

Title	型階層論理における推論のダイアグラム化
Author(s)	吉岡, 卓
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1460
Rights	
Description	Supervisor:東条 敏, 情報科学研究科, 修士

修 士 論 文

型階層論理における推論のダイアグラム化

指導教官 東条 敏 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

吉岡 卓

2001 年 2 月 15 日

要 旨

通常人間が問題解決をするとき、言語的表現以外の様々な表現形態の情報のなかから、適切な表現を選択し変換しながら行っていると考えられる。本研究では、知識表現の一手法として、型階層論理に対するダイアグラムを用いた表現を提案する。

型階層論理とは、計算機上で知識を記述し推論させるために、知識表現に階層構造を持たせたものである。これにより構造的に知識を表現することができ、人工知能の分野では知識表現の目的で広く応用されている。しかし、対象とする世界を広げるほど異なるソートを複数扱わねばならず、推論の手間は増大していく。また、このままの形では計算機内に表現することは困難である。

一方、複雑さを解消したり容易に理解するための手法として、人間は図を用いることがある。図形を用いた表現は詳細に事象を記述するのには向かないが、直感的にその事象の概要を表すことに優れているためである。そのため、視覚的情報を用いた問題解決の方法は古くから研究されてきた。

そこで本研究においては、型階層論理における推論の手間が増大するという問題に対して、異なる形態の知識を同時に扱えるようなダイアグラムの考察を特色とする。本研究では特に、項の性質における抽象度と述語の性質における抽象度といった異なる部分に対するラティスの双対性を考察し、まとめて扱えるセルによる表現を提示する。セルを用いた表現の特色は、各ソートに $\top$ から $\perp$ までの道 (*path*) に応じた素数の積を割り当てることにあり、この方法によってセル上のソートが一意に定まるようにした。また、素数を割り当てることによって、割り算によるサブソート関係の判断が可能となり、サブソート関係にあるか否かを即座に判断できることが分かる。次にセルを用いたダイアグラムにおける描画規則と演繹規則を定義し、ダイアグラムの変形による推論を提案する。これらの定義により、双ラティスを用いるよりも構造化された高速な演繹を期待する。最後に、本研究で示すダイアグラムを実装しその優位性を検証した。特に本稿では、演繹ステップ数及び新しいソートの代入に対して比較を行った。これらの検証により、本稿で提案したセルによる表現の有意性を示す。

目次

1	はじめに	1
2	オーダーソート論理の拡張	3
2.1	オーダーソート論理	3
2.2	型変換の拡張	4
3	図による推論	7
3.1	ダイアグラムの特性	7
3.2	ベン図による表現	9
3.3	変形規則	11
3.4	ベン図を用いた演繹	13
3.5	図の効率性	15
3.6	階層構造の表現	16
3.7	チャートを用いた表現	20
4	双ラティスのダイアグラム化	24
4.1	セルによる表現	24
4.2	Syntax	27
4.2.1	Alphabet	27
4.2.2	Signature	27
4.2.3	セルにおける描画規則	28
4.2.4	セルを用いた推論	33
4.2.5	セルにおける演繹規則	34
4.3	Semantics	37
5	比較検証	41

5.1	図を用いたシステム	41
5.2	セルを用いたシステム	44
5.2.1	セルによるシステムの表現	44
5.2.2	論理積を含む表現	46
5.2.3	論理和を含む表現	47
5.2.4	否定を含む表現	49
5.3	自然演繹法とセル表現の比較	52
5.4	双ラティスとセル表現の比較	53
5.5	新しい宣言に対する負荷	55
6	考察	61

第 1 章

はじめに

計算機上で知識を記述し推論させるために、知識表現に階層構造を持たせる研究が広く行なわれている。オーダーソート論理とは、オブジェクトの集合をソートとし、そのソート間に階層を与えたものである。これにより構造的に知識を表現することができ、人工知能の分野では知識表現の目的で広く応用されている。しかし、名詞であれば名詞に関するソート階層、動詞であれば動詞に関するソート階層などというように、ソートをつくる集合ごとに上位下位の関係を考察しなければならない。このように対象とする世界を広げるほど異なるソートを複数扱わねばならず、推論の手間は増大していく。

一方、複雑さを解消したり容易に理解するための手法として、人間は図を用いることがある。それは言葉や数値を用いるよりも分かり易いことが多いからである。図形を用いた表現は詳細に事象を記述するのには向かないが、直感的にその事象の概要を表すことに優れているためである。そのため、視覚的情報を用いた問題解決の方法は古くから研究されてきた。数学的なアプローチからは、Euler や Venn が図形の構造の発展に寄与した。一方、論理的なアプローチからは、C.S.Peirce が命題、述語、様相論理に対する図による表記法をそれぞれ発展させてきた。しかし一般には、数学の問題に図形を用いることは、正確な証明を与えているとは考えられていない。図形を用いることは、形式的な証明を見つけ出すツールとして以外、利用価値のないものとみなされている。そこで、Shin はベン図の持つ構造と図形操作に着目し、ベン図で正しい推論を行うことができることの健全性と完全性を証明した [8]。また Shin による証明の拡張として、Hammer は、命題論理に対するパースグラフを用いた推論の健全性と完全性を与えている [2]。また、このような図による推論の有効的な特徴に着目し、計算機による推論を記号だけでなく図形を用いて行うことが最近注目されている。Swoboda [7] は、計算機上でベン図を用いた推論を行うための表現について注目しており、頂点と有向辺からなる記述を示した。また、村

田ら [13] は算術問題に着目し、旅人算や植木算といった問題に対して、図形を表示するようなシステムを構築している。清塚ら [8] においては、図形を用いることにより言語表現による推論よりも演繹ステップ数及び戦略において有効であることを示している。

そこで本研究においては、型階層論理における推論の手間が増大するという問題に対して、異なる形態の知識を同時に扱えるようなダイアグラムの考察を特色とする。具体的には、名詞と動詞に関する抽象度を同時に扱えるような形式化を与えたい。ここで、動詞や形容詞に関しては述語として、名詞に関しては項として分類することとする。よって本稿では、項の性質における抽象度と述語の性質における抽象度といった異なる部分に対するラティスの双対性を考察し、まとめて扱える構造を提示する。次に、ダイアグラムの変形規則を定義する。そこで、論理式による推論に対するステップ数の優位性を検証する。また、情報のユニフィケーションにも着目し、表現の抽象度に対して、そのダイアグラム化と証明を与えることが本研究の特色である。本研究により、人間の推論における過程の解釈にも興味深いものと考えられる。また表現の簡略化にも着目し、本研究で提案するダイアグラムが明瞭な表記となることを主眼に、演繹ステップ数減少の有効性を期待する。

本稿は次のような構成をとる。2章は従来の型階層論理を項と述語に適用できるように拡張し、その意味付けを与える。これにより、項と述語が異なるソート階層を持つような論理式を扱えるようになり、実際の知識を記述する際に、より構造的な記述ができるようになる。3章は、2章で定義した型階層論理の拡張をダイアグラムによって形式化するために、ダイアグラムの定義を示す。4章は論理式による推論とダイアグラムによる推論の比較を、実装したシステムによって検討する。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。

第 2 章

オーダーソート論理の拡張

2.1 オーダーソート論理

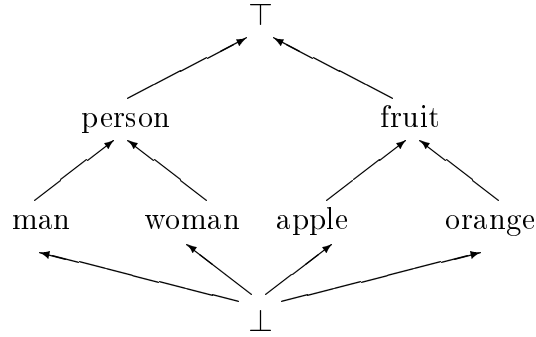
ソート階層は、2 字組 $(S, \sqsubseteq_S)$ で構成される。ここで、 $S(= \{s_1, \dots, s_n\})$ とはソート記号の集合、 $\sqsubseteq_S (\subseteq S \times S)$ とはサブソート関係を表す。また、サブソート関係の要素 $(s_i, s_j) \in \sqsubseteq_S$ は、 $s_i \sqsubseteq_S s_j$ によって表され、これをサブソート宣言という。例えば、次のサブソート宣言の集合、 $\{man \sqsubseteq_S person, woman \sqsubseteq_S person, orange \sqsubseteq_S fruit, apple \sqsubseteq_S fruit\}$ によって、図 2.1 のようなソート階層が構築される。図 2.1 に示されるように、すべての頂点は到達可能な頂点と型関係を持ち、上へいくほど上位概念を示し、下にいくほど下位概念であることを示す。このように、ソートは半順序関係を持つラティスを形成している。本稿で扱う背景知識は、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。こうした概念階層で定義されたものを**型表現**といい、それによって作られる階層を**型階層**という。

ソートとして扱う知識は、表 2.1 に示すように名詞的要素の [項] と動詞または形容詞的要素の [述語] に分けることができる。しかし、一般のオーダーソート論理では述語における型表現を許していない。よって、次に示すような拡張を行う。

表 2.1: 背景知識の分類

分類	形式	特性
項	～は、～の上位（下位）概念である	物、人などの表現
述語	～は、～の上位（下位）概念である	性質、関係、事象などの表現

図 2.1: ソート階層



2.2 型変換の拡張

本稿では、オーダーソート論理を拡張して、項と述語にそれぞれ別の型表現を導入する．ここで、使用する知識は従来の記法であるソート階層を想定している．これにより、簡潔な概念定義と効率的な単一化が期待できる [10]．すなわち本稿で扱う知識は、項と述語の異なる 2 つのソート階層を構築している．項に関する型階層は従来のソート階層にあたり、項の型関係をサブソート関係という．また、述語の型階層を述語の部分関係ということとする．本稿では以後、サブソート関係と述語の部分関係を *implication* として扱うものとし、ソートは最大ソート $\top$ と最小ソート $\perp$ を含むものとする．

拡張 2.1 (項の型変換)

項の型変換 $f_s : T \rightarrow T$ を次のように定義する．

$$f_s(x : s_1) = x : s_2$$

$$f_s(s_2) = s_1$$

$$f_s(s_1) = x : s_1$$

ここで、 T は項の集合、 s_1, s_2 をソートを示し、 $s_1 \sqsubseteq_s s_2$ であるとする．また、 $(x : s)$ とは x が型 s の要素であることを示す．すなわち、 x, s には $\llbracket x \rrbracket \in \llbracket s \rrbracket \subset U$ の関係がある．なお、 $\llbracket \cdot \rrbracket$ は解釈を表し、 U とはすべてのオブジェクトからなる世界を表す．

拡張 2.2 (述語の型変換)

述語の型変換は、述語記号そのものを型表現として見なす． P が述語記号の集合であるとき、 $\sqsubseteq_p (\subseteq P \times P)$ 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、

原始論理式の集合 F に対して、次のように上位方向述語への型変換 $f_p : F \rightarrow F$ を許すものとする。

$$f_p(p_1(t_1, \dots, t_n)) = p_2(t_1, \dots, t_n)$$

ここで、 p_1, p_2 は述語、 $t_1, \dots, t_n$ は項を示し、述語の部分関係が $p_1 \sqsubseteq_p p_2$ であるとする。

なお、述語にはオブジェクトが述語表現をした事実を表す事象的述語と、オブジェクトの持つ属性を持つ属性的述語の 2 種類があるが、これらについては 3 章にて後述する。

以上の拡張により、述語記号 P, Q に対して $P \sqsubseteq_p Q$ ならば、

$$\forall s(P(x : s) \rightarrow Q(x : s))$$

$$\forall s(P(s) \rightarrow Q(s))$$

と定義する。

ソート s_1, s_2 に対して $s_1 \sqsubseteq_p s_2$ ならば、

$$\forall P(P(x : s_1) \rightarrow P(x : s_2))$$

$$\forall P(P(s_2) \rightarrow P(s_1))$$

$$\forall P(P(s_1) \rightarrow P(x : s_1))$$

$$\forall P(P(s_2) \rightarrow P(x : s_2))$$

と定義する。

したがって、述語記号 P, Q に対して $P \sqsubseteq_p Q$ 、ソート s_1, s_2 に対して $s_1 \sqsubseteq_p s_2$ ならば、

$$P(x : s_1) \rightarrow Q(x : s_2)$$

$$P(s_2) \rightarrow Q(s_1)$$

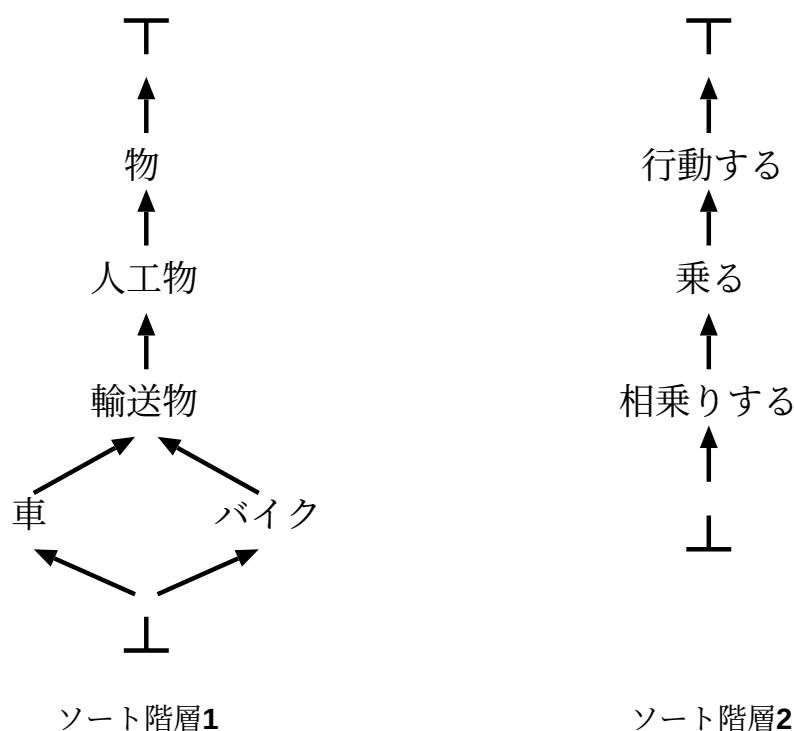
$$P(s_1) \rightarrow Q(x : s_1)$$

$$P(s_2) \rightarrow Q(x : s_2)$$

への implication も可能となる。

従って、この拡張により、従来のオーダーソート論理よりも記述できる知識の量が増えることとなる。しかし、述語の部分関係を考慮することは従来のオーダーソート論理よりも複雑なものとなり、異なるソート階層を同時に考察することは推論の手間の増加へとつながる。

図 2.2: 双ラティス



具体的には、次のような例があげられる。

車に相乗りした. → 物に対して行動した.

