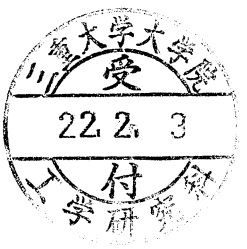
修 士 論 文

型付けによる
ファイル間データ非干渉の検証



平成21年度
三重大学大学院工学研究科
情報工学専攻 博士前期課程
計算機ソフトウェア研究室

座礼 晃一

指導教員 山田俊行

目 次

まえがき

個人情報がコンピュータで管理されるようになり，その管理・操作を行うプログラムには，個人情報漏れを起こさないことが求められる．プログラムが情報漏れを生じないことを保証するために，情報流解析という解析手法を用いて，その解析から得られた情報が非干渉性を満たすかを判定する検証法が用いられる．情報流解析とは，情報の依存先を求める解析であり，非干渉性とは，機密情報を変化させても公開情報に影響がない (変化しない)，つまり公開情報が機密情報に依存しないことを示す性質である．情報流解析に型システムを用いた研究に論文 [GJ05, SM03, Kob03, VS97] がある．[GJ05] では機密度が高い，低いという 2 種類の型情報を用いて，機密度の低い情報が高い情報に依存しない，つまり機密度が高い情報を変化させても，低い情報は変化しないという性質を，プログラムが満たすことを保証する体系を示した．しかし，2 種類の機密度だけでは，検出できない情報漏れが存在する．それを検出するためには機密度を詳細化する必要がある．そこで，本研究では [GJ05] の体系を拡張し，ファイルを扱えるようにした．ファイルにはアクセス権限を設定するので，各ファイルに対してアクセス可能なユーザの集合が定まる．本研究ではユーザの集合を型情報に追加し，ファイル間での情報漏れも起きないことを保証できる体系を提示する．新しく提示する定理は，プログラムが，情報漏れを生じないことを示しており，これは [GJ05] の定理にファイルの概念を取り入れたものである．これは入力プログラムに対して，ファイルのアクセス権限を持つユーザ間でも情報漏れが起きないことを意味する．

第1章　目的

一般に非干渉性とは，データ間に依存関係がないことを表す性質である．機密度の高低を定義し，データをどちらかの機密度に分類することで，機密度の高いデータの変化により，低いデータが変化していないことを示すことに非干渉性は用いられる．ここで機密度の低いデータとは，誰にでも取得可能なデータを指し，高いデータとは特定の者にしか取得できないデータを指す．つまり非干渉でない（干渉する）ということは，誰にでも取得可能なデータから，特定の者にしか取得できないデータを推測可能ということになる．次に，データ間に依存関係がある，簡単なプログラム例を示す．なお変数 l は機密度の低いデータ，h は高いデータを表す．

(1) $l := h$

(2) **if** $h = 1$ **then** l :=2 **else** l :=3 **end**

(1) は機密度の高いデータを直接代入するため，l は h に依存している．つまり l のデータを取得することで h のデータがわかる．(2) は h を直接代入してはいないが，h の値次第で 1 の値に変化が起きてしまう．つまり l のデータを取得することで，条件式 $h = 1$ の真偽が判明する．これにより h のデータ全体ではないものの，一部がわかることになる．これらの依存関係を解析する方法として情報流解析が挙げられる．情報流解析において，(1) では h の値がそのまま l へ流れるので，h から l へ直接的な情報流があるといい，(2) では h の一部 (条件式を満たす値か否か) が l へ流れるため，h から l へ間接的な情報流があるという．情報流解析に類似した解析法にデータフロー解析があるが，データフロー解析では変数間のデータのやりとりを解析するため，(1) の直接的な情報流しか解析できない．このため本研究では情報流解析を用いる．そして，情報流解析には型情報を用いる．変数に付加情報としての型を与え，それを型規則で型付けし，解析を行なう．詳しくは 5 章で述べる．

今回提案する検証法でも，上記プログラムは非干渉性を満たさないものとして扱う．(2) のようなプログラムは，ログイン等の認証において，ID が一致するかどうかでよく利用されるため，そのようなプログラムを検証する場合には，その箇所で情報漏れが生じるという認識が必要である．

機密度の高低は，そのプログラム内でのみ有用な情報であり，プログラムが外部情報にアクセスする場合，その外部情報においても機密度の高低の定義が必要である．例えば外部情報としてファイル $file_1$，$file_2$ を考える．$file_1$ と $file_2$ は，プログラム内では機密度の低い情報とし，ファイルへのアクセスを読み込み，ファイルへの代入を書き込みとして，次のプログラムを考える．

$$file_2 := file_1 + 2$$

これは $file_1$ の情報を読み取って，2 を加算したものを $file_2$ に書き込むプログラムである．つまり $file_1$ の情報が $file_2$ に流れているが，両方ともに機密度が低いので，情報漏れは起きないと [GJ05]

では判定される．しかし，そのファイルのデータを読み取れる者は，通常アクセス権限で設定されており，アクセス権限間での情報漏れを考慮しないので，プログラム内で情報漏れが起きずとも，ファイル間で情報漏れが起きる場合がある．本研究の目的は，[GJ05] の体系にファイルのアクセス権限の概念を導入することで，プログラム内だけでなく，外部情報（ファイル）に関しても情報漏れが起きないことを保証できるように拡張する．

第2章　準備

この章では，扱うプログラム及び，そのプログラムが非干渉性を満たすことを示すために必要な諸定義を示す．

2.1　入力プログラム

入力プログラムの式 e，文 S の構文 [GJ05] を以下に示す．

$$\begin{aligned} e &::= e\ \mathbf{op}\ e \mid id \mid c \\ S &::= \mathbf{if}\ e\ \mathbf{then}\ S\ \mathbf{else}\ S\ \mathbf{end} \\ &\mid \mathbf{while}\ e\ \mathbf{do}\ S\ \mathbf{done} \\ &\mid S; S \mid \mathbf{skip} \mid id := e \end{aligned}$$

c は整数，op は二項演算を表す．if は条件文，while は反復文であり，skip は何もしない文である．id は変数集合 ID の要素であるが，本研究ではファイル変数という概念を扱う．ファイル変数とは，アクセスできるユーザが設定されている変数である．プログラム中でのファイル変数の参照は，ファイル情報の読み取りに当たり，ファイル変数への代入は，ファイルへの書き込みに当たる．なお，ファイル変数集合 F は変数集合 ID の部分集合である．

2.2　値の定義

値には整数集合 $\mathbb{Z}$ と印付き整数集合 $\mathbb{Z}^\sharp$ を用いる．$\mathbb{Z}^\sharp$ を

$$\mathbb{Z}^\sharp = \{x^\sharp | x \in \mathbb{Z}\}$$

と定義する．$\mathbb{Z}^\sharp$ は，機密度の高いデータ集合を表し，$\mathbb{Z}$ は機密度の低いデータ集合を表す．[GJ05] では機密度を変数の型情報に割り当てていたが，本研究では，機密度の高低を型情報に割り当てず，値に割り当てる．なお，$^\sharp$ の有無に関わらず，値は通常の整数と同様の処理で求める．例えば加算の場合は，$+(1, 1^\sharp) = 2^\sharp$ となる．この演算の定義は，式評価用関数 valExp にて明示する．

2.3　型情報の定義

本研究での変数の型はアクセス可能なユーザ集合を表す．例えば変数 id が型情報 $\{u_1, u_2\}$ を持つとき，変数 id はユーザ u_1 と u_2 が読み取れることを表す．ファイル変数にアクセス可能なユー

ザ集合を返す写像を

$$\mathrm{ACC} \quad : \quad \mathrm{F} \to \mathcal{P}(\mathrm{USER})$$

で表す.

2.4 ストアの定義

値ストアσ は変数から値への写像,

$$\sigma \quad : \quad \mathrm{ID} \to \mathbb{Z} \cup \mathbb{Z}^{\sharp}$$

であり，**タグストア**ρ は変数から型への写像

$$\rho \quad : \quad \mathrm{ID} \to \mathcal{P}(\mathrm{USER})$$

である.

