Title	複数の否定を表現するオーダーソート論理の研究
Author(s)	萩原, 信吾
Citation	
Issue Date	2004-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1792
Rights	
Description	Supervisor:東条 敏, 情報科学研究科, 修士

修 士 論 文

複数の否定を表現するオーダーソート論理の研究

指導教官　東条 敏 教授

審査委員主査　東条 敏 教授
審査委員　鳥澤 健太郎 助教授
審査委員　小野 寛晰 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

210068 萩原 信吾

提出年月: 2004 年 2 月

概 要

本稿は現在のオーダーソート論理に対しより人間の思考を反映したものとするために，現在構築されている否定の体系を人の記憶と類似した直観主義論理の体系を持って見直し，再構築するものである．

オーダーソート論理は，導出を元とした推論体系が定義されており，その論理的体系の基盤が確かなものである．また，自然言語を取り扱う上で上位下位概念は非常に重要な位置を占める．なぜならば，人は上位下位概念を用いて，ある語彙が他の語彙の意味を内包しているかどうかで，何を表現しているのかを理解するからである．したがって，知識をどのように階層化表現し，なおかつ，それをどのように論理的に取り扱うのかという問題に対して，オーダーソート論理というものは有用なものであると考えられる．これらのことから，オーダーソート論理は人工知能の分野において広く応用されている．そのオーダーソート論理に否定の表現を加えたものが先行研究の兼岩・東条 [2002] の研究がある．この研究では，オーダーソート論理に弱い否定，強い否定，対立概念による否定，を導入している．

否定というのは，論理的矛盾を引き起こす要素であるので論理構造を考える上では非常に重要なものである．否定表現としては，人間は「好きで無いわけではない」や「嫌いでないが，好きでもない」,「不足している」といったような，文法上の二重否定や，対立する語彙による否定表現，また語彙が接辞の形で否定的意味を内在した表現といったものを頻繁に使用する．前者の例では「好きである」ということを命題 A として表現したならば,「好きでないわけではない」というのは論理式で $\neg\neg A$ と表現可能である．しかしながら，この表現は，一般の古典主義論理の体系上で考えるならば，二重否定の除去が認められているため，A という論理式と同値となる．しかし，上記の例は「好きである」ということを表現しようとした自然言語の表現ではなく,「好きではない」ということを否定しようという意図で用いられた言葉である．したがって，人間の思考を古典主義論理では十分に表現できるとは言い難い．しかし，先行研究が導入している否定の体系というものは古典主義論理の否定体系を元としており，人間の思考にそった表現を実現しているとはいえない．

そこで本研究では，直観主義論理を用いてこの否定体系を再構築し，拡張するものである．直観主義論理では，複数の可能世界によって意味論が構築されている．これにより，

ある世界においては真であるような命題も，他の可能世界では偽である，というような，状況を考慮した人間の知識状態を表現できる．

そのため，命題に関する解釈方法も異なる．直観主義論理の場合では，ある物事が正しいかどうかで判断せず，あくまで主体が人間であるかのように証明解釈という手段を用いてその意味が決定される．つまりは，上記のような命題に対してその真偽値を論じるような場合であったならば，その命題が真である場合その命題に対し「その命題が正しいといえる証明方法を持っている」と解釈される．

顕著な例でいうならば，排中律といわれる公理が古典主義論理には存在する．それは，‘$A \vee \neg A$’ と言ったもので，これは古典主義論理でトートロジーである．しかし，これを直観主義論理の証明解釈にさらして考えた場合は,「すべての命題に関して，それが正しいといえる証明方法があるか，またはそれが正しい場合矛盾を導き出す方法が存在する」と解釈される．そのようなことは到底できるわけではなく，知らないことに関しては，正しいとも，矛盾するとも具体的に述べることはできない．つまり，あらゆる事に関して好きであるか，そうでないかが必ず決定できるわけではないということである．

このように否定の体系を見直した場合，既存の論理体系で導入された他の否定体系に影響を及ぼす．直接影響を受けるのは，弱い否定を表現していた弱い否定ソートである．これは，古典主義論理を基礎としていたため，二重否定の弱い否定ソートと言うものは存在しない．しかしながら，直観主義論理を用いて考えた場合，二重否定の除去が認められていないからそうではない．二重否定がそのまま残ってしまう．したがって，単純に弱い否定ソートのみならず，二重否定の形を持った否定ソートも導入する必要がある．また，強い否定，対立構造これらも，既存の単一の可能世界論に基づいた意味論ではなく，複数の可能世界における意味論に置き換える必要がある．こうすることで，オーダーソート論理により「紫は赤であるとは言えないが，赤でないとも言い切れない」といったような表現が可能となる．

また，既存の研究ではその推論体系として，ヒルベルト流の推論系が提案されていたが，この手法では計算機との相性はよくない．そこで，本研究では計算機に実装する上で相性のよいシーケント計算の体系を用いてその推論系を提案し，またそれを計算機上において実装する．ただし，シーケント計算が可能なことはその論理式がいかなるモデルにおいてもトートロジーかどうかを判断できるだけであるので，実際にモデルを定義し，その任意の可能世界において任意の式の真偽値がどうかということも判断できるようなプログラム，これもまた実装を行う．

目 次

第1章　序論

1.1　導入

人間の思考を形式化しようとしたものが論理学だと考えたとき，自然言語を取り扱う上で上位下位概念と否定という論理的構造は非常に重要な位置を占める．なぜならば，人は上位下位概念を用いて，ある語彙が他の語彙の意味を内包しているかどうかで，何を表現しているのかを理解するからである．具体的な例を挙げるならば,「白」という語彙がどういった意味をなすのかということに対して，上位の語彙である「色」という語彙によってその意味が決定されている部分が大きい．たとえば「可視光線を全て反射した色」と言えば「白」のことをおおむね表すことができるからである．したがって，人工知能の分野において，知識をどのように階層化表現し，なおかつ，それをどのように論理的に取り扱うのかという問題に対して，オーダーソート論理というものは有用なものであると考えられる．

また否定というのは，論理的矛盾を引き起こす要素であるので論理構造を考える上では非常に重要なものである．否定表現としては，人間は「好きで無いわけではない」や「嫌いでないが，好きでもない」,「不足している」といったような，文法上の二重否定や，対立する語彙による否定表現，また語彙が接辞の形で否定的意味を内在した表現といったものを頻繁に使用する．前者の例では「好きである」ということを命題 A として表現したならば，論理式では単純に $\neg\neg A$ と表現可能である．しかしながら，この表現は，一般の古典主義論理の体系上で考えるならば，二重否定の除去が認められているため，A という論理式と同値となる．しかし，上記の例は「好きである」ということを表現しようとした自然言語の表現ではなく,「好きではない」ということを否定しようという意図で用いられた言葉である．したがって，人間の思考を古典主義論理では十分に表現できるとは言い難い．また,「嫌い」という語彙は論理的にどのように表現できるかを考えた場合，単純に「好き」という語彙の否定であると考えた場合,「嫌い」という語彙は前述の A を「好き」という命題というのを利用し，$\neg A$ と考える事もできる．そうした場合,「嫌いでない」と

いうのは $\neg\neg A$ となり,「嫌いでないが，好きでもない」と言う例は矛盾する言葉であるとされてしまう．しかし，人間はこれを矛盾したものとしては理解しない．

そこで，いくつかの否定を考慮したオーダーソート論理としては，先行研究として [12] らの研究がある．この中では [7] らによるオーダーソート論理を拡張し，弱い否定ソート，強い否定ソート，等を導入し，オーダーソート内で表現可能な否定の種類を広げるとともに，体系的に階層の中でのソート間の関係を定義している．しかしながら，導入されている弱い否定の否定体系が古典主義論理の体系であるために，二重否定の除去が認められており，上記例のように「〜でないわけではない」のような人間特有の表現は，正確に表現できない．また，推論体系がヒルベルト流であるために，このままでは計算機による推論に向いていないという問題もある．

このような現状をふまえ，本研究では現状のオーダーソート論理に対して，知識表現の幅を広げるべく様々な否定表現を考慮するとともに，その否定体系の意味論を直観主義論理の立場によって見直す．これによって，オーダーソート内に含まれる否定ソートは先行研究のものとは別の意味を持つこととなり，前述の例のような「〜でない訳ではない」という人間特有の表現を，オーダーソート内に置いてより正確に主張できるようになる．また，直観主義論理には LJ という計算機と相性のよい推論体系がある．これは，LK の推論体系において結論部の論理式がたかだか 1 つという制約を設けたシーケント計算方法である．そこで，計算機に実装を行いやすくするために既存の推論体系を本研究の拡張もふまえた LJ の推論体系にもする．

1.2 本稿の構成

本稿は始めに，基礎知識となる直観主義論理について述べる．その際，直観主義命題論理でその体系の大まかな構造を述べたあと，オーダーソート論理に必要となる，直観主義述語論理に関して述べる．次に，オーダーソート論理について述べる．前半はオーダーソート論理の基礎であり，本稿による拡張を含まない．後半においては，構造的なソート，ソート間関係に関しては，部分的に本稿による先行研究に対する変更も伴う．そして，その後に本稿のオーダーソート論理の拡張を述べた後に，本稿で作成したシステムの説明を行う．そして最後に，結言として，まとめと今後の課題を述べる．

第2章　直観主義論理

直観主義論理とは，既知の知識のみをによる推論を使用する論理体系である．これは古典主義論理とは大きく異なる．古典主義論理ではある命題が与えられた場合，そのことについて必ず真偽が定まる．すなわち，あらゆる事物に関して命題を判断する主体は，命題に関して真偽を判断する完全な知識を持っているのである．このことを表しているのが排中律[1]と呼ばれる公理である．これはある命題が存在した場合，その命題に関して付値が，真かまたは偽である，と主張することは常に正しい，というもので，上記のように A か $\neg A$ かが，必ず定まる事を表している．このことから，古典主義論理では含意も含め，論理式は全て '$\neg$','$\vee$' の組み合わせか，もしくは '$\neg$','$\wedge$' の組み合わせで表すことができる．しかしながら，この変換によって論理式内で人間の感覚にはそぐわない結果が出現する．そのことなどは [10] でも話されており，例えば

> 「もし円周率の中に '7' が 20 個続いた部分があるならば，'7' が 19 個続いた部分がある」

という文を考えた場合，

- 命題 A：「円周率の中に '7' が 20 個続いて出現する」
- 命題 B：「円周率の中に '7' が 19 個続いて出現する」

として，文は論理式で

$$A \supset B$$

と記述可能である．これを古典主義論理で考えたならば，論理式は

$$\neg A \vee B$$

と変換できる．これはすなわち，

[1]排中律：ある命題 A があったとき，$A \vee \neg A$ が恒真であるということ

「円周率の中に‘7’が20個続いて出現しないか，または19個続いて出現する」

という意味になる．上記のような式の変換が許されている以上，古典主義論理では，上記2つの文は同じ意味を表していることになる．しかしながら，これは明らかに人間の感覚にそぐわないものである．この原因が前述の排中律（完全な知識の保有）なのである．人間の感覚にあわない排中律を，公理から取り除いた論理体系が，直観主義論理であるとしばしばいわれる．その場合，論理式はどのように扱われるのかというと，古典主義論理のように含意を変換しない．なぜならば，排中律が成り立たないので変換できないのである．したがって，直観主義論理では，含意‘$\supset$’，論理和$\vee$，論理積$\wedge$それぞれが独立した演算子として扱われる．

そのことからも，直観主義論理は，その意味解釈の方法として，[3],[5],[10]に紹介されているように，古典主義論理とは異なった方法がとられている．古典主義論理では命題に関して,「正しい」もしくは「正しくない」という解釈がなされてきた．しかし，直観主義論理の解釈の一つの方法として証明解釈というものがあり，それはある命題に関して

「正しいと言える具体的な証明方法が存在する」か「正しいとした場合矛盾を導き出す具体的な証明方法がある」

と解釈される．この解釈方法を用いた場合，排中律を考えてみると，$\neg A \vee A$は

「Aが正しいと言える具体的な証明方法が存在する，もしくはAが正しいとした場合矛盾を導き出す具体的な証明方法がある」

と解釈可能である．したがって，円周率の例のように，あらゆるものに具体的な証明方法など存在しないので排中律が成り立たないことがこの解釈方法で解る．そこで，円周率の例文では「円周率中に‘7’が20個連続して出現するという事が具体的に証明できたならば，同時に‘7’が19個連続して出現するという証明を得る事ができる」と解釈される．また$\neg A \vee B$と変換した場合は「円周率中に‘7’が20個連続して出現するという事が証明できると仮定すると矛盾するといえる証明が存在するか，または‘7’が19個連続して出現するという証明を得る事ができる．そして，どちらの証明が得られるか，もしくはともに得られるかが決まっている.」と解釈される．この場合両方の証明も得ることができない事が解るのでこれは恒真でないと解釈できる．

このような直観主義論理の意味論はどのように定義されているのかというと，これはクリプキ・モデルを用いて定義される．このクリプキ・モデルは複数の可能世界を考える．

可能世界が異なるということは，それぞれの可能世界で，成り立っている事実が異なっていることを表現している．ただ，直観主義論理で取り扱うクリプキ・モデルはその可能世界に制約が設けられている．それは，可能世界には順序が決められており，その順序関係によって上位の可能世界で成り立つリテラルの集合は下位の可能世界の成り立つリテラルの集合を内包していなければならない．つまり，直観的には，知識は必ず増えていき，忘れることはないという制約が設けられている．そのことが人間の記憶の形を表現しているともいえる．

2.1 直観主義命題論理

構文論

まず，論理結合子 $\vee, \wedge, \neg, \supset$ と，命題変数 $p, q, r, \ldots$，命題定数 $\top, \bot$，補助記号 ‘,’ を含んだ言語 $\mathcal{L}$ が用いられる．その言語 $\mathcal{L}$ において論理式は帰納的に定義される．命題変数を p とした場合，$p, \neg p$ は論理式である．また，A, B を論理式とした場合 $A \vee B, A \wedge B, A \supset B$ もまた論理式である．

意味論

可能世界集合 $M = \{w, w', w_0, \ldots\}$ とその要素間の順序関係を定義した集合 $R = \{w \leq w', \ldots\}$ の 2 つ組 $F = \langle W, R \rangle$ をクリプキ・フレームという．そして，可能世界に対する付値を与える写像 V とクリプキ・フレームの 3 つ組 $\mathcal{M} = \langle W, R, V \rangle$ によってクリプキ・モデルは定義される．したがって，直観主義命題論理におけるクリプキ・モデルは以下のように定義される．[2]

$\mathcal{M} = \langle M, R, V \rangle$
M：可能世界集合
R：可能世界間の推移的でかつ反射的な二項関係の集合で，関係 ${}_xR_y$ は大小関係の記号 $\leq$ で定義される．
V：命題変数 p において $V(p)$ を M の遺伝的な部分集合に対応させる写像.

