在してはならない．また，$\mu X\varphi$ においては，φ に自由に現れる X にかかる否定記号は偶数個でなければならない．

様相 $\mathbf{o}$ は，大域様相 (global modality) と呼ばれる．

略記法として，$\varphi_1 \wedge \varphi_2$, $\varphi_1 \rightarrow \varphi_2$, $\varphi_1 \leftrightarrow \varphi_2$, $[m]\varphi(\stackrel{\text{def}}{=} \neg\langle m\rangle\neg\varphi)$, $\nu X\varphi(\stackrel{\text{def}}{=} \neg\mu X\neg\varphi[\neg X/X])$ 等は，とくに断らずに用いる．また，$n \in \mathrm{Nom}$ に対し，$\langle\mathbf{o}\rangle(n \wedge \varphi)$ を $@_n\varphi$ で表す．

演算子 $\vee, \langle\rangle, \mu$ について，その双対演算子 $\wedge, [], \nu$ を導入することで，論理式 φ を後述する意味論に関する同値変形により，否定記号が命題記号およびノミナルの直前にしか現れないようにできる．さらに，必要なら命題変数を置き換えることによって，各命題変数は，1 回しか束縛されないようにすることができる．これを正値標準形と呼ぶ．正値標準形の論理式 φ について，以下の 2 条件を満たすとき，φ は，無交代 (alternation-free) である，という．

- φ の任意の $\mu X\varphi_1$ の形の部分論理式について，φ_1 の任意の $\nu Y\varphi_2$ の形の部分論理式に X が現れない．
- φ の任意の $\nu X\varphi_1$ の形の部分論理式について，φ_1 の任意の $\mu Y\varphi_2$ の形の部分論理式に X が現れない．

以下，本稿では，無交代な論理式のみを扱う．

2.3 階層クリプキ構造

階層クリプキ構造 (hierarchical Kripke structure) とは 4 つ組 $\mathcal{K} = (S, R, \lhd, L)$ であって，以下の条件を満たすものである．

- S は空でない集合．無限集合でも良い．S の要素を状態と呼ぶ．
- R は，BMod から $S \times S$ への関数．
- $\lhd$ は，S 上の 2 項関係．S は $\lhd$ に関して，最大元 root を持つ深さ有限の木をなすものとする．$C(s) = \{s' \in S \mid s' \lhd s\}$ と定義する．
- L は，PS∪Nom から $\mathscr{P}(S)$ への関数．$n \in \mathrm{Nom}$ と $C(s) \neq \varnothing$ なる $s \in S$ に対して，$L(n) \cap C(s)$ は単元集合．このとき，$L(n) \cap C(s)$ の唯一の要素を $s.n^{\mathcal{K}}$ で表す．root.$n^{\mathcal{K}}$ は $n^{\mathcal{K}}$ と省略する．また，$n_1^{\mathcal{K}}.\cdots.n_2^{\mathcal{K}}$ を $(n_1.\cdots.n_2)^{\mathcal{K}}$ で表す．

階層クリプキ構造 $\mathcal{K} = (S, R, \lhd, L)$ と $s \in S$ に対して，4 つ組 $\mathcal{K}_s = (S_s, R_s, \lhd_s, L_s)$ を，$\mathcal{K}$ の s への制限と呼ぶ．ただし，$\preceq$ を $\lhd$ の反射推移閉包として，$S_s = \{s' \in S \mid s' \preceq s\}$，$R_s(a) = R(a) \cap (S_s \times S_s)$，$\lhd_s = \lhd_s \cap (S_s \times S_s)$，$L_s(x) = L(x) \cap S_s$ $(x \in \mathrm{PS} \cup \mathrm{Nom})$ である．容易にわかるように，$\mathcal{K}_s$ は，階層クリプキ構造となる．

$\mathcal{K} = (S, R, \lhd, L)$ の付値とは，PV から $\mathscr{P}(S)$ への関数のことである．$\mathcal{K}$ が付値 ρ のもとで $s \in S$ において論理式 φ を満たすという関係 $\mathcal{K}, \rho, s \models \varphi$ を定義する．これは $\rho, s \models \varphi$ と略すこともある．φ に自由変数が現れなければこれは付値には依存しないので，$s \models \varphi$ と書くこともある．同時に，R の定義域を一般の様相 Mod に拡張する．

- $s \models p \iff s \in L(p)$
- $\rho, s \models X \iff s \in \rho(X)$
- $\mathcal{K}, s \models \lceil\varphi\rceil \iff \exists s' \in C(s).\ \mathcal{K}_s, s' \models \varphi$
- $\rho, s \models \neg\varphi \iff \rho, s \not\models \varphi$
- $\rho, s \models \varphi_1 \vee \varphi_2 \iff \rho, s \models \varphi_1$ または $\rho, s \models \varphi_2$
- $\rho, s \models \langle m\rangle\varphi \iff$ $\exists s'.\ (s, s') \in R(m)$ かつ $\rho, s \models \varphi$
- $\rho, s \models \mu X\varphi \iff s \in \bigcap\{S' \subseteq S \mid \{s' \in S \mid \rho[X \mapsto S'], s' \models \varphi\} \subseteq S'\}$