上記の推論は、[車] と [物]、[相乗りする] と [行動する] の概念間に、上位下位のあることにより行われる。例えばそれは、図 2.2 のような階層を持っている。

このように、いくつものソート階層を探索することは処理を複雑なものとする。それは、関係あるオブジェクトの集合においてラティスを生成することにあり、対象とする領域を複数生成することにある。

そこで本稿では、述語と項に分けられる 2 つの対象領域からなるラティスを同時に表記するためのダイアグラムによる形式化を与える。同時に表記することにより、複雑さが解消されるばかりでなく推論のステップ数が軽減することを期待する。

第 3 章

図による推論

3.1 ダイアグラムの特性

我々が問題を考えるとき、言葉や数値を用いる代わりに図形、チャート、テーブルを用いることが多い。それは、言葉や数値による表現よりも図形を用いた表現のほうが、分かり易いことが多いためである。また、直感的な理解により、複雑な状況が容易に理解できるともいえる。図形表現で有名なものに、ベン図やオイラー円があげられる。これらは一般的に数学の証明に対して正しい証明を与えていると考えられていない。しかし、Shin や Hammer らによって論理の整合論理式 (wff) に対応する整合論理図形 (wfd) が考案され、一階述語との健全性と完全性が示された。このことにより、図形を用いた表現は一階述語と同等であることが示されている。

これらの図形に関する研究の背景には Barwise らの図形に関する研究がある。彼らの研究においても図形に関していくつかの定義を行っており、図形一般に関して以下に示すような理想をかかげている [6]。従来の研究は、以下の理想に従って図形に注目することにより、一般の論理との完全性および健全性を保持していると言える。

Idealization 1

図形において表現された各々のオブジェクトは、不明瞭さを解消した表現をもつ。

Idealization 2

well-formed な表現は表現形式において有限で線形であり、かつ故に、記号の有限列で表現される。

Idealization 3

与えられた well-formed オブジェクトの内包は、不確実性を解消したものである。

表 3.1: テーブル表現

A	B	C
	D	

Idealization 4

システムにおけるオブジェクト間の帰結関係は、この syntax 上の関数である。いずれにしても関係を決定付けるような効果的な規則が存在する。

Idealization 5

システムにおける文法と意味論は有限的に明示できる。

また、ここでは具体的に論理式を用いた表現よりも図形を用いた表現のほうが分かりやすい事についても言及する。そのことを説明するために、図形一般が持つ特性について述べる。ここで、表 3.1 に示すような表形式の図形が与えられているとする。このとき、表 3.1 を言葉であらわそうとすると、多数の記述の仕方ができる。例えばそれは、

A は B の左にある。

C は B の右にある。

D の上に B がある。

A は一番上にある。

A は一番左にある。

:

などがあげられる。これは、図形を用いた表現というのは絶対的な位置情報を内在しているためである。一方、言葉を用いた表現というのは相対的な表現を記述することはできるが、オブジェクトの絶対的な位置を記述するのに向かないためである。

ここで逆の例を考えてみる。いま、以下に示すような記述が与えられたとする。

A は B の左にある。

C は B の下にある。

このとき、これらの記述情報から作成される図形は、複数存在する。例えばそれは、表 3.2 や表 3.3 に示されるようなものがあるかもしれない。なぜならば、言葉による表現は完全とはいえ、B と C の間に空白があるかもしれないし、オブジェクトの大きさなども

表 3.2: テーブル 1

A	B
	C

表 3.3: テーブル 2

A	B
	C

言及していないからである．これは，言葉を用いた表現はオブジェクトの絶対的な特徴を記述するの向いていないために，曖昧性がそのまま残ることに起因している．

すなわち，図形を用いた表現は位置や特徴に対して，絶対的な情報を保持することができる．この特徴は図を用いることの重要な点である．このような図形の利点に関して，清塚ら [11] は次のような研究を行っている．その研究によれば，三段論法の証明において記号を用いた証明よりも図形を用いた証明のほうが，演繹ステップ数及び戦略において優位であるという結果がだされている．そのような結果が得られる背景には，ダイアグラムにおける Free Ride の効果が重要な点を占めている．この Free Ride について，以下でベン図を用いて説明する．

3.2 ベン図による表現

ベン図においては，図 3.1 に示すような図形を用いる．左から長方形，閉曲線，影，線とよぶ．また，オブジェクトを指し示すために $\otimes$ と定数記号を用いる．また，線と $\otimes$ を合わせて $\otimes$ -列と呼ぶ．また，線と定数記号を組み合わせて定数列と呼ぶ． $\otimes$ -列と定数列を合わせて列と呼ぶ．長方形はすべての対象からなる個体の集合を表し，閉曲線はその個体の部分集合を表す．影がついている図形の領域は空集合を表している．列がついている図形の領域は空集合でないことを表す．

例えば図 3.2 では， $\otimes$ -列が置かれている 2 つの領域の少なくともいずれかには，個体が存在していることを示している．

閉曲線によって囲まれた領域を基本領域と呼ぶ．また，閉曲線の境界によって区切られ

図 3.1: 使用する図形

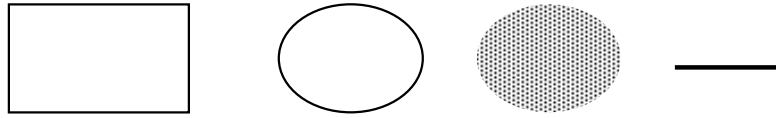
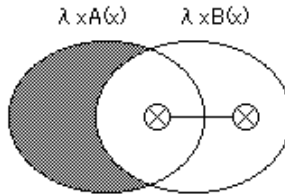


図 3.2: 使用する図形



たこれ以上分割されない領域を最小領域と呼ぶ.

閉曲線にはそれぞれラベルがつけられる. 図 3.2では $\lambda x A$, $\lambda x B$ がそれにあたる. 述語記号に λ と x をつけることでその述語記号の変数である x が自由変数であることを表す. このラベルを集合項と呼ぶ. このラベル付けにより閉曲線の中の領域をその述語が真になる個体の集合に対応付けている. ベン図を用いた演繹法に導入される図形を次のように定義する.

【定義 3.2.1】 (整合論理図形)

整合論理図形 (wfd) とは, 以下にあげる 4 つの条件を満たすものをいう.

- (1) 任意の長方形は wfd である.
- (2) D が wfd であり, また C は D の中不在 1 つのラベルのついた閉曲線であるとする. ここで, 重ね合わせるルールによって D に C を加えることによって得られる図形は wfd である. 重ね合わせるルールとは, C と D の中にある最小領域が一度ずつだけ交わるように重ねるルールである.
- (3) D が wfd であり, b は任意の定数記号とする. 定数列 (b -列) または $\otimes$ -列を D に加えて得られる図形は wfd である. 但し, すべての列が長方形の中にあり D の閉曲線の境界線上に定数あるいは $\otimes$ が置かれてはならない. また, $\otimes$ や b は最小領域の中に 1 つだけしか書けない.

(4) D を wfd とする. D のある閉じた領域に影をつけて得られる図形は wfd である.

【定義 3.2.2】 (図形の領域と集合項の関係)

D を任意の整合論理図形とし, x, y, z を任意の変数とする. 関係 $\cong$ を次の 2 つの条件によって定義する.

- (1) $\lambda x \varphi(x)$ が D の閉曲線にラベル付けされていて, r が閉曲線に囲まれた領域とする.
 y が $\varphi(y)$ の自由変数なら $r \cong \lambda y \varphi(y)$ である.
- (2) もし $r \cong \lambda x \varphi(x)$, $s \cong \lambda y \psi(y)$ であり, $\varphi(z)$, $\psi(z)$ において z が自由変数なら関係 $\cong$ を以下のように定義する.

- $\bar{r} \cong \lambda z \neg \varphi(z)$
- $r \cap s \cong \lambda z (\varphi(z) \wedge \psi(z))$
- $r \cup s \cong \lambda z (\varphi(z) \vee \psi(z))$
- $r - s \cong \lambda z (\varphi(z) \wedge \neg \psi(z))$
- $\bar{r} \cup s \cong \lambda z (\varphi(z) \rightarrow \psi(z))$

3.3 変形規則

〔1〕 ベン図の描画規則

ベン図は次の描画規則をもつ. 以下の規則により図形を変形して新しい図形を生成する. なお, 以下の規則において D は wfd であるとする.

列の一部消去 D の影のついた領域に列の一部が重なっていたら影の中の列を消去する. もし, それによって列が分割されることになれば, それらを線によってつなげる.

列の延長 D の任意の列に新しい線を加える.

図形要素の消去 D に以下の操作をおこなう.

- (1) D の閉曲線を消す.
- (2) ある D の影の領域を消す.
- (3) D の $\otimes$ -列全体を消す.

ただし、(1)を行うとき、閉曲線を消去した後、囲いのなくなった影つき最小領域は消去する。また、同一最小領域内に $\otimes$ あるいは定数が2つ以上ある場合は1つを残して消去し、列が分かれることになればそれらを線によってつなげる。

基本領域の導入 影や列のない閉曲線をもっている長方形もしくは長方形自身自身を生成する。

情報の衝突 D が影がついていて列全体を含んでいる領域を持っているなら任意の図形を生成してよい。

2つの図形の統合 以下の条件をみたす統合により $\text{wfd } D_1$ と $\text{wfd } D_2$ から $\text{wfd } D$ を得ることができる。

- (1) D のラベルの集合は D_1 と D_2 のそれらと同じである。
- (2) D_1 あるいは D_2 のどちらかの領域 r に影がついているならば、 D にある対応するところにも影がついている。逆に D の領域 r に影がついていれば、 D_1 あるいは D_2 のどちらかの対応する領域にもまた影がついている。
- (3) もし D_1 あるいは D_2 のどちらかの領域 r に列があるならば、 D にある対応する領域にも列がある。逆に D の領域 r に列があれば、 D_1 あるいは D_2 のどちらかの対応する領域にもまた列がある。

〔2〕ベン図と論理式間の変形規則

以下の規則によって論理式を図形に、図形を論理式変換することができる。これにより論理式で行う演繹を図形でも行うことができる。以下の規則において D は wfd とする。

$\forall$ -Apply $\text{wfd } D$ と論理式 $\forall x\varphi$ があるとする。 D の中に $r \cong \lambda x\varphi$ となるような領域 r がある時、 $\bar{r}$ の任意の最小領域に影をつけるか、あるいは r への任意の列の付加によって新たな図を得ることができる。

$\forall$ -Observe $r \cong \lambda x\varphi$ のようなある D の領域 r があって $\bar{r}$ に影がついているならば、論理式 $\forall x\varphi$ を D から得ることができる。

$\exists$ -Apply $\text{wfd } D$ と論理式 $\exists x\varphi$ があるとする。 $r \cong \lambda x\varphi$ であるような領域 r があり、 $\otimes$ 列を r のある部分領域に加えることによって、新たな図形を得ることができる。

$\exists$ -Observe $r \cong \lambda x\varphi$ のような D の領域 r があり、 r の部分領域が列を含んでいるか、あるいは、 $\bar{r}$ に影がついていれば、論理式 $\exists x\varphi$ を $\text{wfd } D$ から得ることがで

きる.

Constant–Apply wfd D と論理式 $\varphi(b)$ があるとする. $r \cong \lambda x\varphi(x)$ のような D の領域 r があるとき, b-列をある r を含む領域に加えることによって新しい図形を得ることができる.

Constant–Observe $r \cong \lambda x\varphi(x)$ のような D の領域 r があり, r のある最小領域が b-列を含んでいるか, あるいは, $\bar{r}$ に影がついているかのどちらかであれば, 論理式 $\varphi(b)$ を得ることができる.

Inconsistent–Information ある領域に影があり, その影の上に列の全体があるならば, 任意の論理式を得ることができる.

3.4 ベン図を用いた演繹

整合論理図形の規則を用いてベン図を操作する. 例えば, アリストテレス式三段論法においては以下のような図による推論が行われる. 推論の描画に入る前に, アリストテレス式三段論法の概要を述べる.

次の推論はアリストテレスになる式三段論法の一例である.

大前提 すべての人間は死すべきものである.

小前提 ソクラテスは人間である.

結論 ソクラテスは死すべきものである.

三段論法は大概概念 P , 中概念 M , 小概念 S の3つの概念で表現される. 上の例では「死ぬ」が大概概念であり, 「人間である」が中概念, そして「ソクラテスである」が小概念である. このとき, 3つの概念の包含関係は $S \subseteq M \subseteq P$ である. よって, 上に示した推論は以下のような論理式で表せられる.

$$\frac{\begin{array}{l} \forall x(M(x) \rightarrow P(x)) \\ \exists x(S(x) \wedge M(x)) \end{array}}{\exists x(S(x) \wedge P(x))}$$

アリストテレス式三段論法を, 図 3.3 では自然演繹法を用いて推論した. また, 図 3.4 ではベン図を用いて推論を行った.

ベン図を用いた図 3.4 の証明では最初にそれぞれの前提の論理式の数だけ前提となる図形があるとして証明を行う. 最初2つの図形の統合を用いて新たな図形を生成する. 列の

図 3.3: 自然演繹法による証明

$$\begin{array}{c}
 \frac{[S(a) \wedge M(a)]}{S(a)} \qquad \frac{[S(a) \wedge M(a)]}{M(a)} \qquad \frac{\forall x(M(x) \rightarrow P(x))}{M(a) \rightarrow P(a)} \\
 \hline
 \frac{S(a) \qquad P(a)}{S(a) \wedge P(a)} \\
 \hline
 \frac{\exists x(S(x) \wedge M(x)) \qquad \exists x(S(x) \wedge P(x))}{\exists x(S(x) \wedge P(x))}
 \end{array}$$

図 3.4: ベン図による証明

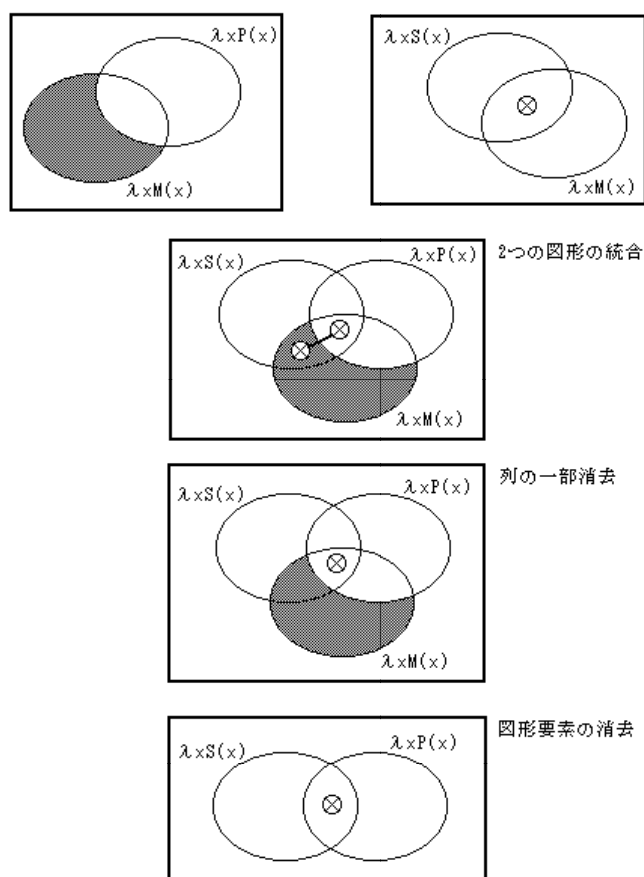
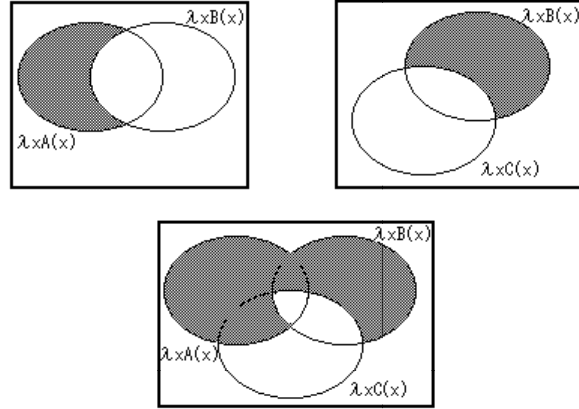


