データベースシステム 102－6
（1995．3．14）

IS-A関係と正規表現を用いたDOT知識ベースシステムの設計および実装

清 一隆† 塚本 昌彦‡ 西尾 章治郎†

† 大阪大学 工学部 情報システム工学科
‡ シャープ株式会社 技術本部 情報技術研究所

概 要

ネットワークやグループウェアなどのマルチユーザのアプリケーションシステムに知識処理機能を付加して，柔軟な運用，管理を行なう研究が行なわれている．このようなことを実現する上で，対象に関するタクソノミの柔軟な表現と，集合を対象とした推論の能力，知識の曖昧性の取り扱いが重要となる．本稿では，これらのアプリケーションにおける推論エンジンとして開発したシステムDOT (Deductive and Object-oriented Term representation) について述べる．DOTでは，ドット記法とIS-A関係を用いてタクソノミを表現し，オートマトンを用いて条件を満たす全解集合を求める．結果は正規表現で表され，その正規表現は再び新たな知識の中で利用できる．知識の曖昧性は多重世界として表現される．その結果，DOTを用いて，前述のような知的アプリケーションの実現が可能になるものと考えられる．

Design and Implementation of DOT Knowledge-base System using IS-A Relation and Regular Expressions

Kazutaka SEI† Masahiko TSUKAMOTO‡ Shojiro NISHIO†

† Department of Information Systems Engineering, Faculty of Engineering, Osaka University
‡ Information Technology Research Laboratories, SHARP Corporation

Abstract

Recently, the status update of multi-user applications such as network applications and groupware systems has been remarkably proceeding, which is due to the enhancement of their functions through knowledge representation and reasoning capability. This enhancement is mainly aiming at achieving more flexible and intelligent management of their operations. For realizing such enhancement, the knowledge representation and reasoning systems employed in those applications should provide rich expressive power on taxonomy, highly reasoning capability on the taxonomy, and ambiguity handling capability of knowledge. However, few systems can provide all of these required capabilities. We have been proposing DOT (Deductive and Object-oriented Term representation) system which represents knowledge based on dot notations and IS-A relations. In this paper, we demonstrate that this DOT system effectively works for providing the above capabilities necessary to advanced application fields, where taxonomy is expressed by IS-A relations between regular expressions of dot notations, the answer to a query can also be expressed by regular expressions of dot notations, and ambiguity of knowledge can be expressed as multiple worlds. As a result, it is possible that the above-mentioned intelligent applications are realized using our DOT system.

1 はじめに

近年，ネットワークアプリケーションやグループウェアなどの主にマルチユーザを対象とする分散アプリケーションにおいて，管理対象やグループの構成メンバなどの対象集合に対して知識表現と推論の枠組を適用することによって，柔軟な運用や知的な管理を行なう研究が行なわれている．例えば，筆者らのグループでは，推論機構を用いたグループ通信を実現するためのメッセージ配送システムMILD[岩室94, 岩室95]を開発している．MILDでは，ネットワークを介して交換する電子メールの配送において,「こんな人にメールを送信したい」,「こんな話題に関するメールを受信したい」といったメールの送信者と受信者双方の意図を反映することを目的として，ユーザ情報とタクソノミを知識ベースとして保持し，送信者は知識ベースに対する問合せを宛先として送信する．問合せとして，和，積，差などの集合演算の記述能力も提供されている．送信されたメールをMILDが受信すると，知識ベースの推論機構を起動して，問合せで表される宛先を具体的なユーザアドレスに変換する．これによって，送信側の宛先のきめ細かで柔軟な指定と受信側の意図の反映を実現している．さらにMILDは，ユーザによるタクソノミや知識の違いを吸収するために，タクソノミや表現形式の変換機構を備えている．

