様相型に基づく情報流解析における
非干渉性の論理関係による一般化とその証明

四熊 尚方　　五十嵐 淳
京都大学 大学院情報学研究科

{naokata,igarashi}@kuis.kyoto-u.ac.jp

概要

情報流解析とは，プログラムの入出力間の依存関係を解析し，秘匿すべき情報の漏洩などを判定する技術である．Miyamoto, Igarashi は，型システムを利用した情報流解析の基礎となる枠組として，型付ラムダ計算 $\lambda_s^{\Box}$ を提案し，情報流解析としての正しさ「非干渉性 (non-interference)」を証明した．本研究では，$\lambda_s^{\Box}$ の非干渉性を，論理関係を用いて一般化し，証明する．さらに，$\lambda_s^{\Box}$ から単純型付ラムダ計算への変換により，この非干渉性を単純型付ラムダ計算における論理関係の基本補題に帰着して示す別証を提示し，その問題点を指摘する．

1　はじめに

1.1　背景

情報流解析とは，プログラム解析の一種で，プログラムの入出力間の依存の仕方を解析し，秘匿すべき情報の漏洩などの有無を判定するために使われる技術である．その一種として，型に基づく情報流解析がある．これは，型理論を利用して，「型付けされたプログラムに対して，その機密性の高い入力を変えても，その機密性の低いところへの出力結果は変わらない」こと (この性質を非干渉性 (non-interference) [4] と呼ぶ) を保証するものである．このような型に基づく情報流解析は，その使用目的や言語の種類・機能に応じて，様々なものが提案されており，その非干渉性の定式化も多岐に渡っている (手続型 [19, 20], 関数型 [5, 1, 13], オブジェクト指向 [11, 2, 3], 並行計算 [15, 6, 7, 12, 8] など).

ラムダ計算 $\lambda_s^{\Box}$ は， そのような型理論に基づいた既存の様々な情報流解析技術の一つの枠組みとして，Miyamoto, Igarashi [10] によって提案された．Miyamoto らは，まず，局所妥当性「可能世界 ℓ から到達できる任意の可能世界で A が成り立つ」を表す様相演算 $\Box_\ell A$ を導入した様相論理の体系[*1]を定義し，それに対し Curry-Howard 同型で対応するように，型付ラムダ計算 $\lambda_s^{\Box}$ を構築した．$\lambda_s^{\Box}$ の主な特徴は以下の 3 つである．最初の特徴は，可能世界を機密性の度合 (レベルと呼ぶ) とみなし，その到達可能関係をレベルの順序関係として，様相演算 $\Box_\ell A$ を，「レベル ℓ 以上でしか見ることのできない， 型 A のデータ型」の意味を持つ型構築子として読み替えることで，機密性に関する情報を型情報の一部として記述できるようになっていることである．型 $\Box_\ell A$ のデータを構築する演算子は，$\mathbf{box}_\ell\, M$ で表され，中身 M を外部から隔離し，レベル ℓ 以上のところでしか中身を取り出すことができないようにする役目を持つ．2 番目の特徴は，型判断に付与

[*1] 直観主義的な S4 を，さらに多重様相命題論理化した体系の一種となっている．

されたレベルにより，型付けされるプログラムが，どのレベルで扱われているものなのかが明示されていることである．3 番目は，型判断のコンテキストが 2 種類あることである．一つは，通常変数コンテキストと呼ばれ，現在の型判断のレベルで使うことのできる変数の宣言列を表している．もう一方は，様相変数コンテキストと呼ばれ，その各変数宣言では，変数の型だけではなく，その変数がどこ以上で使えるかを表すレベルも一緒に宣言されている．そのような変数に対し，$\Box_\ell A$ 型のデータの中身を束縛することで，その中身の情報をプログラム内で用いることができるようになる．

また，Miyamoto, Igarashi は，以上の $\lambda_s^\Box$ の型システムに対し，出力の型が関数型を含まない特殊な形のプログラムに対しては，先述した非干渉性が成立することを簡約意味論のみに基づいて証明した．

1.2 本研究の目標

本研究の目標は，論理関係 (logical relations) による非干渉性の一般化と証明である．それぞれに関し説明する．

非干渉性の一般化　まず，先に述べた非干渉性を，あるレベルの観察者にとっての等価性という観点で，言い直す．あるレベルの観察者は，自分よりレベルの低い情報は，観察して区別できるが，低くない情報は，実際の中身に関わらず区別できない．このとき，ある 2 つの情報が，その観察者にとって区別できないならば，その情報は，その観察者のレベルにおいて等価であると言うことにする．すると，非干渉性は次のように言い直すことができる：「任意にレベル ℓ を一つとる．このとき，型付けられたプログラムは，ℓ において等価な入力に対して，ℓ において等価な出力を返す」[14].

そして，本研究では，Sumii ら [16]，Tse ら [17] の研究を参考に，この等価性を定式化するために，論理関係を用いる．それによって，異なる 2 つのプログラムであっても，その異なる箇所が観察者のレベルよりも高いレベルでの情報であれば，観察可能な挙動が等価なプログラムとして関係づけることができる (ここでいう論理関係とは，型に関して帰納的に定義された，閉じた項上の関係の族であり，各関係は，簡約に関し閉じている)．我々は，この論理関係で表される等価性によって，上で言い直した非干渉性を定式化する．

このように再定式化した非干渉性は，Miyamoto, Igarashi が扱わなかった，出力の型に関数型を含むような一般のプログラムに対しても，成立する．

非干渉性の証明　本研究では，一般化した非干渉性に対し，次の 2 通りの証明方法を考える．

1. $\lambda_s^\Box$ の等価性の定義に基づく直接証明
2. $\lambda_s^\Box$ から単純型付ラムダ計算 $\lambda^\to$への変換によって，$\lambda^\to$における「論理関係の基本補題」に帰着する間接証明

1 の直接証明は完成しており，本論文で説明する．

2 の間接証明を試みる意義は，論理関係によって一般化した非干渉性の本質的な意味を，よく知られた，単純型付ラムダ計算の性質「論理関係の基本補題」から，把握することにある．

我々は，Tse らによる, DCC [1] から System F への変換と，それを用いた DCC の非干渉性の証明法 [17] にならい，$\lambda_s^\Box$ から単純型付ラムダ計算 $\lambda^\to$への変換を定義し，$\lambda_s^\Box$ の一般化した非干渉性の証明を試みた．しかし，この証明方法には問題があり，Tse らの手法にも同様の問題があることを，我々は発見した．本稿では，$\lambda_s^\Box$ の場合に関して，この問題を指摘する．

1.3 本稿の構成

第 2 節では，まず $\lambda_s^{\Box}$ の定義を述べ，次に $\lambda_s^{\Box}$ における等価性を定義し，一般化した非干渉性を述べて，直接的に証明する．第 3 節では，変換先である単純型付ラムダ計算 $\lambda^{\to}$の型システムとその上の等価性を定義してから，変換を定義する．それから，変換の性質を述べ，非干渉性を間接的に証明するために重要な条件「変換が等価性を保存する」ことの証明の問題点を指摘する．第 4 節で関連研究について議論し，第 5 節で結論を述べる．

2 $\lambda_s^{\Box}$ と一般化した非干渉性

$\lambda_s^{\Box}$ は，型に基づく情報流解析の基礎となる枠組として単純型付ラムダ計算を拡張した計算体系であり，Miyamoto, Igarashi によって提案された [10]．本節では，まず，$\lambda_s^{\Box}$ を定義し，その基本的性質を紹介する．それから，$\lambda_s^{\Box}$ 項の等価性を定義し，一般化した $\lambda_s^{\Box}$ の非干渉性の証明を行う．

2.1 構文

最初に，いくつかの前提を述べる．**OVar** は**通常変数**からなる集合であり，その要素を x, y, z で表す．**MVar** は**様相変数**からなる集合であり，その要素を u, v, w で表す．**OVar**, **MVar** はともに可算無限集合であり，共通要素を持たないものとする．また，$(\mathcal{L}, \sqsubseteq)$ はレベルからなる部分順序集合であり，その要素を ℓ で表す．