[2]なお，このような意味論に関しては Beth model[2] と言うようなものもあるが，今回は可能世界に対して固有の対象領域をもうけるため Kripke model を採用した．

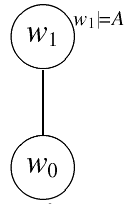


図 2.1 クリプキ・モデルの例 1

これにより，おのおのの論理式における意味論は以下のように定義される．なお，定義内における w,w' はそれぞれ可能世界を表す．また，$w \models A$ とき，論理式 A は可能世界 w で成り立つことを表し，$w \not\models A$ は成り立たない事を表す（$w \models A$ を，論理式 A は可能世界 w で真であるともいう）．なお，w は可能世界，A は論理式，p は命題変数とする．

$w \models p$ ⇔ $w \in V(p)$

$w \models A \wedge B$ ⇔ $w \models A$ かつ，$w \models B$ である

$w \models A \vee B$ ⇔ $w \models A$ または，$w \models B$ である

$w \models A \supset B$ ⇔ $w \in M$ において $w \leq w'$ となる全ての $w' \in M$ で $w' \not\models A$ または，$w' \models B$ が成り立つ

$w \models \neg A$ ⇔ $w' \in M$ において，$w \leq w'$ となる全ての $w' \in M$ で $w' \not\models A$ が成り立つ

これより，$\perp$ を矛盾を表す論理定数として導入した場合，直観主義論理において $\neg A$ は $A \supset \perp$ の略記であることがわかる．

モデルにおける真偽値は以下のように定義されている．あるクリプキ・モデル $\langle M, R, V\rangle$ において，どんな $w \in M$ においても $w \models A$ となり，成り立つような論理式 A があるとき，論理式 A を真であるといい，そうでなければ偽であるという．また，注意すべき事は，A が偽であっても $\neg A$ が真であるとは限らないことである．理解のために簡単に例をあげる．

図 2.1 はクリプキ・モデルのハッセ図であり，その定義は以下のようになる．

$\mathcal{M} = \langle M, R, V\rangle$

$M = \{w_0, w_1\}$

$R = \{w_0 \leq w_1\}$

$V = \{V(A) = \{w_1\}\}$

このようなクリプキ・モデルでは，w_0 で A が成り立っていない $(w_0 \not\models A)$．すなわちすべての可能世界で論理式 A は成り立っていない．よって，論理式 A は偽である．これを $M \not\models A$ のように表す．次に論理式 $\neg A$ を考えてみる．$w \models \neg A$ となるためには可能世界 w 以上のすべての可能世界 w' において $w' \not\models A$ が成り立たなければならないので，図 2.1 ではどの可能世界においても論理式 $\neg A$ は成り立たない．よって論理式 $M \not\models \neg A$ となる．このように，複数の可能世界考えた場合， 論理式 A も $\neg A$ も偽であるようなモデルが容易に存在する．

遺伝性

直観主義論理上のクリプキ・モデル $\mathcal{M} = \langle M, R, V \rangle$ において $w \in M$ であるとき $E \subseteq M$ である M の部分集合 E を考えると，$w \in E$ において $w \leq w'$ の関係が $\mathcal{M}$ で定義されている場合，$w' \in E$ であるような M の部分集合 E は遺伝的であるという．

また，直観主義論理でのクリプキ・モデルにおいては，論理式についても遺伝的であり，ある論理式 ϕ が存在したとき

$$E = \{w \in M \mid M, w \models \phi\}$$

となるような，可能世界の集合 E もまた遺伝的である．

このことはつまり，あるクリプキ・モデルを考え，$w \models A$ であるとき，$w \leq w'$ となるすべての w' で $w' \models A$ が成り立つことを表している．したがって，ある命題 p に対して付値関数 $V(p)$ は，遺伝的な M の部分集合を割り当てる．

エンドワールド

クリプキ・モデルにおけるエンドワールドを考える．エンドワールドとはそれ以上上位の可能世界が存在しない可能世界のことである．

定義 1 あるクリプキ・フレーム $\mathcal{F} = \langle M, R \rangle$ において，$w, w' \in M, w \leq \forall w' \notin R$ の場合のみ，w はエンドワールドであるという．

あるエンドワールド w で $\neg A$ を考えてみる．直観主義論理において $\neg A$ は $\neg A \equiv A \supset \perp$ である．したがって，エンドワールド w で A が成り立っていない場合，それ以上，上位の世界がないので $w \models \neg A$ が成り立つ．これは，エンドワールドでは全ての命題の真偽値が

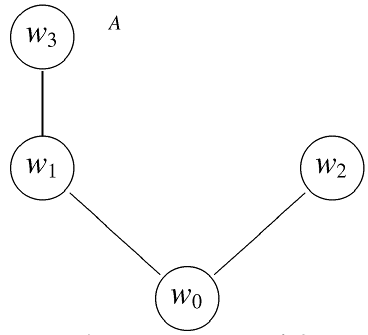


図 2.2 クリプキ・モデルの例：二重否定

決定されていることを表す．これよりエンドワールドでは，古典主義論理で成り立っていた公理が全て成り立つ．すなわち，エンドワールドは古典主義論理同様に，完全な知識が備わった世界ということができる．したがって，可能世界が 1 つだけのクリプキ・モデルを考えた場合，そのモデルでは古典主義論理の公理が全て成り立つ．よって，古典主義論理とは，可能世界が一つのみの直観主義論理ということもできる．このように，直観主義論理は古典主義論理を内包しているのである．

二重否定

直観主義では一般に二重否定の除去は成り立たない．これは排中律が成り立たないのでことから導き出せる．ではクリプキ・モデルではどのようになっているのかというのを例を見て考える．図 2.2 はクリプキ・モデルのハッセ図である．定義は以下のようになる．

$\mathcal{M} = \langle M, R, V \rangle$
$M = \{w_0, w_1, w_2, w_3\}$
$R = \{w_0 \leq w_0, w_1 \leq w_2, w_1 \leq w_3\}$
$V = \{V(A) = \{w_3\}\}$

この例で各世界において成り立つものを考える．命題 A のみに注目した場合，以下のようになる．

$$w_0 \not\models A \tag{2.1}$$
$$w_1 \not\models A \tag{2.2}$$
$$w_2 \not\models A \tag{2.3}$$
$$w_3 \models A \tag{2.4}$$

したがって，$w_0 \models \neg A$ となるためには，$w_0 \not\models A, w_1 \not\models A, w_2 \not\models A, w_3 \not\models A$ であるので，(2.4) と矛盾する．よって $w_0 \not\models \neg A$．また $w_1 \models \neg A, w_3 \models \neg A$ も同様に (2.4) に矛盾する．$w_2 \not\models \neg A$ は (2.3) より矛盾しない．したがって，成り立つ．これより，

$$w_0 \not\models \neg A \tag{2.5}$$

$$w_1 \not\models \neg A \tag{2.6}$$

$$w_2 \models \neg A \tag{2.7}$$

$$w_3 \not\models \neg A \tag{2.8}$$

が成り立つ．次に，二重否定を考える．$w_0 \models \neg\neg A$ が成り立つ条件は $w_0 \not\models \neg A, w_1 \not\models \neg A, w_2 \not\models \neg A, w_3 \not\models \neg A$ である．これは (2.7) より矛盾する．$w_2 \models \neg\neg A$ も同様に (2.7) より矛盾する．$w_1 \models \neg\neg A$ においては (2.6),(2.8) より成り立つ．また遺伝的関係より，w_1 以上の可能世界でこれが成り立つので，以下のようになる．

$$w_0 \not\models \neg\neg A \tag{2.9}$$

$$w_1 \models \neg\neg A \tag{2.10}$$

$$w_2 \not\models \neg\neg A \tag{2.11}$$

$$w_3 \models \neg\neg A \tag{2.12}$$

このように，$\neg\neg A$ の成り立つ可能世界の集合は A が成り立つ可能世界の集合を内包している．$\vdash_i$ で直観主義論理において成り立つことを表す場合，

$$\vdash_i A \supset \neg\neg A$$

が成り立つ．また，(2.2) と (2.11) より

$$\nvdash_i \neg\neg A \supset A$$

であることが解る．よって二重否定の除去は成り立たない．このことは容易に示すことができる．

ある有限のクリプキ・モデル $\mathcal{M} = \langle M, R, V \rangle$ が存在した場合，$a \in M, a \models \neg\neg A, a \not\models A$ が成り立っていると仮定する．$E = \{w | a \leq \forall w, w \in M\}$ とする．このときエンドワールドの可能世界の集合 $W' = \{w' | w' \nleq \exists w'', w', w'' \in M\}$ を考え，$W' \cap E = E'$ とする．$w \in E, w \leq \forall w$ が成り立つ．よって，$w \in E', \forall w \models \neg\neg A$．$E'$ の要素は全てエンドワールドであるので，$w \in E', \forall w \models A$ が成り立つ．したがって $\vdash_i A \supset \neg\neg A$ が成り立つ．また，, $a \not\models A$ より，$\nvdash_i \neg\neg A \supset A$

2.2 直観主義述語論理

今，述語論理における言語 $\mathcal{L}$ について考える．直観主義述語論理では，対象領域 U と言語 $\mathcal{L}$ に属する記号の解釈 I，そしてその解釈 I が依存する可能世界の集合 M によってフレームが作られる．そして，その 3 つ組のフレーム $\langle M, R, U\rangle$ は以下のように定義される．

- $\langle M, R\rangle$ は順序集合
- U は M から，空でないある集合 W のベキ集合への写像で以下の i,ii を満たす.

  i 任意の $w \in M$ に対し，$U(w) \neq \emptyset$

  ii 任意の $w, w' \in M$ に対し，$w \leq w'$ ならば $U(w) \subseteq U(w')$

このようなとき，集合 $U(w)$ を可能世界 w の対象領域という．また，あるフレーム $\langle M, R, U\rangle$ で言語 $\mathcal{L}$ に属するそれぞれの m 変数の述語 P の解釈 I を次のように定義する．

- $P^{I(w)} \subseteq U(w)^m$
- $w, w' \in M, w \leq w'$ であるならば，$P^{I(w)} \subseteq P^{I(w')}$

よって，I がフレーム $\langle M, R, U\rangle$ 上での解釈のばあい，$\langle M, R, U, I\rangle$ の 4 つ組を直観主義述語論理のクリプキ・モデルという．また，解釈 $I(w)$ による述語 P の対象領域を P のドメインと呼び，$P^{I(w)} = [\![P]\!]_w$ と表す．ちなみに，$[\![P]\!]$ とした場合，$[\![P]\!] = \bigcup_{w\in M}[\![P]\!]_w$ とする．

ここで，P を m 変数の述語としたばあい，$P, \neg P$ は論理式である．また A, B を論理式，x を対象変数としたとき，$A \vee B, A \wedge B, A \supset B, \forall x[A], \exists x[A]$ も論理式である．これらの意味論は以下のように定義される．

$$
\begin{array}{lcl}
w \models P(u_1, \ldots, u_m) & \Leftrightarrow & (u_1, \ldots, u_m) \in [\![P]\!]_w \\
w \models A \wedge B & \Leftrightarrow & w \models A \text{ かつ，} w \models B \text{ である} \\
w \models A \vee B & \Leftrightarrow & w \models A \text{ または，} w \models B \text{ である} \\
w \models A \supset B & \Leftrightarrow & w \in M \text{ において } w \leq w' \text{ となる全ての } w' \in M \text{ で } w' \not\models A \\
 & & \text{または，} w' \models B \text{ が成り立つ} \\
w \models \neg A & \Leftrightarrow & w' \in M \text{ において，} w \leq w' \text{ となる全ての } w' \in M \text{ で} \\
 & & w' \not\models A \text{ が成り立つ} \\
w \models \forall xA & \Leftrightarrow & w \leq \forall w' \text{ において } \forall u \in U(w'), w' \models A[\underline{u}/x] \text{ が成り立つ} \\
w \models \exists xA & \Leftrightarrow & \exists u \in U(w), w \models A[\underline{u}/x] \text{ が成り立つ}
\end{array}
$$

また，遺伝的な関係に関しては，クリプキ・モデル $\langle M, R, U, I\rangle$ において $w \in M$ に関し，言語 $\mathcal{L}$ に $U(w)$ の各要素に対する名前を新しい対象定数として加えた 言語 $\mathcal{L}[U(w)]$ を考える．ここで U_0 を $U_0 = \bigcup_{w \in M} U(w)$ とする．このとき，$\mathcal{L}[U_0]$ において閉じた論理式 A を考えたとき $E = \{w \in M \mid M, w \models A\}$ となる集合 E は遺伝的である．

第3章　オーダーソート論理

3.1　ソート階層の定義

ソート階層 $\mathcal{SH} = \langle S, \sqsubseteq \rangle$ は以下のように定義される．

- ソート記号の集合: $S = \{s_1, s_2, \ldots\}$
- サブソート関係: $\sqsubseteq$ $(\subseteq S \times S)$
- 最大ソート: $\top$
- 最小ソート: $\bot$

$s \in S$ のようなソート s を原子ソートという．誤解がない場合を除いてソートとよぶ．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．このように定義されたソートはそれぞれ対象領域を持ち，そのソートの対象領域を表す記号を $[\![\cdot]\!]$ とする．この対象領域はすなわちソートの解釈における対象の集合を表すこととなる．そこで，ソート s に対する対象領域 $[\![s]\!]$ をソート s におけるドメインと呼ぶ．このことに関しては後に詳しく意味を定義する．

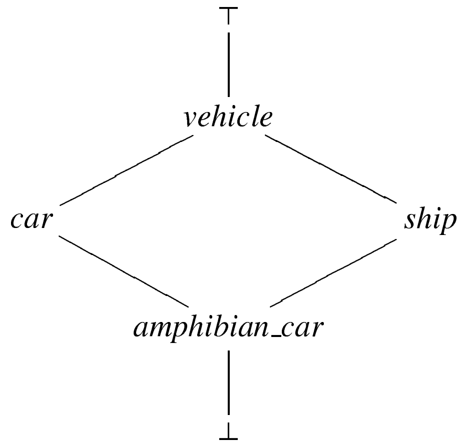
このドメインはソート全てに対して定義され，全てのソートのドメインを真部分集合として持つソートを最大ソート $\top$，また対象領域として空集合を持つソートを $\bot$ として，ソート順序の最大のものと最小のものが定義される．従って，ソートの順序関係は集合の包含関係と見す事ができ，

$$s_i \sqsubseteq s_j$$

とあった場合，これは

$$[\![s_i]\!] \subseteq [\![s_j]\!]$$

という関係を持つ．したがって，このサブソート関係 $\sqsubseteq$ は推移的であり，反射的でもある．これより，



$\mathcal{SH} = \langle S, \sqsubseteq \rangle$

$S = \{vehicle, car, ship, amphibian_{c}ar\}$

$\sqsubseteq = \{ship \sqsubseteq vehicle, car \sqsubseteq vehicle, car \sqsubseteq amphibian\_car, ship \sqsubseteq amphibian\_car\}$

図 3.1 ソート階層の例

- $s_i \sqsubseteq s_j, s_j \sqsubseteq s_k \Rightarrow s_i \sqsubseteq s_k$
- $s_i \sqsubseteq s_j, i = j \Rightarrow s_j \sqsubseteq s_i$

がなりたつ．ソート階層の例を図 3.1 で示す.