図 3.5: Free Ride の例 1



一部消去により影の上の $\otimes$ を消去する．次に図形要素の消去を用いて， $\lambda xM(x)$ のラベルが付けられた図形を消去する．これにより結論となる $\exists x(S(X) \wedge P(X))$ が得られる．

3.5 図の効率性

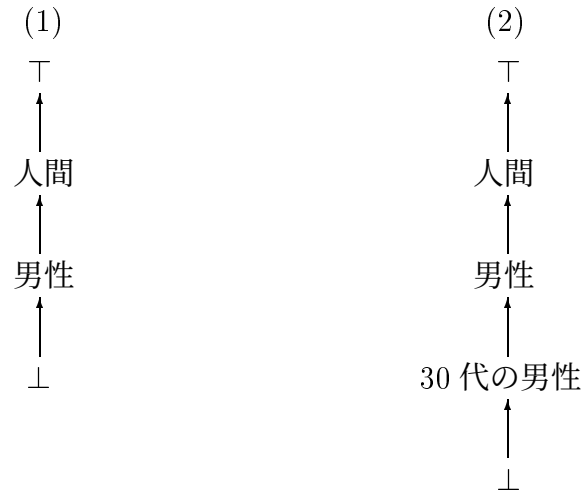
このとき，図 3.3 と図 3.4 において，用いている規則の数を考慮すると，明らかにベン図を用いた表現のほうが効率的である．特にベン図において注目すべき点は 2 つの図形を統合した個所である．統合した図形は既に結論となる図形を内在している．この効果が **Free Ride** と呼ばれる効果である．図 3.5 にその例を示す．図 3.5 は以下の論理式を表現したものである．

$$\frac{\begin{array}{c} \forall x(A(x) \rightarrow B(x)) \\ \forall x(B(x) \rightarrow C(x)) \end{array}}{(\text{Therefore}) \forall x(A(x) \rightarrow C(x))}$$

このとき，前提 1 の図形に前提 2 の図形を統合したものが図 3.5 の下部の図形である．ラベル A の閉曲線と B の閉曲線の関係にラベル C の閉曲線を**加える**だけで，結果として期待するラベル A の閉曲線とラベル C の閉曲線の関係も自動的に明記される．このことが，implication において有効性がみられる結果となる．

図 3.6 にてラティスに対する有効性を確認する．図 3.6 (1) は与えられたラティスとする．このとき，図 3.6 (1) の「 $\perp$ 」と「男性」の間に「40 代の男性」を挿入する．このと

図 3.6: Free Ride の例 2



き,「40 代の男性」は「 $\perp$ 」と「男性」との関係しか考慮していない. しかし,与えられた図形は 3.6 (2) に示すように「40 代の男性」が「人間」の下位概念として含まれる事をしめしている. また「 $\top$ 」の下位概念であることも示している. このように,部分的な情報や個体を新たに加えたときに,全体における個体の位置が定まる効果を Free Ride と呼ぶ. より詳しい形式化の定義は,下嶋 [1] が与えている.

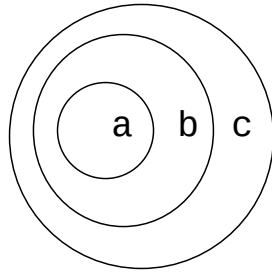
図を用いた表現とは,個々のオブジェクトがオブジェクト全体における個体の位置 (location) を保持しているからであり,特に明記することなしに部分的情報が全体的情報となりえる. このことは,人間が推論を実行する上で非常に有効なものである. そのため本稿では,次章にて型階層論理における問題を再考察する. 型階層論理において問題となっていた異なるラティスを同時に扱わねばならない点に対して,ラティスの双対性を考慮したダイアグラムを提示する.

3.6 階層構造の表現

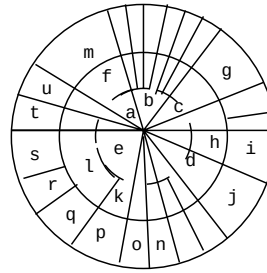
本稿では型階層論理におけるダイアグラムの構築を目的としている. そのため,まず初めに階層構造に適した表現に着目する. 階層構造を表現するには,大まかにいって次の 4 つがあげられる [9].

(1) 領域系の図

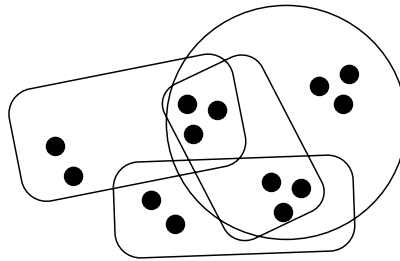
図 3.7: 領域系の図形



representation of nesting



hierarchy by a circle graph



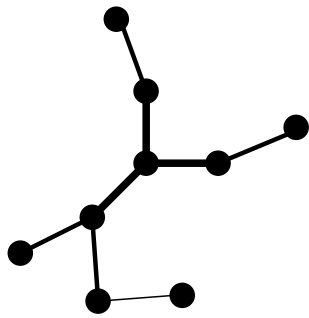
hierarchy containing multiplex relation

- (2) 連結系の図
- (3) 配列系の図
- (4) 要素記号の表現

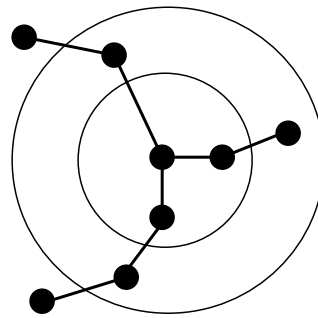
ここで、(1) はそれぞれの部分集合を表す領域図形を、さらに大きな領域図形で囲むことにより上位の部分集合を表現する。すなわち、階層構造は入れ子による空間的表現として描画される。この方法では、関係付けの手段である領域図形そのものが、個々の意味要素を統括する上位の意味内容を担う記号としての役割を果たすことが特徴としてあげられる。しかし、多重関係を示す場合は明確に捉えられない場合もある。これら領域図形の例を図 3.7 に示す。

(2) は分岐 (合流) によって階層関係を表現する。階層関係は木構造として表現できる。これにより、階層の方向性、順序という過程的關係が表現される。また、下位、上位のすべての概念は独立した要素記号で表現される。これらの連結図形の例を図 3.8 に示す。

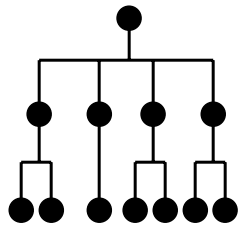
図 3.8: 連結系の図形



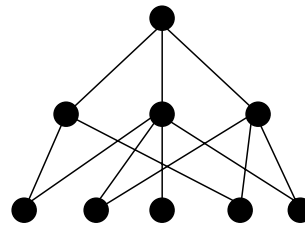
hierarchy by connection diagram



hierarchy included domain diagram

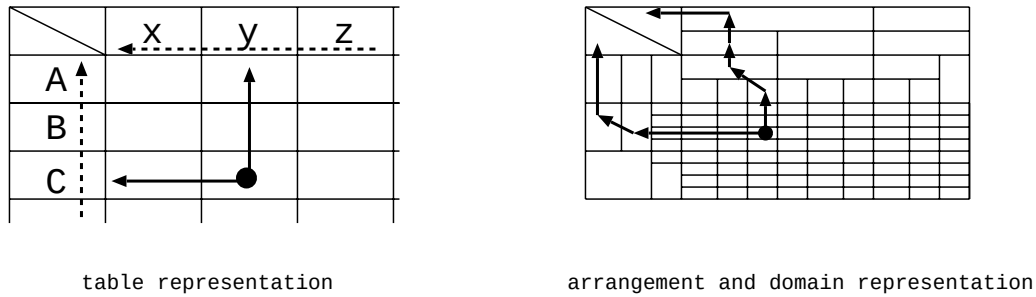


hierarchy by arrangement



hierarchy containing multiplex relation

図 3.9: 配列系の図形



(3) は階層的関係を既にもつ表現である．最も多く用いられる表 (テーブル) は，階層構造のための約束ごとの上に成立している表現形式である．表の基本型は，まず最下位の部分集合が縦，横の配列で表現される．そして，これらの部分集合の意味 (上位概念) を担う要素記号群が描画される．この基本型では表現できる階層が三段階に制限されるので，まず目を活用して，より多段階の表現を展開することも多い．

また，配列では，行もしくは列の並びの順位に，階層のレベルを対応させる表現がある．また，文章やプログラムの表現でも段落としによって意味内容のレベルを表すことも多いが，これも 2 次元の配列に依存するので，配列といってもよい．このように配列による表現は，階層のレベルを明記することができるのが特徴としてあげられる．これらの配列図形の例を図 3.9 に示す．

(4) は要素記号の表現によって階層構造を表す表現である．しかし，表現は難しく，理解しにくい．一般には濃淡や文字の大小などによって，図形に段階的表現をつけて理解させる方法である．

まとめると，領域図形では階層関係が空間的な表現として表される．連結図形では過程的關係が強調される．配列は階層のレベルが明示される特徴をもつ．このように，視覚的に強調される側面が異なるために，これらの組合わせを組み合わせる用いるのが有効であると考えられる．

そこで，今回提案する表現は，配列系の図形として階層レベルを明示しつつ，それらの要素の領域を考察できるような表現とする．Glasgow ら [4] の研究によれば，配列中の任意の領域に対してより詳細な図形を埋め込む (embedded) ような表現も研究されている．よって，ここで提案するダイアグラムは，配列系の図形の個々のます目にオブジェクトを挿入し，その及ぼす範囲を領域で示すようなものが効果的であると考えられる．

3.7 チャートを用いた表現

ここでは、本稿で提案するセル形式の型階層論理の描画に入る前に、チャートを用いた表現に注目する。なぜならば、異なった要素からなる論理の分類に、チャートを用いることは、最も適した表現であるからである。チャートを用いた表現は、非常に使いやすいものがある。チャートは、情報のある種類を表現するのには便利な方法として用いられている。それはまた推論の確かな順序付けにとっても容易に用いることができ、特にオブジェクトの系統だった整合性を期待できる。チャート表現で良く使われる例としては、first name のあとに last name が続く名簿や、犬の名前と、その犬の飼い主といった例があげられる。チャートを用いることは、問題に対する順序を発見することにつながり、前後関係を明らかに扱いやすいものとする。チャートはあまりにも単純な表現であるゆえに論理的に正しい数学モデルを与える方法として学習するのに有効である。ここで、チャートを用いた問題解決の為には、その順序付けが重要となる。そこでチャートを用いるための形式化を Barwise らは以下のように定義した [6]。

ソートを用いたチャート論理とは、選択されたソートとその名前によって形成される。それは以下のように表せる。

$$\begin{array}{l} \text{Sort } \alpha_1 \text{ having names } a_1^1, \dots, a_n^1 \\ \text{Sort } \alpha_2 \text{ having names } a_1^2, \dots, a_n^2 \\ \vdots \\ \text{Sort } \alpha_m \text{ having names } a_1^m, \dots, a_n^m \end{array}$$

このようなソートの選択がなされたとき、図 3.10 に描画されたチャート図は **well-formed chart** と呼ばれる。ただしここで、 $m = 4, n = 3$ とする。

チャートを用いると都合なことに、縦の列の ' α_i ' とラベル付けされた列は '列 α_i ' として参照可能となることである。すなわち、そのようなチャートにおいて、列 α_i の全てのセルの中には α_i という記述がなされずとも、自動的にその性質 α_i を割り振ってくれる。これが well-formed chart の条件制約である。

各ソートを単項の述語記号として扱おうからといって、すべての述語記号をソートとするわけではない。この問題は言語学的な問題をはらんでいるためである。このチャートの

図 3.10: well-formed chart

α_1	α_2	α_3	α_4

モデルのために、我々は identity をもった標準的な 1 階述言語を用いる。これは本来、 $a_3^3 \neq a_3^2$ であるか否かといった問題をはらんでいるためである。これは、項 a, b, c , 関係 R に対して $R(a, b) \wedge \neg R(a, c)$ を考えるようなものである。

すなわち、チャートシステムのモデルは n 項構造 M からなる。ここで M とは、各ソート α に対して、問題を満足させるような言語である。また、 M のドメインは集合 $\{a^M | a \text{ はソート } \alpha \text{ の名前}\}$ としてあらわされる。ここで a^M は M の要素を名称 a に割り当てることをいう。この前提にならい、我々は言語的な公理を、各ソートからちょうど一つ名称付けられた全てのオブジェクトに対して仮定できる。

ここで明らかな点をのぞくとしても、モデルにおいてチャートが真となる意味を述べる必要がある。モデルにおいてチャートが真となるかどうかは、単に同じ名前が同一の列に並ぶか否かといったことである。もしそうならば、それらは同一のものと特定しなければならない。もし、異なる列において同一の名前が存在するならば、それらは異なるものとして扱わねばならない。もしチャートにおいて、これら 2 つの状態が保持されているならば、チャートはモデルにおいて真であるという。Barwise らによれば、チャートによって推論を行うためには、以下のような規則が必要である [6]。

=-Extract

チャートにおいて、もし a と b が同じ列に存在するならば、文章 $a=b$ と推論されることが正当なものであるとする。これはチャートにおける真偽値条件に正当なものである。

=-Apply

1. a はソート α にラベル付けされており, b はソート β にラベル付けされているとする. $a=b$ と a が α の n 列目に位置するならば, b は β の n 列目に位置する.
2. a はソート α にラベル付けされており, b はソート β にラベル付けされているような, $a=b$ が確証されている. このとき, a も b もチャート中にないとす. また, チャートの中には空である列が存在するとする. このとき, その空の列に a と b を各々挿入しても良い.
3. a も b もチャート中には存在しないとする. また, 列 α と列 β の同じ行が空であるとする. このとき, $a=b$ から a を列 α に挿入し, b を列 β に挿入してもよい.

$\neq$ -Extract

もし a と b がチャートにおいて異なる行に存在するならば, $a \neq b$ である.

$\neq$ -Apply

1. a はソート α にラベル付けされており, b はソート β にラベル付けされているとする. a はチャートの n 行目に存在しているが, b は存在していないとする. そして, n 以外の行にも名前なしの部分があるとする. このとき $a \neq b$ より, b をその名前なしの行に挿入できる.
2. a はソート α にラベル付けされており, b はソート β にラベル付けされているとする. a はチャートの n 行目に存在しているが, 列 β の n 行目以外には何も含まれていないものとする. このとき $a \neq b$ より, その行に b を挿入できる.
3. a はソート α にラベル付けされており, b はソート β にラベル付けされているとする. a も b もチャートに存在せず, 列 α, β の異なる行 m, n において α も β も名称がないとする. このとき, そのような行 m, n に対して, a を行 m に, b を行 n に挿入できる.
4. a はソート α にラベル付けされており, b はソート β にラベル付けされているとする. a も b もチャートに存在せず, 列 α, β の異なる行 m, n において各々ひとつだけ名称がないとする. このとき, そのような行 m, n に対して, a を行 m に b を行 n に挿入できる.

Column completion

もし完全に, 列 α の n-1 の名称が列 α に挿入されたならば, 残った名称は余って

いる空白の行に挿入してよい.