$\lambda_s^{\Box}$ の型は，ユニット型，関数型，直積型，直和型，様相型からなる．

定義 2.1.1 (型). $\lambda_s^{\Box}$ の型を次で定義する．

$$A ::= \mathit{unit} \mid A \to A \mid A \times A \mid A + A \mid \Box_\ell A$$

型構築子の結合の優先順位は，$\Box_\ell > \times > + > \to$．また，$\to$ は右結合だとする．

定義 2.1.2 (項). $\lambda_s^{\Box}$ の項を次で定義する．

$$\begin{aligned} M ::= {} & x \mid u \mid () \mid \lambda x\!:\!A.\, M \mid M\, M \mid \langle M, M\rangle \mid \pi_1(M) \mid \pi_2(M) \mid \iota_1(M) \mid \iota_2(M) \\ & \mid (\mathbf{case}\, M\, \mathbf{of}\, \iota_1(x_1) \Rightarrow M \mid \iota_2(x_2) \Rightarrow M) \mid \mathbf{box}_\ell\, M \mid \mathbf{let}\, \mathbf{box}_\ell\, u = M\, \mathbf{in}\, M \end{aligned}$$

また，次節以降，代入が変数の束縛関係を壊さないように，適宜，束縛変数の名前を変えることを予め注意しておく．通常変数 x，様相変数 u への項 M の代入を，それぞれ $[M/x]$, $[M/u]$ で表す．また，通常変数，様相変数への同時代入を，それぞれメタ変数 γ_o, γ_m で表す．

2.2 型システム

$\lambda_s^{\Box}$ の型判断は $\Gamma_m; \Gamma_o \vdash^{\ell} M : A$ という形をしている．この判断は，「レベル ℓ において，様相変数コンテキスト Γ_m と通常変数コンテキスト Γ_o の下で，項 M は型 A を持つ」と読む．このレベル ℓ のことを判断のレベル (あるいは，判断の今のレベル) と呼ぶことにする．Γ_m は，様相変数の宣言 $u ::^{\ell'} A$ の列である．このレベル ℓ' を 様相変数 u の宣言のレベルと呼ぶ．同様に，Γ_o は，通常変数の宣言 $x : A$ の列である．そ

れぞれのコンテキストにおいて宣言する変数の重複はないものとする．また，空のコンテキストを $\cdot$ で表し，$\cdot;\cdot \vdash^{\ell} M : A$ を単に $\vdash^{\ell} M : A$ で表すことにする．

定義 2.2.1 (型付け規則). $\lambda_s^{\Box}$ の型付け規則は以下の通りである．

$$\frac{x : A \in \Gamma_o}{\Gamma_m; \Gamma_o \vdash^{\ell} x : A}\ (\text{T-OVAR}) \qquad \frac{u ::^{\ell'} A \in \Gamma_m \quad \ell' \sqsubseteq \ell}{\Gamma_m; \Gamma_o \vdash^{\ell} u : A}\ (\text{T-MVAR}) \qquad \Gamma_m; \Gamma_o \vdash^{\ell} () : \mathit{unit}\ \ (\text{T-UNIT})$$

$$\frac{\Gamma_m; \Gamma_o, x : A \vdash^{\ell} M : B}{\Gamma_m; \Gamma_o \vdash^{\ell} \lambda x{:}A.\, M : A \to B}\ (\text{T-ABS}) \qquad \frac{\Gamma_m; \Gamma_o \vdash^{\ell} M : A \to B \quad \Gamma_m; \Gamma_o \vdash^{\ell} N : A}{\Gamma_m; \Gamma_o \vdash^{\ell} M\, N : B}\ (\text{T-APP})$$

$$\frac{\Gamma_m; \Gamma_o \vdash^{\ell} M : A \quad \Gamma_m; \Gamma_o \vdash^{\ell} N : B}{\Gamma_m; \Gamma_o \vdash^{\ell} \langle M, N \rangle : A \times B}\ (\text{T-PAIR}) \qquad \frac{\Gamma_m; \Gamma_o \vdash^{\ell} M : A_1 \times A_2 \quad i \in \{1, 2\}}{\Gamma_m; \Gamma_o \vdash^{\ell} \pi_i(M) : A_i}\ (\text{T-PROJ})$$

$$\frac{\Gamma_m; \Gamma_o \vdash^{\ell} M : A_i \quad i \in \{1, 2\}}{\Gamma_m; \Gamma_o \vdash^{\ell} \iota_i(M) : A_1 + A_2}\ (\text{T-INJ})$$

$$\frac{\Gamma_m; \Gamma_o \vdash^{\ell} M : A_1 + A_2 \quad \Gamma_m; \Gamma_o, x_1 : A_1 \vdash^{\ell} N_1 : B \quad \Gamma_m; \Gamma_o, x_2 : A_2 \vdash^{\ell} N_2 : B}{\Gamma_m; \Gamma_o \vdash^{\ell} (\mathbf{case}\, M\, \mathbf{of}\, \iota_1(x_1) \Rightarrow N_1 \,|\, \iota_2(x_2) \Rightarrow N_2) : B}\ (\text{T-CASE})$$

$$\frac{\Gamma_m; \cdot \vdash^{\ell'} M : A}{\Gamma_m; \Gamma_o \vdash^{\ell} \mathbf{box}_{\ell'} M : \Box_{\ell'} A}\ (\text{T-BOX})$$

$$\frac{\Gamma_m; \Gamma_o \vdash^{\ell} M : \Box_{\ell'} A \quad \Gamma_m, u ::^{\ell'} A; \Gamma_o \vdash^{\ell} N : B}{\Gamma_m; \Gamma_o \vdash^{\ell} \mathbf{let\, box}_{\ell'}\, u = M\, \mathbf{in}\, N : B}\ (\text{T-LETBOX})$$

様相型の導入規則 T-Box と 除去規則 T-LetBox に関し説明する．

T-Box の前提の判断における通常変数コンテキストは空でなければならない．なぜなら，通常変数コンテキストで宣言されている変数は，**今**の型判断のレベルで使えるものであり，$\mathbf{box}_{\ell} M$ の中身は，ℓ 以上のレベルでならば，型判断のレベルに依存せず，どこでも同じように，使うことができなければならないからである．これは，述語論理における $\forall$ の導入規則の条件，もしくは，ヒルベルト流様相論理における necessitation 規則の条件に対応する．

T-LetBox において，前提のコンテキストにある様相変数は，T-Mvar の右の前提条件により，その宣言のレベル以上の任意のところでしか使われないことが保証されるので，「型 $\Box_{\ell} A$ を持つ項の中身を，宣言のレベルが ℓ の様相変数に割当てること」は妥当な操作である．

例 2.2.2. $\mathcal{L} = \{\mathsf{L}, \mathsf{H}\}$, $\mathsf{L} \sqsubseteq \mathsf{L}$, $\mathsf{L} \sqsubseteq \mathsf{H}$, $\mathsf{H} \sqsubseteq \mathsf{H}$ とする．また，ブール型を $\mathit{bool} = \mathit{unit} + \mathit{unit}$, $\mathsf{true} = \iota_1()$, $\mathsf{false} = \iota_2()$ と定義する．

$\ell \sqsubseteq \ell'$ となる $\ell, \ell' \in \mathcal{L}$ に対して，型判断 $\vdash^{\ell'} \lambda x{:}\Box_{\ell} \mathit{bool}.\, \mathbf{let\, box}_{\ell}\, u = x\, \mathbf{in}\, u : \Box_{\ell} \mathit{bool} \to \mathit{bool}$ は導出可能である．このように機密性の低い情報をより高いところで使うプログラムに対しては，確かに型チェックが通る．

しかし，$\ell \not\sqsubseteq \ell'$, つまり $\ell = \mathsf{H}$, $\ell' = \mathsf{L}$ の場合は，型付けできない．これは，機密性の度合が高い情報を低いところへ漏らすプログラムは，型チェックを通ることができないことを表している．

