否定的意味が内在するソートを含んだソート階層の論理

兼岩 憲† 東条 敏††

ソート階層を持つ論理において否定的な語彙を使ってソート名を記述したとき，従来の推論方法ではソートの否定情報を推論に反映させるのに十分とは言えない．本研究では，否定を意味するソートがソート階層でどのように位置付けられるかを分析し，その情報を知識ベースに反映できるような枠組みを提案する．否定的要素を含むソートは，自然言語では語彙項目の否定と呼ばれ，(i) 接辞による否定と，(ii) 他の語彙に対する反意を内在する語彙のいずれかに対応し，それらは否定辞（not，no）とは異なった特性を持つ．そのため，否定が内在するソートを語彙項目の否定から考察することで，(i) に対して強い否定の概念に基づいた否定演算子を導入し，語彙そのものが統語論的に否定の要素すら持たない (ii) を，ソート階層で排他性を持つソートとしてとらえる．そのうえでソートによる否定の推論を実現するために，拡張したソート階層に関する導出処理を提供する．そのソート階層では，肯定表現との排他性および全域性（または部分性）を宣言することによって，古典論理の否定や語彙項目の否定の性質を定義することができる．

An Order-sorted Logic with Implicitly Negative Sorts

Ken Kaneiwa† and Satoshi Tojo††

Ordinary order-sorted logics cannot lead to adequate inference for handling the negative meaning of sorts when negative vocabularies are used to denote sort names. In this paper, we propose an inference system in order-sorted logic which can deal with the negative properties of sorts implicitly included in a sort hierarchy. These negations, called lexical negations in linguistics, are classified as (i) negative affix or (ii) lexicon with negative sense, distinguished from the negative particle 'not'. For these, we consider that (i) is regarded as an operator based on strong negation and (ii) is regarded as an exclusive predicate of the antonym in a sort hierarchy. To infer from the conceivable negations as sorts: classical negation, strong negation and antonym, we present clausal resolution rules regarding the extended sort hierarchy in which we can define the properties of these negations by declaring the exclusivity and the totality (or partiality) of affirmation and its antonym.

1. はじめに

オーダーソート論理の特徴は，オブジェクトの集合を表すソートと，ソート間の半順序関係からなるソート階層を導入していることである．ソート階層は，構造的な知識を表現できる利点から人工知能の分野において知識表現の目的で広く応用されている．そのような実用的な利用を考えて，ソート（タイプ）付き論理プログランミング[9] や複雑なソート概念を表現できるように素性構造[4] を導入した言語が研究されてきた．実際に LOGIN[1]，Quixote[22),23]，F-logic[10]，New HELIC-II[12] は実装された言語として，洗練した階層表現を備えている．さらにソート論理は，Walther[19),20] によって導出原理に基づいた完全な体系が提案されているように，その形式的な基盤も保障されている．そのほかにも，ソート項の宣言（term declarations）[15] を導入したソート論理の体系などがある．

ソート階層による知識表現では，概念間の抽象度をサブソート関係で宣言できる．しかし，さまざまな知識を適切に表現して推論するためには，より複雑なソート関係の宣言を必要とする場合がある．特に自然言語の語彙を借りてきてそのままソート名に用いようとすると，古典論理の否定とは別に，そのソート名自体に他のソートに対する否定的要素が内在する可能性がある．たとえば自然言語における語彙項目の否定[14]（(i) 接辞による否定をともなった語，および (ii) 他の語彙に対する反意を内在する語）を考えてみると，そ

† 国立情報学研究所情報学基礎研究系
Foundations of Informatics Research Division, National Institute of Informatics
†† 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

れらがそのままソート名として用いられることによって，ソート階層において他のソートと不適切な関係を導く可能性がある．しかし，従来のオーダーソート論理では，ソート階層上で否定的要素を含むソートを特徴付ける宣言を持たないうえ，ソート代入による推論だけでは，その否定的要素を推論に反映できないのが現状である．

本研究の目的は，オーダーソート論理において否定的なソートを明確化する記法とその推論方法を導入することである．Beierle らの推論システム[3]では，ソートをそのまま単項述語として論理式に出現させて，ソート代入とは別にその単項述語に関する推論規則を導入し，ソート階層と知識ベースとの強い結合を実現している．本研究では，同様に否定的要素を含むソートを特徴付ける宣言をして，同時に論理式に反映する推論を取り入れる．具体的には，あるソートが否定的要素を含む場合，それと意味が対立するソートとの関係を明らかにし，さらに矛盾の定義をソート表現にまで広げた論理を設計する．そのために，否定的なソートを自然言語の語彙項目の否定ととらえることで，(i) 否定的接辞には，部分性を持つ強い否定に基づいた否定演算子を導入する．また，(ii) 反意語は統語論的に否定と判断できないので，ソート階層の中で互いに対立するソートとして排他性を明記する．その結果，古典論理の否定，強い否定および互いに対立するソートが，肯定表現との排他性および全域性（または部分性）によってソート階層に組み込まれる．そのうえでソートによる否定の推論を実現するために，拡張したソート階層に関する導出処理を提供する．

本稿の構成は，以下のとおりである．2 章では，オーダーソート論理の説明を行い，その後でソート階層に含まれる否定的要素とその推論について述べる．3 章では，ソート階層上で否定的要素を含むソートを表現する方法を提案する．4 章では，3 章で述べた否定的なソート表現を持つ論理の構文と意味論を定義し，推論機構の無矛盾性を示す．さらに，推論機構を使って反駁例を示す．最後に 5 章では，本研究の成果と考察について述べる．

2. オーダーソート論理と否定

2.1 ソート階層

ソート階層は，ソート記号の集合 $S = \{s, s_0, s_1, \ldots\}$ とサブソート関係 $\sqsubseteq$ $(\subseteq S \times S)$ により構成される．サブソート関係の要素 $(s_i, s_j) \in \sqsubseteq$ は，$s_i \sqsubseteq s_j$ によって表され，これをサブソート宣言という．たとえば，次のサブソート宣言の集合，

$$\{man \sqsubseteq person,\ woman \sqsubseteq person,$$
$$car \sqsubseteq vehicle,\ ship \sqsubseteq vehicle\}$$

によって，**図 1** のようなソート階層が構築される．各変数はソートによって限定された対象領域を持ち，ソート s の変数は，次のように表記される．

$$x\colon s$$

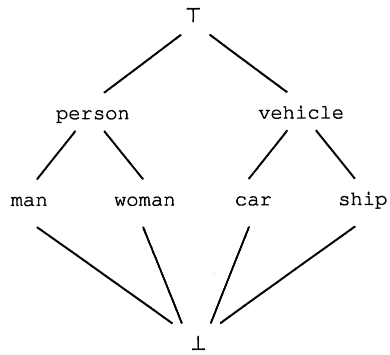

図 1 ソート階層
Fig. 1 A sort hierarchy.

ソートにより限定された述語の引数（すなわち，定数，変数や関数より構成される表現）をソート項という．任意のソート項 t に対して，関数 $Sort[t]$ は項 t のソートを返す関数とする．L_i をリテラル（原子論理式およびその否定のことをリテラルという）とすると，サブソートによるソート代入は次のような規則である．

$$\frac{L_1(x\colon s) \vee \ldots \vee L_n(x\colon s)}{L_1(t) \vee \ldots \vee L_n(t)}\ (substitution)$$

ただし，$Sort[t] \sqsubseteq s$ とする．

さらに従来のオーダーソート論理を拡張して，ソート階層と知識ベースとのつながりをより強化した論理体系が，Beierle ら[3]によって提案された．その論理では，ソートを項の中に出現させるだけでなく，同等な表現力を持つ単項述語としても利用できる．それにともなって，ソート代入とは別に，サブソート関係による述語の包含関係としての推論を導入している．そこでは，サブソート宣言 $s_1 \sqsubseteq s_2$ は，次の包含関係に対応する．

$$s_1(x) \Rightarrow s_2(x)$$

ここで ‘⇒’ は含意（implication）である．このとき，ソートに対応する単項述語 s_1，s_2 をソート述語という．このような考え方から，ソート階層を知識ベースの推論に反映させるために，次に相当する 2 つの推論規則

$$\frac{\neg s_1(t_1) \vee C_1 \quad s_2(t_2) \vee C_2}{\theta(C_1 \vee C_2)}\ (subsort)$$

