

# 民法の証明責任と論理プログラミングの関係について

## Relationship between Burden of Proof in the Japanese Civil Law and Logic Programming

佐藤 健<sup>1\*</sup>

<sup>1</sup> 国立情報学研究所および総研大

<sup>1</sup> National Institute of Informatics and Sokendai

**Abstract:** This paper is discussing about relationship between “burden of proof” in Japanese civil suit and logic programming. In the Japanese civil suit, a burden of proof is a decision making tool by a court when an ultimate fact in the condition of rules is “non liquet”, that is, the state which we cannot conclude that the ultimate fact is true or not even after considering all available pieces of evidence. We show that the burden of proof is closely related with “negation as failure” in logic programming and provide a framework for “burden of proof” in logic programming.

### 1 はじめに

筆者は、現在、論理に基づく人工知能の具体的な応用分野として法学を対象に研究を行なっている。法学で論理的な解析を行う利点として以下の3点が考えられる。

- 暗黙の仮定が明らかになる。論理的には、すべての仮定を明示しなければ推論結果が出てこないが、法律文書のように自然言語文で書かれているとそのような仮定が、暗黙の仮定として現れないことも多く、法的概念の解析が困難になると考える。たとえば、今まで考えられてきた法的推論方法を論理的に解析して、それが機械的に導かれない場合には、隠れた推論図式が存在することが明らかになり、新たな法的知見が得られる。
- 論理的な解析をして法的文書が論理式のような厳密な形で書けるようになれば、法律概念自体を計算機上で扱うことが可能になり、より複雑な法的問題の解析に役に立ち、法律の条文間の矛盾の検出や、2つの法律表現の同一性の証明ができるようになる。たとえば、ある法律の表現が、別な法律の表現と同じであるかどうかについて、論理的な変換を施すことで、明らかになる。
- 論理的な問題と法的な問題を分離することにより法学者は法的な問題に集中できる。論理的な解析をすることにより、形式論理的に解決できる

部分かどうかを明らかにして、論理的な問題と法的な問題の切り分けができるのではないかと考える。もし論理的に解決可能であることがわかれば、論理学でのさまざまな知見を利用できる可能性がある。

### 2 民事訴訟における証明責任

民法は、当事者間の民事的な権利や義務を定めている法律であり、その法律を適用するにあたり、完全情報を仮定して、ある法律上の要件が（例外状況でないことを含めて）満たされればその結論が生じることを決めている。

たとえば、家を賃貸していた場合、賃借人が賃貸人に無断で他の人に転貸したときには、賃貸人は賃借人との賃貸借契約の解除ができる<sup>1</sup>。この場合、民法612条を適用するためには、以下のことが成立していなければならない。

- 賃貸人・賃借人間で賃貸借契約が締結されたこと。
- 賃貸人の家を賃貸借契約に基づき賃借人に引き渡したこと。
- 賃借人・転借人間で賃貸人の家（の一部）について賃貸借契約が締結されたこと。
- 賃借人が転借人に家（の一部）を賃貸借契約に基づき引き渡したこと。

\*連絡先：国立情報学研究所

〒101-8430 千代田区一ツ橋 2-1-2

E-mail: ksato@nii.ac.jp

本論文は、[2]の日本語訳縮小版に加筆したものである。

<sup>1</sup>民法第612条第1項 賃借人は、賃貸人の承諾を得なければ、その賃借権を譲り渡し、又は賃借物を転貸することができない。第2項 賃借人が前項の規定に違反して第三者に賃借物の使用又は収益をさせたときは、賃貸人は、契約の解除をすることができる。

- 転借人がその家（の一部）を使ったこと。
- 賃貸人が賃借人に対して賃貸借契約の解除の意思表示をしたこと。
- 賃貸人が賃借人に対して転貸借について承認を与えていなかったこと。
- 転貸借が賃貸人・賃借人間の信頼関係を破壊していないという例外的状況でないこと。