Name insertion

もし, a がソート α であり, 列 α の名称がない行が完全に空ならば, a を列 α のそのような任意の行に挿入できる.

Empty Chart

空のチャートは正当化なしで断定できる.

Weakening

全ての名称はチャートから削除できる.

以上の規則によってチャートの形式化が行われる. 本稿では, 先に示した型階層論理に対して, チャート表現が有効であると考え. なぜならば, 型階層論理で扱うソート記号はチャートにおける順序表現が明快であるばかりでなく, well-formed chart となる条件制約を満たすことができるためである. しかし, 本稿では双ラティスを扱うことがねらいである. それを表現するために, 列に対するソートと行に対するソートを同時に表現するような well-formed の形式化を与えるものとする. 詳しい定義は次章において示す.

第 4 章

双ラティスのダイアグラム化

4.1 セルによる表現

本稿では、型階層論理に対する表現形態に注目する。型階層論理はラティスによって表現できる。しかし、ラティスによる表現は人間には分かりやすいものの、そのまま計算機内に表現するのは困難である。また、ラティスにおける頂点と辺の数が増加すると、辺の交差数が増加し人間の解釈に困難となる。辺の交差数を θ に対して以下の式が導ける [3].

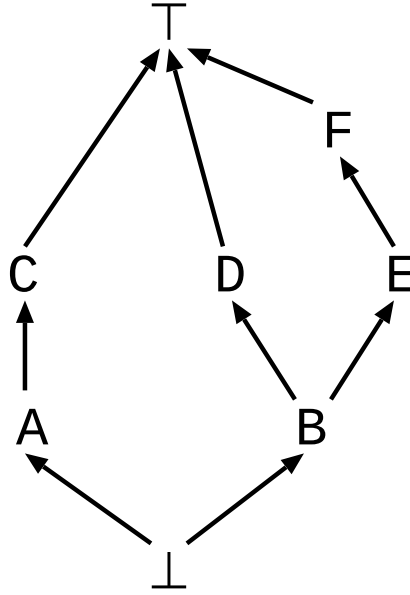
$$\theta \geq \lceil \frac{q}{3p-6} \rceil$$

ここで p とは頂点の数、 q とは辺の数をあらわす。辺の交差数を減らすために、このような制約の元でラティスを作成することは対象とする世界を制限することとなる。実際の問題を扱う上では、人間の理解力にとって交差数が問題になると考えられる。また、ラティスを球面上に描画するとしても、やはり交差する問題は解決できない。

そこで本稿では、辺が交差しないようなセルによる表現を定義し、型階層論理に対する有意性を検証する。ここでいうセルとは 2 次元配列により表現される。このような表現として有名なものに、モルホロジカル・チャートがあげられる。モルホロジカル・チャートは、図によって伝達を意図するばかりでなく、発想あるいは発見の為に考案された表現である。一般に、配列による表現は対比、順序、因果、推移、方向、系統といった関係を意味するのに用いるのが有効とされている [9]。基本的には、複数の部分集合間の対応関係を表す、という配列の特性が表現上の主体的役割を果たしている。このような特性に注目し、セルを以下のように定義する。

セルは複数の領域からなっており、隣接する 2 つの領域間の**境界線 (boundary)** は、上

図 4.1: ソート階層



位下位関係を示すサブソート関係とする．また，各ソート記号は $\top$ から $\perp$ までのパスを示すための index 数値を持ち，2 つの index 数値間に**剰余 0** の関係があるときにサブソート宣言されていると解釈する．

いま，図 4.1 に示すようなソート階層が与えられたとする．このとき，本稿で提案するセルによる表現は表 4.1 のようになる．表 4.1 に記されているように，各ソート記号には index 数値が割り振られている．ここで，index 数値とは素数の階乗の形をしている．index 数値として割り当てられる素数は， $\perp$ から $\top$ までの各パスに対して新たに導入される．このように素数の階乗を割り当てることにより，ラティスにおいて離れた位置にある任意の 2 つのソートがサブソート関係にあるか否かを数値的に判断できる利点がある．また，素数の階乗の形は *Gödel* 数であることより一意性が証明されている¹[12]．各パスに対する素数の割り振りは，例えば図 4.2 のようになる．すなわち，図 4.1 の $\{\perp \rightarrow A \rightarrow C \rightarrow \top\}$ のパスには素数 2 が割り当てられ，深さに従って 2 の階乗が割り当てられていく．また， $\{\perp \rightarrow B \rightarrow D \rightarrow \top\}$ のパスには素数 3 が割り当てられている．このように，異なるパス

¹合成数に対する分解の一意性を次の証明で与える． N が 2 通りの素因数分解がされたとする．このとき， $N = p_1 p_2 \dots p_n = p'_1 p'_2 \dots p'_m$ とすれば， $p'_1 \mid p_1 p_2 \dots p_n$ である．それを p_1 とすれば p_1, p'_1 ともに素数より $p_1 = p'_1$ ．以下同様に $p_2 = p'_2, \dots, p_n = p'_m$ がいえる．

図 4.2: 各パスに対する素数の与え方

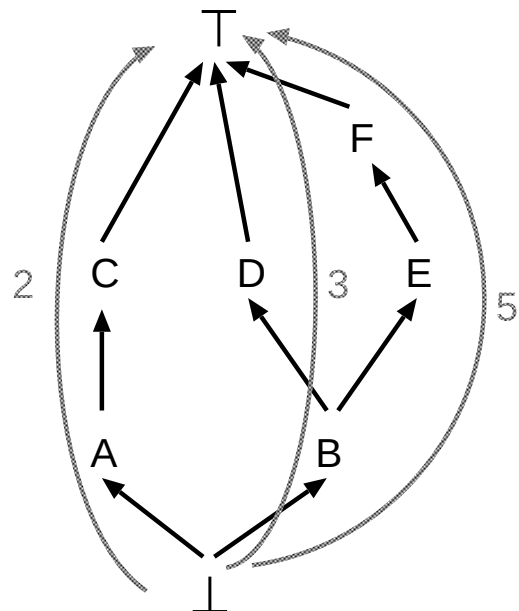


表 4.1: セル表現の例

$\perp$	A^2	C^{2^2}		
	B^3	D^{3^2}		$\top^{2^2 \times 3^2 \times 5^2}$
		$E^{3 \times 5}$	$F^{3 \times 5^2}$	

に応じて異なる素数の階乗を割り振ることにより関係のあるソート記号のみが剰余 0 の関係を持つ。ただし、各パスに対する素数の割り当ては、ソート宣言される順番によって変わる。

ここで示されたセル形式の表現について注目してみる。いま表 4.1において、ソート記号 A, E に着目してみると、index 数値は各々 $2, 3 \times 5$ となる。この index 数値に示されているようにソート記号 E の index 数値 3×5 はソート記号 A の index 数値 2 で割りきることができないので、関係が無いということが分かる。一方、ソート記号 B, F に着目してみると、index 数値は各々 $3, 3 \times 5^2$ であることより、ソート記号 F の index 数値 3×5^2 はソート記号 B の index 数値 3 で割りきることができる。よって、ソート記号 F はソート記号 B の上位概念であることが即座に分かる。

すなわち、 $\perp$ から $\top$ までのパスが異なるということを、素数を用いることによって互いに素という表現に変えた。このことは、計算機上での処理を考慮しているばかりでなく、非常に高速な判別が可能であることを示している。また、双ラティスを表現するためにも必要なものである。一つのラティスを 1 次元上に描画することにより、テーブルによる表現が可能となる。ここで、セルとは、左方向へいくほど上位概念を表す 1 つのラティスと、下方向へいくほど上位概念を表すもう 1 つのラティスからなる表形式のダイアグラムである。次に、具体的な描画のために素数の割り当て方を定義し、セル形式によるダイアグラムの Syntax を定義する。

4.2 Syntax

4.2.1 Alphabet

セルにおける言語 L は以下の記号からなる。

	セル
$P, Q, \dots$	ソート述語
$x, y, \dots$	ソート記号
$2, 3, \dots$	素数
$\neg$	否定記号

このとき、縦に並んだセルを列 (row)、横に並んだセルを行 (column) と呼ぶ。また、行と列に対して、セルとセルの間を境界 (boundary) と呼ぶ。あるセルに隣接したセルを adjacent-cell と呼ぶ。特に左右上下それぞれに対して、adjacent-right-cell, adjacent-left-cell, adjacent-upper-cell, adjacent-lower-cell と呼ぶ。また、ソート述語、ソート記号は各々最大ソート $\top$ と最小ソート $\perp$ を含む。

4.2.2 Signature

組 $\langle S, P, P_N, N, R_S, R_P, D, T \rangle$ をセルにおける言語 L のシグネチャといい、以下のように定義する。

S	ソート記号
P	ソート述語
P_N	素数
N	否定記号
R_S	サブソート関係
R_P	述語の部分関係
D	描画規則
T	演繹規則

ここで、描画規則 (Drawing rules) とは、各ソート記号とソート述語に素数を割り振りセルに描画する規則である。また、演繹規則 (Deduction rules) とはセルに対して論理演算を実行するものである。これらは次で定義する。

4.2.3 セルにおける描画規則

セルの描画規則および書き換え規則を以下のように定義する。

ソート階層 ($S, \sqsubseteq_S$) を考える。このとき、以下の規則に従って与えられたソート階層をセルの中に描画していく。

[規則 1]

- ・セルにおいて空の行、列は任意に増加 (削除) することができる。

[規則 2] (与えられたソート宣言がセルに存在しないとき)

- ・ $P_i, P_j \in S$ ($i \neq j$) に対して、 $P_i \sqsubseteq_S P_j$ が与えられたとき、 P_i も P_j もセルに存在しないならば P_i を P_i^c とし、 P_j を $P_j^{c^2}$ とする。そして、 P_i^c の *right-cell* に $P_j^{c^2}$ がくるように挿入する。また、 $c := \text{new } c$ とする。ここで、 c は素数の最大値を示すカウンターであり、 $\text{new } c$ とは $\text{new } c > c$ なる新たな素数である。

[規則 3] (与えられたソート宣言の下位概念がセルに存在するとき)

- ・ $P_i, P_j \in S$ ($i \neq j$) に対して、セル内に P_i^α が存在し $P_i \sqsubseteq_S P_j$ が与えられたとき、 P_j は P_i^α の *adjacent-right-cell* に $P_j^{\alpha \times \bar{c}}$ として挿入する。ここで、 α とは複数の素数の積であり、 $\bar{c}$ とは α を基準分解した中の素数の最大値を表す。

ただし、 P_i^α の *adjacent-right-cell* に α で割り切れる index を持つソート記号があるときは、 P_j を $P_j^{\alpha \times c}$ として挿入し、 $c := \text{new } c$ とする。

[規則 4] (与えられたソート宣言の上位概念がセルに存在するとき)

・ $P_i, P_j \in S$ ($i \neq j$) に対して, セル内に P_j^α が存在し $P_i \sqsubseteq_S P_j$ が与えられたとき, P_i は P_j^α の adjacent-left-cell に P_i^α として挿入し, P_j^α を $P_j^{\alpha \times \underline{c}}$ と書換える. ここで, $\underline{c}$ は α を基準分解した中の素数の最小値を表す.

ただし, P_j^α の adjacent-left-cell に α を割り切る index を持つソート記号があるときは, P_i を P_i^c として挿入し, P_j^α を $P_j^{\alpha \times c}$ と書換える.

[規則 5] (与えられたソート宣言の上下概念がセルに存在するとき)

・ $P_i, P_j \in S$ ($i \neq j$) に対して, セル内に P_i^α, P_j^β が存在し $P_i \sqsubseteq_S P_j$ が与えられたとき, P_j^β を P_i^α の adjacent-right-cell に $P_j^{\alpha * \beta}$ として挿入する. ここで, $*$ とは最小公倍数をあらわす.

[規則 6]

・ いま index 数値 α, β, γ 間に $\beta | \gamma, \alpha | \beta$ なる関係があるとする. このとき, P_i^β の adjacent-left-cell に P_h^α が存在しないならば P_i^β は adjacent-left-cell に移動可能である. また, P_i^β の adjacent-right-cell に P_j^γ が存在しないならば P_i^β は adjacent-right-cell に移動可能である.

[規則 7]

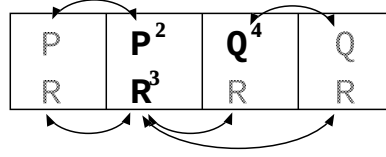
・ 2つのソート階層 S_1, S_2 に対して, S_1 を右方向が上位となるように描画し, S_2 を下方向が上位となるように描画する. このとき, 任意のセル B の要素を E_B とする.

いま, $E_{upper-cellB}$ と $E_{left-cellB}$ が空でないならば, $E_{upper-cellB}$ のソート記号 P_i^α ($1 \leq i \leq n, 1 \leq \alpha \leq k$) と $E_{left-cellB}$ のソート記号 ψ_j^β ($1 \leq j \leq m, 1 \leq \beta \leq l$) から $P_i^\alpha \psi_j^\beta$ をセル B に導きだせる.

以上の7つの規則を用いて, 与えられた2つのソート階層を表形式に変換する. ここで規則6に注目する. 規則6は小さいセルを描画するための規則である. 規則6が行われる場合を図4.3に示した.

ここで, 移動する前と移動後のセルを見ると, セルの与える意味は変化しない. なぜならば, 移動可能な記号は関係のない素数に対してのみ可能となるからである. これはラ

図 4.3: 規則 6



ティスを描画した際に、関係が明記されていない記号の高さが不明瞭であることと同値である。逆に言えば、セルにおける表現はラティスで表す方法と同等の力を持つことを示している。

上の7つの描画規則は、より形式的に以下のようなアルゴリズムを与えることができる。

Algorithm 4 (Prime Number Labeling)

1. [新たな素数を割り振るための素数カウンター]

素数のカウンターを c とする. $c \leftarrow 2$

2. [ソート宣言の実行]

$\alpha \sqsubseteq \beta$ の形のソート宣言をする. そうでないならば, goto Step 4

3. [既にセルに含まれているかの判定]

α が既にセルに存在する. そうでなければ, goto Step 3.2

3.1

β が既にセルに存在すれば, $label\{\beta\} \mid label\{\gamma\}$ なる $\gamma (\ni \beta)$ に対して, $label\{\gamma\} \leftarrow label\{\gamma\} \times label\{\alpha\}$. そうでなければ, $label\{\beta\} \leftarrow c$ とし, $c \leftarrow new\ c$. goto Step2

3.2

β が既にセルに存在すれば, $label\{\alpha\} \leftarrow c$ とし, $label\{\beta\} \mid label\{\gamma\}$ なる γ に対して, $label\{\gamma\} \leftarrow label\{\gamma\} \times c$. また, $c \leftarrow new\ c$ とする. goto Step2

3.3

$label\{\alpha\} \leftarrow c$, $label\{\beta\} \leftarrow c^2$. また, $c \leftarrow new\ c$ とする. goto Step2

4

終了

Algorithm 4 を2回繰り返し, 双ラティスを各々1次元のセルとして保存する. これらを行と列で描画することにより, 双ラティスを反映したセルが得られる. Algorithm 4 の流れを図 4.4に示す.