3.2 ソートと述語

ソートは先行研究 [7] によって単項述語として使用できることがわかっている．したがって，サブソート宣言 $s_i \sqsubseteq s_j$ は次の論理式の包含関係にも対応する.

$$s_i(x) \supset s_j(x)$$

これは $[\![s_i]\!] \subseteq [\![s_j]\!]$ となることからも明らかである．このように $s(x)$ は $x \in [\![s]\!]$ において成り立つ述語である．また，述語の項として出現する対象変数をソートは束縛することができ，上記のように $x \in [\![s]\!]$ であるような対象変数 x を

$$x : s$$

と記述する．このような対象変数をソート項と呼ぶ.

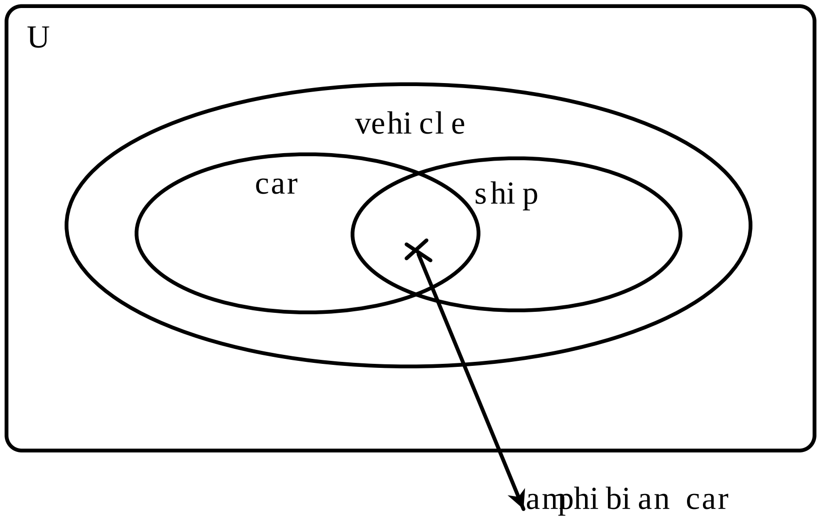


図 3.2 ソートのドメインの関係

3.3　構造的ソート

ソート階層を定義した場合，その順序関係から構造的なソートというものが定義される．それは，ソート階層の定義によって定義されたソート以外に，その順序関係から演繹されるソートのことである．それらには 2 つの種類があり，1 つはソート階層の順序関係における代数的なソートであり，もう 1 つは，意味論によって定義されるソートである．

代数的構造ソート

代数的な構造ソートを演算するに当たっては以下のに有るように，'⊓','⊔' という演算子を用いて表現され，それぞれ下限ソートと上限ソートと呼ぶ．

それぞれの定義は以下のようにする．

[代数的構造ソートの定義]
下限ソート：$s \sqcup s' \stackrel{def}{\equiv} \sup\{s, s'\}$
上限ソート：$s \sqcap s' \stackrel{def}{\equiv} \inf\{s, s'\}$

この場合， sup は上限，inf は下限を返す写像である．そして，この場合の全体の半順序集合は $\mathcal{SH} = \langle S, \sqsubseteq \rangle$ で与えられるソート階層における ⊑ の集合である．これにより代数的に複数のソートの上位概念や下位概念を取り出すことができる．ただしこうした場合，

ソート名のついていない対象が存在する．それらに関しては意味論的なソートによって表現される．

意味論的構造ソート

図 3.3 のソート階層の例を考える．

この例では，便宜上ソート名の横の '()' の内部に，そのソートのドメインを記述している．この図 3.3 の場合を考えると，s_1 のドメインと s_2 のドメインの論理積は $\{b, c\}$ であるが，$s_3 = \inf\{s_1, s_2\}$ と成っており，この s_3 は $[\![s_3]\!] = \{b\}$ である．したがって，$\{b, c\}$ にはソート名がないため，このままではオーダーソート論理内において取り扱うことが出来ない．このような状態は，ソート階層を構成する情報が完全でない場合に起こりうる状態である．つまり，現段階では，対象 c に関してこれ以上の情報が得られていない事が原因である．そこで以下のような意味論的構造ソートを取り扱う構造的ソート「和ソート」と「積ソート」を導入する．[1]

[意味論的構造ソートの意味論]

- 和ソート $[\![s \oplus s']\!] \stackrel{def}{\equiv} [\![s]\!] \cup [\![s']\!]$
- 積ソート $[\![s \otimes s']\!] \stackrel{def}{\equiv} [\![s]\!] \cap [\![s']\!]$

これにより，ソート名の無い集合 $\{b, c\}$ に対して $s_1 \otimes s_2$ というソート名を与えることが出来，これらもソート階層の一部として利用できる．また，上記のような定義によって，ソート階層内部での順序関係も同時に定義可能で，その定義は以下のようになる．

[意味論的構造ソートとソート階層の順序関係]

- $s \oplus s' \sqsupseteq s \sqcup s'$
- $s \otimes s' \sqsubseteq s \sqcap s'$

したがって代数的構造ソート，意味論的構造ソートの双方を考慮することによって，図 3.3 は，演繹から図 3.4 のような構造を持っていると考えることが出来る．

[1]先行研究 [12] における構造ソートでは代数的構造ソートは存在せず $\sqcap, \sqcup$ は，この意味論的構造ソートの意味であった．しかしながら，それでは直接的に下位概念を取り扱うことが出来ないため本稿ではこのように代数的構造ソートと意味論的構造ソートを導入し，それぞれについて定義を行った．したがって前項における代数的構造ソートは本書の拡張である．

$$
\begin{tikzcd}[column sep=small]
 & \top \arrow[d, dash] & \\
 & s_0(\llbracket s_0 \rrbracket = \{a, b, c, d, e\}) \arrow[dl, dash] \arrow[dr, dash] & \\
s_1(\llbracket s_1 \rrbracket = \{a, b, c\}) \arrow[dr, dash] & & s_2(\llbracket s_2 \rrbracket = \{b, c, d\}) \arrow[dl, dash] \\
 & s_3(\llbracket s_3 \rrbracket = \{b\}) \arrow[d, dash] & \\
 & \bot &
\end{tikzcd}
$$

図 3.3 対象を考えたソート階層の例

$$
\begin{tikzcd}[column sep=small]
 & \top \arrow[d, dash] & \\
 & s_0(\llbracket s_0 \rrbracket = \{a, b, c, d, e\}) \arrow[d, dash] & \\
 & s_1 \oplus s_2(\llbracket s_1 \oplus s_2 \rrbracket = \{a, b, c, d\}) \arrow[dl, dash] \arrow[dr, dash] & \\
s_1(\llbracket s_1 \rrbracket = \{a, b, c\}) \arrow[dr, dash] & & s_2(\llbracket s_2 \rrbracket = \{b, c, d\}) \arrow[dl, dash] \\
 & s_1 \otimes s_2(\llbracket s_1 \otimes s_2 \rrbracket = \{b, c\}) \arrow[d, dash] & \\
 & s_3(\llbracket s_3 \rrbracket = \{b\}) \arrow[d, dash] & \\
 & \bot &
\end{tikzcd}
$$

図 3.4 意味論的構造ソートを加えたソート階層の例

このようにして，代数的構造ソートはソート階層において語彙の概念の階層を抽象的に考えることができ，意味論的構造ソートは対象に関して具体的な語彙の関係を考えることができる．

3.4 ソート間関係

オーダーソート論理ではサブソート宣言以外にもソート間の関係が定義される．これは，先行研究の [12] らによって行われたもので，ソート階層を定義しているサブソート関係 $\sqsubseteq$，等号関係 '$=$'，以外に，全域的関係 '$|$'，排他的関係 '$||$' などが有る．これらは表 3.1 のように表される．

表 3.1 ソート間関係

$s \sqsubseteq s'$	サブソート関係
$s = s'$	等号関係
$s\|_{s''} s'$	全域的関係
$s\|\|s'$	排他的関係

サブソート関係はソート階層を構築する関係で，そのソートのドメインの包含関係を表す．また等号関係はそれぞれのソートのドメインが等しいことを表し，全域的関係はそのソートのドメインがサブソートによって完全に覆われていることを，排他的関係はソートのドメインの論理積が空集合であることなどを表す．以下にその意味論を記述する．

[ソート間関係の意味定義]

$$s \sqsubseteq s' \quad \stackrel{def}{\equiv} \quad [\![s]\!] \subseteq [\![s']\!]$$
$$s = s' \quad \stackrel{def}{\equiv} \quad [\![s]\!] = [\![s']\!]$$
$$s|_{s''} s' \quad \stackrel{def}{\equiv} \quad [\![s]\!] \cup [\![s']\!] = [\![s'']\!]$$
$$s||s' \quad \stackrel{def}{\equiv} \quad [\![s]\!] \cap [\![s']\!] = \emptyset$$

この意味論から，これらのソート間関係は以下のような略記であることが解る．

- $s = s' \stackrel{def}{\equiv} s \sqsubseteq s'$ かつ $s' \sqsubseteq s$
- $s|_{s''} s' \stackrel{def}{\equiv} s \sqsubseteq s'$ かつ $s' \sqsubseteq s$ かつ $s'' \sqsubseteq s \oplus s'$
- $s||s' \stackrel{def}{\equiv} s \otimes s'' = \bot$

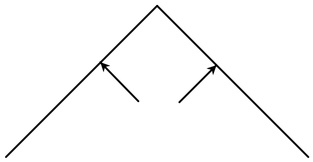


図 3.5 排他的関係

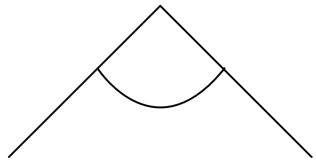


図 3.6 全域的関係

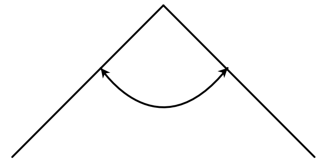


図 3.7 排他的関係・全域的関係

また，全域的，排他的と言うのは，ソート階層のハッセ図では図 3.5, 図 3.6, 図 3.7 のように表現される．

これらを一階述語で考えてみると，排他的関係とは $s \sqsubseteq s''$, $s' \sqsubseteq s''$ とあったときに $s \| s'$ と定義されていたならば，ある対象 a に関して，$s(a)$ が成り立つならば，$\neg s'(a)$ である，と言うことを表している．したがって，

$$s \| s' \Rightarrow \forall x[\neg(s(x) \wedge s'(x))]$$

が成り立つ．よって '$\|$' は交換律が成り立つ．

また全域的関係は $s \sqsubseteq s''$, $s' \sqsubseteq s''$ とあったときに $s|_{s''}s'$ と定義されており，ある対象 a に関して，$s''(a)$ が成り立ち，かつ $\neg s(a)$ つならば，$s'(a)$ である，と言うことを表している．したがって，

$$s|_{s''}s' \Rightarrow \forall x[((s(x) \vee s'(x)) \supset s''(x)) \wedge (s''(x) \supset (s(x) \vee s'(x)))]$$

$((s(x) \vee s'(x)) \supset s''(x))$ はソート階層の定義に含まれているので，したがって

$$s|_{s''}s' \Leftrightarrow \forall x[s''(x) \supset (s(x) \vee s'(x))]$$

が成り立つ．なお $s|_{s''}s' = s'|_{s''}s$ である．これはハッセ図で言うのであるならば，通常はそのサブソート関係よりあるソートが成り立てばそのソートより上の関係にあるソートは全て成り立つとして，上にたどって見ていくが，この場合下方向にたどれることを示している．ただし，サブソートが複数個ある場合は，どれをたどるか選択しなければ意味がない（論理和の関係から）．

表 3.2 オーダーソート論理の記号と一階述語論理の対応

名称	オーダーソート論理を含んだ論理式	一階述語での表現
ソート項	$P(x : s)$	$P(x) \wedge s(x)$
ソート項による全称	$\forall x : sP(x)$	$\forall x[s(x) \supset P(x)]$
ソート項による存在	$\exists x : sP(x)$	$\exists x[s(x) \wedge P(x)]$
サブソート宣言	$s \sqsubseteq s'$	$\forall x[s(x) \supset s'(x)]$
和ソート単項述語	$s \oplus s'(x)$	$s(x) \vee s'(x)$
積ソート単項述語	$s \otimes s'(x)$	$s(x) \wedge s'(x)$

3.5　ソートを用いた論理式

3.5.1　一階述語への変換

先に述べたように，ソート名は単項述語として利用可能なため，それぞれのオーダーソート論理における論理式は一階述語によって表現可能となる．そこで，x を対象変数，s をソート名，P を述語名として，変換の対応を表 3.2 のように定義する．なお，上限ソート，下限ソートに関しては，純粋にオーダーソート論理内でのみの代数演算子であるため対象を考慮する一階述語には変換できない．その理由は $s \sqcup s' = s''$ の場合，

$$[\![s]\!] \cup [\![s']\!] \subset [\![s'']\!]$$

となるため，

$$x \notin [\![s]\!] \cup [\![s']\!]$$

の場合に s, s' から s'' を評価できないからである．したがって，表 3.2 のような式を一階述語論理に変換する際に，式内に上限ソート，下限ソートが含まれていた場合，事前に処理し，別のソート名を当てはめる必要がある．

