法律知識の事象的/属性的読みを区別した推論システム

兼岩 憲[†] 東条 敏[†]

法的推論の研究では，法律知識を計算機上でどのように表現するかが問題となる．従来の法的推論システムは，知識の一回性/広域性や適用範囲の特性に十分対応しておらず，法令文などの記述がシステムに合わせて書き換えられ，本来持っている暗黙の意味や推論力を損なう場合がある．法律知識は事件の記述から形式的な法令文にわたっており，それぞれが意図する意味から適切に推論するためには，自然言語文に現れる事象と属性を区別しなければならない．本稿では，自然言語の解釈に注目し，法令文の事象的/属性的読みからもたらされる述語の上位導出や暗黙対象への量化の違いをサポートした推論システムを提案する．それにより，事象的読みと属性的読みとに依存した柔軟な推論メカニズムを使って法令文の適用が可能になる．さらに，述語のイベント/プロパティの識別，型表現，および型への量化の概念を導入して，読みの特性に基づいた型変換と引数の操作による単一化を備えた論理プログラミングを形式化する．最後に，推論システムを実装し，法律知識の例によって実行した結果を示す．

A Legal Reasoning System with Event and Property Interpretation for Legal Knowledge

KEN KANEIWA[†] and SATOSHI TOJO[†]

In legal reasoning, the knowledge representation is one of the toughest problems, and thus far many subtle representation of natural language have been lost when they have been formalized. In particular, when the legal knowledge is represented in the form of predicate logic, the distinction between one occurrence of an event and a universal property has been neglected and thus the implicit meaning of the original rules (and therefore the ability to make inferences from them) has been lost. In this paper, we distinguish events and properties in predicates, as the problem of quantification. We introduce a legal reasoning system based on order-sorted logic, where we hierarchically sort both of predicates and objects, and distinguish events from properties by superordinate-predicate derivation and quantification of implicit objects. As a result, we would be able to apply legal rules to flexible inference that operate on both properties and events. We implement a logic programming system with unification/resolution mechanisms, where the distinction is realized as the substitution of sort and the manipulation of arguments, and show the feasibility.

1. はじめに

法的推論で扱う知識は，事件の記述から形式的な法令文にわたっていることから，適切な推論のために命題が持ち合わせている自然言語文としての多様な解釈を考慮しなければならない難しさがある．法的推論システムには，主に論理型言語をベースとした推論エンジンが使われているが，その問題に対応して法律知識を直接記述できるような十分な記述力を持っているとはいえない．したがって，実際にシステムを実装するときには，多くの法律知識を計算機に直接的に入力（または記述）することができない．そのために，論理型言語で推論可能なように知識を書き換える必要があり，書き換えによって法律知識が持っていた意味を失う危険性がある．したがって，我々が主張する方法は，法令文の意味する解釈や適用範囲を漏らさずに示せるような記述力と推論力を持った言語によって実現することである．

このような問題に関連して，法的推論に適すような論理型言語の研究がなされている．法的推論システム new HELIC-II[8),9)]では，その推論エンジンに LOGIN[1),2)] の ψ-term を拡張した表現 H-term を持つ論理型言語を開発した．また，演繹データベース言語 *Quixote*[13),14)] を使った法的推論の研究もされている[17)]．一方，代表的な判例ベースシステムである HYPO[4)]はフレーム

† 北陸先端科学技術大学院大学情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology

とファクト的述語（Factual Predicate）の2つからなる事例表現言語を持つが，固定された記述方法であるため論理型言語のような汎用性がなく，知識形態ごとのデータ構造を要求される欠点を持つ．さらに，Grebe[6]は，ファクトとルールを基本とした知識表現を持つシステムで，判例の状況を意味ネットワークで記述している．我々は，法律向けに洗練されたnew HELIC-IIの知識表現から，さらに拡張して，述語表現をイベントとプロパティに区別し，それに基づいた型階層上の推論を備えた型階層論理[15]を新たに構築した．その目標は，自然言語や人が保持している知識を記述し推論することにあった．また，知識表現の研究分野では，時相論理として時間概念を用いて，自然言語における事象や属性の区分を定めた研究[3),7),10)]がなされている．

本研究の目的は，自然言語文の持つ知識の特性に主眼を置き，法令文の事象的/属性的読みからもたらされる述語の上位導出や暗黙対象への量化の違いをサポートした推論システムを提案することである．法令は法律の目的に応じて条文の形で記述されているが，ルールとして利用する観点から，事象的読みと属性的読みの区別が曖昧であると法令文が持っていた一回性/広域性や適用範囲などが引き出せず意図した導出が行えなくなる．2つの読みは，異なった推論結果をもたらし，もし従来の論理型言語で表そうとするならば，読みの違いから得られる推論を考慮してアドホックにわざわざ記述しなければならず，かつ十分対応できるとも限らない．本稿では，我々が提案した型階層論理によって項と述語の型階層，型への量化や述語のイベントとプロパティの区別が，2つの読みに対応し法律知識を元の知識に近い形で特性を保持したままで記述するのに有用であることを示す．

本稿は次のような構成をとる．2章は法律知識を分析して，法的推論における事象と属性の読みの違いを明らかにするとともに，その知識を使って推論を行うための問題を示す．3章は，2章で提起した問題を解決するために型階層論理を使った法律知識の記法と推論方法を提案する．4章は論理プログラムによる構文と推論メカニズムの定義によって推論システムを形式化して，それに基づいて実装したシステムによる実行例を示す．最後に5章では，研究成果と今後の課題について考察する．

2. 法的推論における事象と属性の問題

2.1 法律知識の分析

計算機に入力することを前提に，知識が持つ法律特有の性質を分析する．法律知識は，その特徴から以下のように，事例，法令文および背景知識の3つのタイプ☆に分類することができる．

- 事例：行為，事象や状況などの事実の集まりからなる事件の記述
- 法令文：事例を適用して法的結論（判断）を導く規則
- 背景知識：法律用語以外の日常使われる言葉の概念

これらの知識は，複数の対象と対象間の関係（構造，性質や事象）との2つの要素から構成すると見なすことができる．それぞれ事例，法令文と背景知識の知識形態の違いは，対象と関係の抽象度（または量化），および述語（関係）による言明の特性からきており，その多様性や特殊性が知識表現とその推論に大きく影響する．以上の分類に対して，知識の要素である対象と関係を分析していくことで，法律知識の自然言語文としての特徴を明らかにしていく．

(1) 事 例

事例は，**表1**のように[行為]，[事象]，[状況]に分類できて，それぞれ知識の形式，知識の特性を持っている．事例は，ある事件を詳細に描写した情報であるので，行為，事象や状況の記述に含まれる対象（「知識の形式」の～部分）は，具体的な内容であることが多い．その表現には，対象を特定する情報（たとえば，名前，年齢など）が付加される．**図1**において，左右の幅が大きいほど対象の表現が全称的で抽象的であるとすると，事例の対象は「人」のように抽象的ではなく「20歳の太郎」のように一意に決まる情報（インスタンス）であると分析できる．さらに，行為や事象は，事件に登場する対象による一回性の出来事であり，対象と同様，図1の右枠に示すように時間「X月X日」や場所「太郎の家」が特定に決まる．そのような出来事の集まりと状況が事件を構成する．

(2) 法令文

法令文は，**表2**のように[違法]，[刑罰]，[権利]，[制限]に分類できる．法令文の役割は，多くの事例を適用することにあるので，法律を適用するある幅を持った対象範囲に対して記述される．たとえば，特定の範囲の人々や会社などに対して違法性や権利を明らかにする．[違法]と[刑罰]を示す法令文は，対象範囲の行為や事象に関する事項をルールの条件として含み，特に先に述べた事例内の具体的情報を適用して判決を導

☆ 詳細には，事例は事件ごとに複数に分類されており，法令文は法律の種類に分けられる．

表1 事例の分析
Table 1 An analysis of cases.