最初の要件である「賃貸人・賃借人間に賃貸借契約が締結されていたこと」という要件は、契約の解除の前提としてその解除の対象となる契約がなければならぬから必要である。この要件はそれだけみれば自然な要件のように見えるが、実は契約が締結されるためには、賃貸人・賃借人が未成年者でない等、契約をする法的な能力（民法では「行為能力」と呼んでいる。）があることが前提になっていたり、その契約について当事者間でその契約の意思の合致があることが前提になっていたりしている。このような、民法のところどころにあるすべての関連する前提が成立して初めて賃貸借契約が成立することになる。同様のことが他の要件にもいえ、それらすべての前提を満たした場合に、無断転貸を理由に賃貸借契約を解除できるのである。この例から、民法は、上記のような要件の成立・不成立が確定したときに、どのような権利や法律効果が成立するのかを決めていることがわかる。

これに対して、実際の裁判においては、民法における要件の成立・不成立が証拠不十分などの状況により完全に確定できないことがありうる。演繹論理的に完全に正しい結論を出そうとするならば、この場合には、要件の成否は「わからない」と答えるほかない。しかし、裁判所は裁判の当事者（原告・被告）に対して、その裁判上の紛争について判決を示す応答義務があるため、「わからない」という答えではすまないのである。このため、そのような不完全な状況下で判決を下すため、裁判所は、あらかじめ、ある要件が真偽不明の場合に備え、デフォルトの真偽値を用意している。これが実は証明責任の正体なのである。

通常、証明責任といえ、ある要件について当事者のどちらかがその要件を証明しなければならない責任を負っていて、その責任を果たさないとその要件は偽となり、証明責任を負う方が不利な結論を得ることになるということを意味すると考えられている。正確に言えば、証明責任とは、当事者の側から見ると事実が真偽不明の場合に実体法の適用または不適用が指示される結果、当事者の一方が負う危険ないし不利益のこととされている [4](p.459)。これは、上のように裁判所が不完全情報下でも結論が出せるように各要件にデフォルト値を用意したことが原因となっている。つまり、そのデフォルト値が成立すると不利になる側が、裁判に

勝つために、そのデフォルト値にならないようにその否定を証明しなければならないということである。

上の転貸借の例でいえば、「賃貸人・賃借人間で賃貸借契約が締結されたこと。」「賃貸人の家を賃借人に引き渡したこと。」「賃借人・転借人間で賃貸人の家（の一部）について賃貸借契約が締結されたこと。」「賃借人が転借人に家（の一部）を引き渡したこと。」「転借人がその家（の一部）を使ったこと。」は、もし証拠調べをしてもその真偽値が不明の場合にデフォルト値は偽と設定されている。そうすると、これらすべてを真と証明しない限り、賃貸借契約を解除できないことになる。すなわち、賃貸人が裁判を起こした場合は、その証明に失敗すると敗訴することになる。当事者の側から見れば、賃貸人は上記の要件についての証明責任を負うことになる。さらに、「賃貸人が賃借人に対して転貸借について承認を与えていなかったこと。」や「転貸借が賃貸人・賃借人間の信頼関係を破壊していない」という例外的状況でないこと。」のデフォルト値は真と設定されている。したがって、これらの2つの要件については、契約解除の要件を満たすことが仮定されるため、賃貸人側はこれらについての証明をする必要はなく、「転貸借の承認を与えていたこと。」または「信頼関係を破壊していない」という例外的状況であること。」の証明を行う必要が賃借人側にあることになる。

このような考え方によると、民法の各条文の各要件について、どの要件の真偽値が真又は偽に仮定されるのかが問題になる。もし、あらかじめデフォルト値がわかっていなければ、当事者はどちらが証明責任を負うかわからないので、当事者はどの要件について証拠を集めるかなどの立証活動の計画を立てることができず、裁判が混乱してしまう。このため、裁判所では、あらかじめデフォルト値を決めておくという努力をしており、司法研修所は「要件事実論」[1, 3]という理論を研究・開発している。要件事実論では、民法の各条文の各要件に該当する事実について当事者が主張立証活動を行い、その結果として裁判所がその事実の存否について認定し、それに対応する要件の真偽値を決める。もし、真偽値が不明なものがあれば、証明責任に基づいて、その要件の成立不成立を仮定的に定める。それによりすべての要件の真偽値を決定することで、民法の条文を適用できることになり、それらの条文を適用することで判決を決めることになる。