検証のために，変数には予め値の機密度と型情報を設定する必要がある．変数全体を入力時に機密度の高い情報を格納する変数，低い情報を格納する変数に分割し，それぞれ集合 H_i，L_i で表す．また，出力時に機密度の高い情報を格納してほしい変数 H_o，機密度低い情報を格納してほしい変数 L_o に分割する．そして値ストア，タグストアに関する関数 $\mathrm{init_v}$，$\mathrm{init_t}$ を次のように定義する．

$$\mathrm{init_v}(\sigma)(x) \quad = \quad \begin{cases} (\sigma(x))^{\sharp} & (x \in H_i) \\ \sigma(x) & (x \notin H_i) \end{cases}$$

$$\mathrm{init_t}(\rho)(x) \quad = \quad \begin{cases} \mathrm{ACC}(x) & (x \in \mathrm{F}) \\ \mathrm{USER} & (x \notin \mathrm{F}) \end{cases}$$

$\mathrm{init_v}$ は H_i に属する変数の値を，機密度の高い値として設定する（$^{\sharp}$をつける）関数である．$\mathrm{init_t}$ は各変数がファイル変数にはアクセス権限を持つユーザ集合，それ以外の変数にはユーザ集合全体（全ユーザがアクセス権限を持つ）を割り当てるようにタグストアを設定する関数である．検証には上記関数を適用した値ストア，タグストアを用いる.

2.5 評価関数と規則

ここでは，値の評価と型付けのための評価関数及び規則を提示する．[GJ05] とは異なり，値ストアの規則と，タグストアの規則を分けて定義する.

文実行前の値ストアから文実行後の値ストアを求めるための評価規則を図 2.1 に示す．規則（$\mathrm{ASS_{sec}}$），（$\mathrm{ASS_{pub}}$）は本研究で追加した規則であり，規則（IF）は [GJ05] で提示されている規則と類似した処理をしている．（$\mathrm{ASS_{sec}}$），（$\mathrm{ASS_{pub}}$）で用いられる表記 $\left[id \mapsto v\right]\sigma$ は次の条件を満たす.

$$\left[id \mapsto v\right]\sigma(id) = v$$

$$\mathrm{valExp}(\sigma, e) = v$$

$$\frac{\sigma; \mathrm{v}^{\mathrm{pc}} \vee \mathrm{valSec}(v) \vdash_{\mathrm{v}} S_v \Downarrow \sigma_1}{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} S \Downarrow \mathrm{noExe}(\sigma_1, \mathrm{valSec}(v), S_{\neg v})} \quad \text{(IF)}$$

ただし，$S = \mathbf{if}\, e\, \mathbf{then}\, S_{true}\, \mathbf{else}\, S_{false}\, \mathbf{end}$

$$\frac{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} \mathbf{if}\ e\ \mathbf{then}\ S\ ;\ \mathbf{while}\ e\ \mathbf{do}\ S\ \mathbf{done}\ \mathbf{else}\ \mathbf{skip}\ \mathbf{end} \Downarrow \sigma'}{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} \mathbf{while}\, e\, \mathbf{do}\, S\, \mathbf{done} \Downarrow \sigma'} \quad \text{(WHILE)}$$

$$\frac{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} S_1 \Downarrow \sigma' \qquad \sigma'; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} S_2 \Downarrow \sigma''}{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} id := e \Downarrow \sigma''} \quad \text{(SEQ)}$$

$$\frac{\mathrm{valExp}(\sigma, e) = v \qquad \mathrm{v}^{\mathrm{pc}} = \mathbf{true}}{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} id := e \Downarrow \left[id \mapsto (v)^{\sharp}\right]\sigma} \quad (\mathrm{ASS}_{\mathrm{sec}})$$

$$\frac{\mathrm{valExp}(\sigma, e) = v \qquad \mathrm{v}^{\mathrm{pc}} = \mathbf{false}}{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} id := e \Downarrow \left[id \mapsto v\right]\sigma} \quad (\mathrm{ASS}_{\mathrm{pub}})$$

$$\frac{}{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash_{\mathrm{v}} \mathrm{skip} \Downarrow \sigma} \quad \text{(SKIP)}$$

図 2.1: 評価規則

つまり，値ストアの内容を置き換える表記である．例えば，$\sigma(x) = 0$ を満たす値ストア σ に対して，$[x \mapsto 1]\sigma$ と表記すると，$[x \mapsto 1]\sigma(x) = 1$ となる．また，評価規則に用いられる表記 $\sigma; v^{pc} \vdash_{\mathrm{v}} S \Downarrow \sigma_1$ は，σ, v^{pc} の元で，文 S を実行すると，値ストア σ_1 を得ることを意味する．

$$
\begin{aligned}
\mathrm{valExp}(\sigma, c) &= c \\
\mathrm{valExp}(\sigma, id) &= \sigma(id) \\
\mathrm{valExp}(\sigma, e_1 \,\mathbf{op}\, e_2) &= op^\sharp(\mathrm{valExp}(\sigma, e_1), \mathrm{valExp}(\sigma, e_2))
\end{aligned}
$$

$$
op^\sharp(x, y) = \begin{cases} op(x, y) & (x, y \in \mathbb{Z}) \\ (op(a, y))^\sharp & (a \in \mathbb{Z}, (a)^\sharp = x) \\ (op(x, a))^\sharp & (a \in \mathbb{Z}, (a)^\sharp = y) \\ (op(a, b))^\sharp & (a, b \in \mathbb{Z}, (a)^\sharp = x, (b)^\sharp = y) \end{cases}
$$

$$
(x)^\sharp = \begin{cases} x^\sharp & (x \notin \mathbb{Z}^\sharp) \\ x & (\text{それ以外}) \end{cases}
$$

$$
\mathrm{valSec}(\sigma, v) = \begin{cases} \mathbf{true} & (v \in \mathbb{Z}^\sharp) \\ \mathbf{false} & (v \notin \mathbb{Z}^\sharp) \end{cases}
$$

$op^\sharp$ は 2 引数のどちらかに機密度の高いデータが含まれる場合，返り値も機密度の高いデータとみなす演算である．

評価規則（IF）で用いる noExe 関数の定義を示す．この関数は，if で実行されない側の文（未実行文）に適用する．noExe 関数の適用前後で値ストアが異なるのは，文が $id := e$ の時のみである．

$$
\begin{aligned}
\mathrm{noExe}(\sigma, b, \mathbf{if}\, e \,\mathbf{then}\, S_{true} \,\mathbf{else}\, S_{false} \,\mathbf{end}) &= \mathrm{noExe}(\mathrm{noExe}(\sigma, b, S_{true}), b, S_{false}) \\
\mathrm{noExe}(\sigma, b, \mathbf{while}\, e \,\mathbf{do}\, S \,\mathbf{done}) &= \mathrm{noExe}(\sigma, b, S) \\
\mathrm{noExe}(\sigma, b, S_1; S_2) &= \mathrm{noExe}(\mathrm{noExe}(\sigma, b, S_1), b, S_2) \\
\mathrm{noExe}(\sigma, b, id := e) &= \begin{cases} \sigma & (b = \mathbf{false}) \\ [id \mapsto (\sigma(id))^\sharp]\sigma & (b = \mathbf{true}) \end{cases} \\
\mathrm{noExe}(\sigma, b, \mathbf{skip}) &= \sigma
\end{aligned}
$$

第 2 引数は，文の実行に関わっている機密度が高いかを表している．機密度が高い場合には $true$ となる．例えば，実行前の値ストア σ が $\sigma(l) = 1$,$\sigma(h) = 0^\sharp$ を満たすとき，次のプログラムを考える．

$$
\mathbf{if}\ h = 0\ \mathbf{then}\ h := 0\ \mathbf{else}\ l := 2\ \mathbf{end}
$$

$= (h, 0)$ の評価結果は $true$ なので，$h := 0$ が実行され，$l := 2$ を noExe 関数に適用する．条件式に機密度の高い値 $0^\sharp$ を用いているので，noExe 関数の第 2 引数は $true$ となる．

$$
\mathrm{noExe}(\sigma, true, l := 2) = [l \mapsto (\sigma(l))^\sharp]\sigma
$$

よって最終的な値ストアは $\sigma(l) = 1^\sharp$ となり，l は機密度の高い情報を格納することとなる．これは，条件式に機密度の高い情報が用いられることで，未実行文には，条件式を満たさなかったという情報が流れた，つまり未実行の文も機密度の高い情報との依存関係があるということである．

以上の規則を用いて任意の文 S に対する表記 $[S]^{\mathrm{v}}_{\sigma}$ を次に定義する，

$$\sigma' = [S]^{\mathrm{v}}_{\sigma} \;\Leftrightarrow\; \sigma; false \vdash_{\mathrm{v}} S \Downarrow \sigma'$$