この型システムに対し，以下の補題が成立する．

補題 2.2.3 (型判断のレベル上昇, [10]). $\Gamma_m; \Gamma_o \vdash^{\ell} M : A$ かつ $\ell \sqsubseteq \ell'$ ならば， $\Gamma_m; \Gamma_o \vdash^{\ell'} M : A$. しかも，仮定と結論の型判断の導出木は，判断のレベルの差を除けば，一致する．

補題 2.2.4 (代入補題, [10]). 通常変数と様相変数への代入に関して，それぞれ次が成立する．

1. $\Gamma_m; \Gamma_o \vdash^{\ell} M' : B$ かつ $\Gamma_m; \Gamma_o, x : B \vdash^{\ell} M : A$ ならば，$\Gamma_m; \Gamma_o \vdash^{\ell} [M'/x]M : A$.
2. $\Gamma_m; \cdot \vdash^{\ell'} M' : B$ かつ $\Gamma_m, u ::^{\ell'} B; \Gamma_o \vdash^{\ell} M : A$ ならば，$\Gamma_m; \Gamma_o \vdash^{\ell} [M'/u]M : A$

2.3 操作的意味論

$\lambda_s^{\Box}$ の操作的意味論を簡約関係 $\longrightarrow$ によって定める．

定義 2.3.1 (簡約規則). $\lambda_s^{\Box}$ 項の簡約関係 $\longrightarrow$ を以下の簡約を含む最小の合同関係とする．

$$
\begin{aligned}
(\lambda x{:}A.\, M_1)\, M_2 &\longrightarrow [M_2/x]M_1 \\
\pi_i(\langle M_1,\, M_2\rangle) &\longrightarrow M_i \\
(\mathbf{case}\, \iota_i(M)\, \mathbf{of}\, \iota_1(x_1) \Rightarrow N_1 \,|\, \iota_2(x_2) \Rightarrow N_2) &\longrightarrow [M/x_i]N_i \\
\mathbf{let}\, \mathbf{box}_\ell\, u = \mathbf{box}_\ell\, M\, \mathbf{in}\, N &\longrightarrow [M/u]N
\end{aligned}
$$

この簡約関係に関し，型保存定理 (subject reduction)，合流性，強正規化性が成立することが示されている [10].

2.4 透過的無関数型と非干渉性

Miyamoto, Igarashi は，項の型が次の特殊な場合に関しては，簡約による操作的意味論だけに基いて，非干渉性が証明できることを示した [10].

定義 2.4.1 (透過的な無関数型). 型 A が，レベル ℓ において，透過的な無関数型であることを，次のいずれかの場合に限り言う:

1. A が *unit* である．
2. $A = A_1 \times A_2$ かつ A_1, A_2 がともにレベル ℓ において，透過的な無関数型.
3. $A = A_1 + A_2$ かつ A_1, A_2 がともにレベル ℓ において，透過的な無関数型.
4. $A = \Box_{\ell'} A_0$, $\ell' \sqsubseteq \ell$ かつ A_0 はレベル ℓ において，透過的な無関数型. [*2]

定理 2.4.2 (非干渉性). $u ::^{\ell} A; \cdot \vdash^{\ell'} M : B$, $\ell \not\sqsupseteq \ell'$, かつ B はレベル ℓ において透過的な無関数型ならば，次を満たす唯一の正規形 V が存在する：任意の $\vdash^{\ell} N : A$ に対し，$[N/u]M \longrightarrow^* V$.

2.5 等価性と一般化した非干渉性

本節では，$\lambda_s^{\Box}$ における等価性を論理関係によって定義し，前節の非干渉性を一般化する．今，閉じた項 M, M' がレベル ℓ において型 A を持つとする．このとき，M と M' が「レベル ℓ において等しく見える」ということを，以下で定義する判断 $M \approx_\ell M' : A$ で表わす．この判断は，「レベル ℓ において，項 M, M' は，型 A に対応する等価関係にある」と読む.

[*2] Miyamoto らの透過的な無関数型の定義では，定義 2.4.1 の 4 の場合が，「$A = \Box_{\ell'} A_0$, $\ell' \sqsubseteq \ell$ かつ A_0 は (ℓ ではなく) レベル ℓ' において，透過的な無関数型.」となっている [10]. 定義 2.4.1 は，Miyamoto らの定義の拡張になっており，もとの Miyamoto らの定義においても，定理 2.4.2 が成立する.

定義 2.5.1 (等価性：$M \approx_\ell M' : A$, $V \sim_\ell V' : A$). 論理関係によって，$\lambda_s^{\Box}$ の項の間の等しさを次のように定義する．但し，$\sim_\ell$ は正規形の項に対する等価関係であり，メタ変数 V は正規形の項を表す．

$$\frac{}{() \sim_\ell () : \mathit{unit}} \quad \text{(SBR-Unit)}$$

$$\frac{\forall (M \approx_\ell M' : A_1).\ V\,M \approx_\ell V'\,M' : A_2}{V \sim_\ell V' : A_1 \to A_2} \quad \text{(SBR-Fun)}$$

$$\frac{V_1 \sim_\ell V_1' : A_1 \quad V_2 \sim_\ell V_2' : A_2}{\langle V_1, V_2\rangle \sim_\ell \langle V_1', V_2'\rangle : A_1 \times A_2} \quad \text{(SBR-Pair)}$$

$$\frac{V \sim_\ell V' : A_i}{\iota_i(V) \sim_\ell \iota_i(V') : A_1 + A_2} \quad \text{(SBR-Inj)}$$

$$\frac{\ell' \not\sqsubseteq \ell}{\mathbf{box}_{\ell'}\,V \sim_\ell \mathbf{box}_{\ell'}\,V' : \Box_{\ell'} A} \quad \text{(SBR-Box1)}$$

$$\frac{V \sim_\ell V' : A \quad \ell' \sqsubseteq \ell}{\mathbf{box}_{\ell'}\,V \sim_\ell \mathbf{box}_{\ell'}\,V' : \Box_{\ell'} A} \quad \text{(SBR-Box2)}$$

$$\frac{M \longrightarrow^* V \quad M' \longrightarrow^* V' \quad V \sim_\ell V' : A}{M \approx_\ell M' : A} \quad \text{(SBR-Term)}$$

ここで，$\longrightarrow^*$ は，簡約関係 $\longrightarrow$ の反射的推移的閉包を表している．

上の定義が *well-defined* であることは，型判断のレベル上昇則から保証される．

SBR-Box1，SBR-Box2 は，**あるレベル ℓ の観察者は，自分より低いレベルの情報しか観察できない**ことを表しており，「あるレベルにおいて等しく見える」という関係の性質を反映している．

このとき，明らかに次が成立する．

定理 2.5.2. 各型 A に対応する等価関係 $\approx_\ell$ は，対称律と推移律を満たし，さらに，簡約関係の反射的対称的推移的閉包に関して閉じている．

$\lambda_s^{\Box}$ の一般化した非干渉性を述べる前に，代入に関するいくつか有用な記法の定義を導入しておく．

定義 2.5.3 (通常変数への代入とその等価性：$\gamma_o \models^\ell \Gamma_o$, $\gamma_o \approx_\ell \gamma_o' : \Gamma_o$). γ_o がレベル ℓ における通常変数コンテキスト Γ_o への代入であることを，$\gamma_o \models^\ell \Gamma_o$ で表し，次のときに言う：$\forall x : A \in \Gamma_o$ に対し，$\vdash^\ell \gamma_o(x) : A$.

次に，$\gamma_o, \gamma_o' \models^\ell \Gamma_o$ としたとき，γ_o と γ_o' がレベル ℓ で等価であることを，$\gamma_o \approx_\ell \gamma_o' : \Gamma_o$ で表し，次のときに言う：$\forall x : A \in \Gamma_o$ に対し，$\gamma_o(x) \approx_\ell \gamma_o'(x) : A$.

定義 2.5.4 (様相変数への代入とその等価性：$\gamma_m \models^* \Gamma_m$, $\gamma_m \approx_{*\sqsubseteq\ell} \gamma_m' : \Gamma_m$). γ_m が様相変数コンテキスト Γ_m への代入であることを，$\gamma_m \models^* \Gamma_m$ で表し，次のときに言う：$\forall u ::^\ell A \in \Gamma_m$ に対し，$\vdash^\ell \gamma_m(u) : A$.