第4章　オーダーソート論理における否定表現のモデル

4.1　自然言語における否定表現

自然言語の語彙をそのままソート名として利用することは，オーダーソートを知識表現の手段として利用しようとした場合，ごく自然なことであり，且つ有用である．しかしながら，自然言語には様々な否定表現があり，これらはオーダーソート論理で知識を表現しようとした場合に問題になる．なぜならば，システム上は全てのソート名はただの文字列にすぎないため，何らかの否定的意味を表現したソート表現になっていなければ，矛盾を見知できないからである．まず自然言語の文には “This is not apple.” のような文法上の否定が存在する．これは，ある命題，もしくは述語として表現可能なことに対する否定表現としてとらえることができ，一般に，論理式の ‘¬’ で表現可能である．つまり，上記のような文は，便宜上 ‘This’ にある対象定数 a を割り当て，‘apple’ を $apple(x)$ のような述語として考えた場合，$\neg apple(a)$ として表現できるということである．このような文法上の否定表現（述語否定）を本稿では「弱い否定」という．このような否定を非古典論理の形で取り扱ったものとしては，[4] のようなものがある．

しかし，自然言語において他にも否定表現は多数存在する．本稿のオーダーソート論理のようにそのソート名に自然言語を使用した場合，その語彙項目そのものが否定表現を含む場合が存在する．それは，英語であるならば ‘un-’ や ‘il-’（日本語では「不ー」,「非ー」）などの接辞による否定と，‘soft’,‘hard’ のような否定表現とは語彙から判断できないがその意味が対立している場合の否定表現である．

このような場合，接辞の否定は，接辞を取り除いた語彙にその対象が含まれないことを主張しているので，これは一般に述語項の否定と言われる．つまり，その接辞の否定で表現される語彙によって，その対象は述語否定で表される対象領域のさらに内部においてある主張がなされていると考えることが出来るので，本稿ではこれを「強い否定」と呼ぶ．

接辞の否定は，接辞という否定項目が存在するため，その対立すべき述語対象が比較的

明らかであるが，もっとも表現しがたいのが，対立構造を持つ語彙の否定である．これは文字列だけでは対立するのかが判断できない，意味の上で対立している．したがって，文字列上からは対立の情報を得ることが出来ないため，これには人間がソート階層の定義の段階で何らかの情報を与えてやる必要がある．

4.1.1　文法上の否定（弱い否定）

弱い否定とは，前小節でも有るように，文法上の否定を取り扱う否定である．先行研究においてはこの否定を古典主義論理の否定として取り扱っている．本稿では，これを直観主義論理で見直す必要がある．

なぜ，弱い否定を直観主義論理で見直すかというのは，端的にはオーダーソート論理において表現する場合，その知識表現を正確にするに他ならない．例としてあげられるのが「柔らかくない訳ではない.」のような表現である．一見これは曖昧な表現であり，先行研究においてはこれは「柔らかい」を ‘$soft(x)$’ として表現した場合，まず述語論理で表現し $\neg\neg soft(x)$ となり，$soft(x)$ とされ，その対象 x は ソート $soft$ のドメインに含まれることになる．つまり，先行研究のオーダーソート論理では上記のような「柔らかくない訳ではない.」という対象は「柔らかい」という性質を持った対象であるといってることに他ならない．したがって，実際には曖昧に見える文は，より対象の性質を正確に表現しているといえる．そこで，二重否定を取り扱うことができ，且つ，その意味論を定義づけているクリプキ・モデルが人間の知識状態をうまく表現している，直観主義論理でこれを取り扱う．

この直観主義論理で上記の文を取り扱えば，弱い否定表現を持ったソートを正確に定義でき，なお，その順序関係についても定義出来る．

このような弱い否定を，先行研究ではソート階層内で補ソートと呼び，$\overline{soft}$ として表現していた．しかしながら，本稿でこれは補集合を意味するソートではないため，一重の弱い否定ソートと呼ぶ．また同様に，二重否定に関しても $\overline{\overline{soft}}$ のように記述し，二重の弱い否定ソートと呼ぶ．なお，それぞれのドメインは以下のように定義する．

- $[\![\overline{s}]\!] = [\![\neg s(x)]\!]$
- $[\![\overline{\overline{s}}]\!] = [\![\neg\neg s(x)]\!]$

4.1.2　接辞による否定（強い否定）

‘un-’,‘il-’,‘non-’,‘-less’ などの接辞による否定は文字列として以上の役割を持っている．それは，その接辞は，元の語彙の否定であるという主張を表す，記号として考えられるからである．例えば「これは無色です.」と言った場合，明らかにその接辞である「無ー」の反義語として考えられる「有色」を否定している．しかしながら，接辞による否定はそれだけではない．なぜならば,「これは有色ではない」と言うことと同義ではなく，さらに強い意味を有しているからである．例えば，この接辞の否定をただの否定表現と考えてみる．すなわち，文法上の否定と同じ取り扱いをすると言うことである．そのとき “All things which are numbers are neither colored nor noncolored.”([11]) という文を考えてみる．「ある対象が数字である」という述語を $N(x)$,「有色である」を $C(x)$ とする．そうするとすなわち，上記の文は論理式で

$$\forall x[N(x) \supset (\neg C(x) \wedge \neg\neg C(x))]$$

となる．これを古典主義論理で考えた場合,

$$\forall x[\neg N(x) \vee (\neg C(x) \wedge C(x))]$$

と変形できる．このとき $\neg C(x) \wedge C(x)$ は明らかに矛盾律なので式は $\forall x \neg N(x)$ となる．これは元の例文が「全ての数字は有色でないし，無色でもない」といっていることは「全てのものは数字ではない」と同値であるといっている．また直観主義論理で考えた場合では

$$\forall x[N(x) \supset (\neg C(x) \wedge \neg\neg C(x))]$$

となる．直観主義論理では A を論理式とした場合，$\neg A \wedge \neg\neg A$ は矛盾律であるためやはり古典主義論理の時と同様な答えになってしまう．これらのことからも，接辞による否定は文法上の否定表現とは同一ではなく，別の表現をする必要がある．そこで，本稿ではこのような接辞を ‘~’ で表し，強い否定と呼ぶ．したがって，上記の式は以下のように表記されることとなる．

$$\forall x[N(x) \supset (\neg C(x) \wedge \neg \sim C(x))]$$

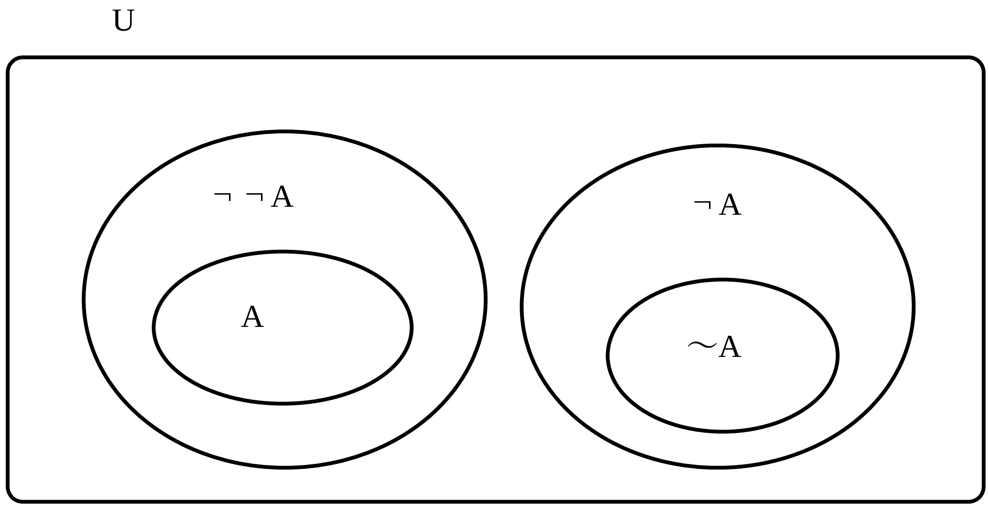


図 4.1 直観主義論理における対象領域

強い否定の領域

強い否定はそれがオペレータのように働くことから，元の述語との関係が深い．実際にその対象領域がどのようになっているのかを考えると図 4.1 のようになる．つまり,「無色（$\sim C(x)$）」がなり立てば，これは明らかに「有色である（$C(x)$）」ということと背反するので「有色ではない（$\neg C(x)$）」も成り立つ．したがって，この強い否定に関してはある論理式を A とした場合，以下の公理が成り立つ．

$$\vdash_i \sim A \supset \neg A$$

ただし，$\nvdash \neg A \supset \sim A$ である．

しかしながら，'~' は接辞の変換であり，一つの述語名の一部でもある．したがって，強い否定の意味論は以下のように定義される．

定義 2 $w \models \sim P(u_0, \ldots, u_n) \overset{def}{\equiv} (u_0, \ldots, u_n) \in [\![\sim P]\!]_w$

4.1.3　語彙の対立による否定

論理式としての対立

否定の一つの形として，対立というものがある．これは、ある語彙とある語彙が同じ対象において成り立つことがないことを表している．たとえば，形容詞であるならば「堅

い」と「柔らかい」などが考えられ，自動詞であるならば「開く」と「閉じる」などが考えられる．このような語彙を述語の述語名として利用した場合，論理式の構文上

$$hard(x) \wedge soft(x)$$

などとすることもできる．しかし，人は “This is hard and soft.” や “I open and close the cap on same time. ” などを見て，その語彙の意味的対立により，矛盾が存在することを理解することが可能である．しかし，それは意味論の話であり，構文上何ら矛盾を引き起こすものではない．従って，このような矛盾を構文上で表現するためには語彙と語彙の対立関係を他の方法で定義する必要があり，その情報を元にその矛盾を引き出す必要がある．

対立する関係を論理式で定義することは容易なことで，

$$w \models \forall x[\neg(hard(x) \wedge soft(x))]$$

と定義すれば，その可能世界ではそれらの語彙が対立関係を持つことになる．これらのことを考えた場合，この対立構造は述語の階層関係によって表現可能である．つまり，hard という述語と soft という述語はそれらの対象領域の積集合が空集合であればよいので，意味論では

$$[\![hard]\!] \cap [\![soft]\!] = \emptyset$$

と定義できる．

ソート階層における対立

ソート階層論理においては先行研究によって

$$hard \otimes soft = \bot$$

となればこれは，前小節の hard と soft が対立することを表現しているのと同義になる事が示されている．なぜならば，その述語の対象領域の論理積が空であればよいからである．しかし，概念的なものとしてもこのような構造は考えることができる．つまり，概念的対立とはそれぞれの語句の共通概念を知らない，と言う事である．これはすなわち代数的構造ソートである，下限，上限ソートによって考えることができる．したがって，

$$hard \sqcap soft = \bot$$

とできる．しかしながら，概念の対立とはこれだけでは不十分で，その対立する概念どうしがある概念内に含まれ，かつその中で対立概念となっていなければならない [6]．例えば ‘hard’ と ‘soft’ のような語彙であれば，人間はこれらが物の程度という概念に含まれることを知っているので，対立する概念としてとらえている．つまり，‘ship’ と言う語彙があっても，上記の $s \otimes s' = \bot$ という定義では対立するソートであるが，対立概念としては人間は考えない．‘ship’ は物の程度の事では無いからである．そこでこのような概念的対立を，$hard \asymp soft \overset{def}{\equiv} hard \otimes soft = \bot$ かつ $hard \sqcup soft \neq \top$ と定義する．つまり概念的な対立構造を持つためには $[\![hard \oplus soft]\!] \subseteq s$ となる s を知らなければならない．しかしながら，必ずしも s があるとは限らない．そこで，$hard \asymp soft$ は $[\![hard \oplus soft]\!] \subseteq s$ となるソート s が存在する，という意味を持たせる．したがってエンドワールドでは s が解ることになるので，

$$s \asymp_{s''} s' \overset{def}{\equiv} \overline{\overline{s \oplus s'}} \sqsubseteq \overline{\overline{s''}} \text{かつ } s \| s'$$

とする．よって，上記定義は

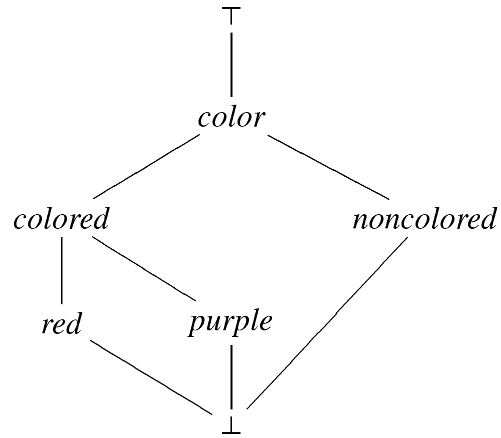
$$s \asymp_{s''} s' \overset{def}{\equiv} \overline{\overline{s}} \sqsubseteq \overline{\overline{s''}} \text{かつ} \overline{\overline{s'}} \sqsubseteq \overline{\overline{s''}} \text{かつ } s \| s'$$

とする．

4.2　ソート階層における否定表現

本節までに導入された否定表現をどのようにソート階層で表現するかを本節では定義する．またその影響によりどのような順序関係が拡張されるかについても，これを拡張する．現時点までに，拡張された否定表現は，弱い否定（一重，二重），強い否定，対立ソート，等が上げられる．これを順を追って定義していく．

このオーダーソート論理ではまず知識の表現として，基本的なソート階層の定義が必要である．そして本稿での拡張はこの定義を元にして，強い否定ソートの順序関係の一部を除き，これを拡張していくものである．



$\mathcal{SH} = \langle S, \sqsubseteq \rangle$

$S = \{color, colored, red, purple, noncolored\}$

$\sqsubseteq = \{colored \sqsubseteq color, noncolored \sqsubseteq color, red \sqsubseteq colored, purple \sqsubseteq colored\}$