(ただし，$s_2 \sqsubseteq s_1$ かつ $\theta(t_1) = \theta(t_2)$)

$$\frac{\neg s(t) \vee C}{\theta(C)} \ (\mathit{sort\ predicate})$$

(ただし，$Sort[\theta(t)] \sqsubseteq s$)

を追加している．ここで C，C_1，C_2 は節形式の論理式，s，s_1，s_2 はソート（もしくはソート述語），θ はソート代入，t，t_1，t_2 は項である．

2.2 自然言語の語彙項目の否定

オーダーソート論理を使った知識表現では，ソート階層に内在的否定と判断できる語彙，つまり**語彙項目の否定**が暗黙のうちに含まれる可能性がある．ただし，推論システムから見るとすべてのソート記号は，単なる文字列にすぎず，それが他のソートと肯定・否定あるいは一般に対立する関係にあることを検知することはできない．しかしながら，推論システムに知識を記述する立場からは，肯定表現 *happy* と語彙項目の否定 *unhappy*☆がともにソート階層に含まれていたならば，互いに反意的であるとして判断すべきであろう．本節ではまず自然言語の語彙項目の否定を観察し，否定的要素が内在するしくみについて考察することとする．

語彙項目の否定は，語句そのものが否定を意味しており，以下の 2 つの種類に分類される[14]．

(i) 否定的意味を有する接辞（in-，un-，non-）による否定：unfix，illogical，incoherent，inactive，impolite，nonselfish，など

(ii) 他の語彙と対立する関係にあるために，その結果否定的意味を内在する反意語：doubt（believe not），deny（approve not），prohibit（permit not），forget（remember not），など

ここで，以上のような自然言語の語彙項目の否定がソート名としてソート階層に含まれた場合，その階層から推論されるべき結果を例示する．

例 1. 接辞による否定

接辞による否定 *unhappy* を含んだソート階層 h_1 を**図 2** に示す．接辞による否定 *unhappy* は，否定表現でありながら *happy* と同じように上位の（抽象的な）語彙 *emotional* の意味を含んでいる．ゆえに，古典論理の否定による $\neg$*happy* から導けない語彙が，次のようにソート階層による上位導出により *unhappy* から推論できる（$\vdash_h$ は，ソート階層 h による左式から右式への推論を示す）．

$\mathit{unhappy}(c) \vdash_{h_1} \mathit{emotional}(c)$

$\neg \mathit{happy}(c) \nvdash_{h_1} \mathit{emotional}(c)$

☆ *un* が付くような語彙は形容詞に多く見られるため，ここでは例としてソート名に *unhappy* のような形容詞を用いている．しかし，*unhappy* がソートとして使われているときには，その意味は「*unhappy* な個体の集まり」となるのが適切である．

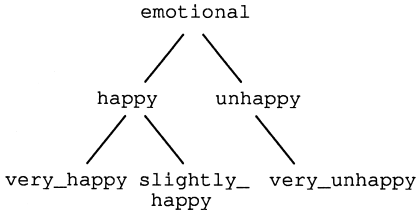

図 2 unhappy を含んだソート階層 h_1
Fig. 2 A sort hierarchy h_1 including unhappy.

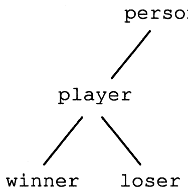

図 3 loser を含んだソート階層 h_2
Fig. 3 A sort hierarchy h_2 including loser.

また，古典論理の否定との関係において，*unhappy* は，より部分的な否定言及であることを反映して，次のように *unhappy* から $\neg$*happy* が推論できる．しかし，その逆は推論できない．

$\mathit{unhappy}(c) \vdash_{h_1} \neg \mathit{happy}(c)$

$\neg \mathit{happy}(c) \nvdash_{h_1} \mathit{unhappy}(c)$

unhappy が部分的で，かつ *emotional* の意味を持つことから，以下の推論も可能である．

$\neg \mathit{emotional}(c) \vdash_{h_1} \neg \mathit{happy}(c) \wedge \neg \mathit{unhappy}(c)$.

例 2. 互いに対立するために内在する否定

図 3 は，*loser* を含んだソート階層 h_2 である．互いに対立するソート *winner* と *loser* は，ともに試合を行った対象（すなわち，*player*）を意味している．しかし，*loser* と同様に *winner* の否定的な対象である $\neg$*winner* は，‘win’ という事態が生じる 1 つのイベント（たとえば 1 つの試合）を越えた否定であり，必ずしも試合をしたことを意味しない．したがって，以下のように *loser*（もちろん，*winner* も）からだけ *player* を導くことができる．

$\mathit{loser}(d) \vdash_{h_2} \mathit{player}(d)$

$\neg \mathit{winner}(d) \nvdash_{h_2} \mathit{player}(d)$

このように，古典論理の否定は言明している単一のイベントを越えたものである[6),16)]．また，$\mathit{loser}(d) \vdash_{h_2} \neg \mathit{winner}(d)$ であるが，その反対の推論は $\neg \mathit{winner}(d) \nvdash_{h_2} \mathit{loset}(d)$ となる．さらに，個体 d がプレイヤでない場合は，次のように *winner* と *loser* がともに成り立たないことが導ける．反対に，個体 d がプレイヤである場合は，*winner* と *loser* は全域的であり，どちらかが成り立つはずである．

$\neg player(d) \vdash_{h_2} \neg winner(d) \wedge \neg loser(d)$
$player(d) \vdash_{h_2} winner(d) \vee loser(d)$

この全域性により，$player(d)$ と $loser(d)$ の否定から $winner(d)$ が導かれ，$winner(d)$ の否定と $loser(d)$ の否定から $player(d)$ の否定が導かれるべきである．

$player(d) \wedge \neg loser(d) \vdash_{h_2} winner(d)$
$\neg winner(d) \wedge \neg loser(d) \vdash_{h_2} \neg player(d)$

ソート階層がこのような内在的否定のソートを含む場合，知識ベースの推論として以上の結果がもたらされることが期待される．しかし，従来のオーダーソート論理では，ソート階層上に否定が内在するソートを特徴的に記述できないため，その性質がソート階層から反映されない．したがって，以上のような上位ソートと古典論理の否定に絡んだ結果をただちに導くことはできない．内在的否定のソートは，それと対立するソートとの排他性を持ち，古典論理の否定とは異なる部分性を備えている．次章から，その特性をソート階層に記述するために，排他性や部分性を宣言できる構造的な記法を提案する．推論システムには，Beierle ら[3] によるソート述語を含んだ推論に基づき，否定的なソートの排他性や全域性に関する推論規則を導入する．

3. 否定的意味が内在するソートを含んだソート階層

本章では，否定的意味が内在するソートを扱うために構造的なソート表現を導入する．その表現には，古典論理の否定や強い否定[17),18)] との関係を定義するために，否定演算子を用いる．それにより，複雑なソート間の関係（排他性や全域性など）を明確にできるようにする．さらに，ソート間の排他性による矛盾を定義し，否定ソートと互いに対立するソートを論理的な推論で扱えるようにする．

3.1 否定とソート間関係

本稿では，接辞による否定がソート名としてソート階層に含まれた場合，これを**否定ソート**と呼ぶことにする．また反意語に関しては，対立するどちらか一方を決めて否定と考えるのは本来適切ではなく，互いに他方の否定と考えるべきであるが，後に述べるようにこの種類のソートは階層内の排反性として表現できるためにどちらが否定であるかに拘泥する必要はない．したがって，このように反意的に働くソートを**互いに対立するソート**と呼ぶ．否定をソート階層に導入する際，たとえばソート名 $winner$ に対して古典論理による論理式の否定 $\neg(winner(c))$ を表すためには，別ソート $\overline{winner}$ を用意し，$\overline{winner}(c)$ としたうえで両者の意味的同値を保証してやる必要がある．後者のように古典論理の否定を実現するソートを**補ソート**と呼ぶ．この意味的同値性は 4.2 節で述べる．

構造的なソートは，以下のように原子ソートを結合子および否定演算子によって構造化したものである．

［構造的なソート］

$angry \sqcap hungry$	：積ソート
$winner \sqcup loser$	：和ソート
$\overline{happy}$	：補ソート（古典論理の否定）
$\sim happy$	：否定ソート（強い否定）
$\top$	：最大ソート
$\perp$	：最小ソート

積ソートと和ソートは，2 つの原子ソートからそれぞれ両方を満たすソートと一方を満たすソートとして構成されたソートである．さらに，補ソートと否定ソートは，古典論理の否定と強い否定に対応するソートである．$\top$ と $\perp$ は，ソート階層における最大要素と最小要素としてのソートである．これらの記法により，構造的なソートを単項述語として使った場合，命題を次のように記述できる．

［命題］

$angry \sqcap hungry(c)\ (=\ angry(c) \wedge hungry(c))$,
$\overline{winner}(d)\ (=\ \neg(winner(d)))$.