要件事実論を賃貸借契約における無断転貸の例で詳しく説明する。

- 原告が、被告に対して原告所有の家を賃貸していたが、被告が無断で彼の姉にその家の部屋を使わせたことを理由として賃貸借契約を解除することを考える<sup>2</sup>。

<sup>2</sup>本論文では、賃貸借契約の終了を contract.end と表す。

- 要件事実論によれば、原告は以下を証明する必要がある。言い換えれば、以下の事実を証明できない場合には、その証明できない事実に対する真偽値が偽となり、解除の条文を適用できなくなり、原告は敗訴することになる。
  - － 原告・被告間に賃貸借契約が締結されていたこと<sup>3</sup>。
  - － 原告の家を賃貸借契約に基づき被告に引き渡したこと<sup>4</sup>。
  - － 被告・彼の姉（転貸人）間に家の部屋の賃貸借契約が締結されていたこと<sup>5</sup>。
  - － 被告が被告の姉に賃貸借契約に基づき家の部屋を引き渡したこと<sup>6</sup>。
  - － 被告の姉がその家の部屋を使ったこと<sup>7</sup>。
  - － 原告が被告に対して賃貸借契約の解除の意思表示をしたこと<sup>8</sup>。
- 要件事実論では、これに対して、被告は解除を妨げる抗弁として以下のどちらかを主張する必要があるとしている。言い換えれば、以下のどちらかの事実を証明できない場合には、その証明できない事実に対する真偽値が偽となり、解除を妨げることができなくなり、被告は敗訴することになる。
  - － 原告が転貸借を承諾していたこと<sup>9</sup>
  - － 転貸借が原告との信頼関係を破壊するには足りないものと評価される事実があること<sup>10</sup>
- 要件事実論では、第二の抗弁に対する再抗弁として以下を原告が主張しなければならないとしている。言い換えれば、以下の事実を証明できない場合には、再抗弁が成立せず、上記の第二の抗弁が成立して、原告が敗訴することになる。
  - － 転貸借が原告との信頼関係を破壊するに足りないものとの評価を障害する事実があること<sup>11</sup>

### 3 要件事実論の論理プログラミングによる定式化

この要件事実論を論理で表現することを考えてみよう。ただし、要件事実論は通常の演繹論理で定式化するのは不可能であることに注意されたい。通常の演繹論理では、情報が付け加わっても今までに得た結論が変わることがない。この性質を「単調性」と呼ぶ。通常の演繹論理では、真偽のわからない命題はわからないままであり、情報が付け加わって確実になったときにのみ真偽を確定していくので、そのような単調性が保たれる。しかし、要件事実論を用いた裁判上の推論はこれと異なり、ある情報状態で要件事実の真偽が不明の場合にその真偽を無理やり決定してしまうのであるから、その後、あらたな事実が判明して仮定された真偽値とは異なる真偽値となり、その結果、それに基づいた推論結果の真偽が変わりうる。たとえば上の転貸借の例でいえば、「賃貸人・賃借人間で賃貸借契約が締結されたこと。」、「賃貸人の家を賃借人に引き渡したこと。」、「賃借人・転借人間で賃貸人の家（の一部）について賃貸借契約が締結されたこと。」、「賃借人が転借人に家（の一部）を引き渡したこと。」、「転借人がその家（の一部）を使ったこと。」および「賃貸人が賃借人に対して解除の意思表示をしたこと。」が証明され、「賃貸人が賃借人に対して転貸借について承認を与えていたこと。」および「転貸借が原告との信頼関係を破壊するには足りないものと評価される事実があること。」について第一審において真偽が不明だったとしよう。すると「賃貸人は契約解除できる」という法的命題は真となる。しかし、控訴審で新たに、「賃貸人が賃借人に対して転貸借について承認を与えていたこと。」が真と判明した場合には、賃借人の抗弁が成立して、「賃貸人は契約解除できる」という法的命題は偽となる。つまり、証明責任による推論は単調性を持たないのである。したがって、このような推論を演繹論理によって定式化することは困難である。