- $R(\mathbf{o}) = S \times S$
- $(s_1, s_2) \in R(\varphi m) \iff \exists s_1' \in C(s_1).\ (s_1', s_2) \in R(m)$ かつ $\mathcal{K}_{s_1}, s_1' \models \varphi$
- $(s_1, s_2) \in R(m\varphi) \iff \exists s_2' \in C(s_2).\ (s_1, s_2') \in R(m)$ かつ $\mathcal{K}_{s_2}, s_2' \models \varphi$
- $R(m_1 \vee m_2) = R(m_1) \cup R(m_2)$
- $R(m_1 \wedge m_2) = R(m_1) \cap R(m_2)$
- $R(m^{-1}) = R(m)^{-1}$

任意の $s \in S$ に関して $\mathcal{K}, s \models \varphi$ となるとき，$\mathcal{K} \models \varphi$ と書く．

任意の $\mathcal{K}$ と $s \in S$ に関して，$\mathcal{K}, s \models \varphi \iff \mathcal{K}, s \models \varphi'$ となるとき，2 つの論理式 φ と φ' は同値であるといい，$\varphi \equiv \varphi'$ と書く．また，2 つの様相 m_1, m_2 が，$R(m_1) = R(m_2)$ を満たすとき，これらは同値であるといい，$m_1 \equiv m_2$ と書く．$(\varphi_1 m)\varphi_2 \equiv \varphi_1(m\varphi_2)$ であることは，簡単に確かめられる: そこで，混乱が生じない場合には，これらを $\varphi_1 m \varphi_2$ と表すことにする．また，$(m\varphi)^{-1} \equiv \varphi^{-1}m^{-1}$, $(\varphi m)^{-1} \equiv m^{-1}\varphi^{-1}$, $(m_1 \wedge m_2)^{-1} \equiv {m_1}^{-1} \wedge {m_2}^{-1}$ であるから，任意の様相は，逆様相演算子が基本様相のみに適用されるような様相と同値になる．

2.4 単層構造

本節では，階層クリプキ構造を，その階層関係を忘れることによって，通常のクリプキ構造とみなす方法を述べる．

まず，単層様相論理式の集合 $\mathrm{Form}^{\mathrm{F}}$ を定義する．これは，概略としては，階層様相論理式の集合 Form

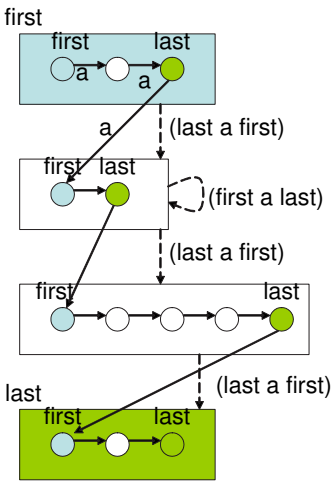

図 2: 階層クリプキ構造 $\mathcal{K}_1$

の構成から，階層様相 $(\varphi m, m\varphi)$ と下位演算子 $(\lceil\varphi\rceil)$ を除いたものである．また，階層様相論理におけるノミナルを並べたものを論理式として記述できる．これはノミナルを最上位階層から順にたどったものとして解釈され，たかだか 1 つの状態において成り立つ．さらに，この先頭に記号「*」をつけることもでき，任意の階層からノミナルをたどったところで成り立つと解釈される．

$$\begin{aligned}&\mathrm{NomSeq} \ni \xi ::= n \mid \xi.n\\ &\mathrm{Mod}^{\mathrm{F}} \ni m ::= a \mid \mathbf{o} \mid m \vee m \mid m \wedge m \mid m^{-1}\\ &\mathrm{Form}^{\mathrm{F}} \ni \varphi ::= p \mid \xi \mid *.\xi \mid X \mid \neg\varphi \mid \varphi \vee \varphi \mid\\ &\langle m\rangle\varphi \mid \mu X\varphi\end{aligned}$$

ただし，$p \in \mathrm{PS}$, $n \in \mathrm{Nom}$, $X \in \mathrm{PV}$, $a \in \mathrm{BMod}$ であり，$\mu X\varphi$ に関する制限は階層様相言語の場合と同様である．

階層クリプキ構造 $\mathcal{K} = (S, R, \triangleleft, L)$ をとる．$\mathrm{SNomSeq} = \{*.\xi \mid \xi \in \mathrm{NomSeq}\}$ として，$L' : \mathrm{PS} \cup \mathrm{NomSeq} \cup \mathrm{SNomSeq} \to \mathscr{P}(S)$ を，以下のように定義する．ただし，$p \in \mathrm{PS}$, $n \in \mathrm{Nom}$, $\xi \in \mathrm{NomSeq} \cup \mathrm{SNomSeq}$.

- $L'(p) = L(p)$
- $L'(n) = L(n) \cap C(\mathrm{root})$
- $L'(*.n) = L(n)$
- $L'(\xi.n) = \{s \in L(n) \mid \exists n' \in L'(\xi).\ s \in C(n')\}$

クリプキ構造 $\mathcal{K}^{\mathrm{F}} = (S, R, L')$ を，階層クリプキ構造 $\mathcal{K}$ に対する単層構造と呼ぶ．

2.5 例

(1) 下位と上位の様相 $\mathrm{Nom} = \{\mathrm{first}, \mathrm{last}\}$, $\mathrm{BMod} = \{a\}$ に対する階層クリプキ構造の例 $\mathcal{K}_1$ を図 2 に示す．この例では，S は，4 個の四角形と 13 個の円 (および図からは省略されている root) から構成されている．2 階層の構造をなしており，各々の円がそれを囲む四角形に対して $\triangleleft$ の関係にある．$R(a)$ が実線の矢印で表示されている．上の 3 つの四角形 s の last と名付けられた円 s' (すなわち，$s' = s.\mathrm{last}^{\mathcal{K}}$) は，下に隣接した四角形の first と名付けられた円に様相 a でリンクされているので，上下の四角形は，様相 (last a first) でリンクされていることになる．図示はしていないが，同じ組合せに対し，様相 (true a true) によるリンクなども存在する．同様に，上から 2 番目の四角形では，first と last が a でリンクされているので，自分自身と (first a last) でリンクされていることになる．