図 4.4: システムの流れ

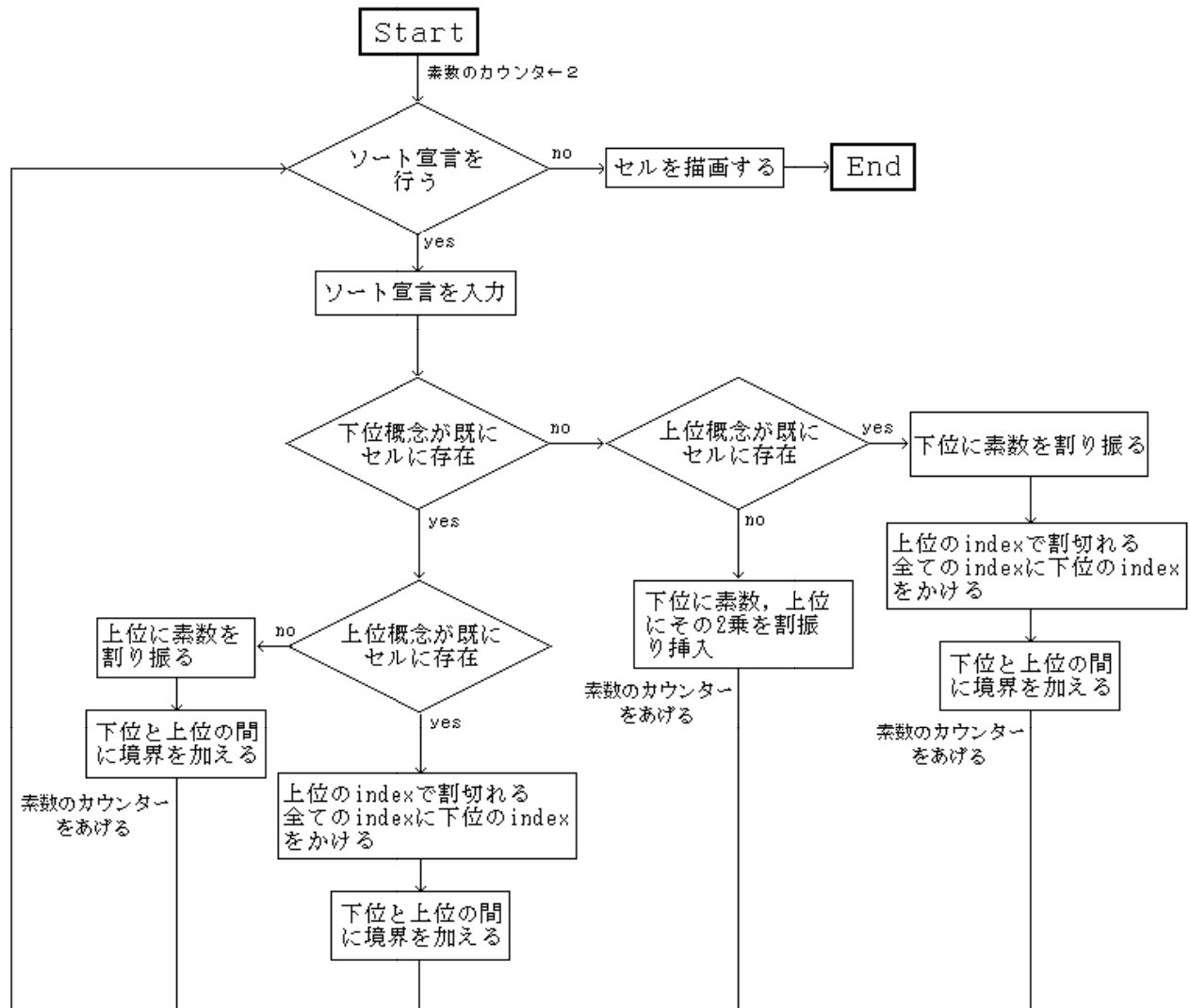
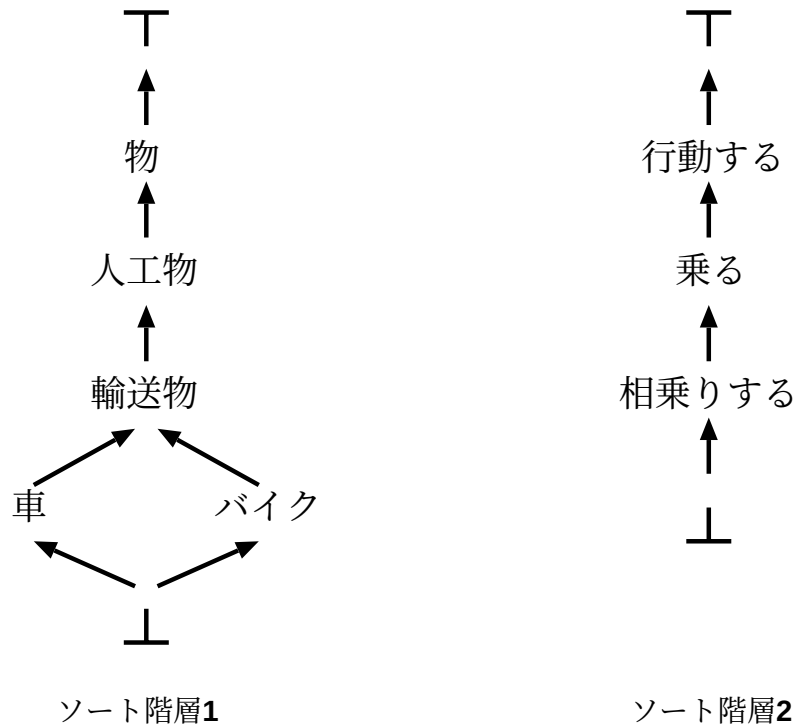


図 4.5: 双ラティスの例



例えば，図 4.5に示すような 2 つのラティスを与えられたとする．このとき，本稿で提案するセルによる表現は表 4.2のように描画できる．このようにテーブルによって描画することにより，斜め方向へも推論が可能となる．

いま，

車に相乗りした → 物に対して行動した

というような推論を行うとき，異なる 2 つのラティスを用いる場合は 車 から 物 まで 車 → 輸送物 → 人工物 → 物 の推論が必要であり，また 相乗りした から 行動した まで 相乗りする → 乗る → 行動する の推論が必要となる．しかし，セルによる表現では **相乗りする²(車²)** と **行動する⁸(物⁴⁸)** を直接比較し，それぞれの index 数値を見ることにより即座に推論が実行される．このような斜め方向への推論を可能とするために，次にセルにおける Semantics を定義する．

表 4.2: セル表現の例

$\perp \perp$	相乗りする ²	乗る ⁴	行動する ⁸	$\top$
車 ²	相乗りする ² (車 ²)	乗る ⁴ (車 ²)	行動する ⁸ (車 ²)	$\top$ (車 ²)
バイク ³	相乗りする ² (バイク ³)	乗る ⁴ (バイク ³)	行動する ⁸ (バイク ³)	$\top$ (バイク ³)
輸送物 ¹²	相乗りする ² (輸送物 ¹²)	乗る ⁴ (輸送物 ¹²)	行動する ⁸ (輸送物 ¹²)	$\top$ (輸送物 ¹²)
人工物 ²⁴	相乗りする ² (人工物 ²⁴)	乗る ⁴ (人工物 ²⁴)	行動する ⁸ (人工物 ²⁴)	$\top$ (人工物 ²⁴)
物 ⁴⁸	相乗りする ² (物 ⁴⁸)	乗る ⁴ (物 ⁴⁸)	行動する ⁸ (物 ⁴⁸)	$\top$ (物 ⁴⁸)
$\top$	相乗りする ² ($\top$)	乗る ⁴ ($\top$)	行動する ⁸ ($\top$)	$\top$ ($\top$)

4.2.4 セルを用いた推論

セルにおける演繹的推論を考える．まず初めに型階層論理において行える推論について考察する．型階層論理では，以下のような推論規則がある．ただしここで， 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．また， $s \parallel s'$
は $(s \sqcap s') = \perp$ を表し排他的であるという． $s \downarrow_{s_i} s'$ は $(s \sqcup s') = s_i$ を表し全域的であると
定義する．

$$\frac{s \sqsubseteq s'}{s'' \sqcap s \sqsubseteq s'' \sqcap s'} \quad (Introduction),$$

$$\frac{s \sqcup s' \sqsubseteq s \sqcup s'' \quad s \parallel s' \quad s \parallel s''}{s' \sqsubseteq s''} \quad (Elimination),$$

$$\frac{s \sqsubseteq s' \quad s' \sqsubseteq s''}{s \sqsubseteq s''} \quad (Translation).$$

また，節形式としての推論規則や，知識ベースに反映させる為の推論規則としては以下の
のものも考えられる．

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

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

$$\frac{(\neg s)(t) \vee C \quad s'(t') \vee C' \quad s' \sqsubseteq s}{\theta(C \vee C')} \quad (\text{Subsort}),$$

$$\frac{(\neg s)(t) \vee C \quad s(t') \vee C'}{\theta(C \vee C')} \quad (\text{Negation}),$$

$$\frac{s \sqcup s'(t) \vee C \quad (\neg s)(t') \vee C'}{\theta(s'(t) \vee C \vee C')} \quad (\text{Disjunction}),$$

$$\frac{s(t) \vee C \quad s'(t') \vee C' \quad s \parallel s'}{\theta(C \vee C')} \quad (\text{Exclusivity}),$$

$$\frac{s_i(t) \vee C \quad (\neg s')(t') \vee C' \quad s \mid_{s_i} s'}{\theta(s(t) \vee C \vee C')} \quad (\text{Totality}).$$

これらの規則は、セルにおいても同様に定義できる。(Introduction) と (Elimination) はセルにおける領域の包含関係に変換できる。それらを、図 4.6 に示す。また、(Translation) に関しても同様に表すことができる。それを図 4.7 に示す。

4.2.5 セルにおける演繹規則

前に示したようなセルの操作を行うために、ここで演繹的推論規則を定義する。図の変形を実行するために、以下のような図形操作の定義を与える。

[規則 1] Dual Lattice Coupling

ソート宣言されたソート述語とソート記号に対して、ソート述語を行、ソート記号を列として描画する。そして、それら両方からなる単項述語をソート述語の行とソー

図 4.6: セル表現における推論

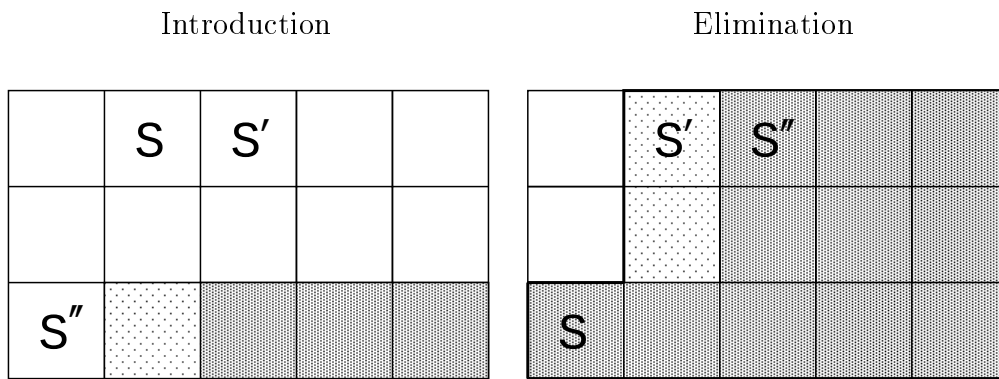
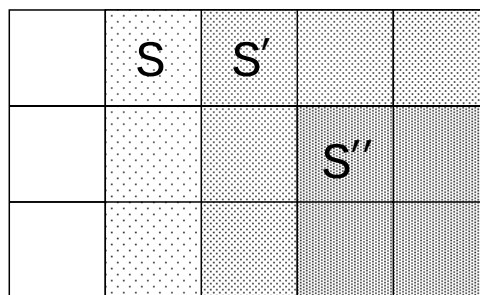


図 4.7: Translation



ト記号の列からなるセルに描画する.

[規則 2] Insertion of Predicate Subsumption

セルに含まれる述語に対して, その述語に割り振られた index 数値が割りきるような index 数値を持つ述語をセルに加える.

[規則 3] Insertion of Sort Subsumption

セルに含まれるソートに対して, そのソートに割り振られた index 数値が割りきるような index 数値を持つソートをセルに加える.

[規則 4] Propagation of Negation

セルに含まれる述語に否定記号が加わるとする. このとき, その述語に割り振られた index 数値が割りきられるような index 数値を持つ述語とソートにも否定記号を加える.

[規則 5] Conjunctive Unification

ある前提 A に対して, 異なる前提 B が論理積で結ばれているとする. このとき, A から得られるセルに, Insertion of Sort Subsumption と Insertion of Predicate Subsumption を B に適用して得られるセルを書き加えることができる.

[規則 6] Disjunctive Unification

ある前提 A に対して, 異なる前提 B が論理和で結ばれているとする. このとき, 述語とソートの各々に対して, A の index 数値と B の index 数値の最小公倍数の index 数値を持つ単項述語をセルに加え, その述語に対して Insertion of Sort Subsumption と Insertion of Predicate Subsumption を適用することができる.

もしくは, A から得られるセルに対して, B に割り当てられている述語とソート両方の index 数値で割りきることができないものを削除する.

[規則 7] Totality

任意の記号に対して, その下位概念が全域的かつ唯一つならばセルに加えて良い.

[規則 8] Weakening

任意の行, 列をセルから削除してもよい.

また、特に否定に関して次の規則を定義する。ただし、全域的であるか否かはユーザーによって与えられているものとする。

[規則 9] Predicate Negation as Totality

任意のセルにおいて、セルに含まれる述語が全域的かつ**全て**否定されれば、その index 数値で割りきることができる直接の述語を否定記号付きでセルに書き加える。

[規則 10] Sort Negation as Totality

任意のセルにおいて、セルに含まれるソートが全域的かつ**全て**否定されれば、その index 数値で割りきることができる直接のソートを否定記号付きでセルに書き加える。

これらの規則により、結論となる式がセル中に存在すれば帰結できることを示せる。特に、規則 1 に関する例を図 4.8 に示す。また規則 2, 3, 4 に関しては各々図 4.9, 図 4.10, 図 4.11 に示す。規則 5, 6 に関しては図 4.6 に示したとおりである。

型階層論理における推論規則は全て、index 数値を用いて判断することもできるが、領域を考えることにより、それらの包含関係がさらに理解しやすいものとなる。この領域という概念は可能世界を描画することが可能である。ベン図において、領域 p の補集合に影 (空集合) が無いことから p が得られるという規則は、 $\neg \Diamond \neg p = \Box p$ を示していることと同値である。それと同様に、セルにおいても集合の概念を用いて、推論を記述できることが利点としてあげられる。また、セルにおいて Weakening が成り立つことは、セルにおける部分領域がラティスを形成することを示している。

4.3 Semantics

セルにおいて斜め方向へ推論させるために、以下のような意味論を与える。このことにより、双ラティスを反映したような推論が効率的に行える。

セル内の $P_i(\in S)$ と $\phi_j(\in S')$ に対して、 $P_i(\phi_j)$ は $P'_i(x : \phi_j)$ を意味する。

ここで、 P_i はソート記号 P'_i を述語として用いることを示している。また $(x:\phi_j)$ とは、 x が型 ϕ の要素であることをしめす。すなわち、 x, ϕ_j には $\llbracket x \rrbracket \in \llbracket \phi_j \rrbracket \subset U$ の関係がある。

図 4.8: ソート述語 P とソート x に対する規則 1

$\perp$	$P^2 S^3$	Q^4	$R^8 T^{12}$	$\top$
x^2				
$y^4 \rightarrow$	$P^2(y^4)$			
z^8				
$\top$				

図 4.9: セル中の $P(x)$ に対する規則 2

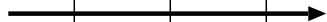
$\perp$	$P^2 S^3$	Q^4	$R^8 T^{12}$	$\top$
x^2				
y^4	$P^2(y^4)$	$Q^4(y^4)$	$R^8(y^4)$ $T^{12}(y^4)$	$\top(y^4)$
z^8				
$\top$				

図 4.10: セル中の $P(x)$ に対する規則 3

$\perp$	$P^2 S^3$	Q^4	$R^8 T^{12}$	$\top$
x^2				
y^4	$P^2(y^4)$	↓		
z^8	$P^2(z^8)$			
$\top$	$P^2(\top)$			

図 4.11: セル中の $R(y)$ に対する規則 4

$\perp$	$P^2 S^3$	Q^4	$R^8 T^{12}$	$\top$
x^2			$\neg R^8(x^2)$	↑
y^4	$\neg P^2(y^4) \neg Q^4(y^4) \neg R^8(y^4)$			
z^8	←			
$\top$				

なお, $\llbracket \cdot \rrbracket$ は解釈を表し, U とはすべてのオブジェクトからなる世界を表す. また, サブソート宣言を implication として扱う.