次に，$\gamma_m, \gamma_m' \models^* \Gamma_m$ としたとき，γ_m と γ_m' がレベル ℓ で等価であることを，$\gamma_m \approx_{*\sqsubseteq\ell} \gamma_m' : \Gamma_m$ で表し，次のときに言う：$\forall u ::^{\ell'} A \in \Gamma_m$ に対し，$\ell' \sqsubseteq \ell$ ならば，$\gamma_m(u) \approx_\ell \gamma_m'(u) : A$.

例 2.5.5. 例 2.2.2 の記法を用いると，

$$[\mathsf{true}/u] \approx_{*\sqsubseteq \mathsf{L}} [\mathsf{false}/u] : \{u ::^{\mathsf{H}} \mathit{bool}\}$$
$$[\mathsf{true}/u] \not\approx_{*\sqsubseteq \mathsf{H}} [\mathsf{false}/u] : \{u ::^{\mathsf{H}} \mathit{bool}\}.$$

また，$\mathbf{box}_{\mathsf{H}}\,u$ に対して，レベル L において等しい代入を作用させた結果は，

$$[\mathsf{true}/u](\mathbf{box}_{\mathsf{H}}\, u) \approx_{\mathsf{L}} [\mathsf{false}/u](\mathbf{box}_{\mathsf{H}}\, u) : \Box_{\mathsf{H}} \mathit{bool}\ .$$

例 2.2.2 で取り上げた項において，$\ell = \ell' = \mathsf{H}$ としたものを，f とすると，

$$f\,(\mathbf{box}_{\mathsf{H}}\,\mathsf{true}) \not\approx_{\mathsf{H}} f\,(\mathbf{box}_{\mathsf{H}}\,\mathsf{false}).$$

一般化した非干渉性を次のように述べることができる．

定理 2.5.6 (一般化した非干渉性). $\Gamma_m; \Gamma_o \vdash^{\ell} M : A\ ,\ \gamma_o \approx_{\ell} \gamma'_o : \Gamma_o\ ,\ \gamma_m \approx_{*\sqsubseteq\ell} \gamma'_m : \Gamma_m$ ならば，

$$\gamma_m\gamma_o\, M \approx_{\ell} \gamma'_m\gamma'_o\, M : A$$

証明. 型判断 $\Gamma_m; \Gamma_o \vdash^{\ell} M : A$ の導出木の大きさに関する帰納法.

最後に用いた導出規則が T-MVAR のとき．型判断の結論は $\Gamma_m; \Gamma_o \vdash^{\ell} u : A$ ．結論の直前の前提条件は，$u ::^{\ell'} A \in \Gamma_m$ かつ $\ell' \sqsubseteq \ell$．従って，$\gamma_m \approx_{*\sqsubseteq\ell} \gamma'_m : \Gamma_m$ の定義より，$\gamma_m\gamma_o\, u \approx_{\ell} \gamma'_m\gamma'_o\, u : A$.

最後に用いた導出規則が T-BOX のとき．型判断の結論は $\Gamma_m; \Gamma_o \vdash^{\ell} \mathbf{box}_{\ell'} M : \Box_{\ell'} A$．結論の直前の前提は $\Gamma_m; \cdot \vdash^{\ell'} M : A$．$\ell' \sqsubseteq \ell$ かどうかで場合分け．$\ell' \sqsubseteq \ell$ ならば，前提に補題 2.2.3 を適用することができ，$\Gamma_m; \cdot \vdash^{\ell} M : A$，かつ，その導出木の大きさは前提と変わらない．従って，帰納法の仮定より，$\gamma_m\gamma_o\, M \approx_{\ell} \gamma'_m\gamma'_o\, M : A$ (M には自由な通常変数がないことに注意)．SBR-BOX2 から，主張が成立．他方，$\ell' \not\sqsubseteq \ell$ ならば，SBR-BOX1 より，成立.

最後に用いた導出規則が T-LETBOX のとき．型判断の結論は $\Gamma_m; \Gamma_o \vdash^{\ell} \mathbf{let}\,\mathbf{box}_{\ell'}\, u = M\,\mathbf{in}\, N : B$. 結論の直前の前提は $\Gamma_m; \Gamma_o \vdash^{\ell} M : \Box_{\ell'} A$ と $\Gamma_m, u ::^{\ell'} A; \Gamma_o \vdash^{\ell} N : B$． 帰納法の仮定より，$\gamma_m\gamma_o\, M \approx_{\ell} \gamma'_m\gamma'_o\, M : \Box_{\ell'} A$．定義より，ある V, V' があって，$\gamma_m\gamma_o\, M \longrightarrow^* \mathbf{box}_{\ell'}\, V$, $\gamma'_m\gamma'_o\, M \longrightarrow^* \mathbf{box}_{\ell'}\, V'$ かつ $\mathbf{box}_{\ell'}\, V \sim_{\ell} \mathbf{box}_{\ell'}\, V' : \Box_{\ell'} A$ ．$\ell' \sqsubseteq \ell$ の場合は SBR-BOX2 より，$\ell' \not\sqsubseteq \ell$ の場合は直ちに，$[V/u] \approx_{*\sqsubseteq\ell} [V'/u] : \{u ::^{\ell'} A\}$ ．従って，帰納法の仮定より，$[V/u]\gamma_m\gamma_o\, N \approx_{\ell} [V'/u]\gamma'_m\gamma'_o\, N : B$．よって，定理 2.5.2 より，主張が成立.

他の場合は容易. □

上の非干渉性の主張は，代入を入力，代入結果を出力と見たとき，「はじめに」で言い直した非干渉性「任意にレベル ℓ を一つとる．このとき，型付けられたプログラムは，ℓ において等価な入力に対して，ℓ において等価な出力を返す」に丁度対応していることが見てとれる．

この定理の系として，次が成立する．

系 2.5.7. 各型 A に対応する等価関係 $\approx_{\ell}$ は，同値関係である．

証明. $\vdash^{\ell} M : A$ に，定理 2.5.6 を適用すれば，$M \approx_{\ell} M : A$ ．従って，$\approx_{\ell}$ は反射律を満たす．これと定理 2.5.2 より主張が成立. □

また，次の定理と上の一般化した非干渉性から，定理 2.4.2 の非干渉性を直ちに導くことができる．

定理 2.5.8. A がレベル ℓ において透過的な無関数型だとする．このとき A に対応する等価関係 $\approx_{\ell}$ は，レベル ℓ において型 A で型付けられる閉じた項上の簡約関係の反射的対称的推移的閉包に，一致する．

証明. 型 A の構造に関する帰納法. □

3 $\lambda_s^{\Box}$ から $\lambda^{\to}$へ

前節で定義した $\lambda_s^{\Box}$ を単純型付ラムダ計算 $\lambda^{\to}$に変換する．変換の基本的なアイデアは，「ℓ 以上の任意のレベルだけで使える型 A の項」を，「基底型 b_ℓ から $A^\dagger$ への関数型の項」とみなすことにある．

このようにすると，例えば，型 $\Box_\ell bool$ のデータは，型 $b_\ell \to bool$ のデータに変換される．変換後のデータの中身のブール値の部分を取り出すためには，b_ℓ 型の項が必要である．逆に，b_ℓ 型の項があれば中身を取り出せる．b_ℓ 型の項は，いわば，秘匿された情報を取り出すための “鍵” であり，b_ℓ 型の項を型付けることができることが，その “鍵” を持つ権限を有していること，つまり，秘匿された情報を取り出すことのできるレベルにいるということに相等すると言える．

また，b_ℓ 型には定数がないものとする．もしあれば，その定数に，エンコード後の型 $b_\ell \to bool$ の項を適用すれば中身を誰でも取り出せることになってしまうからである．[*3]

本節では，まず，変換先である $\lambda^{\to}$を説明し，次に変換を定義し，型付けと評価に関する変換の保存性を述べる．また，等価性/非等価性に関する保存性の証明が上手く行かない理由も論じる．