以下は，$\mathcal{K}_1$ において成り立つ階層論理式の例である．

- $@_{\mathrm{first}}\lceil@_{\mathrm{first}}\langle a\rangle\langle a\rangle\mathrm{last}\rceil$
  二回 $@_{\mathrm{first}}$ が現れているが，最初の first は，上のレベルの状態，すなわち青く塗られた四角形を表す．二つ目の first は，その四角形の中の青く塗られた円を表している．すなわち，この論理式は,「青い四角形の中で，青い円から緑の円までリンク a を 2 回たどって到達できる」ことを表している．
- $@_{\mathrm{first}}\lceil@_{\mathrm{first}}\mu X(\mathrm{last} \vee \langle a\rangle X)\rceil$
  上の論理式の最後の部分をリンク a を何回かたどって到達できる」に変更したもの．
- $@_{\mathrm{first}}\nu Y(\lceil@_{\mathrm{first}}\mu X(\mathrm{last} \vee \langle a\rangle X)\rceil \wedge\langle \mathrm{last}\, a\ \mathrm{first}\rangle Y)$
  上の論理式の内容が，青い四角形から (last a first) リンクでたどれるすべての四角形で成り立つことを表現した mono .

以下は，$\mathcal{K}_1$ において成り立つ単層論理式の例である．

- $@_{\mathrm{first.last}}\mu X(\mathrm{last.first} \vee \langle a\rangle X)$
  first.last は，青い四角形の緑の円，last.first は，緑の四角形の青い円を表す．リンク a をたどることで前者から後者へ到達できるという内容の論理式である．

(2) 階層による遷移関係の制限 $\mathrm{Nom} = \{x\}$, $\mathrm{BMod} = \{a\}$ に対する階層クリプキ構造の例 $\mathcal{K}_2$ を図 3 に示す．様相 (true a true) を b で表すことにす

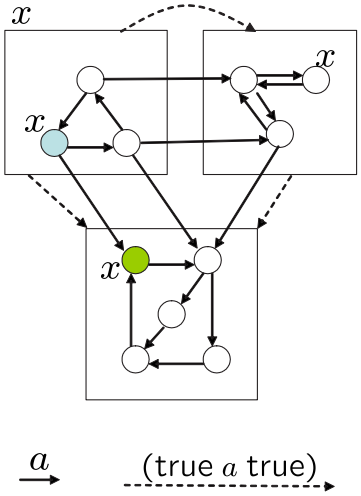

図 3: 階層クリプキ構造 $\mathcal{K}_2$

る．(図中では点線で表示している．)

階層論理式 $@_x\lceil @_x[a]\mu X(x \vee \langle a\rangle X)\rceil$ が $\mathcal{K}_2$ で成立する．青色で表示している円について，四角形の中だけに注目すれば，どの隣の点からも x に戻ってくることができることを表している．

しかし，単層論理式 $@_{x.x}[a]\mu X(x.x \vee \langle a\rangle X)$ は成立しない．四角形の枠を外して考えると，青色の円の隣となる緑色の円から青色の円に戻ってこられないからである．このように，階層論理式を導入することによって，考慮する遷移関係を制限することが可能になっている．これが，単層の場合に比較して階層構造を考える利点の一つである．

階層論理式 $@_x\mu X([b]X)$ が成立する．すなわち，四角形のレベルで x からリンク b をたどると，デッドエンドに到達する．

(3) システムのモデル: チャットサーバ チャットサーバとクライアントを題材にとり，システムの性質を階層論理式を用いて記述する例を示す．階層を使うことにより，部分的な記述を積み上げてシステム全体を記述することができる．

基本様相は a のみとする．様相 a によるリンクは，データの流れを表している．

クライアントのモデル化を図 4 の左上に示した．状態がクライアントであることを意味する命題記号 client を導入する．また，3 つのノミナル in, out, thr を導入する．これらはおのおの入力ポート，出力ポート，処理を行うスレッドを表す．データの流れは，次の論理式 φ_1 で表すことができる．client $\rightarrow \lceil @_{\mathrm{in}}\langle a\rangle \mathrm{thr} \wedge @_{\mathrm{thr}}\langle a\rangle \mathrm{out}\rceil$

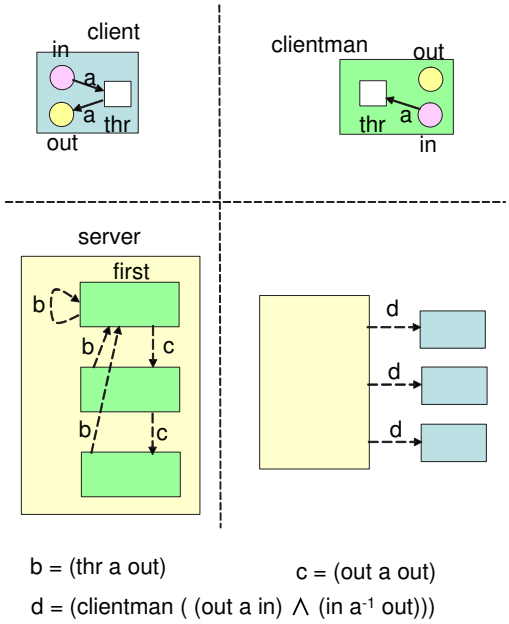

図 4: チャットサーバのモデル化 (1)