そこで本論文では、非単調性をもつ論理的定式化として、論理プログラミングを用いる。論理プログラミングはルールの形をしており、法律の条文との親和性も高く、さらに失敗による否定（Negation as Failure）により非単調性をもっているため、要件事実論の定式化に適していると考えたからである。

以下では、まず要件事実論の数学的定式化を行い、その定式化に合致した論理プログラミングによる定式化を与える。

要件事実論では、要件事実を2つの集合に分ける。一つは、原告に有利になる事実であり、もう一方は、被告に有利になる事実である。ここで原告が求めている判決を「結果」と呼ぶことにする。すると、要件事実

<sup>3</sup>この事実を `agreement_of_lease_contract` で表す。

<sup>4</sup>この事実を `handover_to_lessee` で表す。

<sup>5</sup>この事実を `agreement_of_sublease_contract` で表す。

<sup>6</sup>この事実を `handover_to_sublessee` で表す。

<sup>7</sup>この事実を `using_leased_thing` で表す。

<sup>8</sup>この事実を `manifestation_of_cancellation` で表す。

<sup>9</sup>この事実を `approval_of_sublease` で表す。

<sup>10</sup>この事実を `fact_of_nonabuse_of_confidence` で表す。

<sup>11</sup>この事実を `fact_of_abuse_of_confidence` で表わす。

は、その結果を導く事実とその結果を導かない事実に分けられることになる。

$S$  を要件事実の集合とする。ここで要件事実の分別集合 (*critical set*) を以下のように定義する。

#### 定義 1

- $\emptyset$  は結果を導かない分別集合である。
- $S$  が結果を導かない分別集合であるとして、 $T$  が  $S$  を含む集合とし、 $T$  は結果を導くとする。このとき、 $T \setminus S$  ( $T$  と  $S$  の差集合) のどの要素を取り除いても結果を導かないとすれば  $T$  も分別集合である。なお、 $T \setminus S$  を  $S$  に対する攻撃集合 (*attack set*) と呼ぶ。
- $S$  が結果を導く分別集合であるとして、 $T$  が  $S$  を含む集合とし、 $T$  は結果を導かないとする。このとき、 $T \setminus S$  ( $T$  と  $S$  の差集合) のどの要素を取り除いても結果を導くとすれば  $T$  も分別集合である。なお、 $T \setminus S$  を  $S$  に対する防御集合 (*defense set*) と呼ぶ。

上で、攻撃集合は、結果の成立を導くものであり、防御集合は、結果の成立を妨げるものである。

例 1 上の例では、空集合は上の定義より分別集合である。そして、 $T_1 = \{$

```
agreement_of_lease_contract,  
handover_to_lessee,  
agreement_of_sublease_contract,  
handover_to_sublessee,  
using_leased_thing,  
manifestation_of_cancellation }
```

は分別集合となる。

なぜならば、 $T_1$  は結果である `contract_end` を導き、もし  $T_1 \setminus \emptyset (= T_1)$  の要素の一つをどれか削除した場合には、その集合は、`contract_end` を導かなくなるからである。したがって、 $T_1$  は  $\emptyset$  に対する攻撃集合となる<sup>12</sup>。一方、 $T_2 = T_1 \cup \{ \text{approval\_of\_sublease} \}$  は分別集合となる。なぜならば、 $T_2$  は `contract_end` を導かず、もし  $T_2 \setminus T_1$  の要素の一つをどれか削除した場合には、その削除した集合は、`contract_end` を導くようになるからである。したがって、 $T_2 \setminus T_1 = \{ \text{approval\_of\_sublease} \}$  は  $T_1$  に対する防御集合となる。