続いて，ソート階層を構築するために，ソート間の関係を導入する．サブソート関係 '$\sqsubseteq$' は，ソートの上位下位を示していて，以下はソート s が s' のサブソートであることを宣言している．さらに，ソートの等号関係 '$=$'，ソートの排他関係 '$\|$' とソートの全域関係 '$|_{s_i}$' を次のように表す．

［ソート間関係］

$s \sqsubseteq s'$	：サブソート関係
$s = s'$	：等号関係
$s \parallel s'$	：排他関係
$s\ |_{s_i} s'$	：全域関係

等号関係はソートの同等性を宣言して，排他関係はソート間の不一致を宣言する．さらに，全域関係はソート s，s' の和がソート s_i と同等であることを宣言している．したがって，ソートの等号関係，排他関係と全域関係を次のように定義する．

(i) $s = s' \overset{def}{\Leftrightarrow} s \sqsubseteq s'$ かつ $s' \sqsubseteq s$
(ii) $s \parallel s' \overset{def}{\Leftrightarrow} (s \sqcap s') = \perp$
(iii) $s\ |_{s_i} s' \overset{def}{\Leftrightarrow} (s \sqcup s') = s_i$

本稿では，新しく導入された排他関係と全域関係のソート間関係を**図 4** のように図示する．次のように，2 つのソートのどちらかが成り立つ性質を全域性とい

い，そうでないときを部分性という．さらに，一方のソートだけしか成り立たない性質を排他性という．

［全域性］

$winner \mid_{player} loser$

［部分性］

$(happy \sqcup unhappy) \sqsubseteq emotional$

［排他性］

$happy \parallel unhappy,\ winner \parallel loser$

ソート間関係を使って，ソートの全域性，部分性および排他性を宣言すれば，**図 5** のようなソート階層が構築できる．

3.2　否定が内在するソートの性質

前節のソート表現を用いることで，**表 1** に示す 3 つのタイプの否定（古典論理の否定，強い否定と互いに対立するソート）をソート階層内で扱うことができる．(1) は古典論理の否定に対応する．よって，排中律（$A \vee \neg A$）に基づいて最大ソート $\top$ で全域性かつ排他性の性質を持ち，ソート間の関係による公理で示される．(2) は，強い否定として排他性と部分性が公理として成り立つ．それにより，ソートの等号関係を使い，接辞による否定を $\sim happy$ により定義できる（たとえば，$\sim happy = unhappy$）．互いに対立するソートは，(3) のように排他関係によって宣言される．それぞれの否定は，肯定と否定との関係によりソート階層を構築する公理として，あるいは付加的な宣言として論理体系に導入される．ここでいう公理とは，論理体系において無前提に正しいソート間の関係 ($\vdash s \sqsubseteq s'$) であり，宣言は推論する対象に依存して付加される宣言である．したがって，(2) と (3) は否定の性質以外に公理性においても異なる否定である．

3 つの否定に対する肯定と否定との関係は，それぞれの否定がソート階層上でどのように位置付けられるかを決定する．**図 6** と **図 7** は，(a) 否定が内在するソートを含んだソート階層と (b) 補ソートによる階層（(a) の補ソートにより構築された階層）とを矛盾なく組み合わせた階層 (c) を示している．こうして拡張されたソート階層により，古典論理の否定に加えて，否定が内在するソートを含んだ推論が可能になる．

3.3　矛盾概念の拡張

先に述べたような 3 つの否定（古典論理の否定，強い否定と互いに対立するソート）を実際に導入し，知識ベース上で推論を行うにはまだ問題が残っている．それは，単純に 3 種類の否定演算子を扱う場合とは異なり，対立ソートは否定演算子を持たないために，矛盾の判定や否定間で互いに関連しあう推論を行ううえで困難が生じる．

通常の論理体系では，否定演算子により 2 つの論理式

$A,\ \neg A$

がともに推論されるとき，矛盾するという．これは，$\neg A$ が否定演算子によって統語的に A の否定であることを示しているからである．同様に，対立ソートに関しても，互いに排他的な関係にある 2 つのソート s と s'（たとえば $winner$ と $loser$ のようなソート）が与えられ，その単項のソート述語として

$s(x),\ s'(x)$

がともに推論されるとき，矛盾するとしなければ，否定が内在するソートを反映した推論機構が実現できた

図 4　ソート間関係
Fig. 4　The relations between sorts.

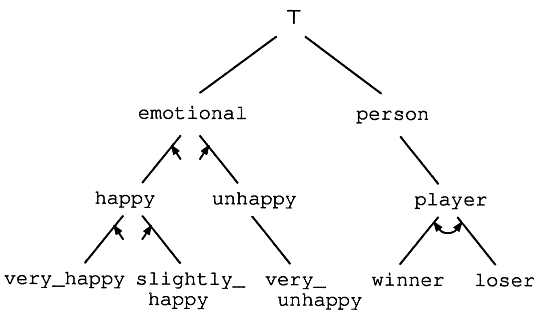

図 5　拡張されたソート階層
Fig. 5　An extended sort hierarchy.

表 1　3 つの否定
Table 1　Three negations.

否定表現	肯定と否定との関係	否定演算子	否定の性質
(1) 補ソート（古典論理の否定）	$\overline{happy} \mid_\top happy$ （公理） $\overline{happy} \parallel happy$	あり	排他性 全域性
(2) 否定ソート（強い否定）	$\sim happy \parallel happy$ （公理） $\sim happy \sqsubseteq \overline{happy}$	あり	排他性 部分性
(3) 互いに対立するソート	$sad \parallel happy$ （宣言）	なし	排他性

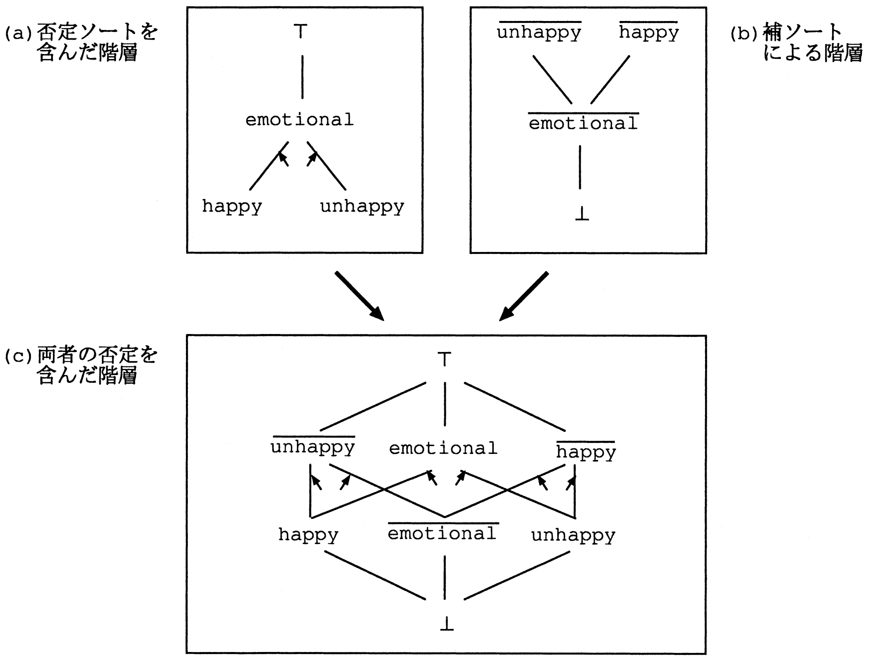

図 6 否定ソートと補ソートによる階層
Fig. 6 A sort hierarchy with negative and complementary sorts.