図 4 の右上はサーバ内のクライアントを管理するオブジェクトのモデル化である．クライアント管理オブジェクトであることを意味する命題記号 clientman を導入する．クライアントの場合と同様に，論理式 $\varphi_2 = \mathrm{clientman} \rightarrow \lceil @_{\mathrm{in}}\langle a\rangle \mathrm{thr}\rceil$ で，データの流れを表現できる．

図 4 の左下はサーバ内でのクライアント管理オブジェクトのリンク状況を示したものである．サーバを示すノミナル server を導入する．また，クライアント管理オブジェクトの先頭を表すノミナル first を導入する．各クライアント管理オブジェクトのスレッドから，first オブジェクトの out ポートにデータが流れるものとする．このことは，$\varphi_3 = @_{\mathrm{server}}\lceil[\mathbf{o}](\langle b\rangle \mathrm{first})\rceil$ と表現できる．ただし，$b = (\mathrm{thr}\ a\ \mathrm{out})$ である．first オブジェクトの out ポートからデータが順にクライアント管理オブジェクトの out ポートに流れていくものとする．このことは，$c = (\mathrm{out}\ a\ \mathrm{out})$ として，論理式 $\varphi_4 = @_{\mathrm{server}}\lceil[\mathbf{o}](\mu X(\mathrm{first} \vee \langle c^{-1}\rangle X))\rceil$ で表現できる．

図 4 の右下はサーバとクライアントの間のリンク状況を示したものである．様相 d は，$d = (\mathrm{clientman}((\mathrm{out\ a\ in}) \wedge (\mathrm{in\ a}^{-1}\ \mathrm{out})))$ と定義されており，クライアント管理オブジェクトの入力と出力が，それぞれクライアントの出力と入力と，互いに反対方向に接続されていることを表している．この様相を用いてリンク状況を論理式 $\varphi_5 = \mathrm{client} \rightarrow \langle d^{-1}\rangle \mathrm{server}$

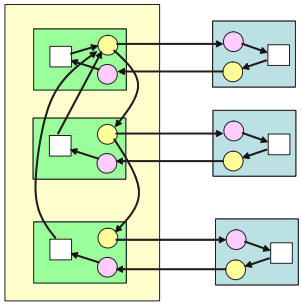

図 5: チャットサーバのモデル化 (2)

で表現することができる．

以上を組み合わせた階層クリプキ構造 $\mathcal{K}_3$ (の主要部分) を図 5 に示す．なお，階層クリプキ構造の定義によれば，最下層以外の状態においてすべてのノミナルが解釈を持っていなければならないが，このモデル化では実際には意味を持たないケースがある．たとえば，クライアントにおいて first は意味がない．これらについては，各状態に属する (図には表示していない) ダミーの状態を用意し，意味のないノミナルは，このダミーによって解釈することにする．

$\mathcal{K}_3$ は，次の論理式 φ を満足する: $[\mathbf{o}]\varphi_1 \wedge @_{\text{server}}\lceil[\mathbf{o}]\varphi_2\rceil \wedge \varphi_3 \wedge \varphi_4 \wedge [\mathbf{o}]\varphi_5$ また，フレッシュなノミナル x, y を導入して，次の単層論理式 ψ を考える: $@_x\text{client} \wedge @_y\text{client} \rightarrow @_{\text{x.out}}\mu X(\text{y.in} \vee \langle a\rangle X)$ これは，クライアント x の出力からクライアント y の入力にデータの流れがつながっていることを意味し，$\mathcal{K}_3$ (でノミナル x, y を解釈するように任意に拡張したもの) で成立する．これはもちろん偶然ではなく，$\mathcal{K}_3$ が φ を満たすことからの帰結である．次節に述べる充足可能性判定手続きを使うことによって,「φ のモデルは常に ψ をみたす」ということを確認することができる．

3 充足可能性判定手続き

φ_{I} を無交代な階層論理式，ψ_{I} を無交代な単層論理式とする．階層クリプキ構造 $\mathcal{K}$ と $s \in C(\text{root})$ が存在して，$\mathcal{K}, s \models \varphi_{\text{I}}$ および $\mathcal{K}^{\text{F}}, s \models \psi_{\text{I}}$ が成り立つとき，$(\varphi_{\text{I}}, \psi_{\text{I}})$ は充足可能である，という．

与えられた φ_{I} と ψ_{I} に対し，$(\varphi_{\text{I}}, \psi_{\text{I}})$ が充足可能であるかどうかを判定する問題は決定可能となる．決定可能手続きは，(階層を持たない) 通常の様相論理における手続きに変更を加えることによって得られる．本節では，手続きの概要を述べる．

まず，階層を持たない様相論理の判定手続き [6] の

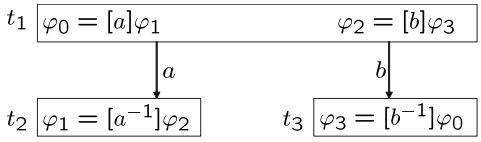

図 6: μ 演算子の無限展開

概略を述べる．これはタブロー法によるものである．充足可能性を判定したい論理式を φ_{I} とする．φ_{I} の閉包と呼ばれる集合 $\text{cl}(\varphi_{\text{I}})$ をとる．これは，φ_{I} を含み，部分論理式をとることについて閉じている最小の集合である．ただし，$\mu X\varphi$ に関しては，部分論理式として φ をとる代わりに $\varphi[\mu X\varphi/X]$ をとる．タブローの各ノード t には，$\text{cl}(\varphi_{\text{I}})$ の部分集合 $\Phi(t)$ が割り当てられ，ノード t において $\Phi(t)$ の各要素が成り立つものと考える．$\Phi(t)$ に ψ と $\neg\psi$ の両方が属するような場合には，ノード t は「矛盾する」と考え，このようなノードは削除する．