2.6 型規則

検証に利用する型規則を図 2.2 に示す．また型規則中 IF で用いられる Φ 関数の定義も示す．型規則に用いられる表記 $\sigma; \rho; T^{pc} \vdash_{\mathrm{t}} S \Downarrow \rho_1$ は，σ, ρ, T^{pc} の元で，文 S を実行すると，ρ_1 を得ることを表す．以後このような ρ_1 を得ることができるとき，ρ は**型付け可能である**という．

$$\begin{aligned}
\Phi(\rho, T, \mathbf{if}\, e\, \mathbf{then}\, S_{true}\, \mathbf{else}\, S_{false}\, \mathbf{end}) &= \Phi(\Phi(\rho, T, S_{true}), T, S_{false}) \\
\Phi(\rho, T, \mathbf{while}\, e\, \mathbf{do}\, S\, \mathbf{done}) &= \Phi(\rho, T, S) \\
\Phi(\rho, T, S_1; S_2) &= \Phi(\Phi(\rho, T, S_1), T, S_2) \\
\Phi(\rho, T, id := e) &= \begin{cases} \left[id \mapsto \rho(id) \cap T\right]\rho & (id \notin F \text{ のとき}) \\ \rho & (id \in F, \rho(id) \subseteq T \text{ のとき}) \end{cases} \\
\Phi(\rho, T, \mathrm{skip}) &= \rho
\end{aligned}$$

型規則では，規則（ASS），(OP）が重要である．規則（OP）では，e_1, e_2 の各型情報の共通部分を取っている．これにより，e_1 と e_2 の両方の値を読み取れるユーザを求める．規則（$\mathrm{ASS_f}$），（$\mathrm{ASS_n}$）はそれぞれ，ファイル変数，それ以外の変数に対する型規則である．T^{pc} は文が実行されることを知ることができるユーザ集合を表す．例えば $T^{pc} = \emptyset$，$\sigma(x) = 1$，$\rho(x) = u_1$ として次のプログラム例を考える．

$$\mathbf{if}\; x = 1\; \mathbf{then}\; x := 2\; \mathbf{else}\; \mathbf{skip}\; \mathbf{end}$$

条件式 $x = 1$ が真であることから，型規則 (IF) より実行先での型付けは

$$\sigma; \rho; \{u_1\} \vdash_{\mathrm{t}} x := 2 \Downarrow \rho'$$

となる．ただし，型規則 (CONS) より $\rho'(x) = \emptyset$ である．ここで，$x := 2$ に関する T^{pc} の値は $\{u_1\}$ なので，これは if 文の条件式の評価結果を，知ることができるユーザは $\{u_1\}$ であることを表す．つまり，ユーザ u_1 だけが，if 文のどちらの分岐先が実行されるかを知ることができる．また，T^e は式 e の値を知ることができるユーザ集合である．以上のことから，型規則（$\mathrm{ASS_f}$），（$\mathrm{ASS_n}$）両方に現れる $T^{pc} \cap T^e$ は，文の実行を知ることができ，かつ式の値を知ることができるユーザ集合を求める処理である．言い替えると，文の実行，式の値を知っていても良いユーザ集合である．さらに，型規則 ($\mathrm{ASS_f}$) について，この規則の適用後，ファイル変数 id の値は，代入文が実行されたという

$$\mathrm{valExp}(\sigma, e) = v$$
$$\rho \vdash_{\mathrm{t}} e \Downarrow T^e$$
$$id \in F \Rightarrow \rho(id) \subseteq T^e$$

$$\frac{\sigma; \rho; T^{pc} \cap T^e \vdash_{\mathrm{t}} S_v \Downarrow \rho'}{\sigma; \rho; T^{pc} \vdash_{\mathrm{t}} \textbf{if } e \textbf{ then } S_{true} \textbf{ else } S_{false} \textbf{ end} \Downarrow \Phi(\rho', T^e, S_{\neg v})} \quad \text{(IF)}$$

$$\frac{\sigma; \rho; T^{pc} \vdash_{\mathrm{t}} \textbf{if } e \textbf{ then } S \ ; \ \textbf{while } e \textbf{ do } S \textbf{ done else skip end} \Downarrow \rho'}{\sigma; \rho; T^{pc} \vdash_{\mathrm{t}} \textbf{while } e \textbf{ do } S \textbf{ done} \Downarrow \rho'} \quad \text{(WHILE)}$$

$$\frac{\sigma; \rho; T^{pc} \vdash_{\mathrm{v}} S_1 \Downarrow \sigma' : \rho' \qquad \sigma'; \rho'; T^{pc} \vdash_{\mathrm{v}} S_2 \Downarrow \sigma'' : \rho''}{\sigma; \rho; T^{pc} \vdash_{\mathrm{v}} id := e \Downarrow \sigma'' : \rho''} \quad \text{(SEQ)}$$

$$\frac{\rho \vdash_{\mathrm{t}} e \Downarrow T^e \qquad id \notin F}{\sigma; \rho; T^{pc} \vdash_{\mathrm{t}} id := e \Downarrow \left[id \mapsto T^{pc} \cap T^e\right]\rho} \quad (\mathrm{ASS_n})$$

$$\frac{\rho \vdash_{\mathrm{t}} e \Downarrow T^e \qquad id \in F \qquad \rho(id) \subseteq T^{pc} \cap T^e}{\sigma; \rho; T^{pc} \vdash id := e \Downarrow \rho} \quad (\mathrm{ASS_f})$$

$$\frac{}{\sigma; \rho; T^{pc} \vdash_{\mathrm{v}} \mathrm{skip} \Downarrow \sigma : \rho} \quad \text{(SKIP)}$$

$$\frac{\rho \vdash_{\mathrm{t}} e_1 \Downarrow T_1 \qquad \rho \vdash_{\mathrm{t}} e_2 \Downarrow T_2}{\rho \vdash_{\mathrm{t}} e_1 \textbf{ op } e_2 \Downarrow op(e_1, e_2) : T_1 \cap T_2} \quad \text{(OP)}$$

$$\frac{}{\rho \vdash_{\mathrm{t}} id \Downarrow \sigma(id) : \rho(id)} \quad \text{(VAR)}$$

$$\frac{}{\rho \vdash_{\mathrm{t}} c \Downarrow c : \emptyset} \quad \text{(CONS)}$$

図 2.2: 型規則

情報 (T^{pc}) と，代入される値 (T^{e}) の両方を情報として持つ．そのため，$T^{pc} \cap T^{e}$ で求めたユーザ集合に含まれないユーザは，文の実行，式の値を知ることができないユーザである．そのユーザがファイル変数 id の値を知ることができるとき，文の実行及び式の値の情報が，ファイル変数 id を経由して，そのユーザに漏れることになる．そのため，型規則 ($\mathrm{ASS_f}$) の前提で，$\rho(id) \subseteq T^{pc} \cap T^{e}$ の判定を行い，情報漏れを防いでいる．また，この型規則 ($\mathrm{ASS_f}$) では型情報の更新を行っていない．これは，アクセス権限をファイル変数の型情報として扱っているためである．例えばファイル変数かどうかに関わらず，適用する代入規則

$$\frac{\rho \vdash_{\mathrm{t}} e \Downarrow T^{e}}{\sigma; \rho; T^{pc} \vdash_{\mathrm{t}} id := e \Downarrow \left[id \mapsto T^{pc} \cap T^{e}\right]\rho} \quad \text{(ASS)}$$

が存在したとする．このとき，$T^{pc} = \emptyset$，ファイル変数 $file_1$,$file_2$ の型情報をそれぞれ，$\{u_1, u_2\}$, $\{u_1\}$ とし，次のプログラムを考える．

$$file_2 := file_1$$

このプログラムの実行後，規則 (ASS) より $file_2$ の型情報は $\{u_1, u_2\}$ となる．そうすると，この代入文より後に $file_2$ にアクセスする際には，$file_2$ のアクセス権限は $\{u_1, u_2\}$ であることになり，アクセス権限が意図しないものへと変更されるからである．

if で用いる Φ 関数は，型情報に関する未実行文のための関数である．適用後のタグストアは型規則 ($\mathrm{ASS_f}$) , ($\mathrm{ASS_n}$) と同様の条件で，$id \in F$ かつ$\rho(id) \not\subseteq T$ のときは，規則 (IF) の前提部分で判定する．

2.7 安全性

この節では本論文で用いる安全性の定義を述べる．$\mathrm{Safe_v}$ は，実行後に機密度の低い情報を格納してほしい変数が，実際に機密度の低い情報を格納していることを表す述語である．この述語は [GJ05] で提示されている安全性と意味合いは同様のものである．

$$\mathrm{Safe_v}([S]^{\mathrm{v}}_{\sigma}) \Leftrightarrow \forall x \in \mathrm{L_o}.\neg \mathrm{valSec}([S]^{\mathrm{v}}_{\sigma}(x))$$