図 4.2 ソート階層の定義例

4.2.1　否定によるソート階層の拡張

弱い否定ソート

弱い否定に関しては，前章までに以下のような対応からソートを拡張することを定義した．

$$w \models \neg s(x) \text{ iff } x \in [\![\overline{s}]\!]_w$$

$$w \models \neg\neg s(x) \text{ iff } x \in [\![\overline{\overline{s}}]\!]_w$$

これらをソート階層の中でどのように表現するのかを考えると，まずオーダーソート論理のサブソート関係を考えると以下のような定義が成り立つことが解っている．

$$s \sqsubseteq s' \Rightarrow s(x) \supset s'(x)$$

したがって，この式の対偶をとると，明らかに

$$s(x) \supset s'(x) \equiv \neg s'(x) \supset \neg s(x)$$

が成り立つことが解る．また $[\![\overline{s}]\!]_w = \{x | w \models \neg s(x)\}$ と表現できることから，

$$\neg s'(x) \supset \neg s(x) \equiv \overline{s'} \sqsubseteq \overline{s}$$

とできる．このことから，一重の弱い否定のソートは元のソート階層の順序を入れ替えたものがその一重の弱い否定のソートの順序関係になることが解る．またこのように対偶をとることは以下のように体系 LJ において証明可能なため直観主義論理上でも成り立つことが解る．

$$\dfrac{\dfrac{\dfrac{A \to A \quad \dfrac{B \to B}{\neg B, B \to}\ (\neg \to)}{A, \neg B, A \supset B \to}\ (\supset\to)}{\neg B, A \supset B \to \neg A}\ (\to \neg)}{A \supset B \to \neg B \supset \neg A}\ (\to\supset)$$

これより，一重の弱い否定ソートを拡張したオーダーソートは図 4.3 のようになる．

弱い否定ソート（二重）

本稿では弱い否定ソートを直観主義論理で見直すため，その弱い否定の形は二重否定の形も存在する．したがって，弱い否定ソートは二重の弱い否定ソートもオーダーソート論理において考慮しなければならない．その際，前章で導入したように，$\neg\neg s(x)$ のソートは $\overline{\overline{s}}$ として表現することにした．直観主義論理では

$$\vdash_i A \supset \neg\neg A$$

が成り立つ．したがって，オーダーソート論理においては，ソート s があった場合，

$$s \sqsubseteq \overline{\overline{s}}$$

が成り立つ．また，$s \sqsubseteq s'$ とあった場合，$\forall x[s(x) \supset s'(x)]$ とでき，$\vdash_i A \supset \neg\neg A$ から

$$\forall x[s(x) \supset s'(x)] \vdash_i \forall x[\neg\neg s(x) \supset \neg\neg s'(x)]$$

が成り立つ．よって，$\mathcal{SH} = \langle S, \sqsubseteq \rangle$ の原子ソートの定義から以下の関係が導かれる．

- $s \in S \Rightarrow s \sqsubseteq \overline{\overline{s}}$
- $s \sqsubseteq s' \Rightarrow \overline{\overline{s}} \sqsubseteq \overline{\overline{s'}}$

したがって，このことから，ソート階層は図 4.4 のように拡張される．

さらに，二重否定に関してはこのような演繹のみの拡張では完全ではない．図 4.5 を考

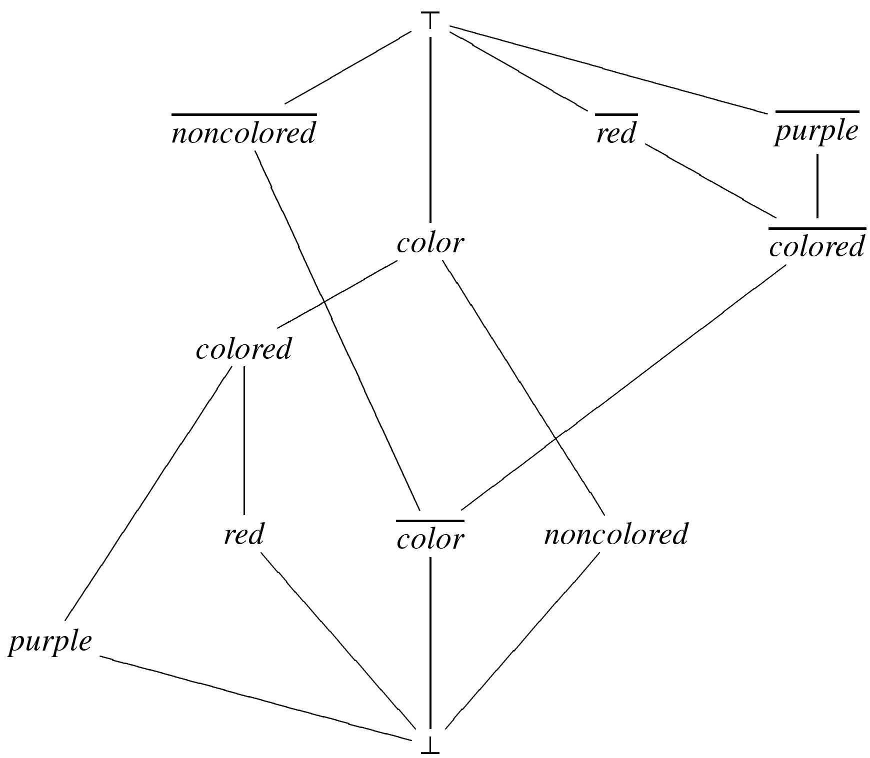


図 4.3 ソート階層の拡張（一重の弱い否定ソート）

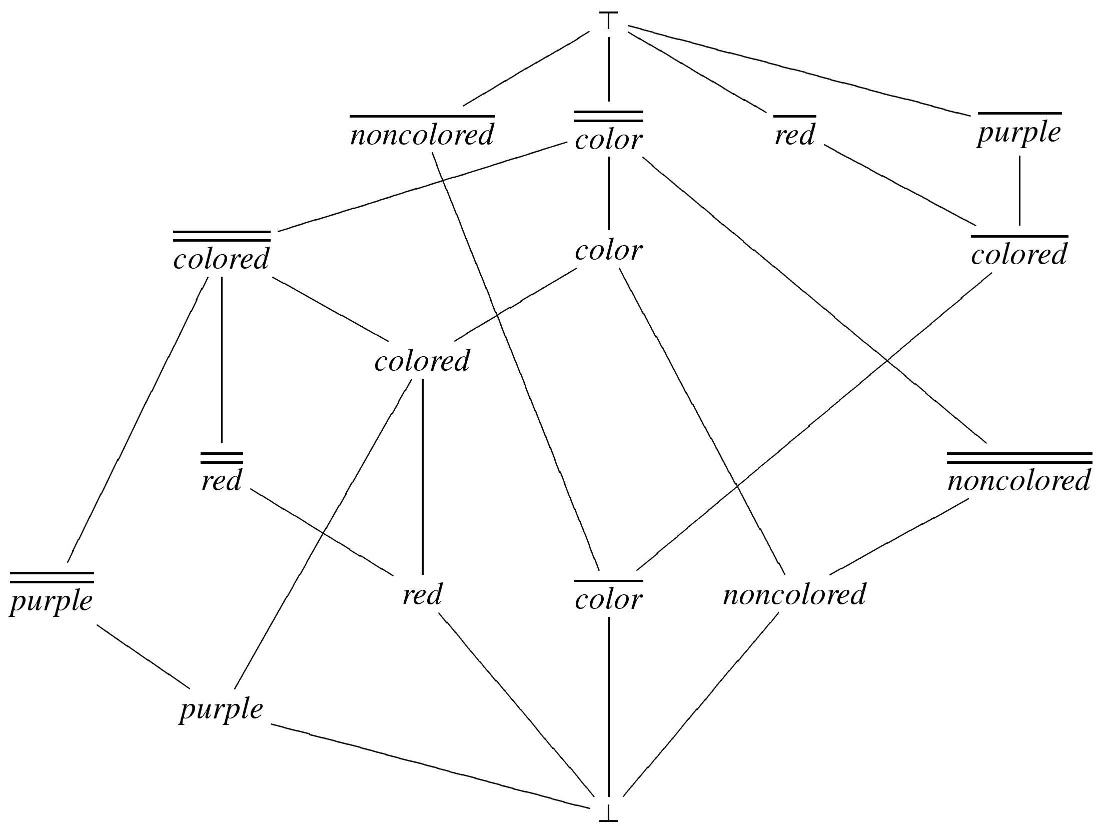
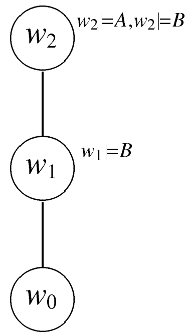


図 4.4 ソート階層の拡張（二重の弱い否定ソート例 1）



$\mathcal{M} = \langle M, R, V \rangle$

$M = \{w_0, w_1, w_2\}$

$R = \{w_0 \leq w_1, w_1 \leq w_2\}$

$V = \{V(A) = \{w_2\}, V(B) = \{w_1, w_2\}\}$

図 4.5 二重否定含意の例

える．このモデルでは，$w_0 \models \neg\neg B$ が成り立つ．また，$w_0 \models \neg\neg A$ も成り立つ．したがって，

$$w_0 \models \neg\neg A \supset \neg\neg B$$

$$w_0 \models \neg\neg B \supset \neg\neg A$$

が成り立つ．また，$w_1 \models B, w_2 \models A$ より，

$$w_0 \models A \supset B$$

も成り立つ．しかしながら，$w_1 \not\models A$ であるため，

$$w_0 \not\models B \supset A$$

となってしまう．つまり，オーダーソート論理において，$s \sqsubseteq s'$ とあった場合，$\overline{\overline{s}} \sqsubseteq \overline{\overline{s'}}$ が成り立つが，$s \not\sqsubseteq s'$ である場合でも $\overline{\overline{s}} \sqsubseteq \overline{\overline{s'}}$ が成り立つ場合が存在する．これは，既存のソート階層の定義からだけでは演繹でだすことができないため，オーダーソート論理を拡張する必要がある．そこで，このように，$s \not\sqsubseteq s'$ でありながら，$\overline{\overline{s}} \sqsubseteq \overline{\overline{s'}}$ が成り立つようなサブソート関係を「弱いサブソート関係」として記号 ‘$\overline{\overline{\sqsubset}}$’ を導入する．そして，原子ソートのソート階層の定義に ‘$\overline{\overline{\sqsubset}}$’ 含んだソート階層の定義を導入したオーダーソート論理を $\mathcal{SH}^+ = \langle S, \sqsubseteq, \sqsubseteq^+ \rangle$ とする．$s \overline{\overline{\sqsubset}} s' \in \sqsubseteq^+$ である．そこで，図 4.4 において,「紫は赤でないわけではない」という表現を許した場合，それを上記の論理でオーダーソート論理を拡張すると図 4.6 のようなソート階層になる．

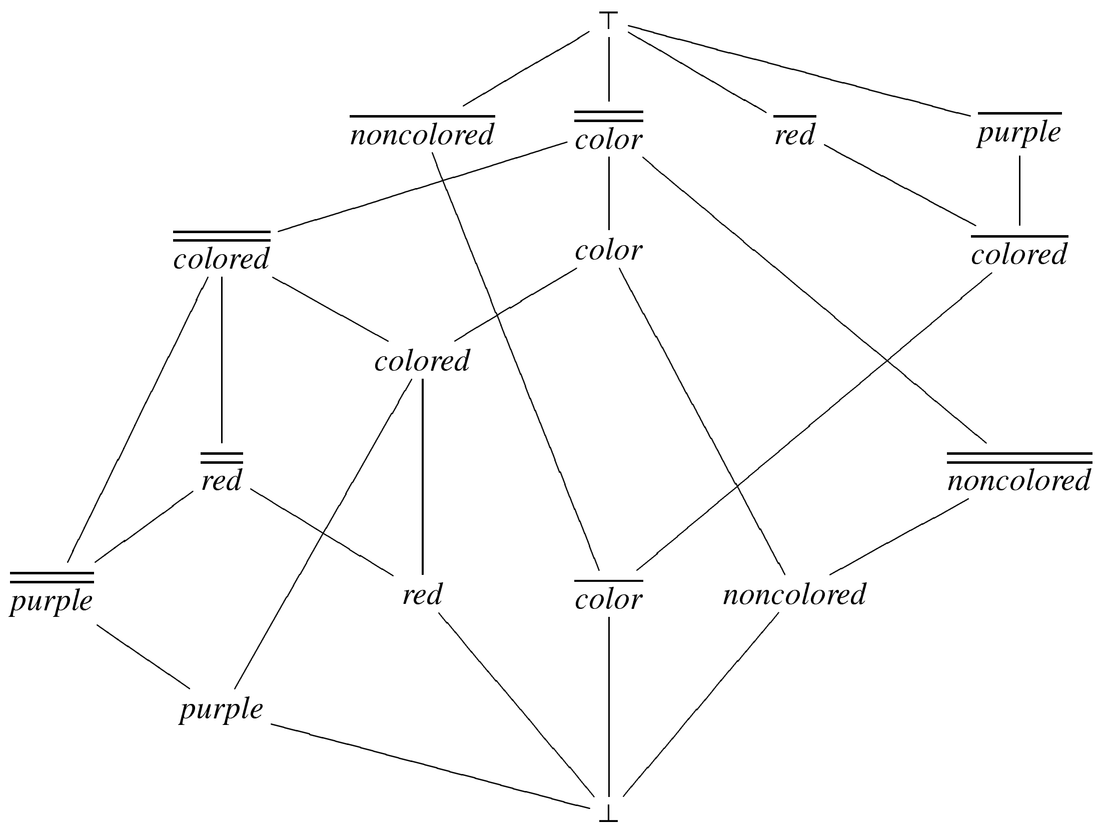


図 4.6 ソート階層の拡張（二重の弱い否定ソート例 2）

強い否定ソート

強い否定ソートは，自然言語の語彙に関して否定表現の接辞を '~' に変換し，それをソート名としてオーダーソート論理で利用したものである．したがって，'noncolored' と言う語彙をソート内で表現した場合，そのソート階層では '~ *colored*' というソート名を使用する．また，強い否定は弱い否定との間には

$$\vdash_i \sim A \supset \neg A$$

したがって，

$$\vdash_i \neg\neg A \supset \neg \sim A$$

$$\vdash_i \neg\neg \sim A \supset \neg A$$

という関係が演繹できるため，$\sim$ *colored* があった場合

$$\overline{\overline{colored}} \sqsubseteq \overline{\sim colored}$$

$$\overline{\overline{\sim colored}} \sqsubseteq \overline{colored}$$

と言うサブソート関係が演繹できる．そして，強い否定はその接辞を除いた語彙と同じ概念の下位概念である．したがって $s \sqsubseteq s'$ であれば $\sim s \sqsubseteq s'$ である．ただし $\sim s$ そのものは 'noncolored' のように語彙なので，ソートの集合の中に定義されていなければそのソート階層で出現することは無い．したがって，強い否定ソートを導入したオーダーソートは図 4.7 のように拡張される．