つぎに，各様相 a について，ノード間に遷移関係を与える．基本的には任意のノード t からノード t' に遷移することができるが，以下の 2 つの場合には遷移ができない．1 つは単純で，ノード t で $[a]\varphi$ が成り立ち，ノード t' では $\neg\varphi$ が成り立つか，反対に t' で $[a^{-1}]\varphi$ が成り立ってノード t では $\neg\varphi$ が成り立つ場合である．

もう 1 つは，最小不動点演算子 μ の無限展開に関係する．例として $\varphi_0 = \mu X([a][a^{-1}][b][b^{-1}]X)$ を考え，$\varphi_3 = [b^{-1}]\varphi_0$, $\varphi_2 = [b]\varphi_3$, $\varphi_1 = [a^{-1}]\varphi_2$ とすると，$\varphi_0 \equiv [a]\varphi_1$ となる．ノード t_1 で φ_0 と φ_2 が，ノード t_2 で φ_1 が，ノード t_3 で，φ_3 が成立しているとする (図 6 参照)．この場合，ノード t_1 から t_2 に a で, ノード t_1 から t_3 に b で遷移することがともに可能であるとすると，μ 演算子の無限展開が生じてしまう．しかし，片方の遷移だけならば問題がないので，遷移不可能と決めてしまうこともできない．

そこで，タブローのノードに μ 演算子に関係する論理式たちの展開の許される方向に関する情報も保持するようにし，この情報が異なる場合には異なるノードとみなすことにする．上の例では，t_1 と t_1' の二つのノードを用意し，成り立つ論理式の集合は同一だが，t_1 では $\varphi_0 \rightarrow \varphi_2$ の展開のみを許し (したがって，t_2 には a で遷移できるが，t_3 には b で遷移できない)，t_1' では逆に $\varphi_2 \rightarrow \varphi_0$ の展開のみを許すようにする．

このようにノード間の遷移関係を定義した後，遷

移関係に関して「矛盾する」ノードを削除していく．たとえば，$\langle a\rangle\varphi$ が成立するはずなのに，a による遷移先のどこでも φ が成り立っていないノードは削除される．このように削除を繰り返して，それ以上削除ができなくなったとき，φ_{I} を成立させるノードが残っていれば充足可能，そうでなければ充足不可能と判定する．以上が，階層がない場合の判定手続きの概要である．実際には，ノミナルや様相の $\wedge$ 演算子に対する考慮が必要であり，上述の手続きを修正する必要があるが，このような修正は階層化と直接関係がないので，ここでは省略する．

ここからは，階層がある場合の判定手続きの拡張について述べる．議論を単純にするため，単層論理式がなく，階層論理式のみの充足可能性を判定する方法について述べるが，単層論理式がある場合への拡張も容易である．

階層がない場合と同じく，タブロー法によって判定を行う．論理式の閉包は，レベルごとにとる．この際，$\lceil\varphi\rceil$ は，上のレベルでは全体を原子論理式として扱い，下のレベルでは φ としてみる．たとえば $\varphi_{\mathrm{I}} = @_x\lceil@_y p\rceil \wedge \langle z\ a\ q\rangle r$ とすると，レベル 1 の閉包 $\mathrm{cl}^1(\varphi_{\mathrm{I}})$ には，$r, \langle z\ a\ q\rangle r, \lceil@_y p\rceil$ などが属し，レベル 2 の閉包 $\mathrm{cl}^2(\varphi_{\mathrm{I}})$ には，y, p, z, q などが属する．$\mathrm{cl}^i(\varphi_{\mathrm{I}}) \neq \varnothing$ となる最大の i を dp と書く．

各 $i \leq \mathrm{dp}$ ごとに，i レベルの「セル」を定義する．階層のない場合のノードと同じく，セル c は，$\mathrm{cl}^i(\varphi_{\mathrm{I}})$ の部分集合 $\Phi(c)$ と μ 演算子に関する展開を許すかどうかの関係 $E(c)$ を保持している．$E(c)$ は非反射的推移的関係であり，$(\varphi,\psi) \in E(c)$ のとき，$\varphi \rightarrow \psi$ の展開が許されると考える．

長さ dp 以下のセルの列をタブローのノードとする[1]．ただし，i 番目のセルは i レベルのものとする．ノード t が列 $c_1 \cdots c_k$ $(k \leq \mathrm{dp})$ であるとき，$\Phi(t) = \Phi(c_k), E(t) = E(c_k)$ と書く．ノード t はノード tc の親であるとみなされる．

次は，タブローの基本様相 a に関する遷移関係である．階層のない場合に述べた 2 種類の遷移不可能な場合に加えて，もう 1 種類の遷移不可能な場合がある．これは，階層クリプキ構造では，下位の遷移関係が上位に影響することに関係している．例えば $\hat{a} = (\mathrm{true}\ a\ \mathrm{true})$ として，$\psi_1 = \neg p \wedge \mu X(p \vee [\hat{a}]X) \wedge \lceil@_x\langle a\rangle y\rceil$ は充足不可能である (図 7)．なぜなら，下位の x と y の間の遷移関係のために，上位で自分自身への $\hat{a}$ による遷移関係が生じ，このため，「どのパスをたどってもいずれは p が成り立つ」ことはできなくなるからである．