これは値に関する安全性であり，型に関しては，型規則に安全性の条件 ($\rho(id) \subseteq T^{pc} \cap T^{e}$ などの部分集合判定を持つ計算) を設けているため，型付け可能であることが安全であることを意味する．

第3章　定理

この章では，これまで示した情報を元に，導くことができる定理を示す．定理を最初に示し，そして定理の意味を述べた後，定理の証明に必要な補題を提示する．定理及び補題の証明は 4 章に示す．

3.1　非干渉性の定理

定理 1 S, 値ストア θ, θ_1, タグストア γ, $\sigma = \mathrm{init_v}(\theta)$, $\sigma_1 = \mathrm{init_v}(\theta_1)$, $\rho = \mathrm{init_t}(\gamma)$, $Safe_v([S]^V_\sigma)$, $Safe_v([S]^V_{\sigma_1})$, ρ が型付け可能である，$\sigma(x) =_{L_i} \sigma_1(x)$ を満たす時，任意の $y \in L_o$，$z \in \mathrm{ID}$ に関して次の 2 つが成り立つ．

$$
\begin{array}{ll}
\text{(公開値)} & [S]^V_\sigma(y) = [S]^V_{\sigma_1}(y) \\
\text{(依存)} & y \in \mathrm{F} \text{ かつ} \rho(y) \cap \rho(z)^c \neq \emptyset \text{ ならば} \\
& [S]^V_{\left[z \mapsto ((\sigma(z))^\sharp)\right]\sigma}(y) \notin \mathbb{Z}^\sharp
\end{array}
$$

この定理では，結論を 2 つ述べている．結論公開値の意味は，[GJ05] で述べられている，σ,σ_1 それぞれの出力に対して，公開情報は常に一致することを示している．定理の前提から実行前の σ,σ_1 に対して，値が異なる可能性があるのは機密情報であり，それぞれの出力に対して公開情報が一致することから，機密情報と公開情報間に情報流がないことを示している．結論情報依存は本研究で導入した性質であり，前提つきの結論である．前提では，アクセス権限に注目しており，z は任意の ID の要素であるが，$init_t$ の定義から，$z \notin \mathrm{F}$ である場合には，$\rho(z)^c = \emptyset$ となる．よって z はファイル変数である．$\rho(y) \cap \rho(z)^c \neq \emptyset$ では，$\rho(y)$ は y のアクセス権限を持つユーザ集合を求めるのに対し，$\rho(z)^c$ は z のアクセス権限を持たないユーザ集合を求めている．これらの共通部分が空でないことから，あるユーザにとって y は公開情報であり，z は機密情報になる．そして情報依存の結論では，z の値を機密情報にしたとしても，出力の y の値は機密情報にならない，つまり z と y 間に情報流がないことを示している．ここで，z が元から機密情報だとすると y は公開情報なので，結論公開情報から，y, z 間に情報流はないことは示されている．よって結論情報依存では，z は公開情報扱いだが，その上でアクセス権限に着目した場合，z と y に機密・公開情報の関係が生じ，その関係上でも情報漏れがないことを示している．

3.2 任意式の評価値に関する補題

補題 1 任意の，式 e，値ストア σ と σ_1 に関して，次の前提が成り立つとき

$$\begin{aligned}&\text{(公開値)} \quad \forall x(\sigma(x), \sigma_1(x) \in \mathbb{Z} \Rightarrow \sigma(x) = \sigma_1(x))\\ &\text{(評価値 1)} \quad \mathrm{valExp}(\sigma, e) \in \mathbb{Z}\\ &\text{(評価値 2)} \quad \mathrm{valExp}(\sigma_1, e) \in \mathbb{Z}\end{aligned}$$

次の式が成り立つ

$$\mathrm{valExp}(\sigma, e) = \mathrm{valExp}(\sigma_1, e)$$

この補題は，2 つの値ストアに関して，どちらにおいても機密度が低い値は，式評価前では等しく (前提 (公開値))，式評価後でも等しい (結論) ことを示すものである．

3.3 任意式の機密情報に関する補題

補題 2 任意の，式 e，値ストア σ，値 v，変数 id に関して，次の前提が成り立つとき

$$\begin{aligned}&\text{(機密)} \quad \sigma(id) \in \mathbb{Z}^\sharp\\ &\text{(使用)} \quad id \in e\end{aligned}$$

次の論理式が成り立つ

$$\mathrm{valExp}(\sigma, e) \in \mathbb{Z}^\sharp$$

この補題は，式の一部に機密情報が用いられた場合，式全体も機密情報であることを示すものである．

3.4 機密度の伝搬に関する補題

補題 3 任意の文 S に関して $\sigma; true \vdash_v S \Downarrow \sigma'$ であるとき，S の導出過程において，任意の S',σ_1, σ_1' に関して $S' \in S$ かつ $\sigma_1; \mathrm{v}^{\mathrm{pc}} \vdash_v S' \Downarrow \sigma_1'$ ならば，$\mathrm{v}^{\mathrm{pc}} = true$ である．

この補題は，実行に関わる機密度は，その後のどの実行においても伝搬されていることを示すものである．

3.5 定理の一般化の補題

補題 4 任意の，文 S，値ストア σ，σ_1，σ'，σ'_1，タグストア $\mathrm{init_t}(\rho)$，$\mathrm{init_t}(\rho')$ に関して，次の前提が成り立つとき

$$
\begin{array}{ll}
\text{(停止 1)} & \sigma; false \vdash S \Downarrow \sigma' \\
\text{(停止 2)} & \sigma_1; false \vdash S \Downarrow \sigma'_1 \\
\text{(公開値)} & \forall x(\sigma(x), \sigma_1(x) \notin \mathbb{Z}^\sharp \Rightarrow \sigma(x) = \sigma_1(x))
\end{array}
$$

次の論理式が成り立つ

$$
\begin{array}{ll}
\text{(評価値)} & \text{任意の } y \text{ に関して，} \\
 & \sigma'(y), \sigma'_1(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\
\text{(値の一致)} & \sigma'(y) = \sigma'_1(y) \\
\text{(情報依存)} & \forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset) \\
 & \Rightarrow [S]^{\mathrm{V}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \notin \mathbb{Z}^\sharp))
\end{array}
$$

定理では，L_i や H_i など，プログラム全体に対して設定されている情報があり，例えば L_i はそのプログラム実行前の機密度の低い変数集合であるが，サブプログラムに対しての実行前の機密度の低い変数集合を示してはいない．そのため，構造に関する帰納法で直接定理を証明することは難しい．この補題は，機密度の低い値というものは，プログラムのどの状態においても等しいことを示しており，上記問題点を解消するために定理を一般化したものである．

第4章　証明

4.1　任意式の評価値に関する補題

補題 5 任意の，式 e，値ストア σ と σ_1 に関して，次の前提が成り立つとき

$$\begin{aligned}
&\text{(公開値)} \quad \forall x(\sigma(x), \sigma_1(x) \in \mathbb{Z} \Rightarrow \sigma(x) = \sigma_1(x)) \\
&\text{(評価値 1)} \quad \mathrm{valExp}(\sigma, e) \in \mathbb{Z} \\
&\text{(評価値 2)} \quad \mathrm{valExp}(\sigma_1, e) \in \mathbb{Z}
\end{aligned}$$

次の式が成り立つ

$$\mathrm{valExp}(\sigma, e) = \mathrm{valExp}(\sigma_1, e)$$

証明 式の構造に関する帰納法で証明する．

1. e が c(定数) のとき

valExp 関数の定義より，

$$\mathrm{valExp}(\sigma, e) = c = \mathrm{valExp}(\sigma_1, c)$$

が成り立つ．

2. e が id のとき

valExp 関数の定義と前提 (評価値 1)，(評価値 2) より，

$$\begin{aligned}
\mathrm{valExp}(\sigma, id) &= \sigma(id) \in \mathbb{Z} \\
\mathrm{valExp}(\sigma_1, id) &= \sigma_1(id) \in \mathbb{Z}
\end{aligned}$$