分類	知識の形式	知識の特性
行為	～が（～を）～した	具体的主体による一回性の行動
事象	～が起きた	主体者以外から発生する一回性の事実
状況	～という状況	事件における物，人，場所や時間などの状況

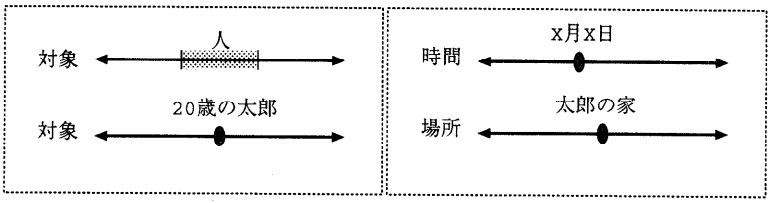


図1 対象の具体性
Fig. 1 The concreteness of objects.

表2 法令文の分析
Table 2 An analysis of the text of a law.

分類	知識の形式	知識の特性
違法	～したならば，～罪である	対象範囲の行為からもたらされる事実
刑罰	～罪ならば，～に処する	対象範囲の行為からもたらされる判決
権利	～ならば，～権利がある	対象範囲が持つ属性
制限	～と～とは，～関係にある	2つ以上の対象範囲を制限する属性

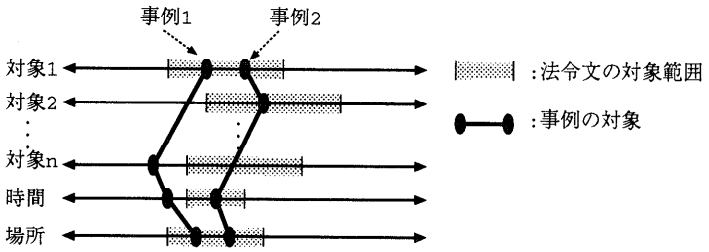


図2 対象から構成する事例と法令文
Fig. 2 A rule and cases built by objects.

く．本稿では，この特徴を持つ法令文を**事象的読み**の法令文と呼ぶ．対して，[権利] と [制限] を示す法令文は，対象の性質や環境を制限し，継続的に成り立つ関係をルール化していて，事例から結論を導く以外に対象範囲の人物などの性質や事実から権利などを示す役割を持つ．このような法令文を**属性的読み**の法令文と呼ぶ．

法令文が，事例の知識形態と異なる点は，(a) インスタンスではなく対象範囲への言明，(b) 事象的/属性的読みの違う役割を持つことである．**図2**のように，法令文は網かけ部分の範囲を持つ対象の集まりととらえられ，事例は対象の点のつながりであるととらえられる．相違点 (b) のように，言明が事象的/属性的に解釈できることは，時相論理の研究で示されている．その研究において，言明を分類する概念として，Allen[3]はイベント/プロセス/プロパティ☆を導入し，また，McDermott[7]は，ファクト/イベントによる区別を行っている．イベントはある時点で起こった事象であり，プロパティ（またはファクト）は，1つの時点（または時区間）を越えて成り立つ事実としている．その考察より，事象的読みの言明を一回性に量化し（$\exists t$），属性的読みの言明を通時的に量化する（$\forall t$）ことが考えられる．本研究の出発点は，このような述

☆ 本稿では，事象と属性はそのような知識そのものを示し，事象的/属性的読みは1つの法令文（または言明）の解釈の仕方を示している．さらに，イベント/プロパティは，論理プログラム言語などに含まれる述語や論理式を区別する名前（identity）として使われる．

表3 背景知識の分析
Table 3 An analysis of the background knowledge.

分類	知識の形式	知識の特性
項	～は，～の上位概念である	物，人などの表現の定義
述語	～は，～の上位概念である	性質，関係や事象などの表現の定義

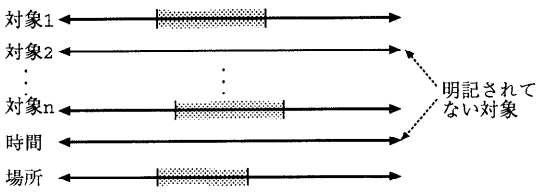


図3 知識表現の不完全さ
Fig. 3 The incompleteness of knowledge representation.

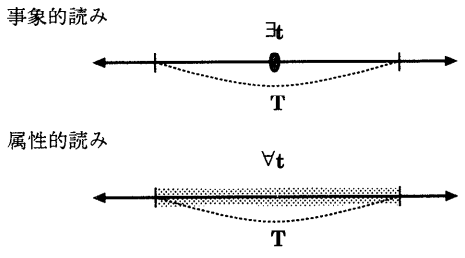


図4 暗黙の対象への量化
Fig. 4 The quantification of implicit objects.

語の特性の区別を量化という観点からとらえることにある．さらに，Allenらの論理が時間関係だけでこの区別を議論しているのに対して，我々の独自の考察は，事象的読み/属性的読みにおいてある言明が成立しているとき，その違いは言明の中に存在する時間に限らない世界，対象などに対する量化の区別によるものだと考える点である．この読みの区別は，柔軟な推論に不可欠である．法令文を実際に計算機に入力するとき，図2とは違って**図3**のようにすべての対象について記述されていない不完全さがあり，すべての記述は現実的に不可能である．しかし我々は，事象的/属性的読みの考えにより，図3で示す暗黙の対象（不明記の対象）に対する扱いを，読みの違いから量化によって判断できると考える．**図4**で示すように，言明が事象的読みのとき，対象は広域的な範囲 T☆の中で存在限量的に振る舞い，属性的読みのときは T の中で全称的に振る舞う特徴を持っている．

(3) 背景知識

背景知識の概念定義は，**表3**のように名詞概念の[項]と動詞または形容詞概念の[述語]に分けることができる．本稿では，背景知識は法律用語以外の一般的な言葉の概念を定義した概念階層とする．背景知識に定義される概念は，概念間の上下関係を示すことで概念とその抽象度を定めることができる基本的なものをいう．こうして概念階層で定義された概念を**型表現**といい，それによって作られた階層構造を**型階層**という．特殊な意味を持ち簡単には定義できない法律用語は法令文の中で宣言される．

☆ T は，直観的にはその対象としてとりうるすべてのものを表していて，その対象の最大の範囲を持ち，対象の種類に依存して大きさが決まる．

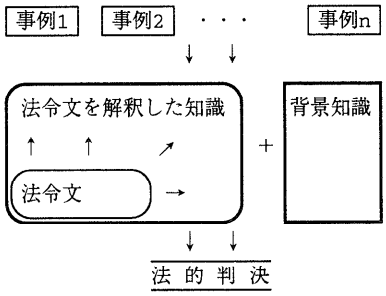


図5 法的推論システム
Fig. 5 A legal reasoning system.