定義 2 要件事実木  $T$  とは、以下の条件を満たすラベル付き有限有向木  $\langle V(T), E(T) \rangle$  である。

1. 根のラベルは、原告が求める「結果」である。そして、根からのすべての子ノードには、“*ATTACK(name)*” というラベル付けがされる。このノードを、攻撃ノード (*attack node*) と呼ぶ。ここで *name* は子ノードの名前を表す。そして、各子ノードには、 $\emptyset$  に対する相異なる攻撃集合が関連付けられる。さらに子ノードからルートノードへの有向辺が  $E(T)$  に含まれる。
2.  $v \in V(T)$  を攻撃ノードとし、 $v$  ノードから有向辺をたどって根ノードに至るノードすべてに関連付けられているすべての集合 (攻撃集合、防御集合の両方) の合併集合を  $T$  とする。このとき  $v$  のすべての子ノードには、“*DEFENSE(name)*” というラベル付けがされる。このノードを、防御ノード (*defense node*) と呼ぶ。そして、各子ノードには、 $T$  に対する相異なる防御集合が関連付けられる。さらに子ノードから  $v$  への有向辺が  $E(T)$  に含まれる。
3.  $v \in V(T)$  を防御ノードとし、 $v$  ノードから有向辺をたどって根ノードに至るノードすべてに関連付けられているすべての集合 (攻撃集合、防御集合の両方) の合併集合を  $T$  とする。このとき  $v$  のすべての子ノードには、“*ATTACK(name)*” というラベル付けがされる。このノードを、攻撃ノード (*attack node*) と呼ぶ。そして、各子ノードには、 $T$  に対する相異なる攻撃集合が関連付けられる。さらに子ノードから  $v$  への有向辺が  $E(T)$  に含まれる。
4. 以上によって定義される最小のグラフが要件事実木である。

無断転貸に基づく解除の例に関する要件事実木を図 1 に示す。

要件事実論では、当事者が主張していない事実が含まれる条文についてはその適用を行わない。これは「その事実を当事者が主張しなければ、判断の基礎とすることはできない」という民事訴訟法における弁論主義の第 1 テーゼによる。これは、人工知能の観点から見ると、推論のための探索空間のカットを当事者の主張によって行うことにより、推論の効率化を図る工夫と見られることもできる。

この制約を表現するため、「主張事実 (*alleged fact*)」という概念を導入する。

要件事実が主張されているとは、原告・被告のどちらかが  $f$  の存否を主張していることをいう。

定義 3 要件事実木  $T$  と主張事実の集合  $A$  に関するブロック図  $B$  とは、要件事実木  $T$  の部分木  $\langle V(T), E(T) \rangle$  である。