であり，前提 (公開値) より $\sigma(id) = \sigma_1(id)$ が成り立つ．よって

$$\mathrm{valExp}(\sigma, id) = \mathrm{valExp}(\sigma_1, id)$$

3. e が e_1 op e_2 のとき

valExp 関数の定義と前提 (評価値 1)，(評価値 2) より，

$$\left.\begin{array}{rcl} \mathrm{valExp}(\sigma, e_1 \,\mathbf{op}\, e_2) & = & op^\sharp(\mathrm{valExp}(\sigma, e_1), \mathrm{valExp}(\sigma, e_2)) \in \mathbb{Z} \\ \mathrm{valExp}(\sigma_1, e_1 \,\mathbf{op}\, e_2) & = & op^\sharp(\mathrm{valExp}(\sigma_1, e_1), \mathrm{valExp}(\sigma_1, e_2)) \in \mathbb{Z} \end{array}\right\} \quad (*)$$

である．$op^\sharp$ の定義より

$$\mathrm{valExp}(\sigma, e_1), \mathrm{valExp}(\sigma, e_2) \in \mathbb{Z}$$
$$\mathrm{valExp}(\sigma_1, e_1), \mathrm{valExp}(\sigma_1, e_2) \in \mathbb{Z}$$

が導かれるので，e_1，e_2 に関して前提 (評価値 1)，(評価値 2) は満たされている．帰納法の仮定より，

$$\mathrm{valExp}(\sigma, e_1) = \mathrm{valExp}(\sigma_1, e_1)$$
$$\mathrm{valExp}(\sigma, e_2) = \mathrm{valExp}(\sigma_1, e_2)$$

が成り立つ．よって $op(v_1, v_2) = op(v_1', v_2')$ であるから，$v = v'$ が成り立つ．以上のことから補題 1 が成り立つ．

4.2 任意式の機密情報に関する補題

補題 6 任意の，式 e，値ストア σ，値 v，変数 id に関して，次の前提が成り立つとき

$$\text{(機密)} \quad \sigma(id) \in \mathbb{Z}$$
$$\text{(使用)} \quad id \in e$$

次の論理式が成り立つ

$$\mathrm{valExp}(\sigma, e) \in \mathbb{Z}^\sharp$$

証明 式 e の構造に関する帰納法で証明する．

e が skip のとき

前提使用を満たさないので命題は成り立つ．

e が id のとき

前提停止と valExp 関数の定義より

$$\mathrm{valExp}(\sigma, id) = \sigma(id) = v$$

よって前提機密より，$\sigma(id) \in \mathbb{Z}^\sharp$ であるから

$$v \in \mathbb{Z}^\sharp$$

が成り立つ.

e が $e_1 \mathrm{op} e_2$ のとき

前提使用より，id は次の論理式の少なくとも 1 つを満たす.

$$id \in e_1$$
$$id \in e_2$$

1. $id \in e_1$ のとき

前提停止から，e_1 に関して，

$$\mathrm{valExp}(\sigma, e_1) = v_1$$

となる v_1 が存在する．このことから前提機密より

$$\begin{aligned} \mathrm{valExp}(\sigma, e_1) &= v_1 \\ \sigma(id) &\in \mathbb{Z}^\sharp \\ id &\in e_1 \end{aligned}$$

が成り立つ．よって帰納法の前提が満たされたので，帰納法の仮定より，

$$v_1 \in \mathbb{Z}^\sharp$$

が成り立つ．よって valExp 関数の定義より，

$$\begin{aligned} \mathrm{valExp}(\sigma, e_1 \text{ op } e_2) &= op^\sharp(v_1, \mathrm{valExp}(\sigma, e_2)) \\ op^\sharp(v_1, \mathrm{valExp}(\sigma, e_2) &= (op(v_1, \mathrm{valExp}(\sigma, e_2)))^\sharp \in \mathbb{Z}^\sharp \end{aligned}$$

が成り立つ．前提停止より，$v = (op(v_1, \mathrm{valExp}(\sigma, e_2)))^\sharp$ であるから

$$v \in \mathbb{Z}^\sharp$$

が成り立つ.

2. $id \in e_2$ のとき $id \in e_1$ **のとき**の議論において，e_1 と e_2 をいれかえることで，$v \in \mathbb{Z}^\sharp$ が成り立つことが言える.

以上のことから補題 2 は成り立つ.

4.3 機密度の伝搬に関する補題

補題 7 任意の文 S に関して $\sigma; true \vdash_v S \Downarrow \sigma'$ であるとき，S の導出過程において，任意の S',σ_1, σ_1' に関して $S' \in S$ かつ $\sigma_1; \mathrm{v}^{\mathrm{pc}} \vdash_v S' \Downarrow \sigma_1'$ ならば，$\mathrm{v}^{\mathrm{pc}} = true$ である．

証明 文 S の構造に関する帰納法で証明する．

1. S が skip のとき

規則 (SKIP) より，導出過程はないので成り立つ．

2. S が id:=e のとき

規則 ($\mathrm{ASS_s ec}$)，($\mathrm{ASS_p ub}$) より，両規則ともに導出過程に S' は存在しないので成り立つ．

3. S が if e then S else S' end のとき

S が実行されるとき

規則 (IF) より，$\sigma; true \vdash_v S \Downarrow \sigma_1'$ となる σ_1' が存在する．S に関して，$\mathrm{v}^{\mathrm{pc}} = true$ である．帰納法の仮定より，S に関して補題は成り立っているので，if の時も補題は成り立つ．

S' が実行されるとき

S が実行されるときにおいて，S が S' に置き換わるだけなので成り立つ．

4. S が while e do S done のとき

S を適用できるのは規則 (WHILE) であるから，その前提から

$$\sigma; true \vdash_v \text{if } e \text{ then } S \text{ ; while } e \text{ do } S \text{ done } else \text{ skip end}$$

となり，$\mathrm{v}^{\mathrm{pc}} = true$ である，任意の if 文に関する文に変換できたので，3 より補題は成り立つ．

5. S が S ; S' のとき

S を適用できるのは規則 (SEQ) であるから，その前提から，

$$\sigma; true \vdash_v S \Downarrow \sigma_2$$
$$\sigma_2; true \vdash_v S' \Downarrow \sigma_2'$$

となる σ_2,σ_2' が存在する．S, S' に関して，それぞれ $\mathrm{v}^{\mathrm{pc}} = true$ である．よって帰納法の仮定より，補題は成り立つ．

4.4 定理の一般化の補題

補題 8 任意の，文 S，値ストア σ，σ_1，σ'，σ_1'，タグストア $\mathrm{init_t}(\rho)$，$\mathrm{init_t}(\rho')$ に関して，次の前提が成り立つとき

(停止 1)　$\sigma; false \vdash S \Downarrow \sigma'$

(停止 2)　$\sigma_1; false \vdash S \Downarrow \sigma_1'$

(公開値)　$\forall x(\sigma(x), \sigma_1(x) \notin \mathbb{Z}^\sharp \Rightarrow \sigma(x) = \sigma_1(x))$

次の論理式が成り立つ

$$
\begin{aligned}
&\text{(評価値)} \quad \text{任意の } y \text{ に関して,} \\
&\qquad \sigma'(y), \sigma_1'(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\
&\text{(値の一致)} \quad \sigma'(y) = \sigma_1'(y) \\
&\text{(情報依存)} \quad \forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset) \\
&\qquad \Rightarrow [S]^{\mathrm{V}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \notin \mathbb{Z}^\sharp))
\end{aligned}
$$

証明　文 S の構造に関する帰納法で証明する.

1. S が skip のとき

実行によるストアの更新はないので,

$$\sigma = \sigma'$$
$$\sigma_1 = \sigma_1'$$
$$\rho = \rho_1$$
$$\rho' = \rho_2$$

が成り立つ. よって前提公開値より,

$$\sigma'(x) = \sigma_1'(x)$$

が導ける. また, 結論情報依存に関して, 仮定から

$$y \neq z$$

であり, ストアの更新がないので, z を変化させても y には影響しない. 前提評価値より $\sigma'(y) \notin \mathbb{Z}^\sharp$ である. よって

$$S([z \mapsto (\sigma(z))^\sharp]^{\mathrm{V}}(y) \notin \mathbb{Z}^\sharp$$

が成り立つ.