3.1 $\lambda^{\to}$：変換先

変換の準備として，まず，変換先である $\lambda^{\to}$とその閉じた項の上の等価性を定義し，それらの性質を紹介する．

定義 3.1.1 (型 B，項 N，コンテキスト Γ). $\lambda^{\to}$の型，項，コンテキストを次で定義する．

$$
\begin{array}{rcl}
B & ::= & b \mid unit \mid B \to B \mid B \times B \mid B + B \\
N & ::= & x \mid () \mid \lambda x{:}B.\, N \mid N\, N \mid \langle N,\, N\rangle \mid \pi_i(N) \mid \iota_i(N) \\
 & & \mid (\mathbf{case}\, N\, \mathbf{of}\, \iota_1(x_1) \Rightarrow N \mid \iota_2(x_2) \Rightarrow N) \\
\Gamma & ::= & \cdot \mid \Gamma,\, x : B
\end{array}
$$

型判断と型付け規則は通常の単純型付ラムダ計算と同じであり，簡約関係は，$\lambda_s^{\Box}$ の簡約関係から，$\Box_\ell$ に関わるものを除いたものである．

b は ($unit$ を除く) 基底型を表すメタ変数であり，b で表わされる基底型自体は，可算無限個あり，それらの基底型に関わる定数はないとする．これ以降，$unit$ は，基底型とは呼ばず，ユニット型と呼ぶことにする．

次に，等価性を定義するのだが，その前にいくつかの約束事を述べる．まず，ρ は，基底型に対し，基底型を含まない型と，その型を持つ正規形の閉じた項上の二項関係を割当てる基底型環境だとする．またこのとき，b^ρ は，基底型 b に対し ρ で割当てた型，もしくは，関係を表し，型 B^ρ は，B の中の基底型を ρ で割当てたそれぞれの型で置き換えたものだとする．

以上により，論理関係を用いて，等価性を定義する．

定義 3.1.2 (等価性： $N \approx N' : B \mid \rho$ ， $U \sim U' : B \mid \rho$). N, N' は，型 B^ρ を持つ閉じた項だとする．このとき，基底型環境 ρ の下で，項 N と N' が，型 B に対応する等価関係[*4]にあることを，$N \approx N' : B \mid \rho$ で表し，次で定義する．但し，$\sim$ は正規形の項に対する等価関係であり，U は正規形の

[*3] このような基底型は一種の型変数とみなせることに注意．実際，Tse ら [17] は，基底型ではなく，型変数を変換に用いている．

[*4] 常に同値関係であるとは限らないが，便宜上本論文では，そのように呼ぶことにする．

項を表すものとする．

$$\frac{(U, U') \in b^\rho}{U \sim U' : b \mid \rho} \quad \text{(STLCR-Var)}$$

$$() \sim () : \mathit{unit} \mid \rho \quad \text{(STLCR-Unit)}$$

$$\frac{\forall (N \approx N' : B_1 \mid \rho).\ \ U\,N \approx U'\,N' : B_2 \mid \rho}{U \sim U' : B_1 \to B_2 \mid \rho} \quad \text{(STLCR-Fun)}$$

$$\frac{U_1 \sim U_1' : B_1 \mid \rho \quad U_2 \sim U_2' : B_2 \mid \rho}{\langle U_1, U_2 \rangle \sim \langle U_1', U_2' \rangle : B_1 \times B_2 \mid \rho} \quad \text{(STLCR-Pair)}$$

$$\frac{U \sim U' : B_i \mid \rho}{\iota_i(U) \sim \iota_i(U') : B_1 + B_2 \mid \rho} \quad \text{(STLCR-Inj)}$$

$$\frac{N \longrightarrow^* U \quad N' \longrightarrow^* U' \quad U \sim U' : B \mid \rho}{N \approx N' : B \mid \rho} \quad \text{(STLCR-Term)}$$

代入に関し，いくつか有用な記法を導入する．

定義 3.1.3 (代入とその等価性：$\gamma \models^\rho \Gamma$, $\gamma \approx \gamma' : \Gamma \mid \rho$). γ が，基底型環境 ρ の下において，コンテキスト Γ への代入であることを，$\gamma \models^\rho \Gamma$ で表し，次のときに言う：$\forall x : B \in \Gamma$ に対し，$\vdash \gamma(x) : B^\rho$ ．

次に，$\gamma, \gamma' \models^\rho \Gamma$ としたとき，γ と γ' が基底型環境 ρ の下で等価であることを，$\gamma \approx \gamma' : \Gamma \mid \rho$ で表し，次のときに言う：$\forall x : B \in \Gamma$ に対し，$\gamma(x) \approx \gamma'(x) : B \mid \rho$.

上で定義した等価性に対し，次の定理が成立する．

定理 3.1.4 (論理関係の基本補題, [9]). $\Gamma \vdash N : B$ ，$\gamma \approx \gamma' : \Gamma \mid \rho$ ならば，

$$\gamma(N^\rho) \approx \gamma'(N^\rho) : B \mid \rho.$$

但し，N^ρ は，N の中の基底型を ρ で割当てたそれぞれの型に置き換えたものとする．

この定理の主張は，「任意に基底型環境を一つとる．このとき，型付けられたプログラムは，基底型環境の下で等価な入力に対して，基底型環境の下で等価な出力を返す」ということであり，非干渉性 (定理 2.5.6) の主張と対応していることに注意．

3.2 変換

$\lambda_s^\square$ から $\lambda^\to$ への変換を定義する．

先に述べた通り「ℓ 以上の任意のレベルでしか使うことのできない項」を基底型 b_ℓ からの関数型の項に変換する．そのような項は，2 種類ある．すなわち，型 $\square_\ell A$ を持つ項と，様相変数コンテキストで宣言された様相変数 $u ::^\ell A$ である．以下，まず，型とコンテキストを変換する．

定義 3.2.1 (型の変換). $A^\dagger$ を型 A の変換とし，次で定義する．

$$
\begin{array}{rcll}
\mathit{unit}^\dagger & = & \mathit{unit} & \text{LT-Unit} \\
(A_1 \times A_2)^\dagger & = & {A_1}^\dagger \times {A_2}^\dagger & \text{LT-Pair} \\
(A_1 + A_2)^\dagger & = & {A_1}^\dagger + {A_2}^\dagger & \text{LT-Sum} \\
(A_1 \to A_2)^\dagger & = & {A_1}^\dagger \to {A_2}^\dagger & \text{LT-Fun} \\
(\Box_\ell A)^\dagger & = & b_\ell \to A^\dagger & \text{LT-Box}
\end{array}
$$

定義 3.2.2 (コンテキストの変換). 様相変数コンテキストと通常変数コンテキストの変換は次で定義する.

$$
\begin{aligned}
(\Gamma_m)^\ddagger &\stackrel{\text{def}}{=} \{u : b_\ell \to A^\dagger \,|\, u ::^\ell A \in \Gamma_m\} \\
(\Gamma_o)^\dagger &\stackrel{\text{def}}{=} \{x : A^\dagger \,|\, x : A \in \Gamma_o\}
\end{aligned}
$$

次に項を変換する.

まず, 変換後, 型 $b_\ell \to A^\dagger (\equiv (\Box_\ell A)^\dagger)$ を持つ項の中身を取り出すために, "鍵" となる自由変数 $k_\ell : b_\ell$ を用意する. また, 変換前の型判断のレベルが ℓ であることを, 変換後のコンテキストに, この $k_\ell : b_\ell$ を追加することで表現する.

さらに, $\ell' \sqsubseteq \ell$ ならば, $b_{\ell'} \to {A'}^\dagger (\equiv (\Box_{\ell'} A)^\dagger)$ を持つ項の中身も取り出せる必要があるので, "鍵" k_ℓ を変換するための変換関数を表す自由変数 $k_{\ell\ell'} : b_\ell \to b_{\ell'}$ も用意して, 型判断の変換後のコンテキストに加える.