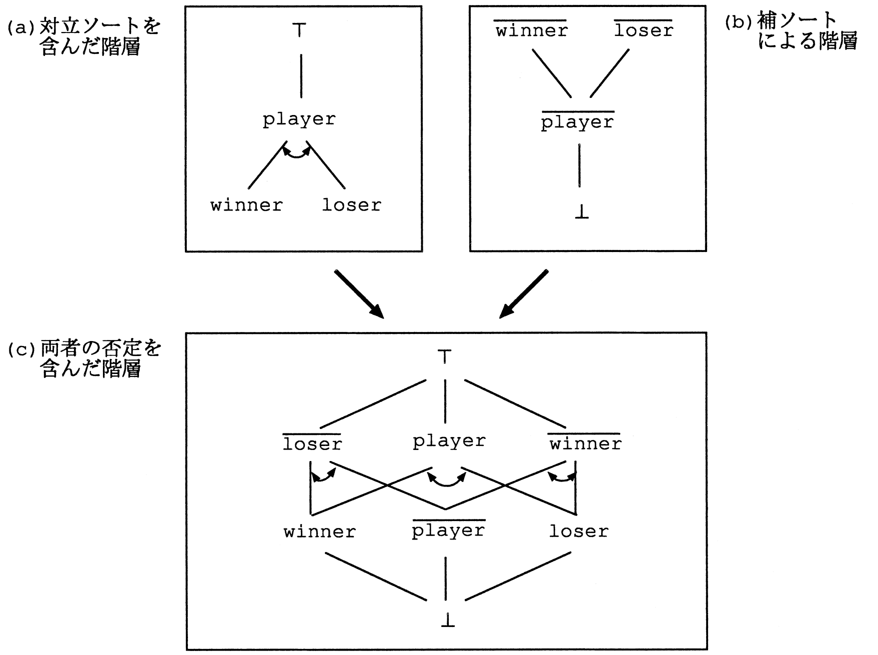

図 7 対立ソートと補ソートによる階層
Fig. 7 A sort hierarchy with opposite and complementary sorts.

とはいえない．これは，ソート s と s' が意味論において排他的であることを主張しているのではなく，否定演算子があらかじめ否定を表す記号として使われているように，形式的にソート s と s' の記号が相反する記号として導入されることを主張しているのである．

しかし，対立ソートは否定演算子を頼りにその肯定

表現の否定とは判断できず，そのため他の否定演算子との関係も認識できない（たとえば，ソート s と s' との矛盾性が判断できなければ，s' と $\overline{s}$，および s と $\overline{s'}$ の関係も判断できないのである）．したがって，ソート階層によって拡張された新たな矛盾の定義を行う．改めて 4.4 節で詳しく定義するが，$s \parallel s'$ となるようなソート s，s' が存在して，$s(t)$ かつ $s'(t)$ が成り立つとき矛盾とする．これにより，対立ソートは否定演算子を持たないにもかかわらず，その矛盾性を統語的に判断できる．

4. 内在的否定のソートを導入した論理

前章で拡張したソート表現に基づき，内在的否定のソートを導入した論理を形式化する．この形式化では，文献 2)，11)，13) の記法を参照する．

4.1 構　　文

$S = \{s, s_0, s_1, \ldots\}$ は原子ソート記号の集合である．

定義 4.1（拡張ソート） 拡張ソートの集合 S^+ は以下を満たす最小の集合である．

(1) $s \in S$ ならば，$s \in S^+$ である．

(2) $s, s' \in S^+$ ならば，$(s \sqcap s')$，$(s \sqcup s')$，$(\overline{s})$，$(\sim s) \in S^+$ である．

F は関数記号 $(f, f_0, f_1, \ldots)$ の集合，C は定数記号 $(c, c_0, c_1, \ldots)$ の集合，P は述語記号 $(p, p_0, p_1, \ldots)$ の集合である．さらに，すべてのソート記号 s に対して，単項のソート述語 $p_s \in P$ が存在すると仮定する．ソート述語 $p_s \in P$ は，特に混乱を招かない限り，そのままソート記号 s で表される（たとえば，$s(t)$）．

定義 4.2（シグネチャ） Dec を非論理記号に対する型宣言とする．そのとき以下の条件を満たすならば，$\Sigma = (S^+, F, P, Dec)$ をシグネチャという．

(1) $(S^+, \sqsubseteq)$ は，$\top$ と $\bot$ を含む拡張ソートの半順序集合である．

(2) $c \in C$ ならば，$c{:} \to s \in Dec$ である．

(3) $f \in F$ のアリティが n ならば，$f{:} s_1 \times \ldots \times s_n \to s \in Dec$ である．

(4) $p \in P$ のアリティが n ならば，$p{:} s_1 \times \ldots \times s_n \in Dec$ である．特に，$p_s \in P$ がソート述語のとき，$p_s{:} \top \in Dec$ である．

シグネチャ Σ の言語 L は，次のアルファベットを含む．V_s はソート s の変数 $(x{:}s, x_0{:}s, x_1{:}s, \ldots)$ の集合であり，V はすべてのソートに対する変数の集合 $\bigcup_{s \in S^+} V_s$ である．$\neg$，$\vee$，$\wedge$，$\Rightarrow$ は論理結合子，$\forall$，$\exists$ は量化記号であり，‘(’，‘)’，‘,’ は，補助記号である．

定義 4.3 言語 L の項は，以下のように帰納的に定義される．

(1) 変数 $x{:}s$ はソート s の項である．

(2) 定数 $c \in C$ かつ $c{:} \to s \in Dec$ のとき，c はソート s の項である．

(3) $t_1, \ldots, t_n$ がソート $s_1, \ldots, s_n$ の項であり，$f \in F$ かつ $f{:} s_1 \times \ldots \times s_n \to s \in Dec$ のとき，$f(t_1, \ldots, t_n)$ はソート s の項である．

定義 4.4 言語 L の論理式は，以下のように帰納的に定義される．

(1) $t_1, \ldots, t_n$ がソート $s_1, \ldots, s_n$ の項であり，$p \in P$ かつ $p{:} s_1 \times \ldots \times s_n \in Dec$ のとき，$p(t_1, \ldots, t_n)$ は（原子）論理式である．

(2) A，B が論理式ならば，$(\neg A)$，$(A \wedge B)$，$(A \vee B)$，$(A \Rightarrow B)$，$(\forall x{:}sA)$，$(\exists x{:}sA)$ は論理式である．

論理式の集合を $\mathcal{F}$ で表す．拡張ソート s，s' に対して，$s \sqsubseteq s'$ をサブソート宣言といい，s が s' のサブソートであることを示す．サブソート宣言の集合を $\mathcal{D} = \{s \sqsubseteq s' | s, s' \in S^+\}$ で表す．さらに，論理式とサブソート宣言の両方を総称して式と呼ぶ．

定義 4.5（ソート代入） 論理式に出現する自由変数 $x_i{:}s_i$ を項 t_i に置き換える操作を代入といい，$\theta = \{x_1{:}s_1/t_1, \ldots, x_n{:}s_n/t_n\}$ で表す．このとき，すべての項 t_i に対して，$Sort[t_i] \sqsubseteq s_i$ ならば，θ はソート代入という．

論理式 A，B に対して，$\theta(A) = \theta(B)$ となるようなソート代入 θ を単一化子という．

4.2 意　味　論

言語 L の式の意味づけを行う．

定義 4.6 シグネチャ Σ が与えられたとき，構造 $M = (U, I)$ とは，次の条件を満たすものである．

(1) U は空でない集合である．

(2) I は次を満たす関数である．

(a) $I(s) \subseteq U$（特に，$I(\top) = U$，$I(\bot) = \emptyset$），
$I(s \sqcap s') = I(s) \cap I(s')$，
$I(s \sqcup s') = I(s) \cup I(s')$，
$I(\overline{s}) = I(\top) - I(s)$，
$I(\sim s) \subseteq I(\top) - I(s)$．

(b) $c \in C$ かつ $c{:} \to s \in Dec$ ならば，$I(c) \in I(s)$，
$f \in F$ かつ $f{:} s_1 \times \ldots \times s_n \to s \in Dec$ ならば，$I(f){:} I(s_1) \times \ldots \times I(s_n) \to I(s)$，
$p \in P$ かつ $p{:} s_1 \times \ldots \times s_n \in Dec$ ならば，$I(p) \subseteq I(s_1) \times \ldots \times I(s_n)$．特に，$I(p_s) = I(s)$．

変数割当ては，$\sigma(x{:}s) \in I(s)$ を満たす関数 $\sigma{:} V \to U$ である．

定義 4.7 解釈 $\mathcal{I} = (M, \sigma)$ が与えられたとき，項

の解釈関数 $[\![\ ]\!]_\sigma$ は，次により定義される．

(1) $[\![x\colon s]\!]_\sigma = \sigma(x\colon s)$,

(2) $[\![c]\!]_\sigma = I(c)$,

(3) $[\![f(t_1,\ldots,t_n)]\!]_\sigma = I(f)([\![t_1]\!]_\sigma,\ldots,[\![t_n]\!]_\sigma)$.