以上により, $P \sqsubseteq_{S_1} Q$ ならば,

$$\forall \phi (P'(x : \phi) \rightarrow Q'(x : \phi))$$

$\phi \sqsubseteq_{S_2} \psi$ ならば,

$$\forall P' (P'(x : \phi) \rightarrow P'(x : \psi))$$

と定義できる.

したがって, セル内において $P(\psi)$ の右下に $Q(\phi)$ があるとき, $P'(x : \psi)$ から $Q'(x : \phi)$ への implication が可能となる. ただし, 述語と項の部分において, 割り当てられた index 数値に剰余 0 の関係があるものとする. より形式的には以下のように定義できる.

4 字組み $\langle S, T, :, \rightarrow \rangle$ を考える. ここで, $\langle S, T, : \rangle$ とは classification であり, ‘ $:$ ’ は, 型 T からトークン S を選択することを示す. $\rightarrow$ は包含関係である.

このとき, $\forall s \in S, \forall t \in T$ に対して,

$$\text{if } s : t \text{ and } t \rightarrow t' \text{ then } s : t' \quad (\text{Strong Facticity})$$

が成り立つ [5].

また, チャートを用いた表現のところで定義したような well-formed の形式化がここでも成り立つ. なぜならば, セルに代入されるソート記号は, 既に順序付けられたソートであり, 同一のものが同じ行, 列には存在しないためである. もし存在するならば, それはセルに挿入される段階で排除され, index 数値の書き換えによって単一化されている. すなわち, well-formed の制約条件を満たしているといえる. 特にセル表現では, 行および列にそれぞれ別のソートを挿入したものとなっている. その際に, ソートが挿入されるセルは単一のソート記号が入り, そのまま名前を保存した形で属性が継承されているので, 制約条件を超えるものではないことがいえる. ただし, ここでは型階層論理の性質上 Column completion などが成り立たないが, その他の規則に関しては成立する. よって, 本稿で提案するセルは well-formed chart といえる.

第 5 章

比較検証

5.1 図を用いたシステム

最近の研究では、計算機における推論に図形を用いて行おうという試みも多く行われている。Swoboda[7] は、オイラー円とベン図を実際に計算機上にインプリメントする際の複雑さについて考察している。その研究では次のように図形を実装している。

オイラー円やベン図を 3 つ組み $\langle \Gamma, \Delta, \Sigma \rangle$ で表す。

ここで、 Γ は整合論理図形の文法集合である。 Δ は推論システム。そして、 Σ はシステムのセマンティクスを表す。また、オイラー円やベン図において用いられる図形言語を定義し、推論に用いられる数学的文法の定義を行うことができる。また、演繹システムの推論規則も定義できる。同様に意味論を付加することもでき、オイラー円とベン図で用いる表現言語の健全性と完全性の証明も与えることができる。そして、これらの図形を実装するための DAG を以下のように形式的に定義する。

オイラー円やベン図の実装形式を 3 つ組み $\langle G, D, S \rangle$ で表す。

ここで、 G は実装システムの文法、 D は実装システムの推論システム、そして S はその意味論である。これら $\langle G, D, S \rangle$ は $\langle \Gamma, \Delta, \Sigma \rangle$ と同等である。 DAG における言語はラベル付けされた頂点と有向辺によって表現される。

例えば、図 5.1 に示されるようなベン図が与えられたとき、ここで提案しているインプリメント表現は図 5.2 のように表現される。ここで、図 5.2 に示される DAG は、全体集合 V が、領域 A, B, A の補集合、 B の補集合によって細分され、考えられる領域は $AB, \bar{A}B, A\bar{B}, \bar{A}\bar{B}$

図 5.1: ベン図による表現

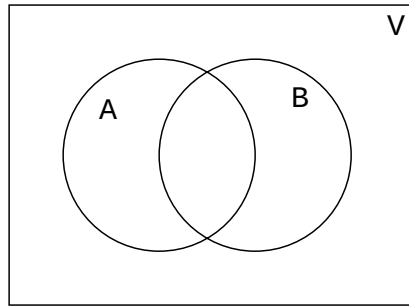
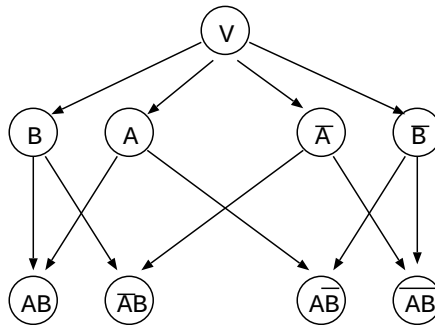


図 5.2: ベン図の実装のための表現



となることを頂点と有向辺を用いて表現している．このような表現により，ベン図の描画規則に従うような DAG の描画規則が 3 つ組み $\langle G, D, S \rangle$ によって定義されている．すなわち， DAG においても完全性と健全性が保証されたものとなっている．インプリメントするという点に関して言えば，リスト処理として実装することは可能となっている．しかし，表現的には**辺の交差および頂点の数**が問題としてあげられる．

一方，村田ら [13] の研究によれば，算術問題に対して問題解決過程を図を表示することによって理解しやすくするようなシステムを構築している．このシステムは，算数の文章題（算術問題）に関して与えられた文章から図形を表示することによって理解しやすいものとしている．いっけん算術問題は高等数学に比べて容易であるように見られるが，実際は図を用いて問題を解決することが多いためである．具体的には，旅人算や植木算に注目している．例えば，以下のような問題が与えられたとき，図 5.3 や図 5.4 のような描画をするシステムを構築している．

図 5.3: 旅人算の描画

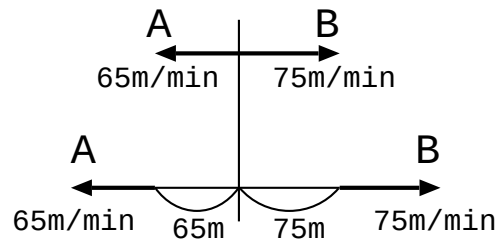
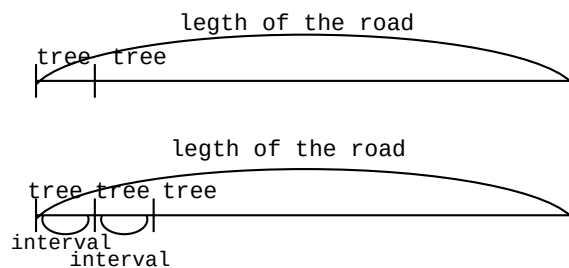


図 5.4: 植木算の描画



[問題 1]

A, B の 2 人が同じところから, 反対方向へ向かって歩き出しました. A は分速 65 メートル, B は分速 75 メートルです. 15 分後にはどれだけ離れますか.

[問題 2]

長さ 120 メートルの道路に 3 メートルおきに木を植えます. 両端にも木を植えると何本の木が必要ですか.

このように, 植木算と旅人算に注目したシステム実現のために, 表 5.1 のような描画表現を定義している.

このような問題解決システムにより, 問題に対する解を得ることが有利なものとなった. しかし, 言語によって与えられた表現は曖昧性や多義性を含むものである. 植木算でいえば, 端に木を植えるか否かなどといった明記されていない状況に対して描画を実現するのが困難となる場合もある.

表 5.1: 図構成ルール

ルール	対応する言語表現
点を描く	木, 杭
線を描く	道路, 電車
距離 x の 2 点を描く	メートルおき, 間隔
ある図上に図を配置する	植える, 置く, 打つ
2 点を結んだ線を描く	〜と〜の間

5.2 セルを用いたシステム

5.2.1 セルによるシステムの表現

今回, 本稿で扱う型階層論理に注目してみれば, 一般の自然言語における曖昧性や多義性も考えられうる. しかし, 概念間の上位下位関係はユーザーによって与えられるものであり, これらの順序付けに関してみれば曖昧性はないものとする. ユーザーが与える概念関係に対する図形の表記の上で行う推論に着目している. また, ベン図やオイラー円といった, 無限の概念や連続量といった概念はそれほど重要ではなく, セルによる表現を構築することは,それほど難解なものではない.

そこで今回, 前章で定義した Syntax と Semantics に基づいて, 計算機上にインプリメントした. 作成したシステムは, ユーザーが 2 つのソート階層に対する複数のソート宣言を行い, それを入力とした. その上で, システムが入力に基づいたテーブルを作成する. このテーブルに対し, ユーザーが推論を行うことができる. ユーザーが任意のソート記号を入力すると, それに対する上位概念を導出したり, ソート記号間の関係を提示する. また, 2 つのソート記号間に上位下位の関係があるときは, 付加的にオイラー円で明記することとした. index 数値にたいしては, 入力されたソート記号を配列に挿入し, 複数のパスが存在するような場合は新たな素数を割り振って配列に代入した. 素数の割り当て方については, 4 章のアルゴリズム 4.1 に示した.

入出力に関して, 具体的には次のような動作を行う. いま, 下に示すようなソート宣言をユーザーが逐次行うとする. すなわち, 図 5.5 に示すようなソート階層が入力されたとする.

図 5.5: ソート階層 1 の入力例

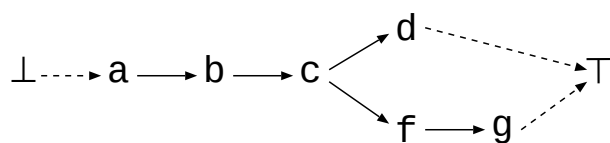
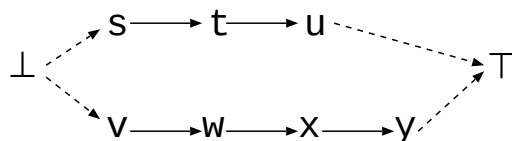


図 5.6: ソート階層 2 の入力例



$$\text{入力} \left\{ \begin{array}{l} a < b \\ b < c \\ c < d \\ c < f \\ f < g \end{array} \right.$$

このとき，システムは入力に基づいて以下のような，ソート階層 1 に関する素数付きの出力をする．

<< sort hierarchy 1 >>
出力 $a_2 \mid b_4 \mid c_8 \mid d_{16} \mid f_{24} \mid g_{72}$

同様に，別のソート階層について，下に示すようなソート宣言をユーザーが逐次行うとする．すなわち，図 5.6 に示すようなソート階層が入力されたとする．

$$\text{入力} \left\{ \begin{array}{l} s < t \\ t < u \\ v < w \\ w < x \\ x < y \end{array} \right.$$

すると同様に，システムは入力に基づいて以下のような，ソート階層 2 に関する素数付

表 5.2: *Planar Dual Lattice*

a2(s2)	a2(v3)	a2(t4)	a2(u8)	a2(w9)	a2(x27)	a2(y81)
b4(s2)	b4(v3)	b4(t4)	b4(u8)	b4(w9)	b4(x27)	b4(y81)
c8(s2)	c8(v3)	c8(t4)	c8(u8)	c8(w9)	c8(x27)	c8(y81)
d16(s2)	d16(v3)	d16(t4)	d16(u8)	d16(w9)	d16(x27)	d16(y81)
f24(s2)	f24(v3)	f24(t4)	f24(u8)	f24(w9)	f24(x27)	f24(y81)
g72(s2)	g72(v3)	g72(t4)	g72(u8)	g72(w9)	g72(x27)	g72(y81)

きの出力をする．

<< sort hierarchy 2 >>

出力 $s2\ v3 \mid t4 \mid u8\ w9 \mid x27 \mid y81$

また，同時に 2 つのソート階層を反映したようなセル表現を行う．この場合，表 5.2 のように出力される．このセル表現が出力されたあとで，ユーザーが推論を行う．

例えば，次のような推論を入力する．

入力 $b(v) \rightarrow b(w)$

このような入力に対しては，*yes* の出力を返し， $b(v)$ の (到達可能な) 上位概念を表示することで，その正当性を見ることができる．出力を表 5.3 に示す．また，上位概念でないものを推論しようとする場合 (例えば $b(v) \rightarrow c(u)$)，同様に表 5.3 を出力し *no* を返す．

5.2.2 論理積を含む表現

次に論理積を含むような上位概念への推論を考える．例えば次のような推論を考える．

入力 $d(v) \wedge a(w) \rightarrow b(w)$

このような入力に対しては，まず $d(v)$ から導出される上位概念を計算し，同時に $a(w)$ から導出される上位概念も計算する．この場合 *yes* の出力を返し， $d(v)$ と $a(w)$ の (到達可能な) **それぞれの上位概念**を表示することで，その正当性を見ることができる．出力を表 5.4

表 5.3: 上位概念への *implication*

$$b(v) \rightarrow b(w) \quad \text{————} \quad \text{yes.}$$

< calculation >

b4(v3)	b4(w9)	b4(x27)	b4(y81)
c8(v3)	c8(w9)	c8(x27)	c8(y81)
d16(v3)	d16(w9)	d16(x27)	d16(y81)
f24(v3)	f24(w9)	f24(x27)	f24(y81)
g72(v3)	g72(w9)	g72(x27)	g72(y81)

表 5.4: *Conjunction* を伴う上位概念への *implication*

$$d(v) \wedge a(w) \rightarrow b(w) \quad \text{————} \quad \text{yes.}$$

< calculation >

	a2(w9)	a2(x27)	a2(y81)
	b4(w9)	b4(x27)	b4(y81)
	c8(w9)	c8(x27)	c8(y81)
d16(v3)	d16(w9)	d16(x27)	d16(y81)
	f24(w9)	f24(x27)	f24(y81)
	g72(w9)	g72(x27)	g72(y81)

に示す。また、上位概念でないものを推論しようとする場合 (例えば $d(v) \vee a(w) \rightarrow f(u)$) は、同様に表 5.4 を出力し *no* を返す。

5.2.3 論理和を含む表現

次に論理和を含むような上位概念への推論を考える。例えば次のような推論を考える。

$$\text{入力} \quad d(v) \vee a(w) \rightarrow d(y)$$

このような入力に対しては、まず $d(v) \vee a(w)$ から導出される共通の上位概念を推論し、その概念から到達可能な上位概念を計算する。この場合 *yes* の出力を返し、 $d(v)$ と $a(w)$ の (到達可能な) **共通の上位概念** を表示することで、その正当性を見ることがで

表 5.5: *Disjunction* を伴う上位概念への *implication*

$$d(v) \vee a(w) \rightarrow d(y) \quad \text{———} \quad yes.$$

$< \textit{calculation} >$

d16(w9)	d16(x27)	d16(y81)
---------	----------	----------

きる．出力を表 5.5に示す．また，上位概念でないものを推論しようとする場合 (例えば $d(v) \vee a(w) \rightarrow f(w)$) は，同様に表 5.5を出力し *no* を返す．

システムにおいて，論理積と論理和に関するアルゴリズムを以下のように記述する．

Algorithm 5.1 (Transformation of Conjunction and Disjunction)

[$P(\alpha)$, $Q(\beta)$ が前提にあるときについて考える.]