対立による否定ソート

対立に関する否定表現は，前章で記述したように，その語彙項目（ソート名）からは判断ができない．したがって，そのソート階層を定義する時点で何らかの情報が無ければその対立構造を定義できない．そこで，前章でその語彙項目が概念的対立構造を持つことを '$\asymp$' で表現した．

- $s \asymp_{s''} s' \equiv \overline{\overline{s}} \sqsubseteq \overline{\overline{s''}}$かつ$\overline{\overline{s'}} \sqsubseteq \overline{\overline{s''}}$かつ $s \| s'$
- $s \| s' \equiv [\![s \otimes s']\!] = \emptyset$

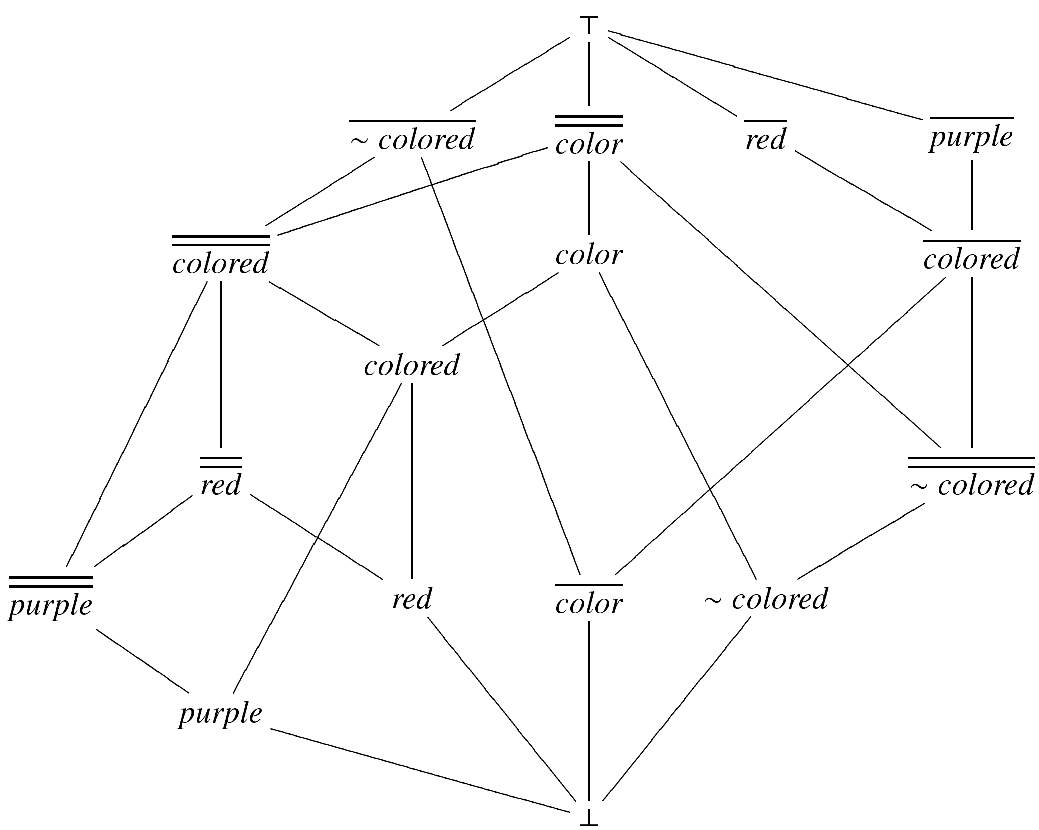


図 4.7 ソート階層の拡張（弱い否定ソートと強い否定ソート）

したがって,

$$s \asymp s' \vdash s \| s'$$

となる．そこで，'$\sim s$' と 's' というソートを述語で考えると，これは

$$\vdash_i \sim s(x) \supset \neg s(x)$$

と定義したため，'$\sim s$' と 's' は $\sim s \| s$ の関係にあり，したがって，対立語彙によって拡張されるサブソート関係は,

$$\overline{\overline{s}} \sqsubseteq \overline{\sim s}$$

$$\overline{\overline{\sim s'}} \sqsubseteq \overline{s}$$

という関係が拡張されることとなる．

このようにして，$\mathcal{SH} = \langle S, \sqsubseteq \rangle$ のソート間関係に $\overline{\overline{\sqsubset}}, \|, |, \asymp,$, ソートに $\sim s, \overline{s}, \overline{\overline{s}}, s \oplus s', s \otimes s', s \sqcup s', s \sqcap s'$(ただしユーザは定義しない) を加えたものを拡張ソート階層 $\mathcal{SH}^{+} = \langle S^{+}, \sqsubseteq^{+} \rangle$ とすると,

$\mathcal{SH}^{+} = \langle S, \sqsubseteq^{+}, \rangle$

$S = \{color, colored, red, purple, noncolored\}$

$\sqsubseteq^{+} = \{colored \sqsubseteq color, noncolored \sqsubseteq color,$

$red \sqsubseteq colored, purple \sqsubseteq colored, purple \overline{\overline{\sqsubset}} red, colored \| noncolored\}$

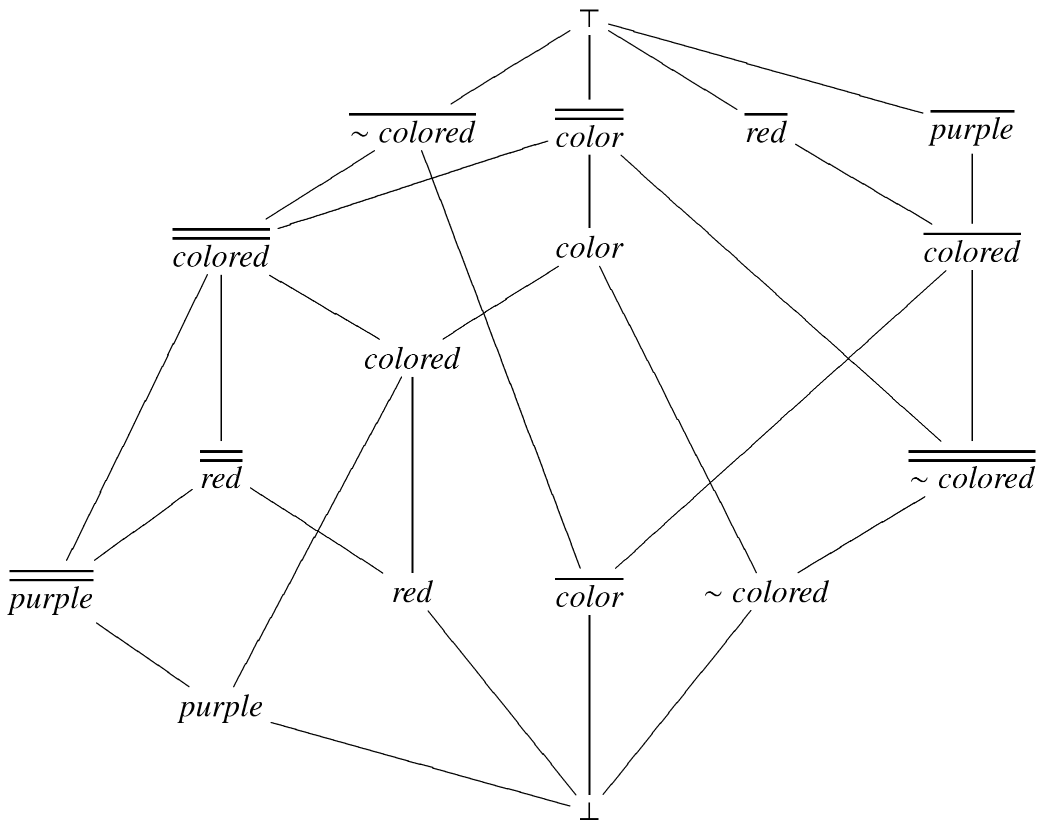
と定義され，これは自動的に,

$\mathcal{SH}^{++} = \langle S^{++}, \sqsubseteq^{++} \rangle$

$S^{++} = \{color,\ colored,\ \sim colored,\ red, purple,\ \overline{color},\ \overline{colored},\ \overline{\sim colored}, \overline{red},\ \overline{purple},$
$\overline{\overline{color}},\ \overline{\overline{colored}},\ \overline{\overline{\sim colored}},\ \overline{\overline{red}},\ \overline{\overline{purple}}\}$

$\sqsubseteq^{++} = \{\sim colored \sqsubseteq color, red \sqsubseteq color, purple \sqsubseteq color,\ color \sqsubseteq \overline{\overline{color}},$
$\sim colored \sqsubseteq \overline{\overline{\sim colored}},\ red \sqsubseteq \overline{\overline{red}},\ purple \sqsubseteq \overline{\overline{purple}}, \overline{\overline{\sim colored}} \sqsubseteq \overline{\overline{color}},$
$\overline{\overline{red}} \sqsubseteq \overline{\overline{color}},\ \overline{\overline{purple}} \sqsubseteq \overline{\overline{color}},\ \overline{color} \sqsubseteq \overline{\sim colored},\ \overline{color} \sqsubseteq \overline{red},\ \overline{color} \sqsubseteq \overline{purple},$
$\overline{\overline{colored}} \sqsubseteq \overline{\sim colored},\ \overline{\overline{\sim colored}} \sqsubseteq \overline{colored},\ colored \| \sim colored\}$

という $\mathcal{SH}^{++} = \langle S^{++}, \sqsubseteq^{++} \rangle$ に拡張される．



$\mathcal{S}^{+} = \langle S, R, S^{+}, R^{+} \rangle$

$S = \{color, \sim colored, red, black\}$

$R = \{\sim colored \sqsubseteq color, red \sqsubseteq color, black \sqsubseteq color\}$

$S^{+} = \{\overline{color}, \overline{\sim colored}, \overline{red}, \overline{\overline{color}}, \overline{\overline{\sim colored}}, \overline{\overline{red}}\}$

$R^{+} = \{color \sqsubseteq \overline{\overline{color}}, \sim colored \sqsubseteq \overline{\overline{\sim colored}}, red \sqsubseteq \overline{\overline{red}}, red \overline{\overline{\sqsubset}} \sim colored, red \asymp black\}$

図 4.8 ソート階層の全ての拡張

第5章　拡張オーダーソート論理の形式化

5.1　構文

$S = \{s, s', s_0, \ldots\}$ は原子ソートの集合である．拡張ソートの集合 S^{++} は以下を満たす最小の集合である．

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

F は関数記号 $(f, f', f_0, \ldots)$ の集合，C は定数記号 $(c, c', c_0, \ldots)$ の集合，P は述語記号 $(p, p', p_0, \ldots)$ の集合である．さらに，全てのソート記号 s に対して，単項のソート述語 $p_s \in P$ が存在する．このような単項のソート述語 p_s は s で表され，$p_s(x)$ は $s(x)$ のようになる．

シグネチャΣ

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

シグネチャΣを含む言語 $\mathcal{L}$ は以下のアルファベットを含む．

- V_s はソート s の変数 $(x : s, x : s', x : s_0, \ldots)$ の集合であり，V は全てのソートに対する変数の集合 $\bigcup_{s \in S^{++}} V_s$ の集合である．
- $\neg, \vee, \wedge, \supset$ は論理結合子，$\forall, \exists$ は量化記号である．
- '(', ')', ',' は補助記号である．

言語 $\mathcal{L}$ の項は以下のように帰納的に定義される．まず，オーダーソート論理の論理式を以下のように帰納的に定義する．

- s が $s \in S^{++}$ である場合，s はオーダーソート論理の論理式である．

- S, S' がオーダーソート論理の論理式である場合，$(S \sqcap S')$, $(S \sqcup S')$, $(S \otimes S')$, $(S \oplus S')$, $(\overline{S})$, $(\overline{\overline{S}})$ もまたオーダーソート論理の論理式である．

そのとき，

- 変数 $x : s$ はソート s の項である．s はオーダーソート論理の論理式である．
- 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．
- $t_0, t_1, \ldots, t_n$ がソート $s_0, s_1, \ldots, s_n$ の項であり，$f \in F$ かつ $f : s_0, s_1, \ldots, s_n \to s \in Dec$ のとき，$f(t_0, t_1, \ldots, t_n)$ はソート s の項である．

言語 $\mathcal{L}$ のオーダーソート論理を含む論理式は以下のように帰納的に定義される．

- $t_0, t_1, \ldots, t_n$ がソート $s_0, s_1, \ldots, s_n$ の項であり，$p \in P$ かつ $p : s_0, s_1, \ldots, s_n \in Dec$ のとき，$p(t_0, t_1, \ldots, t_n)$ はオーダーソート論理を含む原子論理式である．また，オーダーソート論理を含む原子論理式はオーダーソート論理を含む論理式である．
- A, B がオーダーソート論理を含む論理式ならば，$\neg A$, $A \vee B$, $A \wedge B$, $A \supset B$, $\forall x : sA$, $\exists x : sA$ もオーダーソート論理を含む論理式である．

このようなオーダーソート論理を含む論理式の集合を $\mathcal{SF}$ で表す．

$s, s' \in S^{++}$ に対して，$s \sqsubseteq s'$ をサブソート宣言という．また，$s, s' \in S$ に対して $s \overline{\overline{\sqsubset}} s'$ を弱いサブソート宣言という．$s \sqsubseteq s'$ はこのような s が s' のサブソートであることを表し，$s \overline{\overline{\sqsubset}} s'$ は $\overline{\overline{s}}$ が $\overline{\overline{s'}}$ のサブソートであることを表す．したがって，$s \overline{\overline{\sqsubset}} s'$ は $\overline{\overline{s}} \sqsubseteq \overline{\overline{s'}}$ の略記である．したがって，サブソート宣言の集合を $\mathcal{D} = \{s \sqsubseteq s' | s, s' \in S^{++}\}$ で表す．