変数割当て σ に対して，$\sigma[x\colon s/u]$ は変数 $x\colon s$ には u を割り当て，それ以外の変数には σ と同じ要素を割り当てる．さらに，解釈 $(M,\sigma)[x\colon s/u]$ は $(M,\sigma[x\colon s/u])$ を表す．

定義 4.8 言語 L の解釈 $\mathcal{I}=(M,\sigma)$ が与えられたとき，$\mathcal{I}$ と式 $\alpha(\in \mathcal{F}\cup\mathcal{D})$ との充足関係を以下のように定義する．

(1) $\mathcal{I}\models p(t_1,\ldots,t_n)$ iff $([\![t_1]\!]_\sigma,\ldots,[\![t_n]\!]_\sigma)\in I(p)$,

(2) $\mathcal{I}\models(\neg A)$ iff $\mathcal{I}\not\models A$,

(3) $\mathcal{I}\models(A\wedge B)$ iff $\mathcal{I}\models A$ かつ $\mathcal{I}\models B$,

(4) $\mathcal{I}\models(A\vee B)$ iff $\mathcal{I}\models A$ または $\mathcal{I}\models B$,

(5) $\mathcal{I}\models(A\Rightarrow B)$ iff $\mathcal{I}\not\models A$ または $\mathcal{I}\models B$,

(6) $\mathcal{I}\models(\forall x\colon s)A$ iff すべての $a\in I(s)$ に対して，$\mathcal{I}[x\colon s/a]\models A$,

(7) $\mathcal{I}\models(\exists x\colon s)A$ iff ある $a\in I(s)$ が存在して，$\mathcal{I}[x\colon s/a]\models A$,

(8) $\mathcal{I}\models s\sqsubseteq s'$ iff $I(s)\subseteq I(s')$.

解釈 $\mathcal{I}$ に対して，$\mathcal{I}\models\alpha$ であるとき，$\mathcal{I}$ は α のモデルであるという．任意の解釈 $\mathcal{I}$ が α のモデルであるとき，α は恒真であるという．

定義 4.6 と定義 4.8 より，明らかに以下が成り立つ．

(1) $\mathcal{I}\models\neg s(t)$ iff $\mathcal{I}\models\overline{s}(t)$,

(2) $\mathcal{I}\models s(t)\wedge s'(t)$ iff $\mathcal{I}\models s\sqcap s'(t)$,

(3) $\mathcal{I}\models s(t)\vee s'(t)$ iff $\mathcal{I}\models s\sqcup s'(t)$.

4.3 ソート階層の宣言

定義 4.9（ソート階層の宣言） ソート階層の宣言 $H=(S^+,D)$ は次から構成される．

(1) S^+：拡張ソートの集合

(2) D：サブソート宣言による有限集合（$\{s_1\sqsubseteq s_1', s_2\sqsubseteq s_2',\ldots\}$）

ソートの等号関係，排他関係と全域関係は，サブソート宣言を簡略化した記号として導入される．$s=s'$ は $s\sqsubseteq s'$ かつ $s'\sqsubseteq s$ を示し，$s\parallel s'$ は $(s\sqcap s')=\bot$ を簡略化している．さらに，$s\mid_{s_i}s'$ は $(s\sqcup s')=s_i$ を表す．特に，$s\mid_\top s'$ を $s\mid s'$ と表す．

ソート階層の宣言 $H=(S^+,D)$ とある論理式の集合 Δ が与えられたとする．D のすべての元 $s\sqsubseteq s'$ に対して $\mathcal{I}\models s\sqsubseteq s'$ が成り立つとき，$\mathcal{I}$ を D のモデルといい，$\mathcal{I}\models D$ で表す．Δ のすべての元 A 対して $\mathcal{I}\models A$ が成り立つとき，$\mathcal{I}$ を Δ のモデルといい，$\mathcal{I}\models\Delta$ で表す．$\mathcal{I}$ が D と Δ の両方のモデルであるとき，$\mathcal{I}$ は (D,Δ) のモデルであるといい，$\mathcal{I}\models(D,\Delta)$ で表す．(D,Δ) が少なくとも 1 つのモデルを持つとき，(D,Δ) は充足可能であるという．そうでないときは，充足不可能であるという．

4.4 公理と推論規則

公理と推論規則を導入する前に，節形式の論理式を定義する．原子論理式およびその否定のことを，それぞれ正のリテラルと負のリテラルと呼ぶ．

定義 4.10 L_i が正または負のリテラルのとき，以下の形式の論理式を節という．

$$(\forall x_1\colon s_1\cdots\forall x_m\colon s_m)L_1\vee\ldots\vee L_n$$

ただし，$n\geq 0$ であり，$x_1\colon s_1,\ldots,x_m\colon s_m$ は $L_1\vee\ldots\vee L_n$ に出現するすべての変数である．さらに，この節（または節形式の論理式）は推論において $L_1\vee\ldots\vee L_n$ で表される．

節の集合を $\mathcal{C}(\subset\mathcal{F})$ で表す．上記では，定義 4.3 の項と定義 4.4 と論理式に基づいて節が定義された．これにより，節において $x\colon s$ のようなソート付きの項が出現するだけでなく，ソート述語による原子論理式 $p_s(t)$（または $s(t)$）を利用することも可能である．p，p_{s_2} が述語，s_1，s_2 がソート記号とする．たとえば，次の論理式は節である．

$$p(x\colon s_1)\vee\neg p_{s_2}(y\colon s_1)$$

ここで，述語 p_{s_2} はソート s_2 のソート述語であるので，$s_2(y\colon s_1)$ と表してもよい．

ソート階層に関する公理と推論規則 I，および節形式の論理式に適用可能な推論規則 II を次のように与える．s，s'，s'' はソートまたはソート述語，L，L' は正のリテラル，t，t' は項，C，C' は節とする．

公　理：

$(s\sqcap\bot)=\bot$	$(s\sqcup\bot)=s$	$\bot\sqsubseteq s$
$(s\sqcap\top)=s$	$(s\sqcup\top)=\top$	$s\sqsubseteq\top$
$s\sqsubseteq(s\sqcup s')$	$(s\sqcap s')\sqsubseteq s$	$s\sqsubseteq s$
$s\parallel\overline{s}$	$s\parallel\!\sim s$	$s\mid\overline{s}$

推論規則 I：

$$\frac{s\sqsubseteq s'}{s''\sqcap s\sqsubseteq s''\sqcap s'}\ (\mathit{introduction})$$

$$\frac{s\sqcup s'\sqsubseteq s\sqcup s''\quad s\parallel s'\quad s\parallel s''}{s'\sqsubseteq s''}\ (\mathit{elimination})$$

$$\frac{s\sqsubseteq s'\quad s'\sqsubseteq s''}{s\sqsubseteq s''}\ (\mathit{transition})$$

推論規則 II：

$\theta(L)=\theta(L')$ となる単一化子 θ が存在するとき，

$$\frac{\neg L \vee C \quad L' \vee C'}{\theta(C \vee C')} \ (resolution)$$

$$\frac{\neg s(t) \vee C}{\overline{s}(t) \vee C} \ (logical\ negation)$$

$\theta(s(t)) = \theta(s(t'))$ となる単一化子 θ が存在するとき，

$$\frac{\overline{s}(t) \vee C \quad s'(t') \vee C' \quad s' \sqsubseteq s}{\theta(C \vee C')} \ (subsort)$$

$$\frac{\overline{s}(t) \vee C \quad s(t') \vee C'}{\theta(C \vee C')} \ (sort\ negation)$$

$$\frac{s(t) \vee C \quad s'(t') \vee C' \quad s \parallel s'}{\theta(C \vee C')} \ (exclusivity)$$

$$\frac{s_i(t) \vee C \quad \overline{s'}(t') \vee C' \quad s \,|_{s_i} s'}{\theta(s(t) \vee C \vee C')} \ (totality)$$