このように，一般に，分散アプリケーションの柔軟な運用や知的な管理を行なう上で，

1. 対象に関するタクソノミの柔軟な表現
2. 集合を対象とした推論の能力
3. 知識の曖昧性の取り扱い

が重要であると考えられる．しかし，これまでに開発されてきた知識表現と推論システムでは，これらのすべてを提供するものはほとんど見当たらない．

これに対し筆者らは，オブジェクト指向データベースにおける継承の枠組に演繹を導入するために，DOT(Deductive and Object-oriented Term representation)を提案し[塚本92a]，さらにその枠組を，タクソノミに関する知識表現と推論の一般的枠組として拡張してきた[塚本95a, 塚本95b]．DOTは次のような点で特徴付けられる．

1. ドット記法間のIS-A関係，IS-NOT-A関係として知識を表現する．IS-A関係のドメインがドット記法によって拡張されているため，従来のIS-A関係だけでなく，属性値関係，属性値制約，属性値に関する不完全情報などが表現できる[塚本94a].
2. 問合せの結果は正規表現で表され[Tsukamoto91a]，正規表現を用いて再び新たに知識を表現できる[塚本94a, 塚本94b].
3. IS-NOT-A関係を用いて知識の矛盾を表現することができ，効率良い矛盾の検出と解消アルゴリズムが示されている[塚本93]．この手法を用いて，知識の曖昧性を多重世界として表現することができる[劉93b].

これらの特徴は，前述の各々の用件を実現するものと考えられ，分散アプリケーションの柔軟な運用や知的な管理を行なうための推論エンジンとして，DOTシステムの実現が有効であるものと考えられる．

本稿では，DOTをシステムとして実現する上で前述の特徴を活かすことに重点をおき，開発したシステムの設計および実装について論ずる．まず，システムが扱うデータに対して型の概念を導入した．次に，システムの扱うデータは型および識別子を持つものとした．そして，それらのデータを扱う豊富なコマンド群を定義し，各コマンドが各々返り値を持つものとすることによって，コマンドを再帰的に別のコマンド内で使用できるようにした．さらに，システム内で一時的にデータを格納するための変数を用意した．

以下，2章ではDOTの基本的な概念を例を用いて簡単に説明し，3章でシステムの概要を示し，4章で実装について述べる．5章で本稿のまとめと今後の課題について述べる．

2 DOTの概要

本章ではDOTの概要を示す．DOTにおいて，例えば，「'人'の'親'は'人'である．'きんた'は'人'である．'せいじ'は'きんた'の'親'である.」という知識は，

$$\text{人.親} < \text{人} \quad (1)$$
$$\text{きんた} < \text{人} \quad (2)$$
$$\text{せいじ} < \text{きんた.親} \quad (3)$$

と表される．ここで，'$<$' はIS-A 関係を表す．そして，IS-A 関係を，

- $X < X$.
- $X < Y, Y < Z$ならば $X < Z$.
- $X < Y$ならば $X.P < X.P$.

が成り立つ関係としてとらえた．ここで，X, Y, Z は**DOT 式**とよばれるドット記法を用いて表される式，Pはラベルとよばれる属性を表す式である．このIS-A 関係の定義の妥当性に関しては，[塚本 94a] で詳しく論じられている．先ほどの例では，

$$\text{せいじ} < \text{人} \qquad (4)$$

などが成立することになる．

DOT においては，問合せに対する答が正規表現で表されることが示されている [Tsukamoto91b]．「'せいじ' とは何ですか？」，「何が '人' ですか？」という問合せは，各々，

$$? \ \text{せいじ} < X,$$
$$? \ X < \text{人}$$

と表現され，これらに対する答は

$$X : \text{せいじ} + \text{きんた.親} + \text{人.親} + \text{人}$$
$$X : (\text{人} + \text{きんた} + \text{せいじ}).\text{親}^*$$

となる．

また，正規表現上の演算を用いて高度な推論が可能である [Tsukamoto91a]．例えば，

$$0 < even \qquad (5)$$
$$even.s < odd \qquad (6)$$
$$odd.s < even \qquad (7)$$
$$0 < m3 \qquad (8)$$
$$m3.s.s.s < m3 \qquad (9)$$