そして，言語 $\mathcal{L}$ の一階述語論理式は以下のように帰納的に定義される．

- P を m 変数の述語としたばあい，$P, \neg P$ は論理式である．
- A, B を論理式，x を対象変数としたとき，$A \vee B, A \wedge B, A \supset B, \forall x[A], \exists x[A]$ も論理式である．

このような集合を $\mathcal{F}$ とする．そして，$\mathcal{SF}$ にたいして，

$$\phi : \mathcal{SF} \longrightarrow \mathcal{F}$$

のような写像 ϕ をオーダーソート論理の式の一階述語変換という．

5.2　意味論

　言語 $\mathcal{L}$ の意味づけを行う．クリプキ・モデル $\langle M, R, U, I \rangle$ が与えられたとき，意味論は以下のようになる．$I(w)$ は可能世界 w における解釈であり，$U(w)$ は可能世界 w の対象領域である．また P_s はソートの単項述語記号であり，s, s', s'' はオーダーソート論理の式である．

- $w \models p_s(u) \Leftrightarrow u \in p_s^{I(w)}$
- $w \models \sim P_s(u) \Leftrightarrow u \in [\![\sim P]\!]_w$ かつ $w \models \overline{s}(u)$
- $w \models s = s' \Leftrightarrow w \models s \sqsubseteq s'$ かつ $w \models s' \sqsubseteq s$
- $w \models s \otimes s'(x) \Leftrightarrow w \models s(x)$ かつ $w \models s'(x)$
- $w \models s \oplus s'(x) \Leftrightarrow w \models s(x)$ または $w \models s'(x)$
- $w \models \overline{s}(x) \Leftrightarrow w < w', \forall w' \not\models s(x)$
- $w \models \overline{\overline{s}}(x) \Leftrightarrow w < w', \forall w' \not\models \overline{s}(x)$
- $w \models s \sqsubseteq s' \Leftrightarrow w \models \forall x[s(x) \supset s'(x)]$
- $w \models s \overline{\overline{\sqsubset}} s' \Leftrightarrow w \models \overline{\overline{s}} \sqsubseteq \overline{\overline{s'}}$
- $w \models s \| s' \Leftrightarrow w \not\models \forall x s(x)$ かつ $w \not\models \forall x s'(x)$
- $w \models s \sqcap s'(x) \Leftrightarrow w \models s'' \sqsubseteq s, w \models s'' \sqsubseteq s', w \models s''(x)$
- $w \models s \sqcup s'(x) \Leftrightarrow w \models s \sqsubseteq s'', w \models s' \sqsubseteq s'', w \models s''(x)$
- $w \models s \asymp_{s''} s'(x) \Leftrightarrow w \models \overline{\overline{s}} \sqsubseteq \overline{\overline{s''}}$かつ $w \models \overline{\overline{s'}} \sqsubseteq \overline{\overline{s''}}$かつ $w \models s \| s'$

また，$A,$ を B 論理式，P を述語記号とした場合以下のように定義される．

- $w \models P(u_1, \ldots, u_m) \Leftrightarrow (u_1, \ldots, u_m) \in P^{I(w)}$
- $w \models A \wedge B \Leftrightarrow w \models A$ かつ，$w \models B$ である
- $w \models A \vee B \Leftrightarrow w \models A$ または，$w \models B$ である

- $w \models A \supset B \Leftrightarrow w \in M$ において $w \leq w'$ となる全ての $w' \in M$ で $w' \not\models A$ または，$w' \models B$ が成り立つ

- $w \models \neg A \Leftrightarrow w' \in M$ において，$w \leq w'$ となる全ての $w' \in M$ で $w' \not\models A$ が成り立つ

- $w \models \forall xA \Leftrightarrow w \leq \forall w'$ において $\forall u \in U(w'), w' \models A[\underline{u}/x]$ が成り立つ

- $w \models \exists xA \Leftrightarrow \exists u \in U(w), w \models A[\underline{u}/x]$ が成り立つ

したがって，解釈関数 $[\![\cdot]\!]$ は以下のように定義される．

$$[\![s]\!]_w = \{x|w \models s(x)\}$$

特に，$[\![s]\!]$ の時は，$[\![s]\!] = \{x|w \in M, \forall w \models s(x)\}$ である．

5.3 推論系

5.3.1 推論規則

直観主義論理では排中律が認められていないために反駁推論を行うことができない．従って，逆向き推論を行って論理式が成り立つか確かめる．その逆向き推論を行う際に使用する推論規則としてはヒルベルト流体系の推論規則は適していないので，今回直観主義論理でシーケント計算が使用できる体系 LJ にオーダーソート論理に対応する推論規則を提案した．それら提案した推論規則は以下のようなものである．

推論規則

$$\frac{\Pi \to S \quad S', \Gamma \to \Delta}{S \sqsubseteq S', \Gamma, \Pi \to \Delta}\ (\sqsubseteq\to)$$

$$\frac{S\Pi \to S'}{\Pi \to S \sqsubseteq S'}\ (\to\sqsubseteq)$$

$$\frac{\Pi \to S \otimes S'}{S \parallel S', \Pi \to}\ (\parallel\to)$$

$$\frac{S \otimes S', \Pi \to}{\Pi \to S \parallel S'}\ (\to\parallel)$$

$$\frac{S'', S \sqcup S', \Pi \to \Delta}{S \mid_{S''} S', \Pi, \Gamma \to \Delta}\ (|\to)$$

$$\frac{S'', \overline{S}, \Pi \to S' \quad S'', \overline{S'}, \Pi \to S}{\Pi \to S \mid_{S''} S'}\ (\to|)$$

$$\frac{\Pi, S \to \Delta \quad \Pi, S' \to \Delta}{S \oplus S', \Pi \to \Delta}\ (\oplus \to)$$

$$\frac{\Pi \to S}{\Pi \to S \oplus S'}\ (\to \oplus 1)$$

$$\frac{\Pi \to S'}{\Pi \to S \oplus S'} \ (\to \oplus 2)$$

$$\frac{S, \Pi \to \Delta}{S \otimes S', \Pi \to \Delta} \ (\otimes \to 1)$$

$$\frac{S', \Pi \to \Delta}{S \otimes S', \Pi \to \Delta} \ (\otimes \to 2)$$

$$\frac{\Pi \to S \quad \Pi \to S'}{\Pi \to S \otimes S'} \ (\to \otimes)$$

$$\frac{S, \Pi \to}{\Pi \to \overline{S}} \ (\to \text{-})$$

$$\frac{\Pi \to S}{\overline{S}, \Pi \to} \ (\text{-} \to)$$

$$\frac{\overline{S}, \Pi \to \Delta}{\sim S, \Pi \to \Delta} \ (\sim\to)$$

$$\frac{S \sqsubseteq S'', S' \sqsubseteq S'', S'', \Pi \to \Delta}{S \sqsubseteq S'', S' \sqsubseteq S'', S \sqcup S', \Pi \to \Delta} \ (\sqcup \to)$$

$$\frac{S \sqsubseteq S'', S' \sqsubseteq S'', \Pi \to S''}{S \sqsubseteq S'', S' \sqsubseteq S'', \Pi \to S \sqcup S'} \ (\sqcup \to)$$

$$\frac{S'' \sqsubseteq S, S'' \sqsubseteq S', S'', \Pi \to \Delta}{S'' \sqsubseteq S, S'' \sqsubseteq S', S \sqcap S', \Pi \to \Delta} \ (\sqcap \to)$$

$$\frac{S'' \sqsubseteq S', S'' \sqsubseteq S', \Pi \to S''}{S'' \sqsubseteq S, S'' \sqsubseteq S', \Pi \to S \sqcap S'} \ (\sqcap \to)$$

$$\frac{S \| S', S \sqcup S', \Pi \to \Delta}{S \asymp S', \Pi \to \Delta} \ (\asymp\to)$$

推論手順について

推論の目的は 2 つあり，1 つは論理式の評価，もう 1 つはオーダーソート論理の式の評価である．つまり，論理式の評価とはオーダーソートの定義を補助の知識として利用し，そこから対象に関して推論する方法であり，オーダーソート論理の式の評価とは概念階層を評価することである．その場合，論理式の評価では，ϕ はソート階層の定義からオーダーソート論理の式を一階述語に変換する写像であり，A_i, B_j をオーダーソート論理を含む論理式とすると，

$$(1) \cdots \phi(SH^{++}), \phi(A_0), \ldots, \phi(A_n) \to \phi(B_0), \ldots, \phi(B_m)$$

のようなシーケントを用いて計算できる．なお SH^{++} はソート階層の定義から得られる，サブソート関係と拡張された構造的ソート間関係のオーダーソート論理の論理式の略記である．これより，このようなシーケントは一階述語のシーケントによって証明可能である．また，オーダーソート論理の式の評価では，$S_0, \ldots, S_m$ をオーダーソート論理の式と

して

$$(2)\cdots SH^{++} \rightarrow S_0, \ldots, S_m$$

を用いて評価する．この際に，上記の推論規則を適応する．これによって，(1) では

$$\phi(SH^{++}), apple(a), \phi(\forall x : human\ eat(x : food)) \rightarrow eat(a)$$

のような式を評価し，(2) では

$$apple \sqsubseteq fruit, banana \sqsubseteq fruit \rightarrow \overline{fruit} \sqsubseteq \overline{\overline{apple}} \sqcap \overline{banana}$$

のような概念的推論が行える．

LJ のような推論系は証明を抑制する（証明をできなくする）ような対立構造の定義を活用することが難しい．つまり，シーケントの前提部に矛盾が含まれていた場合，これは証明可能になってしまうので，シーケントは矛盾を含まないようにしなければならない．しかしながら，前提部が無矛盾であるかどうかは，保証できない．なぜなら，実際にユーザが入力する情報というのは SH^{++} に関するシーケントの要素以外の部分だからである．そこで，本稿における推論は (1),(2) の様な式を評価する際に，

$$\begin{aligned}\phi(SH^{++}), \phi(A_0), \ldots, \phi(A_n) \rightarrow \\ SH^{++} \rightarrow\end{aligned}$$

このシーケントが証明可能かどうかを検査する必要がある．これが，証明可能であるならば前提部が矛盾を含んでいる．この過程において，対立構造の定義が意味をなす．この過程がなければ，対立構造の定義は Weakning で削除されてしまうので意味をなさないのである．

なお，'$\rightarrow\asymp$' に関しては，ソート階層の定義に依存する．したがって，成り立つ場合は SH^{++} に含まれる．

第6章　拡張推論システムの概要

6.1　処理手順

　システムはソート階層に対する検査プログラムと，前提に対して結論がソート階層を用いて正しいかどうかを推論するシーケント推論プログラム，またモデルを具体的に与えた場合の，式の評価プログラムからなる．処理の概要は以下のようなものである．なおプログラムは [1, 8, 9] を参考に，Prolog 言語を使用して作成した．

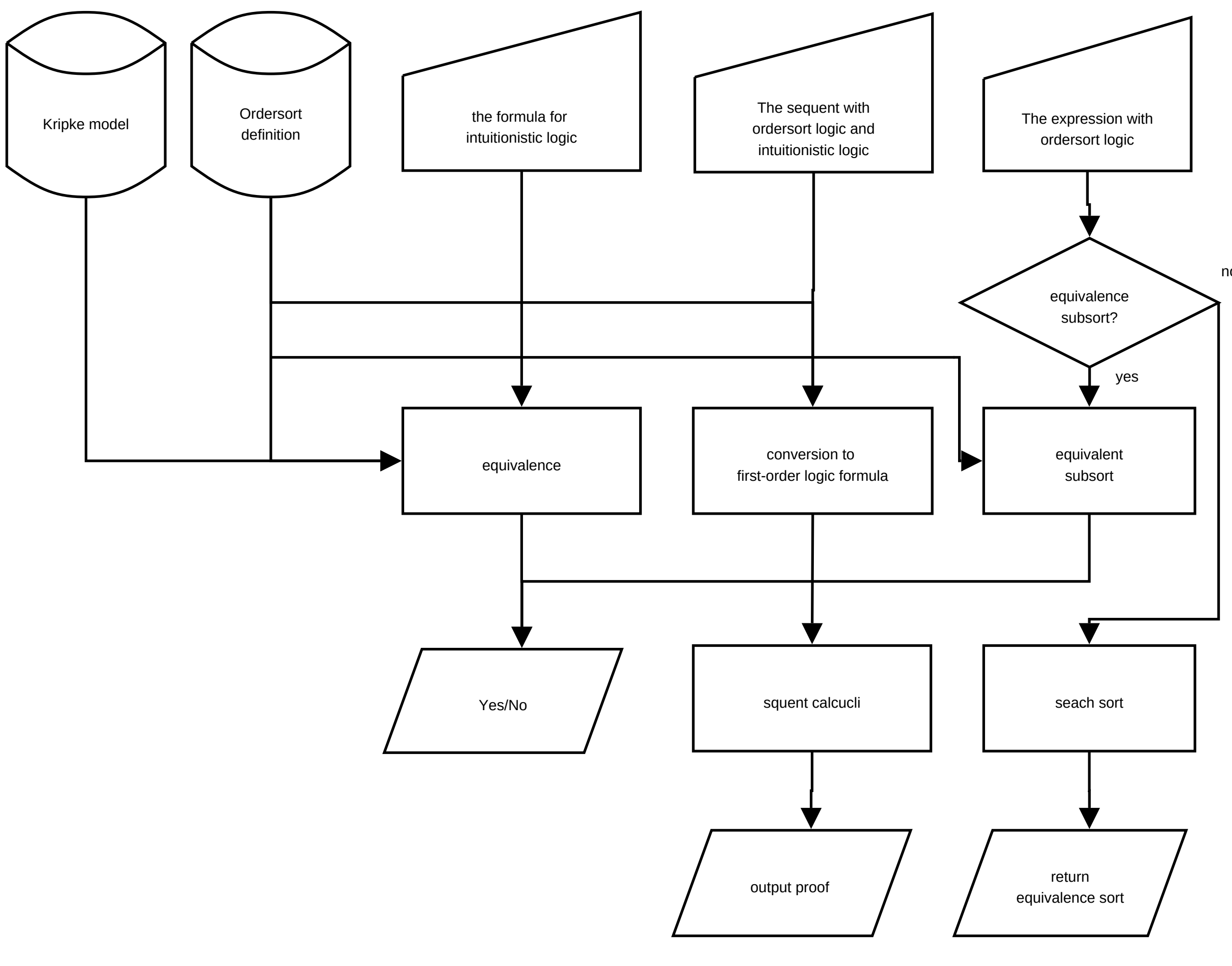


図 6.1 システム概要

なお実行環境は以下の通りである.

```
<Hardware environment(Informed by WCPUID V3 for Win32 Version 3.1a (Build 1089))>
  Processor #1 : Intel Pentium III / 7BA1938E
Internal Clock : 1135.91 MHz
  System Clock :  133.64 MHz
  Memory  Size : 256M Byte
  Memory Clock : ----

    OS Version : Windows 2000  Version 5.00.2195 Service Pack 4
##--- Date 11/25/2003, Time 15:09:43

Software environment
% SWI-Prolog version 5.2.10 by Jan Wielemaker (jan@swi-prolog.org)
% Copyright (c) 1990-2003 University of Amsterdam.
% SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
% and you are welcome to redistribute it under certain conditions.
% Please visit http://www.swi-prolog.org for details.
```