推論規則 (resolution) は，ソート述語を含めた任意の述語からなる節に対して適用可能であり，従来の導出と同じである．推論規則 II のそれ以外は，構造的なソートにより拡張されたソート述語に関する推論規則である．

定義 4.11 ソート階層の宣言 $H = (S^+, D)$ と節集合 Δ から式 $\alpha_n (\in \mathcal{C} \cup \mathcal{D})$ への推論は式の有限列 $\alpha_1, \alpha_2, \ldots, \alpha_n$ であり α_i は次のいずれかである．

(i) D または Δ の元である．

(ii) 公理または，ある式 $\alpha_j (j < i)$ に推論規則を適用した結果である．

このとき，$(D, \Delta) \vdash \alpha_n$ と表す．

補題 4.1 公理は恒真である．

証明 公理 $(s \sqcap \bot) = \bot$ が恒真であることを示す．$\mathcal{I} = (M, \sigma)$ を任意の解釈とする．定義 4.6 より，$I(\bot) = \emptyset$ なので $I(s \sqcap \bot) = I(s) \cap I(\bot) = \emptyset$ より，$\mathcal{I} \models (s \sqcap \bot) \sqsubseteq \bot$ かつ，$\mathcal{I} \models \bot \sqsubseteq (s \sqcap \bot)$ が容易に示せる．他の公理も同様にして導ける． ∎

(D, Δ) の任意のモデルが，式 $\alpha (\in \mathcal{C} \cup \mathcal{D})$ のモデルでもあるとき，式 α は D と Δ の論理的帰結といい，$(D, \Delta) \models \alpha$ と表す．

補題 4.2 節 C のモデルは，$\theta(C)$ のモデルでもある．つまり，$C \models \theta(C)$．

証明 $\mathcal{I} \models \forall x_1{:}\, s_1 \ldots \forall x_m{:}\, s_m C$ とする．ただし，$x_1{:}\, s_1, \ldots, x_m{:}\, s_m$ は，C に現れるすべての自由変数である．このとき，$\mathcal{I} \models \forall y_1{:}\, s'_1 \ldots \forall y_k{:}\, s'_k \theta(C)$ を示したい．そのために，任意の要素 $a_1 \in I(s'_1), \ldots, a_k \in I(s'_k)$ に対して，

(1) $\mathcal{I}[y_1{:}\, s'_1/a_1, \ldots, y_k{:}\, s'_k/a_k] \models \theta(C)$

を示せばよい．仮定より，任意の要素 $b_1 \in I(s_1), \ldots, b_n \in I(s_n)$ に対して，$\mathcal{I}[x_1{:}\, s_1/b_1, \ldots, x_n{:}\, s_n/b_n] \models C$ である．ここで，$b_i = [\![\theta(x{:}\, s_i)]\!]_{\sigma[y_1:s'_1/a_1, \ldots, y_k:s'_k/a_k]}$ と定義する．これにより，(1) が導かれる．したがって，$\mathcal{I} \models \forall y_1{:}\, s'_1 \ldots \forall y_k{:}\, s'_k \theta(C)$ が証明できる． ∎

補題 4.3 推論規則 I, II の結論 α は，その前提 $\{\alpha_1, \ldots, \alpha_n\}$ の論理的帰結である．すなわち，$\{\alpha_1, \ldots, \alpha_n\} \models \alpha$ が成り立つ ($1 \leq n \leq 3$)．

証明 推論規則 (introduction) について示す．解釈 $\mathcal{I} = (M, \sigma)$ が $s \sqsubseteq s'$ を充足すると仮定する．このとき，$I(s) \subseteq I(s')$ が成り立つので，定義 4.6 より $I(s'' \sqcap s) \subseteq I(s'' \sqcap s')$ となる．したがって，$\mathcal{I} \models s'' \sqcap s \sqsubseteq s'' \sqcap s'$ であるので，$s'' \sqcap s \sqsubseteq s'' \sqcap s'$ は $s \sqsubseteq s'$ の論理的帰結である．

推論規則 (elimination) について示す．$I(s) \cup I(s') \subseteq I(s) \cup I(s'')$ かつ $I(s) \cap I(s') = \emptyset$ かつ $I(s) \cap I(s'') = \emptyset$ とする．任意の $d \in I(s')$ について，$d \in I(s) \cup I(s')$ ならば，$d \in I(s) \cup I(s'')$ である．さらに，$I(s) \cap I(s') = \emptyset$ より，$d \notin I(s)$ となる．したがって，$d \in I(s'')$ がいえる．

推論規則 (resolution) について示す．$\mathcal{I} \models C \vee \neg L$ かつ $\mathcal{I} \models L' \vee C'$ を仮定する．このとき，$\theta(L) = \theta(L')$ となる単一化子が存在する．補題 4.2 により，$\mathcal{I} \not\models \theta(L)$ のとき，$\mathcal{I} \not\models \theta(L')$ なので $\mathcal{I} \models \theta(C')$ が成り立つ．$\mathcal{I} \models \theta(L)$ のとき，$\mathcal{I} \not\models \neg\theta(L)$ なので $\mathcal{I} \models \theta(C)$ が成り立つ．したがって，$\mathcal{I} \models \theta(C) \vee \theta(C')$ である．よって，(resolution) の結論は前提の論理的帰結である．

推論規則 (subsort) について示す．$\mathcal{I} \models \overline{s}(t) \vee C$ かつ $\mathcal{I} \models s'(t') \vee C'$ かつ $\mathcal{I} \models s' \sqsubseteq s$ とする．このとき，$\theta(t) = \theta(t')$ となる単一化子が存在する．まず，$[\![\theta(t')]\!]_\sigma \notin I(s)$ の場合を考える．$I(s') \subseteq I(s)$ により，$[\![\theta(t')]\!]_\sigma \notin I(s)$ ならば，$[\![\theta(t')]\!]_\sigma \notin I(s')$ である．したがって，$\mathcal{I} \not\models \theta(s'(t'))$ なので，$\mathcal{I} \models \theta(C')$ が成り立つ．また，$[\![\theta(t')]\!]_\sigma \in I(s)$ の場合，$\theta(t) = \theta(t')$ より $\mathcal{I} \not\models \theta(\overline{s}(t))$ となるので，$\mathcal{I} \models \theta(C)$ が成り立つ．ゆえに，(subsort) の結論は前提の論理的帰結である．

推論規則 (exclusivity) について示す．$\mathcal{I} \models s(t) \vee C$ かつ $\mathcal{I} \models s'(t') \vee C'$ かつ $I(s) \cap I(s') = \emptyset$ とする．このとき，$\theta(t) = \theta(t')$ となる単一化子が存在する．$I(s) \cap I(s') = \emptyset$ により，$\mathcal{I} \models \theta(s(t))$ と $\mathcal{I} \models \theta(s'(t'))$ は同時に成り立つことはないので，仮定により $\mathcal{I} \models \theta(C)$ または $\mathcal{I} \models \theta(C')$ が成り立つはずである．したがって，$\mathcal{I} \models \theta(C) \vee \theta(C')$ が証明で

きる．

推論規則 (totality) について示す．$\mathcal{I} \models s_i(t) \vee C$ かつ $\mathcal{I} \models \overline{s'}(t') \vee C'$ かつ $I(s) \cup I(s') = I(s_i)$ とする．このとき，$\theta(t) = \theta(t')$ となる単一化子が存在する．$I(s) \cup I(s') = I(s_i)$ より，$\mathcal{I} \models s(t) \vee s'(t') \vee C$ である．後は，(resolution) の証明と同様にして，$\mathcal{I} \models \theta(s'(t)) \vee \theta(C) \vee \theta(C')$ が成り立つ．ゆえに，(totality) の結論は前提の論理的帰結である．

さらに，推論規則 (logical negation)，(transition)，(sort negation) についても同様に証明できる． ∎

定理 4.1 $(D, \Delta) \vdash \alpha$ ならば，$(D, \Delta) \models \alpha$ である．

証明 補題 4.1 と補題 4.3 より明らか． ∎

定義 4.12 $H = (S^+, D)$ をソート階層の宣言，Δ を節集合とする．$(D, \Delta) \vdash s \parallel s'$ であるようなあるソート s，s' が存在して，$(D, \Delta) \vdash s(t)$ かつ $(D, \Delta) \vdash s'(t)$ のとき，(D, Δ) は**排他関係上矛盾**するという．また，$(D, \Delta) \vdash A$ かつ $(D, \Delta) \vdash \neg A$ のとき，**論理的に矛盾**するという．