2.2 法律知識の記述例と法令文の解釈

法的推論は，法令文と背景知識が与えられていて，そこに事例を追加して演繹的に法的判決を導くことを目的とする．したがって，我々は**図5**のような法的推論システムを想定する．たとえば，法令文[16]，背景知識と事例は**図6**のようなものである．これらの知識から，我々が期待している演繹結果は，たとえば，(a) 太郎は違法行為をしたか?，(b) 次郎は違法行為をしたか?，(c) 次郎や花子は酒場で酒を飲む権利を持つか? などに対する答である．しかしながら，法令文はいく通りもの解釈ができるような曖昧な記述であったり，その法令の本質部分にしかふれられてなかったりするために，そのままの形では事例を直接適用しても期待する演繹結果を導くのは困難である．そのために，法的推論では法令文に事例を適用できるように条文の解釈を行う．

たとえば，刑法199条殺人罪は，そのままの形では事例(1)に適用できない．なぜならば，事例(1)には太郎が花子を殺したという記述がされていないからである．したがって，殺人罪の解釈には，「人が人に殺意を持ってある行為を行い，死に至らしめたならば殺人である」が追加されるべきである．そして，法令文の

【法令文】
法令文 (1)[刑法 199 条殺人罪]：
人を殺したものは死刑又は無期もしくは 3 年以上の懲役に処する．
法令文 (2)[未成年者飲酒禁止法]：
未成年者が酒類を飲用すること及び酒類の販売業者が未成年に酒類を販売又は供与することを禁止する．
【背景知識】
項 (1)：太郎と花子は，成人である．
項 (2)：次郎は，未成年である．
項 (3)：成人と未成年は，人である．
述語 (1)：殺人，未成年者飲酒と暴行は，違法行為である．
述語 (2)：死ぬと飲むは，正当行為である．
述語 (3)：違法行為と正当行為は，行為である．
【事例】
事例 (1)：「20 歳の男性の会社員，太郎は花子に対する日頃の怨みから自宅へ呼び出し，殺す意志で花子に棒で殴る暴行を加えた．その結果，花子は死亡した．」
事例 (2)：「昨夜，次郎は花子を誘っていっしょに日本酒を飲んだ．花子は 21 歳の成人であるが，次郎は 19 歳の未成年であった．」

図 6 法律知識の記述例
Fig. 6 An example of legal knowledge.

そのままの解釈である「殺人を行った人は，3 年以上の有罪である」によって刑罰を示す．

また，未成年者飲酒禁止法タイプの法令文は，権利や禁止を述べている規則である．その解釈は「未成年は酒類を飲む権利を持たない」となり，肯定的に示すと「成人は酒類を飲む権利を持つ」となる．「飲む権利」という記述は，属性的読みを表していて，言明が広域的に成り立つ特性を持つ．つまり，広域的な属性的読みの解釈を丁寧に記述すると「成人はいかなるときも，いかなる場所でも，酒類を飲む権利を持つ」となるのである．同様にして，残りの部分は「販売業者は，成人に対してのみ酒類を販売する権利を持つ」と解釈できる．

先の知識の分析に基づくと，図 6 の法令文は，次のように事象的読みと属性的読みを含んでいる．

法令文 (1)[刑法 199 条殺人罪]：事象的読み
法令文 (2)[未成年者飲酒禁止法]：属性的読み

法的推論を計算機上で実現するには，**図 7** のように法令文から，その解釈によって対象範囲を明示して，さらに事象的/属性的読みの特性による暗黙の対象を導くことにより，法令文を適切に拡張する必要性がある．

しかし，通常の論理型言語では，これらの読みの違いによる記法や推論を持たない．述語論理をベースにした言語では，アドホックな対処として新たな述語とルールを追加する．たとえば，述語「飲む」に対して「飲める」と「飲んだ」という名前に分けた述語を使って，それぞれの述語の性質にあった推論をその言語で書いたルール文（$A \rightarrow B$ という記述）として追加する．もし，その方法で記述しようとするならば，新たな知識を入力する度に対処する羽目になり，読みの違いによる暗黙知識への量化の扱いにまで，完全に対応できない．

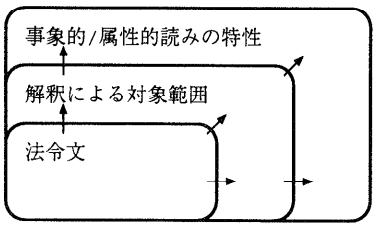
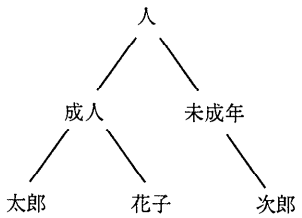


図 7 法令文の拡張
Fig. 7 An extension of rules.



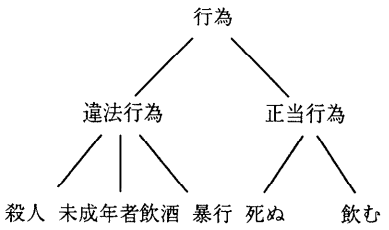


図 8 項と述語の型階層
Fig. 8 The hierarchys of terms and predicates.

3. 型階層論理を用いた法的推論の実現

型階層論理による項や述語の型階層，述語のイベントとプロパティの区別を用いた記述と推論によって，前章で示した問題に対する解法を提案する．

3.1 項と述語の型階層

本稿は，オーダソート論理[5),11),12)]を拡張して，項と述語にそれぞれ別の型表現を導入する．背景知識の概念定義は，オーダソート論理の記法であるソート階層を想定しており，それにより簡潔な概念の定義と効率的な概念の単一化が期待できる．背景知識は，項と述語の 2 つの型階層とし，**図 8** のように木構造で示すことができる．すべての節点は，ノードによって到達可能な節点と型関係を持ち，上へ行くほど上位の概念を示す．このうち，項の型階層がソート階層にあたる．特に，項の型関係を**サブソート関係**といい，述語

の型関係を**述語の部分関係**ということにする．

項の型表現は，LOGIN の ψ-term に基づき，素性構造による属性の記述も許している．たとえば，「20歳，男性，会社員の太郎」は，

太郎 [年齢 → 20, 性別 → 男，職業 → 会社員]

と記述される．'太郎' はソート記号，'年齢' '性別' '職業' は属性ラベルである．S がソート記号の集合であるとき，$\leq_s$ $(\subseteq S \times S)$ をサブソート関係とする．たとえば，サブソート関係 '太郎 $\leq_s$ 人間' を宣言すると，太郎が人間の下位概念を示している．項の記法は，変数の有無によって解釈が分かれ，'x: 太郎' が存在限量子に束縛されて「ある太郎」を意味し，'太郎' が全称的に「すべての太郎」を意味する．型表現には，仕様 3.1 のように型変換が許される．

仕様 3.1（**項の型変換**） T は項の集合，s_1，s_2 がソート，$s_1 \leq_s s_2$ のとき，次のように項の型変換 $f_s : T \to T$ が定義される．

$$f_s(x{:}\, s_1) = x{:}\, s_2$$
$$f_s(s_2) = s_1$$
$$f_s(s_1) = x{:}\, s_1$$

たとえば，$f_s(x{:}$ 太郎$) = x{:}$ 人，$f_s($人$) =$ 太郎 のように変換できる．

述語の型表現は，述語記号そのものを型表現として見なしている．P が述語記号の集合であるとき，$\leq_p$ $(\subseteq P \times P)$ を述語の部分関係とする．述語にも型表現を導入することにより，仕様 3.2 のように，項とは違い上位方向だけであるが述語の型変換が許される．

仕様 3.2（**述語の型変換**） F が原子論理式の集合，述語が p_1, p_2，項が $t_1, ..., t_n$，述語の部分関係が $p_1 \leq_p p_2$ のとき，以下のように，$p_1(t_1, ..., t_n)$ の述語 p_1 に対して型変換 $f_p : F \to F$ が行われる．

$$f_p(p_1(t_1, ..., t_n)) = p_2(t_1, ..., t_n)$$

しかしながら，p_1 と p_2 の引数構成が異なる場合は，述語の型変換だけでは述語の上位導出として不十分である．その解決ために，まず，述語の引数にもその役割を示すラベル $l_1, ..., l_n$ を設け，$p_1(l_1 \Rightarrow t_1, ..., l_n \Rightarrow t_n)$ と記述する．さらに，上位述語を導く際に，引数の操作を行うが，詳しくは，次節のイベントとプロパティの推論において説明する．