- $B$  の根は  $T$  である。

<sup>12</sup>なお、要件事実論では、 $\emptyset$  に対する攻撃集合は「請求原因」と呼んでいる。

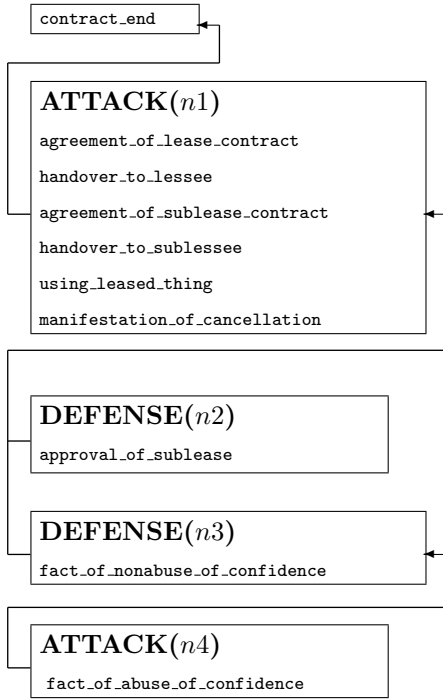


図 1: 要件事実木

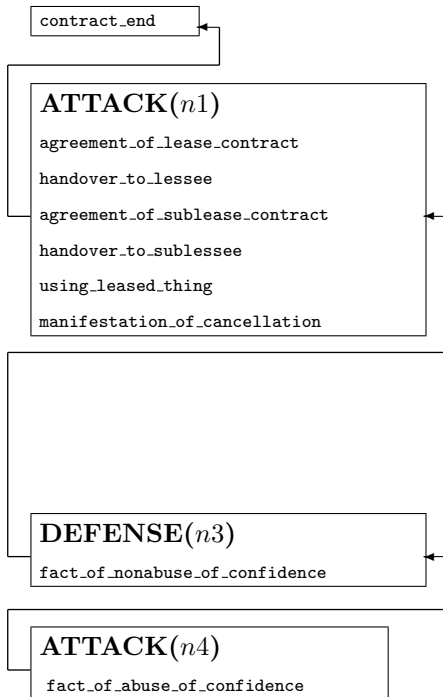


図 2: approval\_of\_sublease の主張のないブロック図

- $v \in V(B)$  のとき  $v' \in V(T)$  が  $v$  の  $T$  における子ノードであって、 $v'$  に関連づけられている攻撃集合または防御集合が  $A$  に含まれているとき、 $v' \in V(B)$  かつ  $(v', v) \in E(B)$  である。

ブロック図は、要件事実木のノードのうち主張のない事実を含むノード以下をすべて削除した木となっている。無断転貸に基づく解除の例で approval\_of\_sublease の主張のない場合のブロック図を図 2 に示す。

さらに、要件事実論を適用する場合は、裁判所は各要件事実  $f$  について当事者の立証活動が終了した時点（口頭弁論終結時）において、 $f$  が成立しているか成立していないかを決定していることが前提である。これは、要件事実  $f$  の確からしさが、裁判官が  $f$  が成立していると考えた確からしさ（「証明度」と呼ぶ）を超えているかどうかで決める。ここでは、主張されている事実集合の中で、証明度を越えた要件事実の集合を  $P$  で表すとする。この  $P$  を用いて原告が求める「結果」が成立するかどうか以下のように判断される。

定義 4 ブロック図におけるノード  $n$  が  $T, A, P$  に関して生きているとは、以下の条件を満たすことをいう。

- $n$  に関連付けられている要件事実がすべて  $P$  に含まれていること。
- $n$  の子ノードで生きていないものがないこと。

$T, A$  に関するブロック図  $B$  が  $P$  の元で根ノードの結果  $c$  を満たしているとは、 $c$  が生きていいことをいい、 $B_T^A, P \vdash c$  と表現する。

なお、もし  $n$  が生きていなければ、死んでいるという。無断転貸に基づく解除の例において、すべてのノードの要件事実が証明度を越えている（すなわち  $P$  にすべて含まれている）とする。すると、ノード  $n4$  は、下に子ノードがないので生きており、そのためその親ノードの  $n3$  は死んでいる。すると、根ノードの子ノードで生きていないため、 $B_T^A, P \vdash \text{contract\_end}$  となる。

## 4 論理プログラミングへの変換

要件事実木  $T$  を以下の論理プログラム  $tr(T)$  に変換することを考える。この変換は、要件事実木のそれぞれのノードを論理プログラムのルールに変換するものである。

- 根ノードの結果を  $c$  とし、 $n_1, \dots, n_m$  を子ノードの名前とすれば、根ノードを変換したルールは以下のようなになる：

$$\text{prove}(c): \neg \text{prove}(n_1).$$

```

    :
    prove(c): -prove(nm).