いずれの矛盾も成り立たないとき，(D, Δ) は無矛盾であるという．

定理 4.2 $H = (S^+, D)$ をソート階層の宣言，Δ を節集合とする．(D, Δ) がモデルを持つならば，(D, Δ) は無矛盾である．

証明 (D, Δ) がモデル $\mathcal{I}$ を持つとする．ここで (D, Δ) が排他関係上矛盾するならば，$(D, \Delta) \vdash s \parallel s'$ であるようなあるソート s，s' が存在して，$(D, \Delta) \vdash s(t)$ かつ $(D, \Delta) \vdash s'(t)$ となるはずである．定理 4.1 より，$\mathcal{I} \models s \parallel s'$ であり，$\mathcal{I} \models s(t)$ かつ $\mathcal{I} \models s'(t)$ である．したがって，$I(s) \cap I(s') = \emptyset$ となり，一方，$[\![t]\!] \in I(s)$ かつ $[\![t]\!] \in I(s')$ が成り立つ．またもしくは，(D, Δ) が論理的に矛盾するならば，$\mathcal{I} \models \neg A$ かつ $\mathcal{I} \models A$ が成り立つ．いずれの場合も仮定に反する．したがって，(D, Δ) は無矛盾である． ∎

反駁は，(D, Δ) から空節 $\emptyset$ の推論である．

系 4.1 $(D, \Delta) \vdash \emptyset$ ならば，$(D, \Delta) \models \emptyset$ である．

証明 推論規則（resolution）が最後に適用されているとき，$\theta(L) = \theta(L')$ が成り立つ単一化子が存在し，$(D, \Delta) \vdash \neg L$ かつ $(D, \Delta) \vdash L'$ が成り立つ．したがって，定理 4.1 より，$(D, \Delta) \models \neg L$ かつ $(D, \Delta) \models L'$ となる．今，(D, Δ) がモデル $\mathcal{I} = (M, \sigma)$ を持つと仮定する．すると，$\mathcal{I} \models \theta(L)$ かつ $\mathcal{I} \not\models \theta(L')(= \theta(L))$ となり矛盾する．したがって，(D, Δ) はモデルを持たないので，$(D, \Delta) \models \emptyset$ が成り立つ．推論規則 II の他が最後に適用されているときも同様である． ∎

次の節では，2.2 節の例に対する反駁導出を示す．

4.5 推 論 例

ソート階層の宣言と節集合からの反駁例を示すために推論図を用いる．推論図は，始式から推論規則を適用していく手順を図示したものである．

定義 4.13 推論図を，次のように定義する[13]．

(1) 始式は推論図である．

(2) $P_1, \ldots, P_n$ がそれぞれ $S_1, \ldots, S_n$ を終式とした推論図のとき，

$$\frac{S_1 \cdots S_n}{S}$$

が推論規則ならば，

$$\frac{P_1 \cdots P_n}{S}$$

は推論図である（$1 \leq n \leq 3$）．

(D, Δ) からの推論図とは，公理もしくは D と Δ のいずれかの元を始式とした推論図である．次の無矛盾な節集合 $S_1, \ldots, S_5$ とソート階層の宣言 D，D' が与えられたとする．

$S_1 = \{unhappy(c)\}$
$S_2 = \{\neg emotional(c)\}$
$S_3 = \{\neg loser(d),\ player(d)\}$
$S_4 = \{\neg winner(d),\ \neg loser(d)\}$
$S_5 = \{loser(d)\}$
$D = \{unhappy = \sim happy,$
$\quad happy \sqsubseteq emotional,$
$\quad unhappy \sqsubseteq emotional\}$
$D' = \{winner \sqsubseteq player,\ loser \sqsubseteq player,$
$\quad winner \mid_{player} loser,\ winner \parallel loser\}$

これに対して次の論理式

$C_1 = \neg happy(c)$
$C_2 = \neg happy(c) \wedge \neg unhappy(c)$
$C_3 = winner(d)$
$C_4 = \neg player(d)$
$C_5 = \neg winner(d)$

の真偽を推論したいときに，その否定形

$C'_1 = happy(c)$
$C'_2 = happy(c) \vee unhappy(c)$
$C'_3 = \neg winner(d)$
$C'_4 = player(d)$
$C'_5 = winner(d)$

を作り，節集合とその否定形の節との和集合 $\Delta_1 = S_1 \cup \{C'_1\}, \ldots, \Delta_5 = S_5 \cup \{C'_5\}$ から反駁を行う．**図 8** の (例 1)〜(例 5) は，(D, Δ_1)，(D, Δ_2)，(D', Δ_3)，(D', Δ_4) と (D', Δ_5) からの反駁を推論図で表している．以上の推論は 2.2 節の終わりで述べたオーダーソート論理の拡張に求められる要件を満たしている．

(例 1)

$$\dfrac{happy(c)\quad unhappy(c)\quad \dfrac{\dfrac{unhappy = \sim happy}{happy \sqcap unhappy = happy \sqcap \sim happy}\quad happy \parallel \sim happy}{happy \parallel unhappy}}{\emptyset}$$

(例 2)

$$\dfrac{\dfrac{happy(c) \lor unhappy(c)\quad \dfrac{\neg emotional(c)}{\overline{emotional}(c)}\quad happy \sqsubseteq emotional}{unhappy(c)}\quad \overline{emotional}(c)\quad unhappy \sqsubseteq emotional}{\emptyset}$$

(例 3)

$$\dfrac{\neg winner(d)\quad \dfrac{\dfrac{\neg loser(d)}{\overline{loser}(d)}\quad player(d)\quad winner\ |_{player} loser}{winner(d)}}{\emptyset}$$

(例 4)

$$\dfrac{\dfrac{player(d)\quad \dfrac{\neg winner(d)}{\overline{winner}(d)}\quad winner\ |_{player} loser}{loser(d)}\quad \neg loser(d)}{\emptyset}$$

(例 5)

$$\dfrac{winner(d)\quad loser(d)\quad \dfrac{\dfrac{winner \sqcup loser \sqsubseteq \top\quad winner\ |\overline{winner}}{winner \sqcup loser \sqsubseteq winner \sqcup \overline{winner}}\quad winner \parallel loser\quad winner \parallel \overline{winner}}{loser \sqsubseteq \overline{winner}}}{\emptyset}$$

図 8　推論図による反駁例
Fig. 8　Examples of refutation.

5. 研究の成果と考察

最後に，本研究の成果と考察について述べる．

完全性に関する考察 系 4.1 により，4.4 節の公理と推論規則を用いた反駁推論が健全であることが証明された．形式的な結果としてはさらに完全性の証明があるが，本稿では今後の課題とする．その証明は，ソート階層に関する推論と節形式の推論の両方の完全性を示すように行われるだろう．特に，前者の完全性は，型付きの λ 計算などで導入されているサブタイプの体系のように，フィルタを使ったモデルの構築によって，完全性を証明する方法が有効だと考えられる．その上に，エルブランモデルを用いた反駁の完全性を示すことで，体系全体の完全性を証明する．

計算効率について 本研究で提案した推論体系では，効率の良い推論のために導出原理に基づいた節形式への推論を採用している．しかしながら，ソート表現に演算子を導入して表現を複雑にした結果，従来のソート論理より推論の計算量が増大することが予想される．これを解決するためには，構造的なソートにも節のように限定された表現などを導入して効率を向上させる方法が考えられる．本稿では，否定的なソートを扱う論理の拡張を目的としており推論の効率化への対応は扱わないが，将来検討する必要がある．

推論体系の妥当性 オーダーソート論理の研究[5),9),15),21)]は，単一のソートしか持たない一階述語論理に複数のソートやソート階層を導入して，ソート付きの論理式に対する推論方法を提案している．さらに，知識表現や論理の自然な発想として，型表現であるソートを単項述語として利用できる論理体系が研究されている[3),7),8)]．それらの研究（Beierle らの論理や Frisch の論理）と本研究の否定ソートを含んだ論理，および通常のオーダーソート論理との比較を**表 2**に示す．

通常のオーダーソート論理では，論理式にソート付きの項表現を許し，ソート階層はサブソートによってのみ宣言される．その延長として Beierle らや本研究のアプローチは，ソート述語や構造的ソートによる拡張を行っている．それに対して，Frisch の論理では，サブソートを使わずソート述語のみを許す一階述語表現の論理式で，ソート情報を記述しており，その式の集合はソート理論と呼ばれる．そのソート理論では，ソート付きの項は出現せず，言明的な知識を表す知識ベースでのみソート付きの項や論理式を用いる．また，あくまでソート述語はソート理論を記述するもので，言明的な知識には利用されない．さらに，ソート理論

表 2 ソート論理の比較
Table 2 Comparison between sorted logics.