3.2 イベントとプロパティ

法令文の事象的/属性的読みに対処するために，型階層論理において述語をイベントとプロパティに区別する方法を導入する．まず，仕様 3.3 のように述語の記法について定義する．

仕様 3.3（**イベントとプロパティ**） 述語はイベントとプロパティとして 2 つの役割を持つ，$p \in P$ のとき，プロパティを示す述語には識別記号 $\sharp$ を付けて $\sharp p$ と示し，イベントには状況を示す記号 $sit{:}{:}$ を付けて $sit{:}{:}p$ と示す．

たとえば，次の 2 つの言明は，前者がイベント型述語により「状況 sit である赤ん坊が泣いている」を示し，後者はプロパティ型述語により「ある赤ん坊が泣く性質を持っている」を示している．

$sit{:}{:}$泣く (主体 $\Rightarrow x{:}$赤ん坊)
$\sharp$泣く (主体 $\Rightarrow x{:}$赤ん坊)

この場合は，項を '$x{:}$赤ん坊' で明記しているので，両方とも「ある赤ん坊」への言及である．もし，一般に「すべての赤ん坊が泣く性質を持つ」を示すならば，'#泣く (主体 ⇒ 赤ん坊)' と記述すればよい．述語の部分関係 '泣く $\leq_p$ 声を出す' が成り立つとき，述語の型変換によって上位述語 '声を出す' が導かれる．イベントとプロパティの区別は，述語 '声を出す' の引数構成が '声を出す (主体 ⇒, 位置 ⇒)' のように述語 '泣く (主体 ⇒)' と異なる場合に，仕様 3.4 のような引数のずれを考慮した述語の上位導出を規則化できる．引数のずれの解消には，2 章で説明した事象的/属性的読みの区別による暗黙対象（明記してない対象）への量化の考え方を利用する．

仕様 3.4（**述語の上位導出**） 述語の上位導出は，(i) 述語の型変換，(ii) 引数の操作（補充/削除）からなる．sit が状況，p_1，p_2，q が述語，s_1，s_2 がソート，x_1，x_2，x_3 が変数，l_1，l_2，l_3 が引数ラベル，$p_1 \leq_p q$，$p_2 \leq_p q$ が述語の部分関係のとき，述語の上位導出は次のように定義される．

〈事象的読みの述語の上位導出〉

$$sit{:}{:}p_1(l_1 \Rightarrow x_1{:}\, s_1)$$
$$\Downarrow \text{引数の補充+型変換}$$
$$sit{:}{:}q(l_1 \Rightarrow x_1{:}\, s_1, l_2 \Rightarrow \underline{x_2{:}\, s_2})$$

$$sit{:}{:}p_2(l_1 \Rightarrow x_1{:}\, s_1, l_2 \Rightarrow x_2{:}\, s_2, l_3 \Rightarrow \underline{x_3{:}\, s_3})$$
$$\Downarrow \text{引数の削除+型変換}$$
$$sit{:}{:}q(l_1 \Rightarrow x_1{:}\, s_1, l_2 \Rightarrow x_2{:}\, s_2)$$

〈属性的読みの述語の上位導出〉

$$\sharp p_1(l_1 \Rightarrow x_1{:}\, s_1)$$
$$\Downarrow \text{引数の補充+型変換}$$
$$\sharp q(l_1 \Rightarrow x_1{:}\, s_1, l_2 \Rightarrow \underline{s_2})$$

$$\sharp p_2(l_1 \Rightarrow x_1{:}\, s_1, l_2 \Rightarrow x_2{:}\, s_2, l_3 \Rightarrow \underline{s_3})$$
$$\Downarrow \text{引数の削除+型変換}$$
$$\sharp q(l_1 \Rightarrow x_1{:}\, s_1, l_2 \Rightarrow x_2{:}\, s_2)$$

上位導出は，述語の型変換による「ならば」の役割

のほかに，上位述語に合わせた引数の追加・削除を行い「ならば」を越えた処理を備えている．その引数操作は，補充・削除される引数内の変数の有無によって量化を区別している．先に述べた述語‘泣く’の上位導出の結果は次のようであり，この解釈は前者が「状況 sit で赤ん坊はある場所において声を出した」，後者は「赤ん坊はすべての場所で声を出す性質を持つ」となる．

sit::声を出す (主体 $\Rightarrow$ x:赤ん坊, 位置 $\Rightarrow$ y:場所)

#声を出す (主体 $\Rightarrow$ x:赤ん坊, 位置 $\Rightarrow$ 場所)

補充される引数の量化の違いは，事象的読みが一回性で限定的であり，属性的読みが広域的であることからきており，自然言語文の解釈に合う結果をもたらしている．

3.3 拡張表現を用いた法的推論

項と述語の型階層，述語のイベントとプロパティの区別による拡張が，法律知識の多様性を記述するのに有効であることを示す．

(1) 具体的対象の記述

事例で示される行為や事象に含まれる対象は，詳細な情報を含んでいる．たとえば，ある違法行為を行った人物「太郎」については，事件によって，年齢，職業，年収，家族，健康状態，上司などの限りない属性を付加する可能性を持っている．その属性の内容次第で，導かれる判決も違ってくる．たとえば，次の項表現が法的に車を運転してよい対象範囲であったとする．

人 [年齢 $\rightarrow$ 18 以上, 免許 $\rightarrow$ あり]

すると，以下の2つの言明は，属性の内容から太郎は自動車を運転する資格を持たないので両方とも違法行為といえる．

sit::運転する (主体 $\Rightarrow$ x:太郎 [年齢 $\rightarrow$ 17],
対象物 $\Rightarrow$ y:車)

sit::運転する (主体 $\Rightarrow$ x:太郎 [免許 $\rightarrow$ なし],
対象物 $\Rightarrow$ y:車)

型階層論理では，素性構造を含む型表現を使って事例の具体的知識と法令文の抽象的知識を記述し，背景知識によってそれぞれを関連付けることができる．

(2) 事象の単一化

事象的読みの記法は，状況 sit の付加によって，属性的読みにはない単一化を可能にする．たとえば，次の2つの事象が成り立つ場合を考える．

sit_1::暴行 (主体 $\Rightarrow$ x:太郎, 対象者 $\Rightarrow$ y:人)

sit_1::暴行 (主体 $\Rightarrow$ x:人, 対象者 $\Rightarrow$ y:花子)

前者の意味は「状況 sit_1 で，太郎がある人を暴行した」であり，後者の意味は「状況 sit_1 で，ある人が花子を暴行した」である．事象的読みでは，2つの事象から次の命題を導くことが可能である．

sit_1::暴行 (主体 $\Rightarrow$ x:太郎, 対象者 $\Rightarrow$ y:花子)

両方とも，同じ状況 sit_1 において起こった同事象であることから以上の命題への単一化が成り立つのである．つまり，1つの状況 sit_1 を付加した上の2つの命題において，同じ引数ラベル‘主体’を持つ項‘x:太郎’と‘x:人’の存在束縛の対象は一意に決まる．

対して，最初の事象の代わりに次の事象が成り立つときを考える．

sit_2::暴行 (主体 $\Rightarrow$ x:次郎, 対象者 $\Rightarrow$ y:人)

しかし，これは状況 sit_2 の事象であり2つ目の事象とは関係ないために，次の命題を導くことはできない．

sit_2::暴行 (主体 $\Rightarrow$ x:次郎, 対象者 $\Rightarrow$ y:花子)