1.1 [ソート述語のカウンター初期化]

ソート述語に対して，配列 A^1 を用意し， $A_i^1 \leftarrow 0$

1.2 [ソートのカウンター初期化]

ソートに対して，配列 A^2 を用意し， $A_j^2 \leftarrow 0$

2 [P_α の上位概念を示す]

前提にある P_α に対して， $A_P^1 \mid A_R^1$, $A_\alpha^2 \mid A_\gamma^2$ なる R と γ が存在する．このとき， $A_R^1 \leftarrow 1$, $A_\gamma^2 \leftarrow 1$.

3 [Q_β の上位概念を示す]

前提にある Q_β に対して， $A_Q^1 \mid A_{R'}^1$, $A_\beta^2 \mid A_{\gamma'}^2$ なる R' と γ' が存在する．このとき， $A_{R'}^1 \leftarrow A_{R'}^1 + 1$, $A_{\gamma'}^2 \leftarrow A_{\gamma'}^2 + 1$.

4 [論理積]

前提の 2 つの論理式が論理積で結ばれていれば $A_i^1 = 0$, $A_j^2 = 0$ なる記号を削除．

5 [論理和]

前提の 2 つの論理式が論理和で結ばれていれば $A_i^1 \leq 1$, $A_j^2 \leq 1$ なる記号を削除．

6

終了

表 5.6: *Planar Dual Lattice involves negation*

$\neg a2(s2)$	$a2(v3)$	$\neg a2(t4)$	$\neg a2(u8)$	$a2(w9)$	$a2(x27)$	$a2(y81)$
$\neg b4(s2)$	$b4(v3)$	$\neg b4(t4)$	$\neg b4(u8)$	$b4(w9)$	$b4(x27)$	$b4(y81)$
$\neg c8(s2)$	$c8(v3)$	$\neg c8(t4)$	$\neg c8(u8)$	$c8(w9)$	$c8(x27)$	$c8(y81)$
$\neg d16(s2)$	$d16(v3)$	$d16(t4)$	$d16(u8)$	$d16(w9)$	$d16(x27)$	$d16(y81)$
$\neg f24(s2)$	$f24(v3)$	$\neg f24(t4)$	$\neg f24(u8)$	$f24(w9)$	$f24(x27)$	$f24(y81)$
$g72(s2)$	$g72(v3)$	$g72(t4)$	$g72(u8)$	$g72(w9)$	$g72(x27)$	$g72(y81)$

5.2.4 否定を含む表現

次に否定について注目する．あるソート記号に対して，それが否定されればそのソート記号から到達可能な全ての下位概念も否定される．例えば，表 5.2 において $\neg f(u)$ という事実が与えられたとき，その出力は表 5.6 のようになる．

いま，セルにおける全てのソート記号からなる集合を PDL とする．このとき，否定に関して以下のように定義する．

$P^\alpha(x^\beta) \in PDL$ が存在するとする．ここで， α に関して基準分解 (canonical decomposition) し， $\alpha = p_1^l p_2^m \dots p_k^n$ (p_i : 素数) となる素数の積とする．このとき， $\neg P^\alpha(x)$ が事実として与えられると，列方向に対して以下の推論が行われる．

$$\begin{aligned}
\neg P^\alpha(x) &= \neg P^{p_1^l p_2^m \dots p_k^n}(x) \\
&\vdash \neg P^{p_1^{l-1} p_2^m \dots p_k^n}(x) \sqcap \neg P^{p_1^{l-2} p_2^m \dots p_k^n}(x) \sqcap \dots \sqcap \neg P^{p_1^0 p_2^m \dots p_k^n}(x) \sqcap \\
&\quad \neg P^{p_1^l p_2^{m-1} \dots p_k^n}(x) \sqcap \neg P^{p_1^{l-1} p_2^{m-1} \dots p_k^n}(x) \sqcap \dots \sqcap \neg P^{p_1^0 p_2^{m-1} \dots p_k^n}(x) \sqcap \\
&\quad \vdots \\
&\quad \neg P^{p_1^l p_2^0 \dots p_k^n}(x) \sqcap \neg P^{p_1^{l-1} p_2^0 \dots p_k^n}(x) \sqcap \dots \sqcap \neg P^{p_1^0 p_2^0 \dots p_k^n}(x) \sqcap \\
&\quad \neg P^{p_1^l p_2^m \dots p_k^{n-1}}(x) \sqcap \neg P^{p_1^{l-1} p_2^m \dots p_k^{n-1}}(x) \sqcap \dots \sqcap \neg P^{p_1^0 p_2^m \dots p_k^{n-1}}(x) \sqcap \\
&\quad \vdots \\
&\quad \vdots \\
&\quad \neg P^{p_1^l p_2^0 \dots p_k^0}(x) \sqcap \neg P^{p_1^{l-1} p_2^0 \dots p_k^0}(x) \sqcap \dots \sqcap \neg P^{p_1^0 p_2^0 \dots p_k^0}(x)
\end{aligned}$$

また， β に関して基準分解し， $\beta = p_1^h p_2^i \dots p_k^j$ (p_i : 素数) となる素数の積とする．このとき， $\neg P(x^\beta)$ が事実として与えられると，行方向に対して以下の推論が行われる．

$$\begin{aligned}
\neg P(x^\beta) &= \neg P(x^{p_1^h p_2^i \dots p_k^j}) \\
&\vdash \neg P(x^{p_1^{h-1} p_2^i \dots p_k^j}) \sqcap \neg P(x^{p_1^{h-2} p_2^i \dots p_k^j}) \sqcap \dots \sqcap \neg P(x^{p_1^0 p_2^i \dots p_k^j}) \sqcap \\
&\quad \neg P(x^{p_1^h p_2^{i-1} \dots p_k^j}) \sqcap \neg P(x^{p_1^{h-1} p_2^{i-1} \dots p_k^j}) \sqcap \dots \sqcap \neg P(x^{p_1^0 p_2^{i-1} \dots p_k^j}) \sqcap \\
&\quad \vdots \\
&\quad \neg P(x^{p_1^h p_2^0 \dots p_k^j}) \sqcap \neg P(x^{p_1^{h-1} p_2^0 \dots p_k^j}) \sqcap \dots \sqcap \neg P(x^{p_1^0 p_2^0 \dots p_k^j}) \sqcap \\
&\quad \neg P(x^{p_1^h p_2^i \dots p_k^{j-1}}) \sqcap \neg P(x^{p_1^{h-1} p_2^i \dots p_k^{j-1}}) \sqcap \dots \sqcap \neg P(x^{p_1^0 p_2^i \dots p_k^{j-1}}) \sqcap \\
&\quad \vdots \\
&\quad \vdots \\
&\quad \neg P(x^{p_1^h p_2^0 \dots p_k^0}) \sqcap \neg P(x^{p_1^{h-1} p_2^0 \dots p_k^0}) \sqcap \dots \sqcap \neg P(x^{p_1^0 p_2^0 \dots p_k^0})
\end{aligned}$$

これらをまとめると, $P^\alpha(x^\beta) \in PDL$ に対して, $\alpha = p_1^l p_2^m \dots p_n^n$, $\beta = p_1^h p_2^i \dots p_n^j$ ならば

$$\begin{aligned}
\neg P^\alpha(x^\beta) &\vdash \neg P^{p_1^{\alpha_1} p_2^{\alpha_2} \dots p_n^{\alpha_n}}(x^{p_1^{\beta_1} p_2^{\beta_2} \dots p_n^{\beta_n}}) \\
&\quad (0 \leq \alpha_1 \leq l, 0 \leq \alpha_2 \leq m, \dots, 0 \leq \alpha_n \leq n) \\
&\quad (0 \leq \beta_1 \leq h, 0 \leq \beta_2 \leq i, \dots, 0 \leq \beta_n \leq j)
\end{aligned}$$

と定義する.

また, システムにおける否定に関するアルゴリズムを以下のように記述する.

Algorithm 5.2 (Transformation of Negation)

[$\neg P(\alpha)$ について考える.]

1.1 [ソート述語のカウンター初期化]

ソート述語に対して配列 A^1 を用意し, $A_i^1 \leftarrow 0$

1.2 [ソートのカウンタ初期化]

ソートに対して配列 A^2 を用意し, $A_j^2 \leftarrow 0$

2 [P_α の下位概念を示す]

前提にある P_α に対して, $A_R^1 \mid A_P^1, A_\gamma^2 \mid A_\alpha^2$ なる R と γ が存在する. このとき, $A_R^1 \leftarrow 1$, $A_\gamma^2 \leftarrow 1$.

3

$A_i^1 = 1, A_j^2 = 1$ なる記号に否定記号を加える.

4

終了

図 5.7: 単純なソート階層

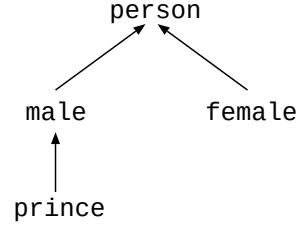


表 5.7: 与えられた *Planar Dual Lattice*

$\perp$	$prince^2$	$male^4$	$person^{36}$	$\top^{72}$
$\perp$	$princess^3$	$female^9$		
$swim^2$	$swim^2(prince^2)$	$swim^2(male^4)$	$swim^2(person^{36})$	$swim^2(\top^{72})$
$walk^3$	$swim^2(princess^3)$	$swim^2(female^9)$	$walk^3(person^{36})$	$walk^3(\top^{72})$
	$walk^3(prince^2)$	$walk^3(male^4)$		
	$walk^3(princess^3)$	$walk^3(female^9)$		
do^6	$do^6(prince^2)$	$do^6(male^4)$	$do^6(person^{36})$	$do^6(\top^{72})$
	$do^6(princess^3)$	$do^6(female^9)$		
$\top^{12}$	$\top^{12}(prince^2)$	$\top^{12}(male^4)$	$\top^{12}(person^{36})$	$\top^{12}(\top^{72})$
	$\top^{12}(princess^3)$	$\top^{12}(female^9)$		

また、否定に関して次のような推論規則が定義できる．いま，図 5.7に示すような単純なソート階層が与えられているとする．

このとき， $\neg male \sqcap \neg female$ という前提に対して，次のような推論が考えられる．

- (1) $\neg male \sqcap \neg female \vdash \neg person$

また， $person \sqcap \neg female$ という前提に対して，次のような推論が考えられる．

- (2) $person \sqcap \neg female \vdash male$

よって，表 5.7に示すようなセルが与えられたとき， $do(prince) \sqcap \neg swim(male) \vdash walk(person)$ を行くと，表 5.8に示すような出力が得られる．このように一見推論が困難と思われる場

表 5.8: $do(prince) \sqcap \neg swim(male) \vdash walk(person)$ の出力

$\perp$	$\neg prince^2$	$\neg male^4$		
$swim^2$	$\neg swim^2(prince^2)$	$\neg swim^2(male^4)$		
$walk^3$	$walk^3(prince^2)$	$walk^3(male^4)$	$walk^3(person^{36})$	$walk^3(\top^{72})$
do^6	$do^6(prince^2)$	$do^6(male^4)$	$do^6(person^{36})$	$do^6(\top^{72})$
$\top^{12}$	$\top^{12}(prince^2)$	$\top^{12}(male^4)$	$\top^{12}(person^{36})$	$\top^{12}(\top^{72})$

合においても，構造的な表現によって推論を行うことができる．

5.3 自然演繹法とセル表現の比較

作成したシステムについて，いくつかの比較を行った．清塚 [11] らの研究では，自然演繹法とベン図の比較を行っていた．この研究にならい，本稿でも同様に比較を行った．

いま，下記に示すような推論を行うときの step 数に注目する．ここで，前提は 4 つあり，求める結論は 1 つである．

$$\begin{array}{c}
 P(x) \rightarrow P(y) , P(y) \rightarrow P(z) \\
 P(z) \rightarrow Q(z) , Q(z) \rightarrow R(z) \\
 \hline
 P(x) \rightarrow R(z)
 \end{array}$$

このとき，自然演繹法では一般に

$$(n + m - 4) \times 7.5 + 1 \text{ [step] 必要であり}$$

上記の場合 16 [step] 必要となる．

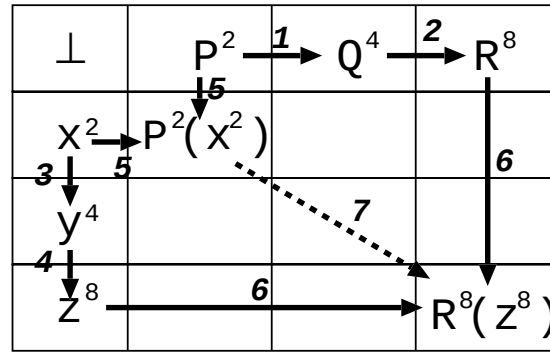
セルによる表現では一般に

$$(n + m + 1) \text{ [step] 必要であり}$$

上記の場合 7 [step] 必要となる．

ここで n , m とは，各ソート階層で演繹に要するソート記号の数である．自然演繹法では，大前提，小前提，結論の組み合わせは 256 種類あり，これを格式と呼ぶ．現在の格式と

図 5.8: 双ラティスによる描画



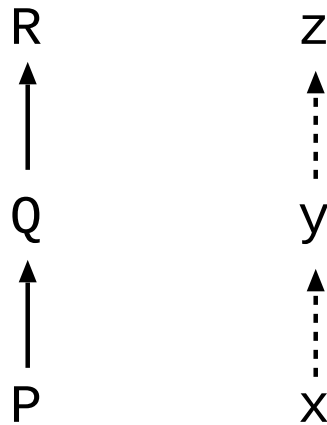
して妥当なものは15種類あり、三段論法を用いた際のそれらの Step 数の平均値は7.5となることより、上記の Step 数が算出される。Step 数は推論の規則を用いた回数について注目している。ただし項と述語の部分に対して、各々定言命題として三段論法を行ったものとする。セルによる表現では、与えられた前提からセル上に描画していくために、 $n+m$ ステップ必要となる。前提がソート階層として与えられ、結論として導出されるものの index 数値を比較すれば、それが妥当な結論として得られるか否かが判断できる。そのため、index 数値の比較として1 Step 必要となる。図 5.8にこの例における推論過程を示す。

これは明らかに、セルによる表現のほうが Step 数において有意である。また、セルによる表現は定まった結論を導くだけでなく、全体の構造に注目している。このことは、与えられた前提から通常気づかないような結論を導出することも示唆している。セルによる表現は Step 数の減少だけでなく、全体の構造を反映したものとなる。結果としてセルによる表現の有意性がみられた。また、図か持つ利点である Free Ride も見られる。このことは、推論を行う上で非常に重要なものである。自然演繹法を用いて推論を行うとき、新たにソート宣言がされる度に Step 数が定数倍で増加するのは、全体の構造が保存されないためである。セルによる表現では Free Ride が期待できるために、新たにソート宣言がなされても結論として得られるものの判断は1 Step で済んでいる。