6.2 入力

システムの入力としては以下の構文で与える．なお,
+ : 直前のトークンの一回以上の繰り返し
| : トークンの条件の追加
' '：文字列（この場合はオペレータの文字列）
'?'：システムからの出力を受け取る項

クリプキ・モデル

PW = 可能世界の名前
PREDICATE = 述語．ただし量化記号を伴わない閉じた素式

```
user_def_W([PW+]).
user_def_R([relation(pw,PW,PW)+]).
user_def_V([node(pw,PW,PREDICATE)+]).

for examples:
user_def_W([w0,w1]).
user_def_R([relation(pw,w0,w1),relation(pw,w1,w1)]).
user_def_V([node(pw,w0,car(a)),node(pw,w0,car(b)),
node(pw,w1,ship(c))]).
```

オーダーソート論理の式を含む論理式

X = 対象変数名（Plorog の変数にならないように）

SORT = ソート名

```
QOP = 'FA' | 'EX'
WOP = '&&' | '||' | '=>'
SOP = '-.' | '~'
BIND = bind(X) | bind(X,SORTEXP)
EXP = pc(fanc(FANC),term([ of(X,SORTEXP) | of(X) +]))
SORTEXP = SORT SFUNC SORT
SFUNC = '|_|' | '|^|' | '(+)' | '(*)'
SEXPRESSION = EXP WOP EXP
SEXPRESSION = SOP EXP
SEXPRESSION = BIND QOP EXP
```

ソート階層の定義

```
user_def_sort([node(os,SORT)+]).
user_def_sr([relation(os,SORT,SORT)+]).
```

```
user_def_er([ex(SORT,SORT) | cnf(SORT,SORT)+]).

for examples:
user_def_sort([
      node(os,purple),
      node(os,red),
      node(os,colored),
      node(os,'~' colored),
      node(os,color),
      node(os,top_node),
      node(os,end_node)
     ]).

user_def_sr([
    relation(os,top_node,color),
    relation(os,color,colored),
    relation(os,color, '~'colored),
    relation(os,colored,red),
    relation(os,colored,purple),
    relation(os,purple,end_node),
    relation(os,red,end_node)
    ]).
```

実行

一階述語変換

```
os_permutation(SORTEXP,FOEXP?).
```

LJ

```
ev(FOEXP,FOEXP).
```

クリプキ・モデルにおける式の検証

```
evaluate(PW,FOEXP).
```

第7章　システムによる検証

概念階層の検査の例

この検査対象は図 4.7 を対象に行っているので，そちらを参照すると直観的にわかりやすい．図 4.7 で確かめられるような結果が以下で検査できる．なお，ソート階層の定義部分は SH^{++} として略記しており，必要に応じて適宜，式を取り出している．

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{SH}^{++}, purple \to purple \qquad \dfrac{\dfrac{\mathcal{SH}^{++}, colored \to colored \qquad \dfrac{\mathcal{SH}^{++}, color \to color}{\mathcal{SH}^{++}, color, \overline{color} \to}\ left\text{-}}{\mathcal{SH}^{++}, colored \sqsubseteq color, colored, \overline{color} \to}\ left \sqsubseteq}{\mathcal{SH}^{++}, colored, \overline{color} \to}}{\mathcal{SH}^{++}, purple \sqsubseteq colored, \overline{color}, purple \to}\ left \sqsubseteq}{\mathcal{SH}^{++}, \overline{color}, purple \to}}{\mathcal{SH}^{++}, \overline{color} \to \overline{purple}}\ right\text{-}}{\mathcal{SH}^{++}, \overline{\overline{purple}}, \overline{color} \to}\ left\text{-}}{\mathcal{SH}^{++}, \overline{\overline{purple}} \to \overline{\overline{color}}}\ right\text{-}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{SH}^{++}, red \to red \qquad \dfrac{\dfrac{\mathcal{SH}^{++}, colored \to colored \qquad \dfrac{\mathcal{SH}^{++}, color \to color}{\mathcal{SH}^{++}, color, \overline{color} \to}\ left\text{-}}{\mathcal{SH}^{++}, colored \sqsubseteq color, colored, \overline{color} \to}\ left \sqsubseteq}{\mathcal{SH}^{++}, colored, \overline{color} \to}}{\mathcal{SH}^{++}, red \sqsubseteq colored, \overline{color}, red \to}\ left \sqsubseteq}{\mathcal{SH}^{++}, \overline{color}, red \to}}{\mathcal{SH}^{++}, \overline{color} \to \overline{red}}\ right\text{-}}{\mathcal{SH}^{++}, \overline{\overline{red}}, \overline{color} \to}\ left\text{-}}{\mathcal{SH}^{++}, \overline{\overline{red}} \to \overline{\overline{color}}}\ right\text{-}
}{
\dfrac{\mathcal{SH}^{++}, \overline{\overline{purple}} \sqcup red \to \overline{\overline{color}}}{\mathcal{SH}^{++} \to \overline{\overline{purple}} \sqcup red \sqsubseteq \overline{\overline{color}}}\ right \sqsubseteq
}\ left \sqcup
$$

実行例

また以下に，実際のプログラムの実行例も示す．

図 7.1 はオーダーソートの表示，図 7.2, 図 7.3, 図 7.4 はオーダーソート論理の論理式の評価の過程と，実行結果．図 7.5, 図 7.6 は一階述語論理に対応した LJ の実行結果．図 7.7, 図 7.8 は $\forall x : apple \sqcup banana\ [apple \sqcup banana(y : plant)]$ を一階述語の表現に変えている．結果，$\forall x[fruit(x) \supset (fruit(y) \land plant(y))]$ と言う式に変換されている．これらを組み合わせて，検査を行った．なおこのときに使用した簡単なオーダーソートは

$\mathcal{SH}^{++} = \langle S^{++}, R^{++} \rangle$

$S^{++} = \{fruit, apple, banana, plant\}$

$R^{++} = \{apple \sqsubseteq fruit, banana \sqsubseteq fruit, fruit \sqsubseteq plant\}$

である．

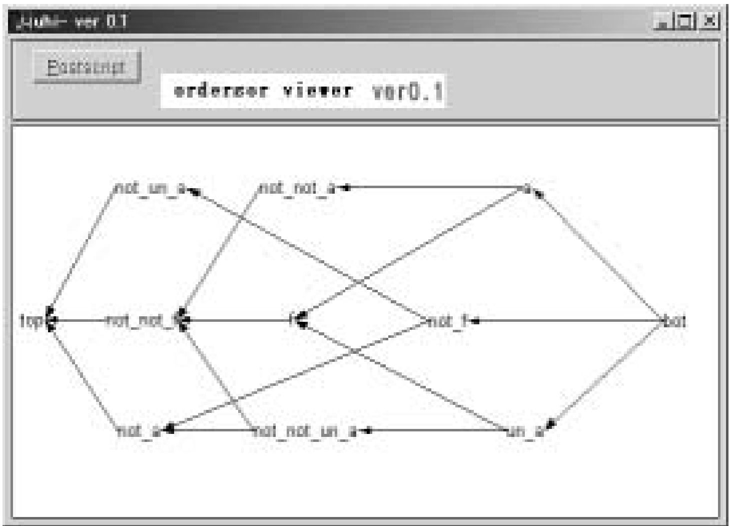


図 7.1 ソート階層の表示例

図 7.2 オーダーソート論理の論理式の評価例 1

図 7.3 オーダーソート論理の論理式の評価例 2.1

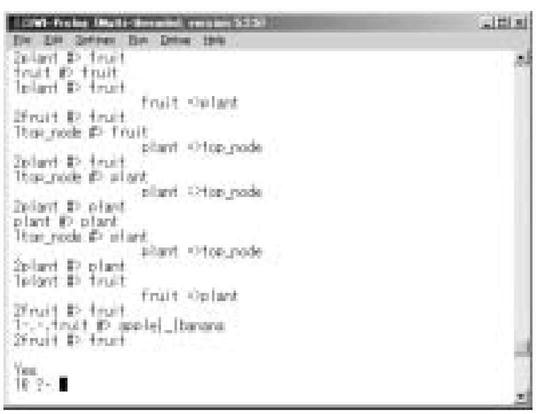


図 7.4 オーダーソート論理の論理式の評価例 2.2

図 7.5 LJ 例 1

図 7.6 LJ 例 2

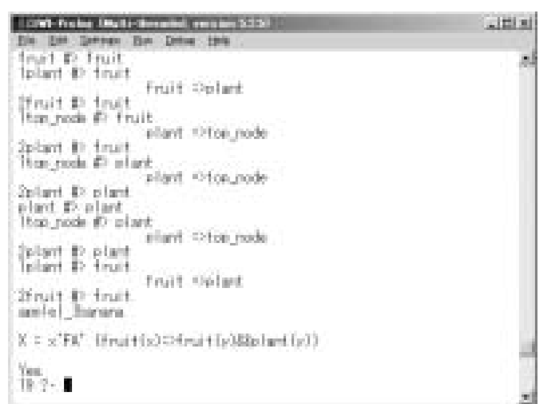


図 7.7 一階述語変換の例 1

図 7.8 一階述語変換の例 2

第8章　結言

本稿では自然言語の語彙をそのソート名として利用したオーダーソート論理に関して，直観主義論理の立場に立ち，その否定体系を見直し，推論系をあたえた．これによって，既存の手法では二重否定のような表現が正確にオーダーソートに反映されなかった点を解消し，より自然言語の意図したところを正確に表現できるようになった．特に，二重否定のみのサブソート関係は，緩やかな概念の上下関係を表現し，よりオーダーソート論理の表現力を増している．また，今回全てのオーダーソート論理の式を一階述語に変換することで，LJ によって完全な計算が可能となった．これは計算機への実装の面で非常に大きな利点である．

しかしながら，現時点では，ソート階層の中に出現するソートは，対象の性質などを表す，いわゆる形容詞や名詞のような単項のソート述語として使用可能な語彙に限定されている．これは，その品詞によって今回取り扱った内在的否定を表す接辞の否定の意味が変わってしまい，また，複数の項を持つものをオーダーソート内で表現しようとすると，そのサブソート関係が煩雑になってしまうこともあるからである．例えば，‘unhappy’ と言う語彙を考えると，本稿ではこれを $\sim happy$ というソート名としてソート階層内で利用する．この場合，その $\sim happy$ は $happy$ と対立し，$\neg happy$ を導きだし，$happy \sqsubseteq S(S \neq happy)$ となる S の性質を全て併せ持つようなソートという関係を導き出していた．しかしながら，‘uninstall’ のような動詞に接辞の否定が結合している語彙を考えると，これは本来の接辞のない動詞の逆の動作を表す．したがって，直ちに $\neg install$ を導くことは危険である．

また，ソート階層に含まれる品詞も現時点では全て統一する必要もある．これは例えば,「硬い」という形容詞と「水」,「液体」という名詞が混在したようなソート階層を想像するとわかりやすい.「硬水」というのは水のカルシウムとマグネシウムの含有量が高いことを表し，一般にその物質の形状変化に必要な力の大きさを表すものではない．このとき「硬水」は「液体」のサブソートであり，また「硬い」のサブソートにもなる．しかし,「硬い液体」というのは奇異な表現であることはすぐに解る.「液体」が「硬くない」と言うソートのサブソートにあるべきであると人間は考えるからである．このように，ある品

詞と他の品詞が結合した場合その語彙の意味が定まらないことに，問題がある．したがってソート階層における品詞を統一している．しかしながら，この際には複数のオーダーソートを同時に利用しなければならず，かつそれぞれのソート階層ごとの関係も定義する必要があり煩雑性が増す.

今後の課題としては以下のようになる．ソート階層において否定を表現することの意味は，ある対象が何らかのソートに含まれる場合，同時にどのようなソートに含まれるか，どのようなソートには含まれないことが保証されるかが直観的にわかりやすい点にある．そのことにおいて，自然言語の否定の表現は多種多様で，まだ全てを表現したとはとても言い難い．したがって，まだここで表現された否定表現だけでは自然言語を処理して推論させるには不十分である．また，一階述語のシーケント計算はその完全性，健全性において保証されている手法であるが，計算量が大きいため，よりソート階層を利用して計算量を減らせるものと考え，その面での改良も必要となる.

参考文献

[1] *The Art of Prolog*. The MIT Press, 1986.

[2] *Structures for Semantics*. Kluwer Academic Publishers, 1991.

[3] **情報科学における論理**. 日本評論社, 1994.

[4] *What is negation?* Kluwer Academic Publishers, 1999.

[5] **論理学をつくる**. 名古屋大学出版会, 2000.

[6] *A Natural History of Negation*, chapter 5. CSLI Pblications, 2001.

[7] C. Beierle, U. Hedtstuck, U. Pletat, P. H. Schmitt, and J. Siekmann. An order-sorted logic for knowledge representation systems. *Artificial Intelligence*, 55:149–191, June 1992.

[8] Michael Gelfond. Representing knowledge in a-prolog. 2001.

[9] Kentaro KIKUCHI. *Gentzen Style Sequent Calculi for Some Subsystems of Intuitionistic Logic*. PhD thesis, Japan Advanced Institte of Science and Technology, 2002.

[10] Dirk van Dalen. Intuitionistic logic. In *The Blackwell Guide to Philosophical Logic*, chapter 11, pages 224–257. 1986.

[11] Heinrich Wansing. Intuitionistic logic. In *The Blackwell Guide to Philosophical Logic*, chapter 18, pages 415–435. 1986.

[12] 兼岩憲，東条敏. 否定的意味が内在するソートを含んだソート階層論理. **情報処理学会誌**, 2002.