ソート論理の種類	項と論理式	ソート階層	推論体系の拡張
オーダーソート論理	ソート付き	サブソート	なし
Beierle の論理	ソート付き + ソート述語	サブソート	2 つ規則を追加
Frisch の論理	構造的ソート付き	ソート理論	なし
否定ソートの論理	構造的ソート付き + ソート述語	拡張サブソート	5 つ規則を追加

はソートシグネチャの代わりの役割を果たして，ソート代入においてその情報が利用される．ここで重要なのは，Frisch の論理でもソート理論で複雑なソートを宣言できるが，本研究には次の点で新規性がある．

- 否定ソートを扱うために導入された，排他性や全域性の宣言
- 拡張ソート間関係に関する推論体系
- 構造的ソート（すなわち，排他性や全域性）に対応した節の導出体系

本研究では，否定的なソート情報を扱うのを目的に，Beierle らの論理に基づいた拡張を行っている．そのために，排他性や全域性を宣言できる構造的なソート表現を導入し，その上に拡張ソート間関係に関する推論体系も提案している．また，節の推論規則を 5 つ追加して，構造的ソートに対応した節による導出を可能にした．Frisch の方法ではソート情報を一階述語論理で表し，特別なソートの推論メカニズムは導入しないという立場である．本研究で形式化した拡張ソート間関係は Frisch の体系におけるソート理論により，一階述語論理で変換することが可能である．しかしながら，知識表現で宣言的な知識と階層的な知識が区別されているように，ソート階層を論理式と区別し，変数を含まない内包的な表現で簡潔に表して推論するには，本研究の論理と推論体系が必要である．たとえば，互いに対立するソートを，$\neg(happy(x) \wedge unhappy(x))$ のように記述することは可能である．しかし本研究のように構造的なソートによって表現し推論すれば，次の利点を得ることができる．1 つは，変数を含まない拡張ソート間関係の推論に置き代わった部分では，単一化やソート代入による負荷のかかる推論を回避できる．その結果，複雑なソート情報に付随して増加する計算量への軽減をもたらす．もう 1 つは，別の視点を持った言明的な知識と階層的な知識が，それぞれの特徴を生かした方法で表現され，それぞれ独立して推論アルゴリズムを持つことができる．これにより，知識表現言語として実用を想定した推論体系への寄与が期待できる．

終わりに 本研究では，構造的なソート表現やソート間関係による拡張によって，否定が内在するソートがソート階層に含まれているときに，その否定の特性を扱うことのできる推論体系を実現した．否定演算子や反意語を構造的なソート表現の一部として見なすことで，ソート階層に含まれる否定的ソートを認識できる．特に，古典論理の否定，強い否定および互いに対立するソートをソート階層の中で関係付け，肯定表現との排他性および全域性から，それぞれの否定の特徴に沿った推論をもたらす．それにより，ある語彙が抽象的な表現のサブソートとして肯定的な役割を持ちながら，相反するソートに対してはその排他性から統語的に否定の役割を持たせることができた．さらに，ソートの排他関係によって矛盾の定義が拡張され，否定演算子では表現できない反意語を含めて推論の無矛盾性がいえるようになった．その結果，ソート階層による知識表現の利便性を備えながら，否定表現の拡張を実現している．

本稿で提案した推論体系は，一般的な節形式に対してであるが，今後の展望として計算機上で実現させるとき，存在量化記号をスコーレム化したうえで，ソート間関係とホーン節による論理プログラミング言語の形式化が考えられる．しかし，ヘッド部のリテラルに否定的記述が可能なので，すべてのプログラム節の集合が無矛盾だとは限らない．そのとき，矛盾するプログラムをどう扱うかが今後の課題となる．

参 考 文 献

1) Aït-Kaci, H. and Nasr, R.: LOGIN: A Logic Programming Language with Built-In Inheritance, *Journal of Logic Programming*, Vol.3, No.3, pp.185–215 (1986).
2) Basin, D., Matthews, S. and Vigano, L.: Labelled Propositional Modal logics: Theory and Practice, *Journal of Logic Computation*, Vol.7, No.6, pp.685–717 (1997).
3) Beierle, C., Hedtsuck, U., Pletat, U., Schmitt, P. and Siekmann, J.: An order-sorted logic for knowledge representation systems, *Artificial Intelligence*, Vol.55, pp.149–191 (1992).
4) Carpenter, B.: *The Logic of Typed Feature Structure*, Cambridge University Press (1992).
5) Cohn, A.G.: Taxonomic reasoning with many

sorted logics, *Artificial Intelligence Review*, Vol.3, pp.89–128 (1989).

6) Fauconnier, G.: *Espas Mentaux*, Editions de Minuit (1984). 坂原，水光，田窪，三藤（訳）：メンタルスペース，白水社 (1987).
7) Frisch, A.M.: A Genaral Framework for Sorted Deduction: Fundamental Results on Hybrid Reasoning, *Proc. 1st International Conference on Principles of Knowledge Represntation and Reasoning* (1989).
8) Frisch, A.M.: The substitutional framwork for sorted deduction: Fundamental results on hybrid reasoning, *Artificial Intelligence*, Vol.49, pp.161–198 (1991).
9) Hill, P.M. and Topor, R.W.: A semantics for typed logic programs, *Types in Logic Programming*, Pfenning, F. (Ed.), MIT Press (1992).
10) Kifer, M., Lausen, G. and Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages, *J. ACM*, Vol.42, No.4, pp.741–843 (1995).
11) 森下真一：知識と推論，共立出版 (1994).
12) Nitta, K., Tojo, S., et al.: Knowledge Representation of New Helic II, *Workshop on Legal Application of Logic Programming, ICLP '94* (1994).
13) 小野寛晰：情報科学における論理，日本評論社 (1994).
14) 太田　朗：否定の意味，大修館書店 (1980).
15) Schmidt-Schauss, M.: *Computational Aspects of an Order-Sorted Logic with Term Declarations*, Springer-Verlag (1989).
16) Shoham, Y.: *Reasoning about Change*, The MIT Press (1988).
17) Wagner, G.: Logic Programming with Strong Negation and Inexact Predicates, *Journal of Logic Computation*, Vol.1, No.6, pp.835–859 (1991).
18) Wagner, G.: *Vivid Logic: Knowledge-Based Reasoning with Two Kinds of Negation*, Springer-Verlag (1994).
19) Walther, C.: A Mechanical Solution of Schuber's Steamroller by Many-Sorted Resolution, *Artificial Intelligence*, Vol.26, No.2, pp.217–224 (1985).
20) Walther, C.: Many-Sorted Unification, *Journal of the Association for Computing Machinery*, Vol.35, p.1 (1988).
21) Weibel, T.: An Order-Sorted Resolution in Theory and Practice, *Theoretical Computer Science*, Vol.185, No.2, pp.393–410 (1997).
22) Yasukawa, H., Tsuda, H. and Yokota, K.: Objects, Properies, and Modules in $\mathcal{QUIXOTE}$, *Proc. FGCS '92*, pp.257–268 (1992).
23) Yokota, K.: Quixote: A Constraint Based Approach to a Deductive Object-Oriented Database, Ph.D. Thesis, Kyoto University (1994).

(平成 12 年 3 月 10 日受付)
(平成 14 年 2 月 13 日採録)

兼岩　憲（正会員）

1993〜1996 年富士通（株）勤務．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．2001 年同大学院情報科学研究科博士後期課程修了．現在，国立情報学研究所情報学基礎研究系助手．論理プログラミング，ソート論理および知識表現に関する研究に従事．ソフトウェア科学会，電子情報通信学会，人工知能学会，ALP，ASL 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 年北陸先端科学技術大学院大学情報科学研究科助教授，2000 年同教授．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，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，Folli 各会員．