という知識を考える．'s' は「次の数」を表す．'$m3$' は「3 の倍数」を表す．このとき，

$$? \ X < odd,$$
$$? \ X.s < m3$$

の答えに含まれる正規表現

$$(0.s + even.s + odd).(s.s)^*,$$
$$(0 + m3).s.s.(s.s.s)^*$$

の積集合を計算することにより，両方の問合せを同時に満足する答え

$$0.s.s.s.s.s.(s.s.s.s.s.s)^*$$

が得られる．

さらに，IS-NOT-A 関係を表現に組み入れて拡張したシステムにおいて，矛盾の検出と解消のアルゴリズムが示されている [塚本 93]．そのために，

- $X < Y, X \not< Y$ならば矛盾

という規則を導入している．ここで，'$\not<$' はIS-NOT-A 関係を表す．例えば，先ほどの (1), (2), (3) の知識に

$$\text{きんた.親} \not< \text{人} \qquad (10)$$
$$\text{せいじ} \not< \text{人.親} \qquad (11)$$

を追加すると直観的に矛盾が生ずる．このとき，すべての無矛盾な極大部分集合

$$\{\{(1),(2),(3)\}, \{(1),(2),(11)\}, \{(2),(3),(10)\},$$
$$\{(2),(10),(11)\}, \{(1),(3),(10),(11)\} \ \}$$

を効率よく求める方法が示されている．

また，DOT において，知識表現の中で正規表現を使うことも可能であることが示されている．正規表現を用いて表された知識をそれと等価な正規表現を用いない知識に変換する**A 変換**とよばれる知識の変換を利用して，問合せ処理が可能となる．例えば，

$$0.(s.s)^* < even \qquad (12)$$
$$0.s.(s.s)^* < odd \qquad (13)$$

は，A 変換により

$$0 < s_1 \qquad (12a)$$
$$s_1.s.s < s_1 \qquad (12b)$$
$$s_1 < even \qquad (12c)$$
$$0.s < s_2 \qquad (13a)$$
$$s_2.s.s < s_2 \qquad (13b)$$
$$s_2 < odd \qquad (13c)$$

に変換される．これによって前述の問合せ処理方式により，例えば，

$$? \ X.s < odd.s.s$$

という問合せに対して

$$X : 0.s.s.(s.s)^* + odd.s$$

と答えることが可能である．答えの中で，s_1, s_2の関わる部分は簡単な操作によって削除される．これによって，正規表現というクラスで閉じた推論システムが構築できた．つまり，問合せに対する答を用いてインクリメンタルに知識を追加することが可能である．

IS-NOT-A 関係の表現の中での正規表現の使用も，N 変換と呼ばれる変換を用いることによって可能となる．例えば，

$$0.(s.s)^* \not< odd \qquad (14)$$

という記述を許し，これによって，

$$0.s.s.s.s < odd \qquad (15)$$

などの知識を追加すると矛盾が生じることになり，整合性のある知識管理が可能となる．

3 DOT システム

本章では，2章で述べた DOT の諸相をシステムとして実現する一つの方法を示す．

3.1 データと型

システムにおいて取り扱うデータには次のものがある．

- オブジェクト名 (ON)．知識表現の対称 (オブジェクト) に付与された名前．英大文字で始まらず，'&', '—', '(', ')', ',', '.', '+', '*', '_' を含まない文字列で表される．'きんた', '人', '数', '0' など．

- ラベル (L)．オブジェクトに付随する属性を表す名前．英大文字で始まらず，'&', '—', '(', ')', ',', '.', '+', '*', '_' を含まない文字列で表される．オブジェクト名と同一の名前をラベルとして使用できるが，オブジェクト名 a とラベル a の間にはシステム上何ら特別な関係は存在しないものとする．'親', '年齢', '次の数' など．