この事象的読み特有の単一化は，事件の同じ状況における多数の事実から，法令文へ適用するためのより具体的な命題を数多く導き出すことができるようになる．

(3) 事象的/属性的読みによる導出

2.2節で説明した事象的/属性的読みを含む法令文を解釈し拡張させることで，柔軟な導出が得られることを示す．法令文を解釈するポイントは，条文の汎用性を保持しつつ，対象範囲を厳密に記述し直すことにある．条文は，当てはまる対象範囲を型表現によって明確にすることで，事例に含まれる表現の多くを背景知識の型階層に従って適用できるようになる．

図 9 は，図 6 の法令文に解釈を与え型階層論理で記述した例である．〈刑法 199 条殺人罪〉の解釈は「人が人に殺意を持ってある行為を行い，死に至らしめたならば殺人である」であったので，述語‘行為’‘死ぬ’‘殺意’を使ってルール化している．そして，‘殺人 (主体 $\Rightarrow$ x:人, 対象者 $\Rightarrow$ y:人)’と‘行為 (主体 $\Rightarrow$ x:人, 対象者 $\Rightarrow$ y:人)’の記述から，殺人は人から人への行為であると対象範囲を定めている．これにより，型表現「人」に属すすべての表現が法令の対象範囲となる．同じように，述語に対しても型表現「行為」によって，特定の行為によらず，人を死に至らしめたときに殺人が成り立つのである．もし，間接的に「毒を盛った」行為でもその行為によって相手が死んだならば，このルールを適用可能である．

ほかに，〈未成年者飲酒禁止法〉のように権利や禁止を示す法令文には，イベントとプロパティの区別が有用である．この条文には，‘飲む’の事象的読みに対する違法性と，‘飲む’の属性的読みによる権利や禁止との両方のルールが要求される．そのため，図 9 では，述語をイベント‘飲む (...)’とプロパティ‘#飲む (...)’とに分けたルールを書いている．**図 10** のよう

〈刑法 199 条殺人罪〉:
殺人 (主体 ⇒ x: 人, 対象者 ⇒ y: 人) ← 行為 (主体 ⇒ x: 人, 対象者 ⇒ y: 人),
死ぬ (主体 ⇒ y: 人), 殺意 (主体 ⇒ x: 人, 対象者 ⇒ y: 人).
有罪 (主体 ⇒ x: 人, 懲役 ⇒ 3 年以上) ← 殺人 (主体 ⇒ x: 人).

〈未成年者飲酒禁止法〉:
未成年者飲酒 (主体 ⇒ x: 未成年) ← 飲む (主体 ⇒ x: 未成年, 対象物 ⇒ y: 酒).
#飲む (主体 ⇒ 成人, 対象物 ⇒ 酒).
#販売 (主体 ⇒ 販売業者, 対象者 ⇒ 成人, 対象物 ⇒ 酒).

図 9 解釈を与えた法令文の記述例
Fig. 9 An example of legal rules with an interpretation.

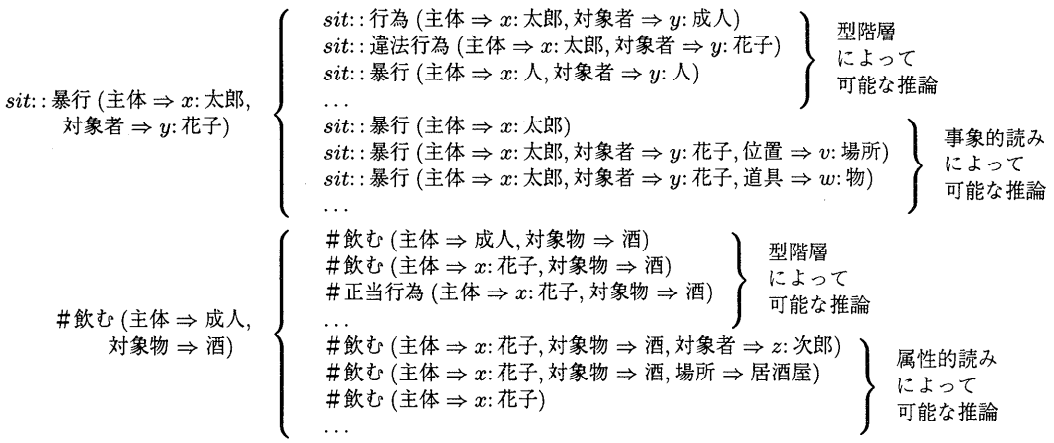


図 10 事象的/属性的読みから導出可能な命題
Fig. 10 The derivable propositions from event and property interpretation.

に, '#飲む (...)' という属性的読みの言及から, 全称の量化に基づいて引数の補充・削除を行い, 導出可能な命題が得られる. この結果は, '#飲む (主体 ⇒ 成人, 対象物 ⇒ 酒)' の属性的読みから, 上位導出の定義に従い '#飲む (主体 ⇒ x: 花子, 対象物 ⇒ 酒, 対象者 ⇒ 人)', '#飲む (主体 ⇒ x: 花子, 対象物 ⇒ 酒, 場所 ⇒ 場所)' が導かれることによる. 意味として,「成人である花子は, いかなる場所で, 誰とでも酒を飲む権利を持つ」を導いている.

事象的読みでは, 量化の区別により全称的な引数ではなく, 図 10 のように存在限量子を表す引数に対して補充・削除した命題を導いている. 上位導出の定義に従い 'sit:: 暴行 (主体 ⇒ x: 太郎, 対象者 ⇒ 花子)' から, 'sit:: 暴行 (主体 ⇒ x: 太郎, 対象者 ⇒ y: 花子, 位置 ⇒ v: 場所)' と 'sit:: 暴行 (主体 ⇒ x: 太郎, 対象者 ⇒ y: 花子, 道具 ⇒ w: 物)' を導くことによる. この意味は,「太郎がある場所で花子に暴行した」と「太郎はある物を使って花子に暴行した」である.

論理型言語による法的推論では事例を法令文に適用するとき, 述語の型階層上で述語間の引数構成が違うのと同様に, 必ずしも事例と法令文で使われている述語間の引数構成が一致しないので法令文はそのまま適用できない場合が多い. 本稿の提案では, 図 10 の導出可能な命題のように自然言語文の解釈に基づいて柔軟な導出が可能なので, 結果として法令文は拡張されており, 適用できる事例が格段に増えると考える.

4. 推論システム

本章では, 前章までの提案に基づいて, 論理プログラム[18]による推論システムを形式化する.

4.1 論理プログラムの設計

まず, 論理プログラムの構文について説明する. S はソート記号の集合, V は変数の集合 $(x, y, z, ...)$, L_s はソート属性ラベルの集合, $Sit(sit_1, sit_2, ...)$ は状況の集合である.

定義 4.1 (**項の定義**) 項の集合 $Term$ は以下のように定義される.

(1) $s \in S, x \in V$ であるならば, s と x: s は項である.

(2) $s \in S$, $t_1, ..., t_n \in Term$, $x \in V$, $l_1, ..., l_n \in L_s$ であるならば, $s[l_1 \to t_1, ..., l_n \to t_n]$ と x: $s[l_1 \to t_1, ..., l_n \to t_n]$ は項である.