5.4 双ラティスとセル表現の比較

次に、双ラティスを用いる場合とセルによる表現の違いに注目する。双ラティスを用いた表現では、先の例は図 5.9のように描画できる。図 5.9を用いて、例における結論を導

図 5.9: 双ラティスによる描画



出するためには述語部分 P, Q, R の関係を走査し、項部分 x, y, z の関係を走査することが必要となる。すなわち、与えられた階層に対して結論を導出するために双ラティスを別々に走査する必要がある。

しかし、本稿で提案するセル形式では別々に走査する必要は無く、結論として導出したものの index 数値の比較のみで判断できる。この場合でいえば、双ラティスを用いるとソート宣言以外に 4 Step 必要となり、セルによる表現では 1 Step で済む。このことから本稿で提案するセルによる表現は非常に有意であるといえる。また、左右方向が 1 つのソート階層の下位上位の関係、上下方向が別のソート階層の下位上位の関係という意味付けが行われている。これは単純に双ラティスをまとめたものではなく、このことが step 数減少にもつながる要因となっている。また、実際の問題をとく上で、双ラティスによる表現は各々のラティスにおける辺の交差が問題となる。あまりにも辺が交差すると、人間がその上位下位関係を理解するのに困難となる。しかし、セルによる表現では辺が交差するという問題が生じない結果となっている。

これらの比較から分かることは、自然演繹法よりも Free Ride の効果によって、大幅な演繹 Step 数の減少が見られた。図形による表現は、全体として構造化された表現となるため、個々のオブジェクトの相関関係をみるのに適している。また、従来どおりの双ラティスを用いるよりも、ステップ数減少の効果も得られた。これは、1 つのソート階層に対する関係を宣言すると、同時に他のソート階層も考慮することになるためである。すなわち、本稿で提案したセル表現は、双ラティスを反映している簡潔な表現といえる。また、本システムでは推論部分に関してのみ、オイラー円を用いた表現も考慮した。なぜな

らば、オイラー円は包含関係を明記するのに優れているためである。与えられた各ソート記号に関して、その概念を持つオブジェクトを抽象的な円で描くことは図形を考察する上での理想に関しても有意義なものとなるからである。ただし、今回のオイラー円による表示は付加的なものである。なぜならば、ベン図やオイラー円といった表現は4つ以上のラベルに関しては表現が困難となるためである。この点に関する構造的な解釈は今後に委ねるものとした。

5.5 新しい宣言に対する負荷

今までの比較は推論における手間を示してきたが、ここでは表現の違いに対するソート宣言の手間を示す。新たにソート宣言が加わる時、今までの表現を再構築するにはどれほどの労力が必要となるかをみる。特にここでは、言語による表記と、単純なテーブルを用いた場合と、本稿で提案するセルによる表現について比較する。いま、以下に示すようなソート宣言が行われたとする。

述語に関する宣言

- (1) $P_1 \sqsubseteq P_2$
- (2) $P_3 \sqsubseteq P_4$
- (3) $P_1 \sqsubseteq P_3$

ソート宣言

- (1) $s_2 \sqsubseteq s_1$
- (2) $s_1 \sqsubseteq s_5$
- (3) $s_3 \sqsubseteq s_4$

これらは、言語による表記である。このような言語表現に対して、推移的關係を得ることができる。すなわち述語に関しては、(2)と(3)より新たな $P_1 \sqsubseteq P_4$ が推移的關係として得られる。ソートに関しては、(1)と(2)より新たな $s_2 \sqsubseteq s_5$ が推移的關係として得られる。

また、関係あるソート記号に印をつけることによって、テーブル表現が可能となる。図 5.10 に述語、図 5.11 にソートのテーブル表現を示した。ここで、○の印がついているものは、**直接**の上位下位の関係があることを示している。よって○に対して、テーブルの

図 5.10: 述語に関するテーブル表現

sort sub sort	P_1	P_2	P_3	P_4
P_1	○	○	○	
P_2		○		
P_3			○	○
P_4				○

表 5.9: セルによる表現

$\perp \perp$	P_1^2	P_2^4 P_3^6	P_4^{18}	$\top$
s_2^2	$P_1^2(s_2^2)$	$P_2^4(s_2^2)$ $P_3^6(s_2^2)$	$P_4^{18}(s_2^2)$	$\top(s_2^2)$
s_3^3	$P_1^2(s_3^3)$	$P_2^4(s_3^3)$ $P_3^6(s_3^3)$	$P_4^{18}(s_3^3)$	$\top(s_3^3)$
s_1^4	$P_1^2(s_1^4)$	$P_2^4(s_1^4)$ $P_3^6(s_1^4)$	$P_4^{18}(s_1^4)$	$\top(s_1^4)$
s_4^9	$P_1^2(s_4^9)$	$P_2^4(s_4^9)$ $P_3^6(s_4^9)$	$P_4^{18}(s_4^9)$	$\top(s_4^9)$
s_5^8	$P_1^2(s_5^8)$	$P_2^4(s_5^8)$ $P_3^6(s_5^8)$	$P_4^{18}(s_5^8)$	$\top(s_5^8)$
$\top$	$P_1^2(\top)$	$P_2^4(\top)$ $P_3^6(\top)$	$P_4^{18}(\top)$	$\top(\top)$

各ます目に従うことにより推移的關係を得る．

また，本稿で提案するセルによる表現は表 5.9に示した．セルにおいては素数の剰余によって推移的關係が判別可能となっている．

ここで，新たにソート宣言が行われるとする．例えば，次のようなソート宣言を考える．

述語に関する宣言

(4) $P_2 \sqsubseteq P_3$

図 5.11: ソートに関するテーブル表現

sort sub sort	S ₁	S ₂	S ₃	S ₄	S ₅
S ₁	○				○
S ₂	○	○			
S ₃			○	○	
S ₄				○	
S ₅					○

ソート宣言

$$(4) \quad s_6 \sqsubseteq s_2$$

このとき、言語による表現では、述語に関して (1), (2), (3) を参照することが必要となる。また、ソートに関しても同様に (1), (2), (3) を参照する必要がある。すなわち述語に関しては、

$$(1) \text{ と } (4) \text{ より, } P_1 \sqsubseteq P_3$$

$$(2) \text{ と } (3) \text{ より, } P_1 \sqsubseteq P_4$$

$$(2) \text{ と } (4) \text{ より, } P_2 \sqsubseteq P_4$$

を得る。

またソートに関しては、

$$(1) \text{ と } (2) \text{ より, } s_2 \sqsubseteq s_5$$

$$(4) \text{ と } (1) \text{ より, } s_6 \sqsubseteq s_1$$

$$(4) \text{ と } (1) \text{ と } (2) \text{ より, } s_6 \sqsubseteq s_5$$

を得る。

これらの判別には非常に多くの労力がかかる。単純な計算をするならば、あるソート宣

図 5.12: 述語に関するテーブル表現 2

sort sub sort	P_1	P_2	P_3	P_4
P_1	○	○	○	
P_2		○	○	
P_3			○	○
P_4				○

言の右辺に現れているものと、他のソート宣言の左辺に現れているソート記号を判別するのに $n(n-1)$ 回の判別がかかる．よって述語とソートに対する言語による判別は、新たなソート記号が 1 つ加わる度に $n(n-1)$ 回必要となる．

また、テーブルによる表現では、新たなソート記号を反映させるのは容易である．述語、ソートに関して図 5.12, 図 5.13 に示す．しかし、全体的な構造として推移的關係に注目すると、労力のかかる事がわかる．ある ○ に対して、その列の記号に対応する行の記号を走査することを繰り返すことが必要となる．例えば、図 5.12 において、 P_2 行、 P_3 列の ○ に注目する．このとき、○ の列は P_3 なので、 P_3 の行に注目する．ここで、 P_3 行、 P_4 列に ○ があることから $P_2 \subseteq P_4$ が得られる．このような走査を繰り返すためには、1 個の ○ に対して、 n 回の手間が必要となる．ここで n とはソート記号の数である．推移的關係があるほど、この走査は**指数関数的に増加**していく．

最後にセルによる表現に注目する．セルによる表現では、新たに表 5.10 に示すような関係が得られる．ここで変化したのは、記号の**部分集合**である P_3, P_4 と s_2, s_1, s_5 の index 数値である．これは逆にいえば、パスに応じた素数の割り振りにより関係ある記号のみが影響をうけた事を示している．また、推移的關係を考察するのも非常に高速に行える．これは、前に見た言語的表現やテーブルによる表現と違って、全体を再構築すること無く書き換えられることを示している．これらの比較により、次の結論が得られる．

図 5.13: ソートに関するテーブル表現 2

sort sub sort	S_1	S_2	S_3	S_4	S_5	S_6
S_1	○				○	
S_2	○	○				
S_3			○	○		
S_4				○		
S_5					○	
S_6		○				○

表 5.10: セルによる表現 2

$\perp \perp$	P_1^2	P_2^4	P_3^8	P_4^{12}	$\top$
s_6^2	$P_1^2(s_6^2)$	$P_2^4(s_6^2)$	$P_3^8(s_6^2)$	$P_4^{12}(s_6^2)$	$\top(s_6^2)$
s_2^4	$P_1^2(s_2^4)$	$P_2^4(s_2^4)$	$P_3^8(s_2^4)$	$P_4^{12}(s_2^4)$	$\top(s_2^4)$
s_3^3	$P_1^2(s_3^3)$	$P_2^4(s_3^3)$	$P_3^8(s_3^3)$	$P_4^{12}(s_3^3)$	$\top(s_3^3)$
s_1^8	$P_1^2(s_1^8)$	$P_2^4(s_1^8)$	$P_3^8(s_1^8)$	$P_4^{12}(s_1^8)$	$\top(s_1^8)$
s_4^9	$P_1^2(s_4^9)$	$P_2^4(s_4^9)$	$P_3^8(s_4^9)$	$P_4^{12}(s_4^9)$	$\top(s_4^9)$
s_5^{16}	$P_1^2(s_5^{16})$	$P_2^4(s_5^{16})$	$P_3^8(s_5^{16})$	$P_4^{12}(s_5^{16})$	$\top(s_5^{16})$
$\top$	$P_1^2(\top)$	$P_2^4(\top)$	$P_3^8(\top)$	$P_4^{12}(\top)$	$\top(\top)$

言語やテーブルを用いた表現では、新たな宣言がなされる度に、全体に注目しなければならない。一方セルによる表現では、新たな宣言に関係ある部分集合のみを注目することができる。これは処理の高速化につながる。また、言語やテーブルによる表現では、新たな宣言を加えたあとの演繹処理も多大な労力がかかる。しかし、セルでは容易に行うことができる。

第 6 章

考察

人間が問題解決をするとき、言語的表現で行われるだけでなく、図や数値データなど、さまざまな表現形態の情報のなかから適切な表現を選択し、変換しながら行っていると考えられる。推論を行う上でも同様に、問題の前提となる構造を理解するとき、様々な表現形態を扱っていると考えられる。本研究は、そのような知的活動の解明に対して重要なものであると考えられる。

今回本研究において、従来の型階層論理に対する双ラティス構造を反映したダイアグラムを提案した。本稿で提案したセルによる表現は well-formed chart に準じた表現となっている。このようなダイアグラムによる表現を用いることにより、ソート宣言を相対的な部分的構造としてではなく、全体的な構造を保持した宣言へと変換している。これは図が持つ位置情報の効果であり、ソート宣言がより有意なものとして扱われている。また論理演算に対して、セルを単なる個別のものとして扱うことなく、領域として考察できることがベン図やオイラー円に劣らない表現であることを示している。このセルによる表現で、自然演繹法や従来の異なるラティスを各々用いることよりも有意であることを示した。また、言語による表現やテーブルを用いた表現と異なり、新たなソート宣言に対して部分集合のみに注目することも利点としてあげられる。これは処理の高速化という面で、非常に有効なものとなる。

ただし、線分や円といった連続量を表す表現に関して、考察の余地が多く残されている。セルにおける推論規則を領域における包含関係としてとらえたが、より抽象的な表現として円を用いると、ソートの意味がさらに付加されると考えられる。このことは限量子の扱いにおいても今後の課題として重要なものであると考えられる。また、実際に素数を計算すると非常に大きな数となり、計算機内部の処理が困難となる。そこで、素数の代わりにシンボリックに扱える群を用いることも今後の目的とする。

将来的には、図における論理体系の構築を考えている。文字のみを用いた論理に注目するのではなく、図における言語記号を用いた論理を対象にしたいと考える。より抽象的なシンボルを用いた論理体系は興味深いだけでなく、人間の思考により近いものと考えられる。なぜならば、人間の脳という有限の領域において、抽象的なシンボルを用いることの意義は大きいと考える。今回本稿で述べたような処理の高速化を考えても、図を用いることの有意性は大きい。今後はこのような点に注目して、図における論理体系の研究を行う。

参考文献

- [1] Atsushi Shimojima, Operational Constraints in Diagrammatic Reasoning, Logical Reasoning with Diagrams, Oxford University Press, pp27-48, 1996.
- [2] Eric Hammer, Perceant Graphs for Propositional Logic, LOGICAL REASONING WITH DIAGRAMS, Oxford University Press, pp129-147, 1996.
- [3] Gary Chartrand Ortrud R.Oellermann, APPLIED AND ALGORITHMIC GRAPH THEORY, McGraw-Hill, pp247-281, 1993.
- [4] J.I.Glasgow D.Papadias, Computational Imagery, Diagrammatic reasoning:cognitive and computational perspective; ed.by Janice Glasgow. N.Nari Narayanan, B. Chandrasekaran, MIT Press, pp435-480,1995.
- [5] Jerry Seligman, Perspective in Situation Theory, Situaion Theory and its Applications Volume 1, pp146-191, 1990.
- [6] Jon Barwise and Eric Hammer, Diagrams and the Concept of Logical System, Logical Reasoning with Diagrams, Oxford University Press, pp49-78, 1996.
- [7] Nikolaus G. Swoboda, Implementing Euler/Venn Reasoning System, Reasoning with Diagrammatic Representation, Papers from the 1997 AAI Fall Symposium Technical Report FS-97-03.
- [8] Sun-Joo Shin, Situation-Theoretic Account of Valid Reasoning with Venn Diagrams, LOGICAL REASONING WITH DIAGRAMS, Oxford University Press, pp81-108, 1996.
- [9] 出原 栄一 吉田 武生 渥美 浩章, 図の体系 -図的思考とその表現-, 日科技連出版社, 1986.

- [10] 兼岩 憲 東条 敏, 法律知識の事象的/属性的読みを区別した推論システム, 情報処理学会, 40, 7, 1999.
- [11] 清塚 謙助 沢村 一, 図を用いた推論の複雑性に関する考察, 人工知能学会, Vol.14, No.4, 1999.
- [12] 高木 貞治, 初等整数論講義 第2版, 共立出版株式会社, 1999.
- [13] 村田 剛志 加藤 裕樹 志村 正道 沼尾 正行, 図を用いた算術問題解決システム DIPS, 電子情報通信学会 D-II, Vol.J82-D-II, No.1, pp.75-82, 1999.