- ラベル式 (LE)．ラベルの 0 個以上の有限列．ラベル名の重複も可能．'ラベル.…. ラベル' の形式で表される．特に 'empty' は空ラベル式を表す．'親. 親. 名前', '次の数. 次の数. 次の数' など．

- DOT 式 (DE)．オブジェクト名とラベル式の組．'オブジェクト名. ラベル式' の形式で表される．'きんた. 親. 親. 名前', '0. 次の数. 次の数. 次の数' など．

- 正規表現 (RE)．DOT 式の正規表現．'+', '*' および括弧 '(', ')' を用いて通常の方法 [塚本 94a] で表す．'(きんた+へいた). 親*. 名前', '0.((次の数. 次の数. 次の数)* + (次の数. 次の数)*). 次の数' など．

- 知識要素 (KE)．'正規表現 is-a 正規表現', '正規表現 is-an 正規表現', '正規表現 is-not-a 正規表現', '正規表現 is-not-an 正規表現' のいずれかの形式で表される．最初の二つは IS-A 関係を表す．後の二つは IS-NOT-A 関係を表す．ともに両者の間には管理上の違いはない．

- 世界 (W)．知識要素の有限集合．'{ 知識要素, …, 知識要素 }' の形式で表される．空集合は 'empty' で表す．

- 多重世界 (MW)．世界の有限集合．'{ 世界, …, 世界 }' の形式で表される．空集合は 'empty' で表す．

各データは，生成時にデータ ID と呼ばれる識別子が割り振られる．データ ID はシステム内でそのデータを一意に識別する非負整数値であり，'_識別子' の形式でシステム内でデータ値を参照できる．また，各データには，オブジェクト名型，ラベル型，DOT 式型，正規表現型，ラベル式型，知識要素型，世界型，多重世界型のいずれかの型が静的に割り振られる．ただし，文脈によってオブジェクト名を DOT 式として参照したり，DOT 式型を正規表現型として参照したりできる．

データ ID とデータの関係は明示的な消去や更新を行わない限り，システム終了後も存続する．

変数 データを格納するための変数を定義することができる．変数の型はデータを代入した時点で決まる．英大文字で始まる '&', '—', '(', ')', ',', '.', '+', '*', '_' を含まない文字列で表される．例えば，'Number', 'World1' など．

変数とデータの束縛は一時的なものであり，システム終了時は消去される．ただし，ファイルへの明

示的な保存，ファイルからの読み出しを行うこともできる．

3.2 コマンド

システムはインタープリタとして動作する．システムを起動するとプロンプトが表示され，ユーザからのコマンドを受け付ける．コマンドは評価され，何らかのシステム動作を実行し，値を返す．コマンドを別のコマンドの中で使用することもできる．その場合，内側のコマンドが先に評価され，その返り値を外側のコマンドの引き数として使用される．この場合，内側のコマンドの返り値と外側のコマンドの引き数の型が一致しなければならない．

コマンドには，生成コマンド，代入コマンド，知識操作コマンド，問合せコマンド，世界操作コマンド，正規表現操作コマンド，システムコマンドがある．以下では，各コマンドの表現形式と動作内容を示す．表現形式の中で，型 X の式 a あるいはコマンド a は $[a{:}X]$ と表すものとする．例えば，$[a{:}\mathrm{RE}]$ は正規表現を表す式 a あるいは正規表現を返り値とするコマンド a を表す．返り値の型 X を明示するために，コマンドの表現形式の先頭に '(X)' と記す．

生成コマンド． 生成コマンドは次の形式で表される．

(RE)	DOT 式の正規表現
(KE)	[r_1: RE] is-a [r_2: RE]
(KE)	[r_1: RE] is-an [r_2: RE]
(KE)	[r_1: RE] is-not-a [r_2: RE]
(KE)	[r_1: RE] is-not-an [r_2: RE]
(W)	empty
(W)	{ [k_1: KE], $\cdots$, [k_n: KE] }
(MW)	empty
(MW)	{ [w_1: W], $\cdots$, [w_n: W] }