2. S が id:=e のとき

・$y = id$ のとき

前提停止 1, 2 と規則 $(ASSIGN_{pub})$ の前提から,

$$\mathrm{valExp}(\sigma, e) = v'$$
$$\mathrm{valExp}(\sigma_1, e) = v_1'$$

となる v', v_1' が存在する．これは，σ,σ_1, v', v_1', e が補題 1 の前提 (停止 1，2) を満たすことを意味する．次に前提停止 1，2 と規則 ($ASSIGN_{pub}$) の結論から，

$$[y \mapsto v']\sigma = \sigma' \tag{4.1}$$

$$[y \mapsto v_1']\sigma_1 = \sigma_1' \tag{4.2}$$

よって $\quad [y \mapsto v']\sigma(y) = v' \tag{4.3}$

$$[y \mapsto v_1']\sigma_1(y) = v_1' \tag{4.4}$$

ゆえに $\quad \sigma'(y) = v'$

$$\sigma_1'(y) = v_1'$$

ここで結果の前提評価値より，

$$\sigma'(y) = v' \notin \mathbb{Z}^\sharp$$

$$\sigma_1'(y) = v_1' \notin \mathbb{Z}^\sharp$$

が成り立つ．これは，v', v_1' が補題 1 の前提 (評価値 1，2) を満たすことを意味する．また，補題 1 の前提公開値は補題 4 の前提公開値と同じ論理式なので成り立つ．よって補題 1 の前提がすべて満たされたので，$v' = v_1'$ が成り立つ．よって (1)，(2)，(3)，(4) より

$$[y \mapsto v']\sigma(y) = [y \mapsto v_1']\sigma_1(y)$$

$$\sigma'(y) = \sigma_1'(y)$$

が導ける．また，結論情報依存に関して，

$$S([z \mapsto (\sigma(z))^\sharp]^{\mathrm{v}}(y) \in \mathbb{Z}^\sharp$$

が成り立つと仮定し，背理法で証明する．規則 ($ASSIGN_{pub}$) から，$\mathrm{valExp}(\sigma, e) \in \mathbb{Z}^\sharp$ を満たす．つまり式 e 内に z が用いられる必要がある．ここで，型規則 ($ASSIGN_f$) の前提から，

$$\rho(y) \subseteq T^e \cap T^{pc}$$

が言えるが，式 e 内に z が用いられることから，$T^e \subseteq \rho(z)$ を満たさなければならない．つまり，

$$\rho(y) \subseteq T^e \subseteq \rho(z)$$

が成り立たなければならないが，これは結論情報依存の仮定 $rho(y) \cap rho(z)^c \neq \emptyset$ に矛盾する．よって，

$$S([z \mapsto (\sigma(z))^\sharp]^{\mathrm{v}}(y) \notin \mathbb{Z}^\sharp$$

が成り立つ

・$y \neq id$ のとき

y の更新が行なわれないので，前提停止 1，2 より

$$\sigma(y) = \sigma'(y)\sigma_1(y) = \sigma_1'(y)$$

が成り立つ．よって結論の前提評価値から，

$$\sigma'(y),\sigma'_1(y) \notin \mathbb{Z}^\sharp$$
$$\sigma(y),\sigma_1(y) \notin \mathbb{Z}^\sharp$$

が導ける．ゆえに前提公開値から，

$$\sigma'(y) = \sigma'_1(y)$$

が成り立つことが言える．また，結論情報依存に関して，

$$\sigma(y) \notin \mathbb{Z}^\sharp$$

である．y の更新はないので，

$$\sigma(y) = [S]^{\mathrm{V}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y)$$

が成り立ち，結論情報依存の前提より，$y \neq z$ であることから，

$$[S]^{\mathrm{V}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \notin \mathbb{Z}^\sharp$$

が成り立つ．

3. S が $S_1\,;\,S_2$ のとき

S_1 に関して，前提停止 1，2 と，規則 (SEQUENCE) より，

$$\sigma; false \vdash S_1 \Downarrow \theta$$
$$\sigma_1; false \vdash S_1 \Downarrow \theta_1$$

を満たす θ,θ_1 が存在する (帰納法の前提停止 1，2)．また，S_2 に関して同様に

$$\theta; false \vdash S_1 \Downarrow \theta'$$
$$\theta_1; false \vdash S_1 \Downarrow \theta'_1$$

を満たす θ',θ'_1 が存在する (帰納法の前提停止 1，2)．ここで規則 (SEQUENCE) より，

$$\sigma' = \theta' \tag{4.5}$$
$$\sigma'_1 = \theta'_1 \tag{4.6}$$

である．S の前提公開値と S_1 の帰納法の前提公開値は同じ論理式なので，S_1 に関して帰納法の前提を満たしている．よって帰納法の仮定から，

$$\theta(y),\theta_1(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\ \theta(y) = \theta_1(y) \wedge \\ \forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset) \\ \Rightarrow [S_1]^{\mathrm{V}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \notin \mathbb{Z}^\sharp))$$

が導かれる．このことから，

$$\theta(y), \theta_1('y) \notin \mathbb{Z}^\sharp \Leftrightarrow \theta(y) = \theta_1(y)$$

が成り立つので，S_2 に関して帰納法の前提を満たしている．よって帰納法の仮定から，

$$\begin{aligned}
&\theta'(y), \theta_1'(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\
&\theta'(y) = \theta_1'(y) \wedge \\
&\forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset) \\
&\Rightarrow [S_2]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\theta}(y) \notin \mathbb{Z}^\sharp))
\end{aligned}$$

が成り立つ．ここで，(5)(6) より

$$\begin{aligned}
\sigma'(y) &= \theta'(y) \\
\sigma_1'(y) &= \theta_1'(y)
\end{aligned}$$

であるから，$\sigma'(y) = \sigma_1'(y)$ が成り立つ．次に結論情報依存に関して，

$$[S]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \in \mathbb{Z}^\sharp$$

が成り立つと仮定し，背理法で証明する．S_2 に関する帰納法の仮定より，任意の z' に関して

$$\begin{aligned}
&\rho(y) \cap \rho(z')^c \neq \emptyset \Rightarrow \\
&[S_2]^{\mathrm{v}}_{\left[z' \mapsto (\sigma(z'))^\sharp\right]\sigma}(y) \in \mathbb{Z}^\sharp
\end{aligned}$$

が成り立つので，

$$\begin{aligned}
&\rho(y) \cap \rho(a)^c = \emptyset \\
&[S_2]^{\mathrm{v}}_{\left[a \mapsto (\sigma(a))^\sharp\right]\sigma}(y) \in \mathbb{Z}^\sharp
\end{aligned}$$

を満たす変数 a が存在する．さらに S_1 に関する帰納法の仮定から，

$$\begin{aligned}
&\rho(a') \cap \rho(z)^c \neq \emptyset \Rightarrow \\
&[S_1]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(a') \notin \mathbb{Z}^\sharp
\end{aligned}$$

が成り立つので，変数 a は

$$\begin{aligned}
&\rho(a) \cap \rho(z)^c = \emptyset \\
&[S_1]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(a) \notin \mathbb{Z}^\sharp
\end{aligned}$$

を共に満たす．ここで変数 a は

$$\begin{aligned}
&\rho(a) \cap \rho(z)^c = \emptyset \\
&\rho(y) \cap \rho(a)^c = \emptyset
\end{aligned}$$

を満たすことから，

$$\rho(a) = \rho(z)$$
$$\rho(y) = \rho(a)$$

が成り立つ．これは結論情報依存の仮定 $\rho(y) \cap \rho(z)^c \neq \emptyset$ より，

$$\rho(a) \cap \rho(a)^c \neq \emptyset$$

となり矛盾する．

4. S が $\mathbf{if}\, e\, \mathbf{then}\, S_1\, \mathbf{else}\, S_2\, \mathbf{end}$ のとき

前提停止 1，2 と規則 (IF) の前提から，条件式 e に対して，

$$\mathrm{valExp}(\sigma, e) = v$$
$$\mathrm{valExp}(\sigma_1, e) = v'$$

となる v，v' が存在する．また，S_1，S_2 に関する帰納法の前提と S に関する前提は同じものを指しているので，S_1，S_2 について帰納法の仮定を使用できる．