定義 3.2.3 (鍵とその変換関数). $\mathcal{L}_{kv} \stackrel{\text{def}}{=} \{k_\ell : b_\ell \,|\, \ell \in \mathcal{L}\}$, $\mathcal{L}_{kc} \stackrel{\text{def}}{=} \{k_{\ell\ell'} : b_\ell \to b_{\ell'} \,|\, \ell' \sqsubseteq \ell\}$

定義 3.2.4 (項の変換). $(\Gamma_{level},\, M)^{*\ell}$ を, レベル ℓ におけるレベルコンテキスト Γ_{level} の下での項 M の変換とし, 次で定義する. 但し, Γ_{level} は, 宣言 $u : \ell$ の列である.

$$
\begin{array}{rcll}
(\Gamma_{level},\, x)^{*\ell} & = & x & \text{LE-Ovar} \\
(\Gamma_{level},\, u)^{*\ell} & = & u\,(k_{\ell\ell'}\, k_\ell), \quad \mathit{if} \quad u : \ell' \in \Gamma_{level},\, \ell' \sqsubseteq \ell & \text{LE-Mvar} \\
(\Gamma_{level},\, ())^{*\ell} & = & () & \text{LE-Unit} \\
(\Gamma_{level},\, \lambda x{:}A.\, M)^{*\ell} & = & \lambda x{:}A^\dagger.\, (\Gamma_{level},\, M)^{*\ell} & \text{LE-Fun} \\
(\Gamma_{level},\, M_1\, M_2)^{*\ell} & = & (\Gamma_{level},\, M_1)^{*\ell}\, (\Gamma_{level},\, M_2)^{*\ell} & \text{LE-App} \\
(\Gamma_{level},\, \langle M_1,\, M_2\rangle)^{*\ell} & = & \langle (\Gamma_{level},\, M_1)^{*\ell},\, (\Gamma_{level},\, M_2)^{*\ell}\rangle & \text{LE-Pair} \\
(\Gamma_{level},\, \pi_i(M))^{*\ell} & = & \pi_i((\Gamma_{level},\, M)^{*\ell}) & \text{LE-Proj} \\
(\Gamma_{level},\, \iota_i(M))^{*\ell} & = & \iota_i((\Gamma_{level},\, M)^{*\ell}) & \text{LE-Inj} \\
(\Gamma_{level},\, (\mathbf{case}\, M_1\, \mathbf{of}\, \iota_1(x_1) \Rightarrow M_2 \,|\, \iota_2(x_2) \Rightarrow M_3))^{*\ell} & & & \\
= \quad (\mathbf{case}\, (\Gamma_{level},\, M_1)^{*\ell}\, \mathbf{of}\, \iota_1(x_1) \Rightarrow (\Gamma_{level},\, M_2)^{*\ell} \,|\, \iota_2(x_2) \Rightarrow (\Gamma_{level},\, M_3)^{*\ell}) & & & \text{LE-Case} \\
(\Gamma_{level},\, \mathbf{box}_{\ell'}\, M)^{*\ell} & = & \lambda k_{\ell'}{:}b_{\ell'}.\, (\Gamma_{level},\, M)^{*\ell'} & \text{LE-Box} \\
(\Gamma_{level},\, \mathbf{let}\, \mathbf{box}_{\ell'}\, u \,=\, M_1\, \mathbf{in}\, M_2)^{*\ell} & = & [(\Gamma_{level},\, M_1)^{*\ell}/u](\Gamma_{level} + \{u : \ell'\},\, M_2)^{*\ell} & \text{LE-LetBox}
\end{array}
$$

LE-Mvar において, 様相変数 u の変換は, Γ_{level} において宣言されている u のレベル ℓ' と, 変換の際に使える "鍵" のレベル ℓ に依存する.

例 3.2.5. 例 2.2.2 の記法を用いる. $u ::^{\mathsf{H}} \mathit{bool},\, u' ::^{\mathsf{L}} \mathit{bool}\,;\, x : \Box_{\mathsf{H}} \mathit{bool} \vdash^{\mathsf{L}} \mathbf{box}_{\mathsf{H}}\, u : \Box_{\mathsf{H}} \mathit{bool}$ のコンテキ

ストと項の変換はそれぞれ次のようになる．

$$\{u ::^{\mathsf{H}} bool,\, u' ::^{\mathsf{L}} bool\}^{\ddagger} = \{u : b_{\mathsf{H}} \to bool,\, u' : b_{\mathsf{H}} \to bool\}$$
$$\{x : \Box_{\mathsf{H}} A\}^{\dagger} = \{x : b_{\mathsf{H}} \to A^{\dagger}\}$$
$$(\{u : \mathsf{H},\, u' : \mathsf{L}\},\, \mathbf{box}_{\mathsf{H}}\, u)^{*\,\mathsf{L}} = \lambda k_{\mathsf{H}} : b_{\mathsf{H}}.\, u\,(k_{\mathsf{HH}}\, k_{\mathsf{H}})$$

また，変換後にコンテキストに追加される “鍵” とその変換関数は，

$$k_{\mathsf{L}} : b_{\mathsf{L}},\, k_{\mathsf{HL}} : b_{\mathsf{HL}} \to b_{\mathsf{HL}},\, k_{\mathsf{LL}} : b_{\mathsf{LL}} \to b_{\mathsf{LL}},\, k_{\mathsf{HH}} : b_{\mathsf{HH}} \to b_{\mathsf{HH}}$$

3.3 変換の保存性：型付けと評価

この変換に関して，型付けと評価が保存されることを述べる．

$\mathcal{L}(\Gamma_m)$ を様相変数コンテキスト Γ_m から型情報だけを除去した形のレベルコンテキストを表すとする．このとき，型付けが次の形で保存される．

定理 3.3.1 (変換の型付け保存性). $\Gamma_m;\, \Gamma_o \vdash^{\ell} M : A$ ならば，$\mathcal{L}_{kc},\, k_\ell : b_\ell,\, \Gamma_m^{\ddagger},\, \Gamma_o^{\dagger} \vdash (\mathcal{L}(\Gamma_m),\, M)^{*\,\ell} : A^{\dagger}$

証明. 型判断　$\Gamma_m;\, \Gamma_o \vdash^{\ell} M : A$　の構造に関する帰納法. □

“鍵” の変換関数を表す自由変数の具体化を代入で行う．その結果が合成に関して可換になるように，次のような代入に関する条件を設ける．

定義 3.3.2 (定数可換). $\gamma_{kc} \models^{\rho} \mathcal{L}_{kc}$ とする．このとき，γ_{kc} が定数可換であるとは，任意のレベル ℓ' に対し，ある閉じた $\lambda^{\to}$ 項 N が存在して，任意の $k_{\ell\ell'} : b_\ell \to b_{\ell'} \in \mathcal{L}_{kc}$ に対し，$\gamma_{kc}(k_{\ell\ell'}) = \lambda k_\ell : b_\ell{}^{\rho}.\, N$ のときに言い，$\gamma_{kc} \models^{\rho}_{cc} \mathcal{L}_{kc}$ で表す．

また，部分項 $k_{\ell\ell'}\, M$ の代入後に生じる簡約基 $\gamma_{kc}(k_{\ell\ell'})\, \gamma_{kc}\, M$ の簡約を $\longrightarrow_{cc}$ で表す．

このとき，任意の $\ell_1 \sqsubseteq \ell_2 \sqsubseteq \ell_3$ に対し，ある N があって，$\gamma_{kc}(k_{\ell_3\ell_1})\, x \longrightarrow_{cc} N$ かつ $\gamma_{kc}(k_{\ell_3\ell_2})\,(\gamma_{kc}(k_{\ell_2\ell_1})\, x) \longrightarrow_{cc} N$．よって具体化した変換関数の可換性が成立する．

この可換性により，以下の定理で述べるように，項の変換結果の値が，レベルによらないで一意に決まる．この証明は，変換が，レベルに依存する部分は，様相変数の場合 LE-MVAR だけであることことから自明である．

定理 3.3.3 (変換のレベル普遍性). $(\mathcal{L}_{level},\, M)^{*\,\ell}$ が存在し，$\ell \sqsubseteq \ell'$ かつ $\gamma_{kc} \models^{\rho}_{cc} \mathcal{L}_{kc}$ ならば，ある項 N があって，$\gamma_{kc}\,((\mathcal{L}_{level},\, M)^{*\,\ell})^{\rho} \longrightarrow^{*}_{cc} N,\, \gamma_{kc}\,((\mathcal{L}_{level},\, M)^{*\,\ell'})^{\rho} \longrightarrow^{*}_{cc} N$.