$P(p_1, p_2, ...)$ は述語記号の集合, L_p は述語引数

ラベルの集合，♯ はプロパティ型述語の識別記号であり，♯ のない述語はイベント型である．$p \in P$, $t_1,...,t_n \in Term$, $l_1,...,l_n \in L_p$, $sit \in Sit$ であるならば，$\sharp p(l_1 \Rightarrow t_1,...,l_n \Rightarrow t_n)$，および $sit::p(l_1 \Rightarrow t_1,...,l_n \Rightarrow t_n)$ はアトムである．

次に，節形式によってプログラムとゴール節を定義する．一階述語論理の言語において，節は $\neg B_1 \vee ... \vee \neg B_n \vee A_1 \vee ... \vee A_m$（$B_1,...B_n, A_1,...,A_m$ はアトムとする）の形をした論理式である．論理プログラミングにおいて，節は論理的に同値である論理式 $B_1,...,B_n \rightarrow A_1,...,A_m$ で表現される．

定義 4.2（**プログラム**） $B_1,...B_n, H$ をアトムとすると，プログラム節 C_i は，次のような形式で表される．

$$C_i : B_1,...,B_n \rightarrow H$$

$B_1,...,B_n$ はプログラム節のボディ部といい，H はヘッド部という．特に，ボディ部が空のプログラム節を単位節という．プログラム $\mathcal{P}$ はプログラム節の有限集合である．

$$\mathcal{P} = \{C_1,...,C_n\}$$

定義 4.3（**ゴール節**） ゴール節 $\mathcal{G}$ は，ヘッド部を持たない節 $B_1,...,B_n \rightarrow$ である．

本稿で示す言語には，サブソート関係，述語の部分関係，述語引数ラベルのスコープの宣言を推論のための初期データとして導入している．

定義 4.4（**型関係の宣言**） $s_1, s_2 \in S$, $p_1, p_2 \in P$，サブソート関係 $\leq_s$，述語の部分関係 $\leq_p$ のとき，型関係の宣言 HS_i, HP_i は次のように表される．

(1) HS_i：$s_1 \leq_s s_2$

(2) HP_i：$p_1 \leq_p p_2$

定義 4.5（**述語引数ラベルのスコープ**） $l \in L_p$, $s \in S$ のとき，述語引数ラベルのスコープの宣言 LS_i は次のように表される．

$$LS_i : l \Rightarrow s$$

定義 4.6（**初期データ**） 初期データ $\mathcal{I}$ は，型関係の宣言 HS_i, HP_j，述語引数ラベルのスコープの宣言 LS_k の集合である．

$$\mathcal{I} = \{HS_1,...,HS_m, HP_1,...,HP_n\} \cup \{LS_1,...,LS_h\}$$

4.2 推論方法

プログラムと初期データを使った単一化と導出について説明する．

4.2.1 単一化

プログラム $\mathcal{P}$ に含まれる任意の2つのアトムを単一化するために，項の置換を定義する．置換は，変数名，ソート，量化および素性構造に関して行われる．変数名以外は項の量化に影響するので，項の置換は，節におけるアトムの正負に依存する．先に述べたように，プログラム節 $B_1,...,B_n \rightarrow H$ は，$\neg B_1 \vee ... \vee \neg B_n \vee H$ と同値である．プログラム節のボディ部のアトム B_i は否定を意味し，ヘッド部のアトム H は肯定を意味する．したがって，項の置換では，どちらに含まれるアトムであるか場合分けする．

定義 4.7（**項の置換**） 置換 θ は項から項への写像であり，項 t から項 r への置換は $\theta = \{r/t\}$ と書き表す．A がアトム，$t, r \in Term$ であるとき，$A\{r/t\}$ は，A に現れる項への置換を表している．B はボディ部のアトムであり，H はヘッド部のアトムであり，A はその一方であるとき，各条件において置換は以下のように定義される．

(1) $x_1, x_2 \in V$, $s \in S$ のとき，$A\{x_2{:}s/x_1{:}s\}$

(2) $s_1, s_2 \in S$, $s_1 \leq_s s_2$ のとき，$H\{s_1/s_2\}$，または $B\{x{:}s_1/x{:}s_2\}$

(3) $y \in V$, $s \in S$ のとき，$H\{y{:}s/s\}$，または $B\{s/y{:}s\}$

素性構造は，空もしくは有限個のラベルと属性値の列 $(l_1 \rightarrow t_1,...,l_m \rightarrow t_m (m > 0))$ であり，$ATTS$ で表す．

(4) $l_i \in L_s$, $t_i \in Term, s \in S$ のとき，$H\{s[ATTS + (l_i \rightarrow t_i)]/s[ATTS]\}$，または $B\{x{:}s[ATTS + (l_i \rightarrow t_i)]/x{:}s[ATTS]\}$

定義 4.7 の役割を説明する．(1) は x_1 から x_2 への変数名の置換である．(2) は，$s_1 \leq_s s_2$ により，s_2 からそのサブソート s_1 への置換である．(3) は，アトム H では，全称表現 s から存在表現 $x{:}s$ へ置換し，アトム B では，その反対方向へ置換する．(4) は，素性構造☆の置換であり，属性 $l_i \rightarrow t_i$ を追加してより具体的な項へ置換され，$l_i \rightarrow t_i$ を削除してより抽象的な項へ置換される．

定義 4.8（**同事象における項の置換**） $sit \in Sit$, $t_1,...,t_n, r_1,...,r_n \in Term$, $p \in P$, $l_1,...,l_n \in L_p$ であり，ヘッド部のアトム $sit::p(l_1 \rightarrow t_1,...,l_n \rightarrow t_n)$ とボディ部のアトム $sit::p(l_1 \rightarrow r_1,...,l_n \rightarrow r_n)$ が，同じ状況，述語記号と引数構成を持つとき，それぞれの項 t_k, r_k に対して以下の置換が成り立つ．

(1) $s_1, s_2 \in S$, $x \in V$, $s_1 \leq_s s_2$, t_k, r_k がそれぞれ $x{:}s_1$, $x{:}s_2$ のとき，$\{x{:}s_1/x{:}s_2\}$

(2) $s_1, s_2 \in S$, $s_1 \leq_s s_2$, $x \in V$, t_k, r_k がそれぞれ s_1, s_2 のとき，$\{s_2/s_1\}$

(3) $s_1, s_2 \in S$, $s_1 \leq_s s_2$, t_k, r_k がそれぞれ $x{:}s_1, s_2$

☆ なお，この素性構造は，LOGIN の ψ-term に基づいた属性表記で項の一部分である．同じくラベルを持つ述語引数ラベルと引数値とは，別の表記である．

のとき，$\{s_2/x{:}\,s_1\}$

$ATOM_p$ は，$\mathcal{P}\cup\{\mathcal{G}\}$ 内の述語 $p, \sharp p$ からなるすべてのアトムの集合とする．

定義 4.9（述語の引数構成） プログラム $\mathcal{P}$ とゴール節 $\mathcal{G}$ が与えられたとき，述語 p の引数構成 $arg(p)$ は次のような引数ラベルの集合である．

$$arg(p) = \{l_i \in L_p | \forall[p(l_1 \to t_1, ..., l_n \to t_n)] \in ATOM_p\}.$$

ASが有限のラベルと引数（$l_1 \Rightarrow t_1, ..., l_m \Rightarrow t_m\,(m > 0)$）のとき，$AS_p$ は AS から $arg(p)$ に含まれないラベルとその引数を削除したものとする．

定義 4.10（述語の上位導出） プログラム $\mathcal{P}$，ゴール節 $\mathcal{G}$ と初期データ $\mathcal{I}$ が与えられたとする．$l_1, ..., l_m \in L_p$，$p_1, p_2 \in P$，$s_i \in S$，$x_i \in V$，$(LS_i : l_i \Rightarrow s_i) \in \mathcal{I}$，$t_1, ..., t_m \in Term$，$AS = \{l_1 \Rightarrow t_1, ..., l_m \Rightarrow t_m\}\ (m > 0)$，$L_{AS} = \{l_1, ..., l_n\}$ であるとき，述語の上位導出 σ は以下のように定義される．