4.1 $v, v' \in \mathbb{Z}^\sharp$ のとき

ある変数に対するプログラム中の最後の代入文が，その変数の最終的な値を決めるので，任意の id，式 e_1 に関して

$$S = S_1\ id := e_1\ S_2$$
$$\text{ただし}, id := e_1 \notin S_2$$

を満たす代入文 $id := e_1$ が存在するかの場合分けを行う．

4.1.1 代入文が存在するとき

代入文が σ の元で実行されるとき

実行されるので，S の評価の導出中に代入文の評価が現れる．また，$v \in \mathbb{Z}^\sharp$ より，$\mathrm{valSec}(\sigma, e) = false$ である．実行する文 S_{true} に関して，規則 (IF) の前提より，

$$\sigma; false \vee true \vdash_v S_{true} \Downarrow \theta$$

となる値ストア θ が存在する．よって補題 3 より，ある値ストア θ_1,θ_1' に関して，

$$\theta_1; \mathrm{v}^{\mathrm{pc}} \vdash_v id := e' \Downarrow \theta_1'$$

が成り立ち，$\mathrm{v}^{\mathrm{pc}} = true$ である．よって，e' の評価結果を v_e' とすると，規則 ($\mathrm{ASS}_{\mathrm{sec}}$) より，

$$\left[id \mapsto (v_e')^\sharp\right]\theta_1 = \theta_1'$$
$$\theta_1'(id) \in \mathbb{Z}^\sharp$$

が成り立つ．よって id は前提評価値を満たさない変数となる．

代入文が σ の元で実行されないとき

代入文が未実行であることから，規則 (IF) の結論より，ある θ', b に関する $\mathrm{noExe}(\theta', b, S_{\neg v'})$(ただし $id := e' \in S_{\neg v'}$) が S の評価の導出中に存在する．noExe の定義より，

$$noExe(\sigma, b, id := e) \quad = \quad \begin{cases} \sigma(\text{ただし}, b = false) \\ [id \mapsto ((\sigma(id))^\sharp)]\sigma(\text{ただし}, b = true) \end{cases}$$

である．$b = true$ ならば id は前提評価値を満たさない変数となる．$b = false$ のとき，規則 (IF) より，

$$\frac{\mathrm{valExp}(\theta, e_2) = v' \qquad \theta; false \vee \mathrm{v}^{\mathrm{pc}} \vdash_v S_{v'} \Downarrow \theta'}{\theta; \mathrm{v}^{\mathrm{pc}} \vdash_v \textbf{if}\, e_2 \,\textbf{then}\, S_{true} \,\textbf{else}\, S_{false} \,\textbf{end} \Downarrow noExe(\theta', \mathrm{valSec}(v'), S_{\neg v'})} \quad \text{(IF)}$$

となる θ' が存在し，$noExe(\theta', b, id := e) = \theta'$ である．σ_1 に関して，θ' に該当する値ストアを θ'_1, θ に該当する値ストアを θ_1 とする．

$\sigma \neq \theta$,$\sigma_1 \neq \theta_1$ のとき，各規則のうち，条件を満たすような規則は規則 (SEQ) のみである．よって，ある $S'_1; S'_2 \in S$ に関して

$$\frac{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash S'_1 \Downarrow \theta'_1 \qquad \sigma_1; \mathrm{v}^{\mathrm{pc}} \vdash S'_2 \Downarrow \theta}{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash S'_1 \,;\, S'_2 \Downarrow \theta}$$

$$\frac{\sigma_1; \mathrm{v}^{\mathrm{pc}} \vdash S'_1 \Downarrow \theta'_2 \qquad \sigma_1; \mathrm{v}^{\mathrm{pc}} \vdash S'_2 \Downarrow \theta_1}{\sigma; \mathrm{v}^{\mathrm{pc}} \vdash S'_1 \,;\, S'_2 \Downarrow \theta_1}$$

を満たす θ'_1,θ'_2 が存在する．$S = S_1; S_2$ の時と同様の議論から，

$$\begin{aligned} &\theta(y), \theta_1(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\ &\theta(y) = \theta_1(y) \\ &\forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset) \\ &\Rightarrow [S'_1; S'_2]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \notin \mathbb{Z}^\sharp)) \end{aligned}$$

が成り立つ．よって，θ,θ_1 は帰納法の仮定の前提を満たすことから，帰納法の仮定より，任意の y に対して，

$$\begin{aligned} &\theta'(y), \theta'_1(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\ &\theta'(y) = \theta'_1(y) \\ &\forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset) \\ &\Rightarrow [S_{v'}]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\theta}(y) \notin \mathbb{Z}^\sharp)) \end{aligned}$$

が成り立つ．noExe 関数の定義より，

$$noExe(\theta', false, S_{\neg v'}) = \theta'$$
$$noExe(\theta'_1, false, S_{\neg v'}) = \theta'_1$$

であり，$id = e'$ 以降に id の代入文が存在しないことから，ストアの更新がないので，

$$\theta'(id) = \sigma'(id)$$
$$\theta'_1(id) = \sigma'_1(id)$$

が成り立つ．よって，

$$\sigma'(id) = \sigma'_1(id)$$

である．次に結論情報依存に関して，

$$[S]^{\mathrm{V}}_{\left[z\mapsto(\sigma(z))^\sharp\right]\sigma}(id) \in \mathbb{Z}^\sharp$$

が成り立つと仮定し，背理法で証明する．帰納法の仮定から，

$$\theta'(id),\theta'_1(id) \notin \mathbb{Z}^\sharp \Rightarrow (\\ \theta'(id) = \theta'_1(id)\\ \forall z \in ID((id \in F \wedge \rho(id) \cap \rho(z)^c \neq \emptyset)\\ \Rightarrow [S_{v'}]^{\mathrm{V}}_{\left[z\mapsto(\sigma(z))^\sharp\right]\theta}(id) \notin \mathbb{Z}^\sharp))$$

であるので，

$$\rho(id) \cap \rho(a)^c = \emptyset \text{ かつ}\\ [S_{v'}]^{\mathrm{V}}_{\left[a\mapsto(\sigma(a))^\sharp\right]\theta}(id) \notin \mathbb{Z}^\sharp$$

を満たす変数 a が存在する．また S'_2 に関して帰納法の仮定より，任意の y に関して

$$\theta(y),\theta_1(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\ \theta(y) = \theta_1(y)\\ \forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset)\\ \Rightarrow [S'_2]^{\mathrm{V}}_{\left[z\mapsto(\sigma(z))^\sharp\right]\sigma_1}(y) \notin \mathbb{Z}^\sharp))$$

であるので，

$$\rho(a) \cap \rho(z')^c = \emptyset \text{ かつ}\\ [S'_2]^{\mathrm{V}}_{\left[z'\mapsto(\sigma(z'))^\sharp\right]\sigma_1}(a) \notin \mathbb{Z}^\sharp$$

が成り立つ必要がある．さらに S'_1 に関して，同様の議論から，

$$\rho(z') \cap \rho(z)^c = \emptyset \text{ かつ}\\ [S'_1]^{\mathrm{V}}_{\left[z\mapsto(\sigma(z))^\sharp\right]\sigma}(z') \notin \mathbb{Z}^\sharp$$

が成り立つ必要がある．よって，

$$\begin{aligned} \rho(id) &= \rho(a) \\ \rho(a) &= \rho(z') \\ \rho(z') &= \rho(z) \\ \text{ゆえに，} \quad \rho(id) &= \rho(z) \end{aligned}$$

が導かれる．このことから，結論情報依存の仮定において，

$$\rho(z) \cap \rho(z)^c \neq \emptyset$$

となり矛盾する．よって，

$$[S]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(id) \notin \mathbb{Z}^\sharp$$

が成り立つ．

$\sigma = \theta, \sigma_1 = \theta_1$ のとき，先程と同様に，規則 (IF) に関して，

$$\frac{\mathrm{valExp}(\theta, e_2) = v' \qquad \theta; false \vee \mathrm{v}^{\mathrm{pc}} \vdash_v S_{v'} \Downarrow \theta'}{\theta; \mathrm{v}^{\mathrm{pc}} \vdash_v \mathbf{if}\, e_2\, \mathbf{then}\, S_{true}\, \mathbf{else}\, S_{false}\, \mathbf{end} \Downarrow noExe(\theta', \mathrm{valSec}(v'), S_{\neg v'})} \qquad (IF)$$

となる θ' が存在し，$noExe(\theta', b, id := e) = \theta'$ である．σ_1 に関して，θ' に該当する値ストアを θ'_1，θ に該当する値ストアを θ_1 とする．

帰納法の仮定から，任意の y に関して，

$$\begin{aligned} &\theta'(y), \theta'_1(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\ &\theta'(y) = \theta'_1(y) \\ &\forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset) \\ &\Rightarrow [S_{v'}]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\theta}(y) \notin \mathbb{Z}^\sharp)) \end{aligned}$$

が成り立つ．noExe 関数の定義より，

$$\begin{aligned} noExe(\theta', false, S_{\neg v'}) &= \theta' \\ noExe(\theta'_1, false, S_{\neg v'}) &= \theta'_1 \end{aligned}$$