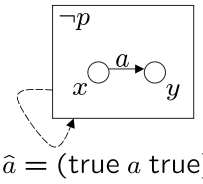

図 7: 論理式 ψ_1 は充足不可能

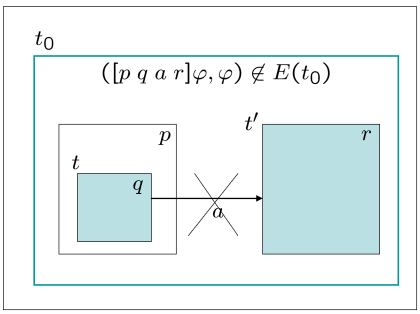

図 8: 遷移の禁止判定

この事情を反映させるために，ノード t からノード t' への様相 a による遷移の有無の決定には，次の操作が必要となる．t と t' の共通の祖先となる各ノード t_0，および $([b]\varphi,\varphi) \notin E(t_0)$ となる各 φ について，t_0 からみた t, t' 間の a による遷移が b に「適合する」とき，この遷移を禁止する (図 8)．適合することの詳細な定義は省略するが，例を挙げれば，$p \in \Phi(c), q \in \Phi(c')$ のとき，t からみた tc, tc' 間の a による遷移は $(p\ a\ q)$ や $(\mathrm{true}\ a\ \mathrm{true})$ には適合するが，$(\mathrm{true\ true}\ a\ \mathrm{true})$ や $(p\ a'\ q)$ $(a \neq a')$ には適合しない．

基本様相に関する遷移関係が定まれば，ノード間の複合様相に関する遷移関係も定義することができる．すなわち，組み合わせるもとになる様相に関する遷移関係が適切に存在し，かつ，3 種類の禁止条件に抵触しない時，その複合様相による遷移が存在するものとする．

これ以降の判定手続きは，階層がない場合と同様である．

4 クリプキ構造の変換操作とその最弱事前条件

システムに関して検証したい事項にはいろいろなタイプがあるが，典型的なものの 1 つに，対象に対す

[1] 実際にはノミナルに関する考慮のため，少し定義に変更が必要である．

る操作に関して対象に関するある性質が保存されることの検証がある．検証を行おうとする対象は，階層クリプキ構造によってモデル化することになるので，対象に関する性質は，階層論理式に対応し，対象に対する操作は，クリプキ構造の変換に対応する．考察する操作に対応するクリプキ構造の変換に関する論理式の事前条件 (可能であれば最弱事前条件) を与える論理式が決定できれば，前節に述べた充足可能性判定 (恒真性判定) を用いることで，性質が保存されることの検証が可能になる．

当然，すべての操作に関する事前条件が決定できるわけではない．どのような操作を考えるべきかは，検証を行いたい問題に依存する．我々は [8] において，ポインタによって構成されたデータ構造を扱うプログラムの検証のために，(階層的でない) クリプキ構造によるモデル化を行った．その際にいくつかのクリプキ構造の変換に関して最弱事前条件を決定した．これらに対応する階層クリプキ構造に関する変換についても，最弱事前条件を決定することができる．本稿では，そのうちの 1 つの変換を例として述べるが，他の変換についても考え方は同様である．

4.1 局所的最弱事前条件

階層クリプキ構造全体のなすクラスを HK で表す．f を HK から HK への関数とする．本稿で検討する f では，$\mathcal{K} \in \mathrm{HK}$ に対して，$S_{f(\mathcal{K})} = S_{\mathcal{K}}$, $\lhd_{f(\mathcal{K})} = \lhd_{\mathcal{K}}$ が成り立つ．階層論理式 φ に対して階層論理式 ψ が，任意の $\mathcal{K} \in \mathrm{HK}$ と任意の $s \in S_{\mathcal{K}}$ に対して，$\mathcal{K}, s \models \psi \iff f(\mathcal{K}), s \models \varphi$ となるとき，ψ を φ の局所的最弱事前条件と呼び，$\mathrm{wpl}(f, \varphi)$ で表す．

容易にわかるように，$\mathrm{wpl}(f, \varphi)$ が存在するとき，$\mathcal{K} \in \mathrm{HK}$ に関する条件「$\mathcal{K} \models \mathrm{wpl}(f, \varphi)$」は，$f$ に対する条件「$\mathcal{K} \models \varphi$」の最弱事前条件となる．ここで考える変換では，$\mathrm{wpl}(f, \varphi)$ を求めることによって最弱事前条件が求められる．

4.2 ノミナルの解釈の変更

本節では，ノミナル x の解釈を，ノミナル y で指定される状態に変更する f を検討する．階層が無ければ，x と y を指定することで f が定まるが，階層があるので，さらに，この変更をどこで行うか指定する必要がある．この指定には，ノミナルの列 $\xi = n_1, \ldots, n_k$ $(k \geq 0)$ を用いる．すなわち，$f(\mathcal{K})$ は，$\mathcal{K}$ で，$L(x) \cap C(\xi^{\mathcal{K}})$ の唯一の要素が $(\xi y)^{\mathcal{K}}$ となるように変更を加え，その他の構成要素はそのままとしたものである (図 9)．この f を，$f_{\xi,x,y}$ で表すことにする．誤解の生じないときには x と y は省略する．また，$k = 0$ のときの列 (空列) を ϵ で表す．