さらに，定数可換な変換関数の下，変換は次の形で評価を保存する．

定理 3.3.4 (変換の簡約保存性). $\Gamma_m;\, \Gamma_o \vdash^{\ell} M : A$ ，$M \longrightarrow M'$ かつ $\gamma_{kc} \models^{\rho}_{cc} \mathcal{L}_{kc}$ ならば，ある $\lambda^{\to}$ 項 $N,\, N'$ が存在して，$\gamma_{kc}\,((\mathcal{L}(\Gamma_m),\, M)^{*\,\ell})^{\rho} \longrightarrow^{*}_{cc} N' \longrightarrow^{*} N$，$\gamma_{kc}\,((\mathcal{L}(\Gamma_m),\, M')^{*\,\ell})^{\rho} \longrightarrow^{*}_{cc} N$.

但し，$N,\, N'$ は簡約 $\longrightarrow_{cc}$ に関して正規形である．

証明. 項 M の構造に関する帰納法．簡約の定義に従い場合分け. □

	$\mathbf{box}_{\ell'}\, V \sim_\ell \mathbf{box}_{\ell'}\, V' : \Box_{\ell'} A'$ となる条件	$\lambda x\!:\!unit.\, U \sim \lambda y\!:\!unit.\, U' : b_{\ell'} \to A'^{\dagger} \mid \rho_\ell$ となる条件
$\ell' \sqsubseteq \ell$	$V \sim_\ell V' : A'$	$[()/x]U \approx [()/y]U' : A'^{\dagger} \mid \rho_\ell$
$\ell' \not\sqsubseteq \ell$	なし	なし

表 1　$\lambda_s^{\Box}$ と $\lambda^{\to}$の等価性の対応

3.4　考察：等価性/非等価性の保存性

本節では，まず，各レベル ℓ に対応した基底型環境を構築し，その環境の下での等価性が，$\lambda_s^{\Box}$ のレベル ℓ における等価性に対応して見えることを説明する．次に，変換において，それらの等価性が同値であるという主張を述べ，その証明を考察し，問題点を指摘する．

変換が等価性/非等価性を保存するように，$\lambda^{\to}$における基底型環境を次のように定める．

定義 3.4.1 (レベル ℓ に対応する基底型環境 ρ_ℓ). 基底型環境 ρ_ℓ が基底型に割当てる型は，すべて $unit$ 型だとする．ρ_ℓ が基底型 $b_{\ell'}$ に割当てる関係を次のようにとる．

$$b_{\ell'}^{\rho_\ell} \stackrel{\text{def}}{=} \begin{cases} \emptyset & \ell' \not\sqsubseteq \ell \text{ の場合} \\ unit \times unit & \ell' \sqsubseteq \ell \text{ の場合} \end{cases}$$

但し，$\emptyset$ は空の関係を，$unit \times unit$ は定数項 () 同士の関係を，それぞれ表すとする．

このとき，$\lambda_s^{\Box}$ でのレベル ℓ における型 A に対応する等価関係と，$\lambda^{\to}$での基底型環境 ρ_ℓ における型 $A^{\dagger}$ に対応する等価関係が，対応する．特に，型 A が $\Box_{\ell'} A'$ という形の場合が，重要であり，2 つの等価性の定義が対応していることを，表 1 から見てとることができる．$(\Box_{\ell'} A')^{\dagger} = b_{\ell'} \to A'^{\dagger}$, $(\Gamma_{level}, \mathbf{box}_{\ell'}\, M)^{*\,\ell} = \lambda k_{\ell'}\!:\!b_{\ell'}.\, (\Gamma_{level}, M)^{*\,\ell'}$ であることに注意,

変換において等価性/非等価性が保存されるという主張を述べる前に，変換後の “鍵” とその変換関数の具体化を定義する．

定義 3.4.2 (鍵への代入). $\gamma_{keys} \models^{\rho_\ell} \mathcal{L}_{kc}, \mathcal{L}_{kv}$ を次で定義する．

$$\gamma_{keys}(x) \stackrel{\text{def}}{=} \begin{cases} \lambda y\!:\!unit.\, () & x = k_{\ell\ell'} \text{ の場合} \\ () & x = k_\ell \text{ の場合} \end{cases}$$

明らかに，$\gamma_{keys} \models_{cc}^{\rho_\ell} \mathcal{L}_{kc}$ であることに注意.

このとき次の主張を示したい.

主張 3.4.3 (等価性/非等価性の保存). $M \approx_\ell M' : A$ *iff* $\gamma_{keys}\,((\cdot, M)^{*\,\ell})^{\rho_\ell} \approx \gamma_{keys}\,((\cdot, M')^{*\,\ell})^{\rho_\ell} : A^{\dagger} \mid \rho_\ell$

この主張が示せれば，一般化した非干渉性 (定理 2.5.6) を，論理関係の基本補題 (定理 3.1.4) に次のように帰着することによって，間接的に証明できることが分かっている：(1) 等価性の保存性より，非干渉性の条件を変換した結果は，論理関係の基本補題の条件を満たすことが分かる．(2) 変換結果に論理関係の基本補題を適用する．(3) 非等価性の保存性により，(2) で得られた結果から非干渉性の結論が導ける．

主張 3.4.3 に出てくる等価関係はともに，論理関係を用いて定義されている．それゆえ，その証明には型に関する帰納法を使うのが自然であると思われる．そこで，閉じた $\lambda_s^{\Box}$ 項と $\lambda^{\to}$項の間に，変換による項の対応含む論理関係 $M \Rightarrow_\ell N : A$ を定義し，主張 3.4.3 の代わりに，次の主張を示すことを試みる．

主張 3.4.4. $M \Rightarrow_\ell N : A$, $M' \Rightarrow_\ell N' : A$ とする．このとき，

$$M \approx_\ell M' : A \ \ \textit{iff} \ \ N \approx N' : A^\dagger \mid \rho_\ell \ .$$

しかし，上の証明方針には問題がある．主張の左から右を示すのに，型に関する帰納法の仮定が，関数型の場合は，上手く使えないという問題である．

この問題をもう少し詳しく説明する．まず， $V \Rightarrow_\ell U : A_1 \to A_2$, $V' \Rightarrow_\ell U' : A_1 \to A_2$, $V \sim_\ell V' : A_1 \to A_2$ と仮定する．示したいことは， $U \sim U' : A_1^\dagger \to A_2^\dagger \mid \rho_\ell$ となる．これを言うには，関数型に対応する論理関係の定義から，任意の $N \sim N' : A_1^\dagger \mid \rho_\ell$ に対し， $U\,N \approx U'\,N' : A_2^\dagger \mid \rho_\ell$ であることを示せば十分である．しかし，ここで，型 A_1 と A_2 の帰納法の仮定を使うことができない．なぜなら，N と N' に対し，必ずしも $M \Rightarrow_\ell N : A_1$, $M' \Rightarrow_\ell N' : A_1$ を満たすような $\lambda_s^{\Box}$ 項 M，M' があるとは限らないからである (このような項がある性質を仮に全射性と呼ぶことにする)．もし，このような M と M' があれば，型 A_1 の帰納法の仮定の右から左を使うことによって，$M \approx_\ell M' : A_1$ が言え，関数型に対応する論理関係の定義と，型 A_2 の帰納法の仮定から， $U\,N \approx U'\,N' : A_2^\dagger \mid \rho_\ell$ を示すことができる．

まとめると，変換を含み，かつ，全射性を満たすような論理関係 $\Rightarrow_\ell$ があれば，変換の等価性/非等価性の保存性を証明することができる．

しかし，本論文で定義した変換では，この全射性は満たされないことが分かっている．例えば，$\ell_1 \not\sqsubseteq \ell_2$ とすると，型 $\Box_{\ell_1}\mathsf{void} \to \Box_{\ell_1}\Box_{\ell_2}\mathsf{void}$ (ただし，void は空集合に対応する型) で型付けられる閉じた $\lambda_s^{\Box}$ 項は存在しないが，変換後の型 $(b_{\ell_1} \to \mathsf{void}) \to b_{\ell_1} \to b_{\ell_2} \to \mathsf{void}$ で型付可能な閉項は明らかに存在する．従って，全射性への反例になっていることが分かる．