よって，この関数適用以降に id に関する代入文が存在しないことから，

$$\begin{aligned} \theta'(id) &= \theta'_1(id) \\ \text{よって，} \quad \sigma'(id) &= \sigma'_1(id) \end{aligned}$$

が成り立つ．次に結論情報依存に関して，

$$[S]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(id) \in \mathbb{Z}^\sharp$$

が成り立つと仮定し，背理法で証明する．$S_{\neg v'}$ に関する帰納法の仮定より，任意の y に関して，

$$\theta'(y),\theta'_1(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\ \theta'(y) = \theta'_1(y)\\ \forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset)\\ \Rightarrow [S_{v'}]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \notin \mathbb{Z}^\sharp))$$

が成り立つ．ここで，$S_{v'}$ の実行以降，id の値が変化しないことから，

$$[S_{v'}]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(id) = [S]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(id)$$

よって，$$[S_{v'}]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \in Z^\sharp$$

が導ける．これは帰納法の仮定と矛盾する．よって，

$$[S]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(id) \notin \mathbb{Z}^\sharp$$

が成り立つ．

4.1.2 代入文が存在しないとき

代入文がないことから，id に関するストアの更新がないため，skip と同様の議論から成り立つことが言える．

4.2 $v, v' \notin \mathbb{Z}^\sharp$ のとき

$$\mathrm{valSec}(\sigma, e) = false$$

$$\mathrm{valSec}(\sigma_1, e) = false$$

であるから，σ,σ_1 に関して，前提停止 (1)(2) と規則 (IF) より，

$$\frac{\mathrm{valExp}(\sigma, e) = v \qquad \sigma; false \vdash_v S_v \Downarrow \theta}{\sigma; false \vdash_v \mathbf{if}\, e\, \mathbf{then}\, S_1\, \mathbf{else}\, S_2\, \mathbf{end} \Downarrow noExe(\theta, false, S_{\neg v})} \quad \text{(IF)}$$

$$\frac{\mathrm{valExp}(\sigma, e) = v \qquad \sigma_1; false \vdash_v S_v \Downarrow \theta_1}{\sigma_1; false \vdash_v \mathbf{if}\, e\, \mathbf{then}\, S_1\, \mathbf{else}\, S_2\, \mathbf{end} \Downarrow noExe(\theta_1, false, S_{\neg v})} \quad \text{(IF)}$$

を満たす θ,θ_1 が存在する．

$$\sigma; false \vdash_v S_v \Downarrow \theta$$

$$\sigma_1; false \vdash_v S_v \Downarrow \theta_1$$

であるから，帰納法の仮定より，任意の y に関して，

$$\theta(y),\theta_1(y) \notin \mathbb{Z}^\sharp \Rightarrow (\\ \theta(y) = \theta_1(y) \quad \wedge\\ \forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset)\\ \Rightarrow [S_v]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \notin \mathbb{Z}^\sharp))$$

が成り立つ．ここで，

$$noExe(\theta, false, S_y) = \sigma'$$
$$noExe(\theta_1, false, S_y) = \sigma'_1$$

であり，noExe の定義から，$b = false$ の時，関数適用前後で値ストアに差異はないので，

$$\theta = \sigma'$$
$$\theta_1 = \sigma'_1$$

である．よって，任意の y に対して，

$$\sigma'(y), \sigma'_1(y) \notin \mathbb{Z}^\sharp \Rightarrow ($$
$$\sigma'(y) = \sigma'_1(y) \quad \wedge$$
$$\forall z \in ID((y \in F \wedge \rho(y) \cap \rho(z)^c \neq \emptyset)$$
$$\Rightarrow [S_v]^{\mathrm{v}}_{\left[z \mapsto (\sigma(z))^\sharp\right]\sigma}(y) \notin \mathbb{Z}^\sharp))$$

が成り立つ．

5. S が while e do S done のとき

評価規則より，if 文に展開できるので，if 文の証明と同様の議論から導ける．

以上のことから補題は成り立つ．

4.5 非干渉性の定理

定理 2 S, 値ストア θ, θ_1, タグストア γ, $\sigma = \mathrm{init_v}(\theta)$, $\sigma_1 = \mathrm{init_v}(\theta_1)$, $\rho = \mathrm{init_t}(\gamma)$, $Safe_v([S]^V_\sigma)$, $Safe_v([S]^V_{\sigma_1})$, ρ が型付け可能である，$\sigma(x) =_{L_i} \sigma_1(x)$ を満たす時，任意の $y \in L_o$, $z \in \mathrm{ID}$ に関して次の 2 つが成り立つ．

(公開値) $[S]^V_\sigma(y) = [S]^V_{\sigma_1}(y)$

(依存) $y \in \mathrm{F}$ かつ $\rho(y) \cap \rho(z)^c \neq \emptyset$ ならば
$[S]^V_{\left[z \mapsto ((\sigma(z))^\sharp)\right]\sigma}(y) \notin \mathbb{Z}^\sharp$

証明

$Safe_v([S]^{\mathrm{v}}_\sigma)$, $Safe_v([S]^{\mathrm{v}}_{\sigma_1})$ より，定理の一般化に関する補題の停止 1，2 を満たす．また，$Safe_v([S]^{\mathrm{v}}_\sigma)$, $Safe_v([S]^{\mathrm{v}}_{\sigma_1})$ より，すべての変数 x に関して，$x \in L_i \Rightarrow (\sigma(x), \sigma_1(x) \in \mathbb{Z}^\sharp)$ である．よって，$\sigma(x) =_{L_i} \sigma_1(x)$ から，定理の一般化に関する補題の公開値を満たす．ゆえに定理の一般化に関する補題の前提をすべて満たしたので，定理の一般化に関する補題が成り立つ．$Safe_v([S]^{\mathrm{v}}_\sigma)$, $Safe_v([S]^{\mathrm{v}}_{\sigma_1})$ と $Safe_v$ の定義より，$y \in L_o$ を満たす y に関して，$[S]^{\mathrm{v}}_\sigma(y)$, $[S]^{\mathrm{v}}_{\sigma_1}(y) \notin \mathbb{Z}^\sharp$ である．よって，

$$[S]^{\mathrm{v}}_\sigma(y) = [S]^{\mathrm{v}}_{\sigma_1}(y)$$
$$y \in \mathrm{F} \text{ かつ} \rho(y) \cap \rho(z)^c \neq \emptyset \text{ ならば}$$
$$[S]^{\mathrm{v}}_{\left[z \mapsto ((\sigma(z))^\sharp)\right]\sigma}(y) \notin \mathbb{Z}^\sharp$$

第5章　結論

ファイルにアクセス可能なユーザ集合を型情報として追加することで，プログラム内の変数間の情報漏れだけでなく，外部の情報に関しても情報漏れを検出できる体系を提示した．これにより，プログラム内の変数に機密度の高低を設定する視点と，アクセス可能なユーザ集合を設定する視点の2視点から，情報漏れを検査できるため，プログラムに対して情報漏れが起きないことをより強く保証できる．ただし，今回の手法では，ファイルやユーザは存在するものとして扱っているため，未作成のファイルやユーザに対しては，保証できない問題点がある．また，if文の解析において，実行・未実行両方の分岐先を解析しているため，計算時間が膨大になることも大きな問題点である．よって未作成のファイル，ユーザに対しても情報漏れを保証，解析の計算時間を減少することが今後の課題である．

謝辞

本研究を述べるに当たり，日頃から根気強く，熱心にご指導いただき，研究以外の点においても大変お世話になりました，山田俊行講師にこの場をお借りして大きな感謝を申し上げます．また，講義を通して良き指導をしてくださった大山口通夫教授，学生生活のサポートをしてくださった落合美子事務員にも併せて感謝いたします．そして，何気ない疑問にも丁寧にお答えくださり，勉学の励みを与えてくれた研究室の諸氏に感謝いたします．

関連図書

[GJ05] Gurvan Le Guernic and Thomas Jensen. Monitoring information flow. *In Proceedings Workshop on Foundations of Computer Security*, pp. 19–30, 2005.

[Kob03] 小林直樹, 白根慶太. 低レベル言語のための情報流解析の型システム. コンピュータソフトウェア, Vol. 20, No. 2, pp. 2–21, 2003.

[SM03] A. Sabelfeld and Andrew C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 1, 2003.

[VS97] Dennis Volpano and Geoffrey Smith. A type-based approach to program security. In *In Proceedings 7th Int'l Joint Conference FASE on the Theory and Practice of Software Development*, pp. 607–621. LNCS 1214, 1997.