```

- ノード  $n$  の名前が  $name_n$  であって、そのノードに関連付けられている要件事実の集合が  $\{f_1^n, \dots, f_{m_n}^n\}$  であって、 $n$  の子ノードの名前が  $d_1^n, \dots, d_{k_n}^n$  のとき、 $n$  を変換したルールは以下ようになる。

```

    prove(namen): -
      prove(f1n), ..., prove(fmnn),
      not prove(d1n),
      :
      not prove(dknn).
    prove(fin): -alleged(fin), plausible(fin).
    (1 ≤ i ≤ mn)

```

ここで、*not* は「失敗による否定」を表す。

さらに  $\mathcal{A}$ 、 $\mathcal{P}$  にある要件事実をそれぞれ以下のプログラム  $tr(\mathcal{A})$  and  $tr(\mathcal{P})$  に変換する。

```

tr(ℳ) = {alleged(f).|f ∈ ℳ}
tr(ℙ) = {plausible(f).|f ∈ ℙ}

```

例 2 無断転貸に基づく解除の例の変換論理プログラム  $tr(\mathcal{T})$  は以下ようになる。

```

prove(contract_end) :- prove(n1).
prove(n1) :-
  prove(agreement_of_lease_contract),
  prove(handover_to_lessee),
  prove(agreement_of_sublease_contract),
  prove(handover_to_sublessee),
  prove(using_leased_thing),
  prove(manifestation_of_cancellation),
  not prove(n2), not prove(n3).
prove(agreement_of_lease_contract) :-
  alleged(agreement_of_lease_contract),
  plausible(agreement_of_lease_contract).
:
prove(n2) :-
  prove(approval_of_sublease) .
prove(approval_of_sublease) :-
  alleged(approval_of_sublease),
  plausible(approval_of_sublease).
prove(n3) :-
  prove(fact_of_nonabuse_of_confidence),
  not prove(n4).
prove(fact_of_nonabuse_of_confidence) :-
  alleged(fact_of_nonabuse_of_confidence),
  plausible(fact_of_nonabuse_of_confidence).
prove(n4) :-

```

```

  prove(fact_of_abuse_of_confidence).
  prove(fact_of_abuse_of_confidence) :-
    alleged(fact_of_abuse_of_confidence),
    plausible(fact_of_abuse_of_confidence).

```

以下の定理は要件事実木での推論結果と変換された論理プログラムの推論結果が一致することを示す。

定理 1  $\mathcal{T}$  を要件事実木  $\mathcal{A}$ 、 $\mathcal{P}$  をそれぞれ主張事実、証明度を超えた事実とし、 $tr(\mathcal{T})$ 、 $tr(\mathcal{A})$ 、 $tr(\mathcal{P})$  をそれぞれを変換した論理プログラミングとする。 $B$  を  $\mathcal{T}$ 、 $\mathcal{A}$  に関するブロック図としたとき、 $B_{\mathcal{T}}^{\mathcal{A}}, \mathcal{P} \vdash c$  のとき、かつそのときに限り  $tr(\mathcal{T}) \cup tr(\mathcal{A}) \cup tr(\mathcal{P})$  のモデル  $M$  に対して、 $M \models prove(c)$  となる。

## 5 おわりに

本稿では要件事実論に数学的な意味を与え、それが論理プログラミングによって計算できることを示した。今後の課題としては以下がある。

- 民法に記載されていない要件をどのように導いていくか。たとえば、無断転貸の例では、実は、`fact_of_nonabuse_of_confidence` という要件は、民法の条文には記載されておらず、最高裁判所が付加した要件である。このような隠れた要件をどのように発見するかが問題である。
- 民法においては条文によっては証明責任が明らかでない場合がある。このような場合にどのように証明責任を分配するかが問題である。

謝辞 本研究は国立情報学研究所のグランドチャレンジ経費によりサポートされている。また、本論文に対して貴重なコメントをいただいた国立情報学研究所の浅井健人、久保田理広、古川昂宗、中村恵、西貝吉晃各特任研究員に感謝します。

## 参考文献

- [1] 伊藤滋夫, 要件事実の基礎, 有斐閣 (2001)
- [2] Ken Satoh, Masahiro Kubota, Yoshiaki Nishigai, Chiaki Takano, “Translating the Japanese Presupposed Ultimate Fact Theory into Logic Programming”, JURIX 2009, pp. 162–171 (2009).
- [3] 改訂問題研究要件事実, 司法研修所編, 法曹会 (2007).
- [4] 高橋宏志, 重点講義民事訴訟法 上, 有斐閣 (2005).