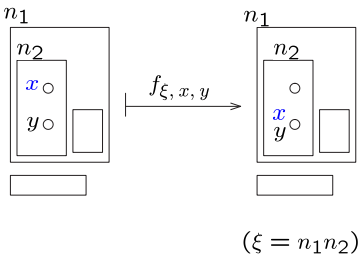

図 9: 変換 $f_{\xi,x,y}$

wpl は次のように計算することができる．ただし，$\xi \in \mathrm{Nom}^*$, $p \in \mathrm{PS}$, $X \in \mathrm{PV}$, $n, z \in \mathrm{Nom}$, $x' \in \mathrm{Nom} \setminus \{x\}$ とする．また，$\varphi_1 \to \varphi_2; \varphi_3$ は，$(\varphi_1 \wedge \varphi_2) \vee (\neg\varphi_1 \wedge \varphi_3)$ の略記である．

- $\mathrm{wpl}(f_\xi, p) = p$
- $\mathrm{wpl}(f_\epsilon, x) = y$
- $\mathrm{wpl}(f_\epsilon, x') = x'$
- $\mathrm{wpl}(f_{n\xi}, z) = z$
- $\mathrm{wpl}(f_\xi, X) = X$
- $\mathrm{wpl}(f_\epsilon, \lceil\varphi\rceil) = \lceil\varphi\rceil$
- $\mathrm{wpl}(f_{n\xi}, \lceil\varphi\rceil) = n \to \lceil\mathrm{wpl}(f_\xi, \varphi)\rceil; \lceil\varphi\rceil$
- $\mathrm{wpl}(f_\epsilon, @_x\varphi) = @_y \mathrm{wpl}(f_\epsilon, \varphi)$
- $\mathrm{wpl}(f_\epsilon, @_{x'}\varphi) = @_{x'} \mathrm{wpl}(f_\epsilon, \varphi)$
- $\mathrm{wpl}(f_{n\xi}, @_z\varphi) = @_z \mathrm{wpl}(f_n\xi, \varphi)$
- $\mathrm{wpl}(f_\xi, \neg\varphi) = \neg\mathrm{wpl}(f_\xi, \varphi)$
- $\mathrm{wpl}(f_\xi, \varphi_1 \vee \varphi_2) = \mathrm{wpl}(f_\xi, \varphi_1) \vee \mathrm{wpl}(f_\xi, \varphi_2)$
- $\mathrm{wpl}(f_\epsilon, \langle m\rangle\varphi) = \langle m\rangle \mathrm{wpl}(f_\epsilon, \varphi)$
- $\mathrm{wpl}(f_{n\xi}, \langle m\rangle\varphi) =$
  $(n \wedge \langle \mathrm{mm}(n\xi, n\xi, m)\rangle(n \wedge \mathrm{wpl}(f_n\xi, \varphi)))$
  $\vee(\neg n \wedge \langle \mathrm{mm}(\epsilon, n\xi, m)\rangle(n \wedge \mathrm{wpl}(f_n\xi, \varphi)))$
  $\vee(n \wedge \langle \mathrm{mm}(n\xi, \epsilon, m)\rangle(\neg n \wedge \mathrm{wpl}(f_n\xi, \varphi)))$
  $\vee(\neg n \wedge \langle \mathrm{mm}(\epsilon, \epsilon, m)\rangle(\neg n \wedge \mathrm{wpl}(f_n\xi, \varphi)))$
- $\mathrm{wpl}(f_\xi, \mu X\varphi) = \mu X \mathrm{wpl}(f_\xi, \varphi)$

ただし，$m \in \mathrm{Mod}$, $\xi, \eta \in \mathrm{Mod}^*$ に対して $\mathrm{mm}(\xi, \eta, m) \in \mathrm{Mod}$ は以下のように ($\mathrm{wpl}(f_\cdot, \varphi)$ と同時帰納的に) 定義されるものである．以下で，$a \in \mathrm{BMod}$ とする．

- $\mathrm{mm}(\xi, \eta, a) = a$
- $\mathrm{mm}(\epsilon, \eta, \psi m) = \psi\, \mathrm{mm}(\epsilon, \eta, m)$
- $\mathrm{mm}(n\xi, \eta, \psi m) =$
  $((n \wedge \mathrm{wpl}(f_{n\xi}, \psi))\, \mathrm{mm}(\xi, \eta, m))$

$\vee((\neg n \wedge \mathrm{wpl}(f_{n\xi}, \psi))\,\mathrm{mm}(\epsilon, \eta, m))$

- $\mathrm{mm}(\xi, \epsilon, m\psi) = \mathrm{mm}(\epsilon, \eta, m)\,\psi$
- $\mathrm{mm}(\xi, n\eta, m\psi) =$
  $(\mathrm{mm}(\xi, \eta, m)\,(n \wedge \mathrm{wpl}(f_{n\eta}, \psi)))$
  $\vee(\mathrm{mm}(\xi, \epsilon, m)\,(\neg n \wedge \mathrm{wpl}(f_{n\eta}, \psi)))$
- $\mathrm{mm}(\xi, \eta, m_1 \vee m_2)$
  $= \mathrm{mm}(\xi, \eta, m_1) \vee \mathrm{mm}(\xi, \eta, m_2)$
- $\mathrm{mm}(\xi, \eta, m_1 \wedge m_2)$
  $= \mathrm{mm}(\xi, \eta, m_1) \wedge \mathrm{mm}(\xi, \eta, m_2)$
- $\mathrm{mm}(\xi, \eta, m^{-1}) = \mathrm{mm}(\xi, \eta, m)^{-1}$