(1) $ALS = arg(p_1) - L_{AS} \neq \phi$ のとき，

$$p_1(AS) \overset{\sigma}{\Longrightarrow} p_1(AS + \{l_i \Rightarrow x_i{:}\,s_i | l_i \in ALS\})$$
$$\sharp p_1(AS) \overset{\sigma}{\Longrightarrow} \sharp p_1(AS + \{l_i \Rightarrow s_i | l_i \in ALS\})$$

(2) $p_1 \leq_p p_2$，$arg(p) - L_{AS} = \phi$，$ALS = arg(p_2) - L_{AS}$ のとき，

$$p_1(AS) \overset{\sigma}{\Longrightarrow} p_2(AS_{p_2} + \{l_i \Rightarrow x_i{:}\,s_i | l_i \in ALS\})$$
$$\sharp p_1(AS) \overset{\sigma}{\Longrightarrow} \sharp p_2(AS_{p_2} + \{l_i \Rightarrow s_i | l_i \in ALS\})$$

定義 4.11（単一化） A_1，A_2 がアトムであるとき，単一化は，以下の等式が成り立つような述語の上位導出 σ と置換 θ の適用である．

$$(A_1\sigma)\theta = (A_2\sigma)\theta$$

4.2.2 導　　出

導出については，PROLOG で行われている通常の方法[18]を用いる．ただし，導出原理において，単一化は定義 4.11 に従った方法を使用する．

定義 4.12（導出原理） $B_1, ..., B_n$ と $D_1, ..., D_m$ が空または有限のアトムの列，H が空またはアトム，A_1，A_2 がアトム，θ が項の置換，σ が述語の上位導出のとき，次のように上式の 2 つの節から下式の節を導く操作を導出原理という．

$$\frac{B_1, ..., B_n \to A_1 \quad A_2, D_1, ..., D_m \to H}{(B_1, ..., B_n, D_1, ..., D_m \to H)\theta}$$

（ただし，$(A_1\sigma)\theta = (A_2\sigma)\theta$ が成り立つ単一化が存在するときに限る）．

定義 4.13（導出） 導出は，初期データ $\mathcal{I}$，プログラム $\mathcal{P}$ とゴール節 $\mathcal{G}$ ($\mathcal{I}\cup\mathcal{P}\cup\{\mathcal{G}\}$) から，導出原理を適用して，次ゴール $\mathcal{G}'$ を求めて空節が導かれるまで繰り返す．

4.3 実 行 結 果

前節の形式化に基づいて SICStus Prolog^{TM} ver.3.0 上で論理プログラム言語を開発した．2 章の法律知識の記述例を入力し，実行した結果を示す．そのとき，自然言語で書かれた事例や法令文を，本言語の構文に従って人手により入力する．

まず，プログラムデータについて説明する．プログラムデータは，述語引数ラベルの宣言，サブソート関係，述語の部分関係およびプログラム節から構成される．ラベルの定義は，ラベル名とソート名によって宣言される．以下の `agt=person.` は，ラベル `agt` の引数スコープとして `person` を示している．

```
% 述語引数ラベルの宣言
agt=person.          coagt=person.
obj=thing.           tool=thing.
place=space.
```

サブソート関係は，2 つのソート名と記号 `<` によって宣言される．左側が下位ソート，右側が上位ソートを示すので，`taro<adult` は `taro` の上位概念として `adult` を定義している．

```
% サブソート関係
taro<adult.          jiro<minor.
hanako<adult.        adult<person.
minor<person.        beer<alcoholic.
sake<alcoholic.      alcoholic<thing.
bat<thing.           bar<space.
```

述語の部分関係は，2 つの述語名と記号 `<-` によって宣言される．左側が下位述語，右側が上位述語を示すので，`hit<-illegal_act` は，`hit` の上位述語として `illegal_act` を定義している．

```
% 述語の部分関係
die<-legal_act.    drink<-legal_act.
hit<-illegal_act.
homicide<-illegal_act.
minor_drinking<-illegal_act.
legal_act<-act.    illegal_act<-act.
```

次の単位節 `sit1::hit(...)` は，状況 `sit1` と述語 `hit(...)` によって，事象を表している．

```
% 事例の記述
% 事象，行為
sit1::hit(agt=X:taro[age=20],
          coagt=Y:hanako,tool=Z:bat,
                        place=V:home).
sit1::die(agt=Y:hanako).
```

```
sit1::intent_to_murder(agt=X:taro,
                       coagt=Y:hanako).
sit2::drink(agt=X:jiro,obj=Y:sake).
```

法令文は，次のプログラム節で宣言される．包含関係 <- によってヘッド部とボディ部を結んでいる．未成年者飲酒禁止法の最終行は，# 記号で示したプロパティ型述語によって「成人は酒類を飲む権利を持つ」を表している．

```
%% 法令文
% 刑法 199 条殺人罪
S::homicide(agt=X:person,
            coagt=Y:person) <-
S::act(agt=X:person,coagt=Y:person),
S::die(agt=Y:person),
S::intent_to_murder(agt=X:person,
                    coagt=Y:person).
    % 事象的読み

%未成年者飲酒禁止法
S::minor_drinking(agt=X:minor) <-
S::drink(agt=X:minor,
         obj=Y:alcoholic).
    % 事象的読み
#drink(agt=adult,obj=alcoholic).
    % 属性的読み
```

これらのプログラムデータによる実行結果を次に示す．最初に,「20 歳の太郎が殺人をしたか?」の質問を行うと yes が返ってくる．次に,「太郎は違法行為をしたか?」の質問に対しても yes が返ってくる．

```
?-sit1::homicide(agt=X:taro[age=20]).
yes
?-sit1::illegal_act(agt=X:taro).
yes
```

プログラムデータ上では，属性的読みによって法令を解釈したルール「成人は酒類を飲む権利を持つ」が宣言されている．それに対して，次のように「花子は酒類を飲む権利を持つか?」の質問を行うと yes が返ってくる．さらに，より具体的な 2 つ目の質問「花子は酒場でビールを飲む権利を持つか?」に対しても yes を返す．ルールでは，飲む場所については言明されていないが，暗黙対象への量化によって引数が補充されてこのような結果を推論できる．

```
?-#drink(agt=X:hanako,
         obj=Y:alcoholic).
yes
?-#drink(agt=X:hanako,
         obj=Y:beer,place=Z:bar).
yes
?-sit2::drink(agt=X:hanako,
              obj=Y:alcoholic).
no
```

2 つ目の質問についての推論過程を説明する．まず，次のゴール節から推論をスタートする．

```
?-#drink(agt=X:hanako,obj=Y:beer,
                      place=Z:bar).
```

ゴール節に導出原理を適用するために，プログラムからヘッド部にゴール節と同じ述語または上位述語を持つプログラム節をとってくる．この場合は，以下のような同じ述語を持つ単位節が見つかる．

```
#drink(agt=adult,obj=alcoholic).
```

次に，導出原理において 2 つのアトムを単一化する．最初，以下のように述語の上位導出 σ を適用して，述語の引数を追加する．

σ = { #drink(agt=adult,obj=alcoholic,place=space)/#drink(agt=adult,obj=alcoholic) }

さらに，項の置換 θ を行う．

θ = { X:hanako/adult, Y:beer/alcoholic, Z:bar/space }

すると，単位節は次のように書き換えられる．

```
#drink(agt=X:hanako,obj=Y:beer,
                    place=Z:bar).
```

この単位節とゴール節に対する導出原理の適用によって空節が導かれるので，反駁が成功し，質問の結果 yes を返す．

ほかには，次郎の飲酒に対して,「次郎は違法行為をしたか?」の質問は yes となる．同様に「次郎は酒類を飲む権利を持つか?」の質問は no となる．

```
?-sit2::illegal_act(agt=X:jiro).
yes
?-#drink(agt=X:jiro,
         obj=Y:alcoholic).
no
```