詳細は 3.1 節の型の説明の中で述べた通りである．

代入コマンド． 代入コマンドは次の形式で表される．

(X)	変数 = データ ID
(X)	変数 = コマンド

指定されたデータおよびコマンドを評価した値が変数に代入される．変数および返り値の型 X は，データおよびコマンドに従って決まる．MW 型のデータを代入する際に，矛盾を含む世界が含まれる場合，矛盾解消が自動的に行なわれる．

例えば，

X = *num.s* is-a *num*
Y = 0 is-a *num*
W = {X, Y}

は，二つの知識要素 X, Y を定義し，X, Y からなる世界 W を定義している．

知識入力コマンド． 知識入力コマンドは次の形式で表される．

(MW)	add [k:KE] to [w:W]
(MW)	add [k:KE] to [m:MW]

最初のコマンドは，知識要素 k を世界 w に追加し，その結果を多重世界として返すものである．二番目のコマンドは，知識要素 k を多重世界 m の要素である各々の世界に追加し，その和を多重世界として返すものである．両者において，世界に矛盾が発生した場合は，自動的に更新が行われる．w および m が変数の場合，変数の束縛は自動的に新たに生成された多重世界に変更される．

例えば，0123 を世界型のデータ ID とするとき，

add せいじ is-a きんた. 親 to _0123

は 'せいじ is-a きんた. 親' を世界_0123 に追加するコマンドである．システムは新たな多重世界を生成し，多重世界のデータ ID を返す．ここで，変数 X が_0123 に束縛されているとき，

add せいじ is-a きんた. 親 to X

とすると，前述と同様の多重世界が生成される．そのデータ ID を_0124 とすると，X の束縛は_0124 に自動的に変更される．

問合せコマンド 問合せコマンドは次の形式で表される．

(RE) ?X: X is-a [d:DE] in [w:W]
(RE) ?X: X.[l:LE] is-a [d:DE] in [w:W]
(RE) ?X: [d:DE] is-a X in [w:W]
(RE) ?X: [d:DE] is-a X.[l:LE] in [w:W]
(MW) ? [d:DE] is-a [d':DE] in [w:W]

最初の四つのコマンドは各々，世界wにおいて，X is-a d，$X.l$ is-a d，d is-a X，d is-a $X.l$を満たすXの全体を表す正規表現を返す．最後のコマンドは，世界wにおいて，d is-a d'が成立するかどうかを調べ，その根拠を多重世界として返す．最初の四つのコマンドにおいて'?'の後ろのXはこのコマンドの内部でのみ使用されるローカル変数であり，変数と同様の構文で表される任意の文字列が使用可能である．変数名と一致してもかまわないが，その場合は式の内部でその変数が参照できなくなる．これらのコマンドにおいて，'is-a'は'is-an'としてもよい．

例えば，

?X: X is-a 親,
?X: きんた. 親 is-a X,
?X: X.s is-a *even*,
? せいじ is-a きんた. 親

は，各々，何が人ですか?，きんたの親は何ですか?，次の数が偶数となるのは何ですか?，せいじはきんたの親ですか?という問合せを表す．これらに対する答えは，各々，

(きんた+せいじ+人). 親*,
きんた. 親+人. 親+人,
$(odd + even.s + 0.s).(s.s)^*$,
_0254 : MW {{_0122 : KE, _0123 : KE},
{_0028 : KE}}

となる．最後の答えは，'せいじ is-a きんた. 親'が，知識要素_0122と知識要素_0123から，あるいは知識要素_0028から導出されることを示している．

世界および多重世界の操作コマンド　世界および多重世界の操作コマンドは次の形式で表される．

(MW) delete [w:W] from [m:MW]
(MW) delete [m:MW] from [m':MW]
(MW) marge [m_1:MW], ⋯, [m_n:MW]