5 表現力に関する議論

本研究では，階層を持った対象のモデル化を行うために階層化したクリプキ構造を考え，これを意味論として持つ論理を構築した．

もちろんこのアプローチは唯一のものではなく，通常のクリプキ構造を用いても，階層を持つ対象をモデル化することができる．そのためには，我々の関係 $\triangleleft$ に相当する特別な様相 parent (ないし，その逆様相 child) を導入すればよい．$(c_1, c_2) \in R(\mathrm{parent})$ のとき，c_2 は c_1 のひとつ上の階層であると解釈することになる．

この方式 (parent 様相方式と呼ぶことにする) の場合，我々が展開した議論を行うためには，以下の道具立てが必要になる．

- 各階層における特定の要素 (オブジェクトのフィールド) を表すために，我々はノミナルを使用した．各状態に付随したクリプキ構造を考えることで，これが可能になった．階層がない場合，ノミナルは全体で 1 箇所しか指すことができない．このため，フィールドを表す特別な様相 $n_1, n_2, \ldots$ を導入することになる．この様相に関しては，その解釈が (部分) 関数となる必要がある．$((c, c'), (c, c'') \in R(n_1) \implies c' = c'')$
- ある階層内での不動点演算を行うためには，階層内に閉じたリンクだけをとりだしてくる必要がある．このためには様相の連接 $m_1; m_2$ を定義すれば $((c, c') \in R(m_1; m_2) \iff \exists c''.\ (c, c'') \in R(m_1), (c'', c') \in R(m_2))$ よい．様相 a の階層内に閉じた遷移関係は，様相 $(\mathrm{parent}; \mathrm{parent}^{-1})\wedge a$ で表現できる．

実際，これだけの準備をすれば，本稿で定義した階層様相論理式は，parent 様相方式の論理式に翻訳することができる．その意味で，parent 様相方式の方がより強力である．我々の方式には，以下のメリットがあると考える．

- 階層構造の表現がより自然である．
- 充足可能性 (したがって恒真性) 判定が決定可能である．parent 様相方式の場合は，様相 parent やフィールドを表す様相の解釈が (部分) 関数でなければならない．この場合，ノミナルと逆様相の存在の元では，一般には充足可能性は決定不能である [1]．別の言い方をすれば，parent 様相方式の決定可能な部分体系を定めたことになる．

6 おわりに

本研究では，階層状に構築されたシステムを自然に記述し，検証するための枠組みとして，階層様相論理を定義した．ベースとなる様相論理については，ノミナル，逆様相，様相の共通部分と和集合，不動点演算子などを導入することによって，記述力を向上させている．検証のために基本的な道具となる充足可能性判定手続きといくつかの操作に関する最弱事前条件が，これらの拡張の下でも求められることを示した．

今後の研究課題としては，以下のものがあげられる．

- 効率的な充足可能性判定アルゴリズム – 本稿の付録で示した一般的な充足可能性判定手続きをそのまま実行するには，膨大な時間が必要になる．実際に検証のために充足可能性を判定したい論理式に特徴的な性質を使ったより効率的な手続きを考案する必要がある．
- さまざまな操作に関する最弱事前条件の決定 – 本稿では，従来研究での検証の際に現れた操作について，階層化された設定での最弱事前条件を求めた．サーバ-クライアントシステム，ネットワークプロトコルなど，他の階層構造を持つ対象をモデル化する際に現れる操作についても検討することが必要である．

参考文献

[1] Bonatti, P. A. and Peron, A.: On the undecidability of logics with converse, nominals, recursion and counting, Vol. 158(2004), pp. 75–96.

[2] Drewes, F., Hoffmann, B., and Plump, D.: Hierarchical Graph Transformation, *Journal of Computer and System Sciences*, Vol. 64(2002), pp. 249–283.

[3] Hagiya, M., Takahashi, K., Yamamoto, M., and Sato, T.: Analysis of Synchronous and Asynchronous Cellular Automata using Abstraction by

Temporal Logic, *FLOPS2004: The Seventh Functional and Logic Programming Symposium*, Lecture Notes in Computer Science, Vol. 2998, 2004, pp. 7–21.

[4] Jensen, O. and Milner, R.: Bigraphs and mobile processes, Technical Report 570, Computer Laboratory, University of Cambridge, 2003.

[5] Rumbaugh, J., Jacobson, I., and Booch, G.: *The Unified Modelling Language Reference Manual*, Addison-Wesley, 1999.

[6] Tanabe, Y., Takahashi, K., Yamamoto, M., Tozawa, A., and Hagiya, M.: A Decision Procedure for the Alternation-free Two-way Modal μ-calculus, *TABLEAUX 2005*, Lecture Notes in Artificial Intelligence, Vol. 3702, 2005, pp. 277–291.

[7] Ueda, K. and Kato, N.: LMNtal: a language model with links and membranes, *Proc. Fifth Int. Workshop on Membrane Computing (WMC 2004)*, LNCS, Vol. 3365, Springer, 2005, pp. 110–125.

[8] 田辺良則, 関澤俊弦, 湯浅能史, 高橋孝一: 様相論理を使用したヒープ検証方式, 第 *3* 回ディペンダブルソフトウェアワークショップ *(DSW'06)*, 2006, pp. 39–50.