このようにして，項と述語の型階層によって，事例と法令文からより抽象的もしくは具体的な別の表現による質問の答を導くことができる．さらに，述語のイベント/プロパティの区別が人間の直観に合った推論を実現しているとともに，命題の解釈を事象的読みまたは属性的読みのどちらかであるか明確にしている．

我々が実現した論理プログラミング言語は，オーダソート論理に基づいていることから，ソートの単一化による推論の効率化が期待できる．しかしながら，従来とは違ったイベントとプロパティの区別，および述

語の型表現によって単一化の処理が複雑になっている．したがって，実装上の課題として従来の推論の効率を損なわないような単一化アルゴリズムが必要である．

5. おわりに

本稿では，事象と属性が持つ限定的な意味と広域的な意味を用いて，法令文が意図する暗黙の読みを導き出す手法を提案した．通常では区別できない言明の事象的読みと属性的読みに対して，本稿では，述語のイベント/プロパティの識別，型表現，および型への量化の概念を導入して，読みの特性に基づいた型変換と引数の操作による単一化を備えた論理プログラミングを形式化した．さらに，推論システムとして，それらの拡張を取り入れた論理型言語を実装し，法律知識を入力して推論できることを示した．

本研究の法的推論に対する成果は次の2点である．

第1に，我々が提案した推論システムは，2章の図5で示したように，元の法令文に基づき拡充された法令文の解釈による推論を，3章の図10のように推論の過程で具体化した．法令文は，多くの事例を適用できるように，その解釈と汎用性を持たなければルールとしての役割に耐えられない．本システムでは，1つの命題からルールに対して図10のような多くの導出結果を実現している．つまり，我々が導入した型表現とイベント/プロパティの区別によって，法律の様々な形態の知識に対応し，知識の構造と特性による柔軟な推論が可能になった．特に，法令文の2つの読みはそれぞれの読みが持つ特性から，その言明が暗黙に保持している情報を導き出すことができる．そのうえ，型階層の推論との組合せにより，法令文の汎用性を適切に表すことができた．以上の成果は，本研究の原点である new-HELIC-II の知識表現では不十分であった，法律知識を自然言語としてとらえたとき，頻繁に発生する事象的読みと属性的読みによる推論の違いへの対処である．

第2に，法的推論システムを構築するときに直面するデータ（事例，法令文や背景知識）の入力の難しさに対して，型表現とイベント/プロパティの区別により本来の記述を極力崩さないような知識表現を実現した．拡張された記述力を持つ知識表現によって，より正確に，より曖昧性を少なくしたデータ入力ができる．法令文などを一階述語論理の記法で苦心して書き換えたような記述は自然な表現とはいえない．さらに，本システムは法律知識に適応する表現力を持ちながら，固有のアプリケーションとして実現されていないので，特定の法律知識の形式によらないで利用できる利点がある．

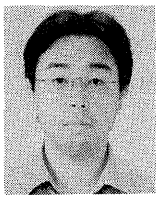
本稿は，知識表現の観点から法律知識の特徴を記述できる論理型言語の拡張を行ったが，否定の拡張についてはふれていない．法令文などの記述には否定的表現は多く含まれる．否定的表現は，文否定と動詞的否定などの作用領域によって区別されるものだけでなく，記述的には肯定であるが，他の表現と否定関係にある意味的な否定も存在する．否定の扱いは，本稿で述べた法令文の事象的/属性的読みや，それに基づいた型階層の推論にも影響を及ぼすと考えられる．したがって，それらを論理型言語で記述する際には，否定の多様性も考慮した推論も必要となってくるだろう．そうした拡張を法的推論として実現するために，本システムへの多様な否定表現の導入を現在検討中である．

参考文献

1) Ait-Kaci, H. and Nasr, R.: LOGIN: A Logic Programming Language with Built-In Inheritance, *Journal of Logic Programming*, pp.185–215 (1986).
2) Ait-Kaci, H. and Podelski, A.: Towards a Meaning of LIFE, *Journal of Logic Programming*, pp.195–234 (1993).
3) Allen, J.F.: Towards a General Theory of Action and Time, *Artificial Intelligence*, Vol.23, pp.123–154 (1984).
4) Ashley, K.: Reasoning with cases and hypotheticals in HYPO, *Int. J. Man-Machine Studies*, pp.753–796 (1991).
5) Beiercle, C., Hedtsuck, U., Pletat, U., Schmit, P. and Siekmann, J.: An order-sorted logic for knowledge representation systems, *Artificial Intelligence*, Vol.55, pp.149–191 (1992).
6) Branting, L.: Building explanation from rules rules and structured cases, *Int. J. Man-Machine Studies*, pp.797–837 (1991).
7) McDermott, D.V.: A temporal logic for reasoning about processes and plans, *Cognitive Science*, Vol.6, pp.101–155 (1982).
8) Nitta, K., Ohtake, Y., Maeda, S., Ono, M., Osaki, H. and Sakane, K.: HELIC-II: Legal Reasoning System on the Parallel Inference Machine, *New Gener. comput.*, Vol.11, No.34, pp.423–448 (1993).
9) Nitta, K., Tojo, S., et al.: Knowledge Representation of New Helic II, *Workshop on Legal Application of Logic Programming,ICLP '94* (1994).
10) Shoham, Y.: *Reasoning about Change*, The MIT Press (1988).
11) Walter, C.: A Mechanical Solution of Schu-

ber's Steamroller by Many-Sorted Resolution, *Artificial Intelligence*, Vol.26, No.2, pp.217–224 (1985).
12) Walter, C.: Many-Sorted Unification, *Journal of the Association for Computing Machinery*, Vol.35, p.1 (1988).
13) Yasukawa, H., Tsuda, H. and Yokota, K.: Objects, Properies, and Modules in *Quixote*, *Proc. FGCS'92*, pp.257–268 (1992).
14) Yokota, K.: Quixote: A Constraint Based Approach to a Deductive Object-Oriented Database (1994).
15) 兼岩 憲，東条 敏：述語の型階層と二面性を導入した状況推論システム，電子情報通信学会技術研究報告，pp.47–54 (1997).
16) 中野目善則：刑法の解説，一橋出版 (1997).
17) 東条 敏，Wong, S.，新田克巳，横田一正：状況理論による法的推論の形式化，情報処理学会論文誌，Vol.36, No.1, pp.51–59 (1995).
18) 有川節夫，原口 誠：述語論理と論理プログラミング，オーム社 (1988).

(平成10年12月2日受付)
(平成11年 5 月7日採録)



兼岩 憲（学生会員）

1970年生．1998年北陸先端科学技術大学院大学情報科学研究科修士課程修了．現在，同大学院情報科学研究科博士後期課程在学中．論理プログラミングに関する研究に従事．人工知能学会，ソフトウェア科学会，電子情報通信学会各会員．

東条 敏（正会員）

1981年東京大学工学部計数工学科卒業．1983年東京大学大学院工学系研究科修了．同年三菱総合研究所入社．1986～1988年，米国カーネギーメロン大学機械翻訳センター客員研究員．1995年4月より北陸先端科学技術大学院大学情報科学研究科助教授．1997～1998年ドイツ・シュトゥットガルト大学客員研究員．博士（工学）．自然言語の形式意味論，オーダーソート論理，法的推論，マルチエージェントの研究に従事，その他人工知能一般に興味を持つ．人工知能学会，ソフトウェア科学会，言語処理学会，認知科学会，ACL各会員．