最初のコマンドは多重世界mから世界wを削除する．二番目のコマンドは多重世界m'から多重世界mの各要素を削除する．最後のコマンドは多重世界m_1,⋯,m_nをマージする．

正規表現操作コマンド　正規表現操作コマンドは次のように表される．

(RE) [r:RE] | [r':RE]
(RE) [r:RE] & [r':RE]
(RE) [r:RE] / [r':RE]

各々，正規集合の和，積，差を表す．

システムコマンド　表示コマンドは次のように表される．

(X) show データ ID
(X) show コマンド

最初のコマンドは，データIDで表されるデータの詳細を表示する．二番目のコマンドは，引数のコマンドを評価したときの返り値の詳細を表示する．

システムの終了コマンドは次のように表される．

(X) quit

システムを終了する．返り値は意味を持たない．

変数束縛の消去コマンドは次のように表される．

(X) clear 変数
(X) clear

最初のコマンドは，指定された変数の束縛を消去する．返り値は指定された変数に束縛されていた値である．二番目のコマンドは，システム中のすべての変数の束縛を消去する．返り値は意味を持たない．

データの消去コマンドは次のように表される．

(X) delete データ ID
(X) delete コマンド

指定されたデータを消去する．返り値は意味を持たない．

変数束縛のファイルへの保存/読み込みコマンド

は次のように表される.

(X) save 変数 as "filename"
(X) save as "filename"
(X) load "filename"

最初のコマンドは指定された変数をファイルに保存する. 返り値はその変数の評価値である. 二番目のコマンドはすべての変数束縛をファイルに保存する. 返り値は意味を持たない. 最後のコマンドはファイルから変数束縛を読み込む. 返り値は変数の評価値である.

4 システムの実装

UNIX ワークステーションおよびDOS/V パーソナルコンピュータにおいてオンメモリのプロトタイプシステムを実装した. システムはC およびC++言語を用いて記述し, GNU プロジェクトのG++およびBorland C++でコンパイルした. ソースコードは約10,000行, 作成したクラスは約30個である.

実装にあたっては, オブジェクト関連グラフ[清 94]に基づく問合せ処理アルゴリズム, および多重世界を世界論理式の最簡形式により表現する手法[劉 93b]に基づく世界計算アルゴリズムを用いることでシステムの高速化を行なっている. また, 正規表現を含む知識要素は入力時にA 変換, N 変換を行い, 変換前の知識要素と変換後の知識要素の集合の両者を保持し, これらの全体に対して一つのデータ ID を割り振ることでDOT の知識に対応している.

図1に実行例を示す. '>' はシステムのプロンプトであり, それに続く部分がユーザの入力である. ユーザ入力に対して, システムは入力コマンドを評価し, 返り値のデータ ID と型を表示している.

5 おわりに

本稿ではドット記法とIS-A 関係を用いてタクソノミを表現し, オートマトンを用いて推論を行なう知識ベースシステムDOT の設計および実装を行なった.

今後の課題として, 条件分岐, 繰り返しなどの制御コマンドの追加によるプログラミングへの応用, システムの完全化の実現, データ管理のデータベース化などのシステムの改良があげられる.

```
DOT Knowledge-base System Interpreter
Version 1.00
> X = {0 is-an even, even.s.s is-an even}
_00200 : W
> Y = {0 is-an m3, m3.s.s.s is-an m3}
_00201 : W
> World = marge X, Y
_00202 : W
> add (?X:X is-an even & ?X:X is-an m3) is-an m6
to World
_00204 : MW
> show World
World =
_00204 : MW {
_00203 : W {
_00100 : KE 0 is-an even,
_00101 : KE even.s.s is-an even,
_00102 : KE 0 is-an m3,
_00103 : KE m3.s.s.s is-an m3,
_00104 : KE 0.(s.s.s.s.s.s)* is-an m6
}}
> ? 0.s.s.s.s.s.s.s.s.s.s.s is-an m6 in _00203
_00205 : MW
> show _00205
_00205 : MW {
_00206 : W {
_00104 : KE 0.(s.s.s.s.s.s)* is-an m6
}}
> _
```

図 1: 実行例

現在, 推論能力を向上させるためにIS-NOT-A 関係の問合せ, 非単調推論, 新しい推論規則などの理論的課題についても取り組んでいる. また, オブジェクト指向プログラミングにおいてDOT を利用することについても検討している[liu92, 劉 93a, 柳沢 95a, 柳沢 95b].