このことに関しては，まだ憶測の段階だが，変換が，型付け規則 T-Box の前提条件「通常変数コンテキストは空」を考慮していないこと，つまり, $\mathbf{box}_\ell M$ の中身は，ℓ 以上のレベルでならば，型判断のレベルに依存せず，どこでも同じように，使うことができなければならないということを，変換自体に反映させきれていないことに原因があるのではないかと推測している．これに対する解決案として，Washburn, Weirich[21] による高階抽象構文の F_ω への変換が参考にならないかと考えている．Washburn らは，直観主義的な必然性に対応する様相型の変換を，F_ω における型抽象で実現しており，様相型の前提条件「変数コンテキストは空」が，型抽象の前提条件「変数コンテキストに型抽象に用いる型変数が現れていない」に丁度対応することを示している．

4 関連研究

型システムに基づく情報流解析の非干渉性 型システムに基づく情報流解析の非干渉性の証明には，様々なものがある．例えば，Heintze と Riecke [5] や Abadi ら [1] は，表示的意味論を利用し， SLam に対する非干渉性を示した．Pottier と Simonet [13] は，特殊な操作的意味論を用いることで，Core ML に対する非干渉性を証明した．さらに，Miyamoto と Igarashi [10] は，非決定的かつ full な簡約システムの下では，$\lambda_s^{\Box}$ に対する非干渉性が，特定の型の場合には，単純な簡約意味論だけに基づいて導けることを示した．また，Sumii らは，cryptographic λ-calculus において，項モデル上の論理関係によって，暗号の秘匿性を定式化し証明し

た [16]．Tse らも同様に項モデル上の論理関係によって，DCC [1] の非干渉性を定式化した [17]．本研究では，Sumii ら，Tse らの研究を参考に，$\lambda_s^{\Box}$ において，論理関係を用いることで，一般の型判断で型付けられる項に対しても，同様の非干渉性が成立することを証明した．

$\lambda^{\to}$への変換とその論理関係の保存性　Tse らは，DCC から System F への変換を定義した [17]．その変換は，Tse ら自身がその論文の中で述べている通り，型抽象・型適用を用いておらず，実質，$\lambda^{\to}$への変換となっている．

Tse らは，DCC から System F への変換により，DCC における非干渉性を System F における parametricity (本研究では，このかわりに，論理関係の基本補題を用いた) に帰着して証明することを試みた (証明の詳細は，上記論文のテクニカルレポート版 [18] にある)．しかし，その証明は，本論文で考察したものと同様な問題点を含んでおり，変換の全射性も示されていないため，非干渉性の証明にはなっていない．

5　おわりに

我々は，$\lambda_s^{\Box}$ において，論理関係によって等価性を定義し，それにより非干渉性を一般化し，証明した．また，$\lambda_s^{\Box}$ から 単純型付ラムダ計算 $\lambda^{\to}$への変換を定義し，それが型付けと評価を保存することを示した．しかし，変換において，等価性が$\lambda^{\to}$における等価性として保存されることの証明は，現時点ではできていない．その他に，我々は変換の全射性が成立しないことを指摘した．

今後の課題として，p. 13 最後で述べた予想に基づき，より忠実な変換の構成に取り組みたいと考えている．

謝辞

論理関係の保存性の問題に関し，快く相談に乗って下さり，示唆に富むアドバイスをして頂いた東北大学の住井英二郎氏と，有益なコメントを下さった査読者の方々に，心より感謝致します．

参考文献

[1] Martín Abadi, Anindya Banerjee, Nevin Heintze, and Jon G. Riecke. A core calculus of dependency. In *POPL '99: Proceedings of 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 147–160, New York, NY, USA, 1999. ACM Press.

[2] Anindya Banerjee and David A. Naumann. Secure information flow and pointer confinement in a Java-like language. In *CSFW '02: Proceedings of 15th IEEE Computer Security Foundations Workshop*, pp. 253–267, 2002.

[3] Gilles Barthe and Bernard P. Serpette. Partial evaluation and non-interference for object calculi. In A. Middeldorp and T. Sato, editors, *Fuji International Symposium on Functional and Logic Programming*, Vol. 1722 of *Lecture Notes in Computer Science*, pp. 53–67, Tsukuba, Japan, 1999. Springer-Verlag.

[4] J. Goguen and J. Meseguer. Security policies and security models. In *Proceedings of IEEE Symposium on Security and Privacy*, pp. 11–20, 1982.

[5] Nevin Heintze and Jon G. Riecke. The SLam calculus: programming with secrecy and integrity.

In *POPL '98: Proceedings of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 365–377, 1998.

[6] Kohei Honda, Vasco Thudichum Vasconcelos, and Nobuko Yoshida. Secure information flow as typed process behaviour. In Gert Smolka, editor, *ESOP '00: Proceedings of European Symposium on Programming*, Vol. 1782 of *Lecture Notes in Computer Science*, pp. 180–199. Springer, 2000.

[7] Kohei Honda and Nobuko Yoshida. A uniform type structure for secure information flow. In *POPL '02: Proceedings of 29th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 81–92, 2002.

[8] Naoki Kobayashi. Type-based information flow analysis for the pi-calculus. *Acta Informatica*, Vol. 42, No. 4-5, pp. 291–347, 2005.

[9] John C. Mitchell. *Foundations for Programming Languages*. MIT Press, 1996.

[10] Kenji Miyamoto and Atsushi Igarashi. A modal foundation for secure information flow. In *FCS '04: Proceedings of Workshop on Foundations of Computer Security*, pp. 187–203, June 2004.

[11] Andrew C. Myers. Jflow: Practical mostly-static information flow control. In *POPL'99: Proceedings of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 228–241, 1999.

[12] François Pottier. A simple view of type-secure information flow in the pi-calculus. In *CSFW '02: Proceedings of IEEE Computer Security Foundations Workshop*, pp. 320–330, 2002.

[13] François Pottier and Vincent Simonet. Information flow inference for ML. *ACM Transactions on Programming Languages and Systems*, Vol. 25, No. 1, pp. 117–158, 2003.

[14] Andrei Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications, special issue on Formal Methods for Security*, Vol. 21, No. 1, pp. 5–19, January 2003.

[15] Geoffrey Smith and Dennis M. Volpano. Secure information flow in a multi-threaded imperative language. In *POPL '98: Proceedings of ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 355–364, 1998.

[16] Eijiro Sumii and Benjamin C. Pierce. Logical relations for encryption. *Journal of Computer Security*, Vol. 11, No. 4, pp. 521–554, 2003.

[17] Stephen Tse and Steve Zdancewic. Translating dependency into parametricity. In *ICFP '04: Proceedings of 9th ACM SIGPLAN International Conference on Functional Programming*, pp. 115–125, New York, NY, USA, 2004. ACM Press.

[18] Stephen Tse and Steve Zdancewic. Translating dependency into parametricity. Technical Report MIS-CIS-04-01, University of Pennsylvania, 2004. Extended version of [17].

[19] Dennis M. Volpano, Cynthia E. Irvine, and Geoffrey Smith. A sound type system for secure flow analysis. *Journal of Computer Security*, Vol. 4, No. 2/3, pp. 167–188, 1996.

[20] Dennis M. Volpano and Geoffrey Smith. A type-based approach to program security. In Michel Bidoit and Max Dauchet, editors, *Proceedings of International Joint Conference on the Theory and Practice of Software Deveopment*, Vol. 1214 of *Lecture Notes in Computer Science*, pp. 607–621, 1997.

[21] Geoffrey Washburn and Stephanie Weirich. Boxes go bananas: Encoding higher-order abstract syn-

tax with parametric polymorphism. In *Proceedings of the Eighth ACM SIGPLAN International Conference on Functional Programming*, pp. 249–262, Uppsala, Sweden, August 2003. ACM SIGPLAN.