参考文献

[岩室 94] 岩室元典, 田中理恵子, 塚本昌彦, 西尾章治郎: 推論機構を用いたメール分配システム, 情報処理学会研究会報告(マルチメディア通信と分散処理研究会 DPS 66-16), Vol.94, No.56, pp.91-96 (1994).

[岩室 95] 岩室元典, 田中理恵子, 塚本昌彦, 西尾章治郎: 配送先に関する知識ベースを用いたメッセージ配送システム, 第 6 回データ工学ワークショップ (DEWS'95) (1995, 発表予定)

[liu92] Liu, B., Nishio, S., Tsukamoto, M., and Miyahara, H.: Design and Implementation of an Object-Base System Based on DOT Expression, Ohsuga, S., Kangassalo, H., Jaakkola, H., Hori, K., and Yonezaki, N. (Eds.): Information Modeling and Knowledge Bases: Foundation, Theory and Applications, IOS Press, pp.586-601 (1992).

[劉 93a] 劉渤江, 塚本昌彦, 西尾章治郎, 宮原秀夫: DOTPL : IS-A 知識に基づく動的オブジェクト識

別と継承機構をもつOODBPL, コンピュータソフトウェア, Vol.10, No.6, pp.54-64 (1993).
[劉 93b] 劉渤江, 牟田道弘, 塚本昌彦, 西尾章治郎: 多重世界の概念を導入したDOT推論システムの構築, 人工知能学会第7回全国大会 (1993).
[清 94] 清一隆, 塚本昌彦, 西尾章治郎: ドット記法とIS-A関係を用いた推論システムの高速化に関する研究, 人工知能学会第8回全国大会 (1994).
[塚本 92a] 塚本昌彦, 西尾章治郎, 長谷川利治: DOT記法とIS-A関係にもとづく項表現を用いた演繹・オブジェクト指向データベースモデル, コンピュータソフトウェア, Vol.9, No.1, pp.10-26 (1992).
[塚本 93] 塚本昌彦, 西尾章治郎: ドット記法とIS-A関係を用いた知識ベースにおける知識の従属性と更新処理, 人工知能学会誌, Vol.8, No.6, pp.769-777 (1993).
[塚本 94a] 塚本昌彦, 西尾章治郎: 正規表現を用いた継承システム, 人工知能学会誌, Vol.9, No.4, pp.115-123 (1994).
[塚本 94b] 塚本昌彦, 西尾章治郎: 正規表現を用いた継承システムにおける知識の無矛盾性, 人工知能学会誌, Vol.9, No.5, pp.122-128 (1994).
[塚本 95a] 塚本昌彦, 西尾章治郎: ドット記法とIS-A関係を用いた知識推論のための完全システム, 人工知能学会誌, Vol.10, No.2 (1995, 掲載予定).
[塚本 95b] 塚本昌彦, 西尾章治郎: ドット記法とIS-A関係を用いた知識表現システムDOT, 人工知能学会誌, Vol.10, No.2 (1995, 掲載予定).
[Tsukamoto91a] Tsukamoto, M., Nishio, S., and Fujio, M.: DOT : A Term Representation using DOT Algebra for Knowledge-bases, Delebel, C., Kifer, M., and Masunaga, Y. (Eds.): Deductive and Object-Oriented Data-bases, Springer-Verlag, pp.391-410 (1991).
[Tsukamoto91b] Tsukamoto, M., Nishio, S., Fujio, M., and Miyamoto, M. : Query Processing for a Knowledge-base Using DOT Algebra, *Proc. of the 1st Int'l Workshop on Interoperability of Multi-Database Systems*, pp.46-53 (1991).
[柳沢 95a] 柳沢豊, 塚本昌彦, 西尾章治郎: 動的継承演繹機構のオブジェクト指向プログラミング言語への導入, 大堀淳 (編): オブジェクト指向コンピューティング III, 近代科学社 (1995, 掲載予定).
[柳沢 95b] 柳沢豊, 塚本昌彦, 劉渤江, 西尾章治郎: 知識ベース独立のための演繹オブジェクト指向プログラミング, 人工知能学会誌, Vol.10, No.6 (1995, 掲載予定).

付録: コマンドの構文

システムが入力として受け付けるコマンドの構文は次のBNFにおける<COMMAND>で表される.

```
<SYMBOL> ::= <英大文字で始まらず, '&', '-', '(', ')',
     ',', '.', '+', '*', '_' を含まない文字列>
<VAR> ::= <英大文字> <SYNBOL>
<DID> ::= '_' <SYNBOL>
<ON> ::= <SYMBOL>
<L> ::= <SYMBOL>
<LE> ::= <L> | <LE> '.' <L>
<DE> ::= <ON> | <ON> '.' <LE>

<LRE> ::= <LE>
 | <LRE> '.' <LRE>
 | <LRE> '+' <LRE>
 | <LRE> '*'
 | '(' <LRE> ')'

<RE> ::= <DE>
 | <RE> '.' <LRE>
 | <RE> '+' <RE>
 | '(' <RE> ')'
 | <RE_Q>
 | <RE_OP>
 | <VAR>
 | <DID>

<REL> ::= <IS_A> | <IS_NOT_A>
<IS_A> ::= 'is-a' | 'is-an'
<IS_NOT_A> ::= 'is-not-a' |  'is-not-an'
<KE> ::= <RE> <REL> <RE> | <VAR> | <DID>

<W> ::= empty
 | '{' <KE> ( ',' <KE> )* '}'
 | <VAR>
 | <DID>

<MW> ::= '{' <W> ( ',' <W> )* '}'
 | <REASON_Q>
 | <VAR>
 | <DID>

<QUERY> ::= <RE_Q> | <REASON_Q>

<RE_Q> ::= '?' <VAR> ':' <VAR> <IS_A> <DE> 'in' <W>
 | '?' <VAR> ':' <VAR> '.' <LE> <IS_A> <DE> 'in' <W>
 | '?' <VAR> ':' <DE> <IS_A> <VAR> 'in' <W>
 | '?' <VAR> ':' <DE> <IS_A> <VAR> '.' <LE> 'in' <W>

<REASON_Q> ::= '?' <DE> <IS_A> <DE> 'in' <W>
 | '?' <KE> 'in' <W>

<RE_OP> ::= <RE> '|' <RE>
 | <RE> '&' <RE>
 | <RE> '-' <RE>

<MW_OP> ::= 'add' <KE> 'to' <MW>
 | 'delete' <MW> 'from' <MW>
 | 'merge' <MW> ( ',' <MW> )*

<SYSTEM> ::= 'show' <VAR>
 | 'show' <COMMAND>
 | 'show' <DID>
 | 'quit'
 | 'clear' <VAR>
 | 'clear'
 | 'delete' <VAR>
 | 'delete' <COMMAND>
 | 'delete' <DID>
 | 'save' <VAR> 'as' filename
 | 'save' 'as' filename
 | 'load' filename

<ASSIGN> ::= <VAR> '=' <COMMAND>
 | <VAR> '=' <DID>

<COMMAND> ::= <QUERY>
 | <MW_OP>
 | <RE_OP>
 | <SYSTEM>
 | <ASSIGN>
```