

Kappaにおける非正規形演繹 データベース

第 4 研究室

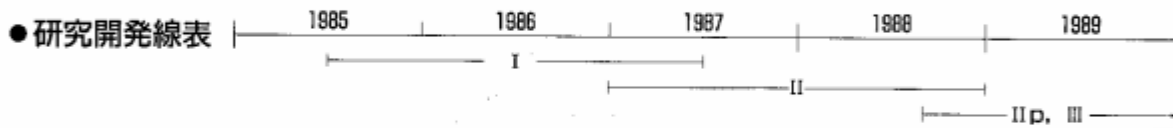
横 田 一 正

Kappaの概要

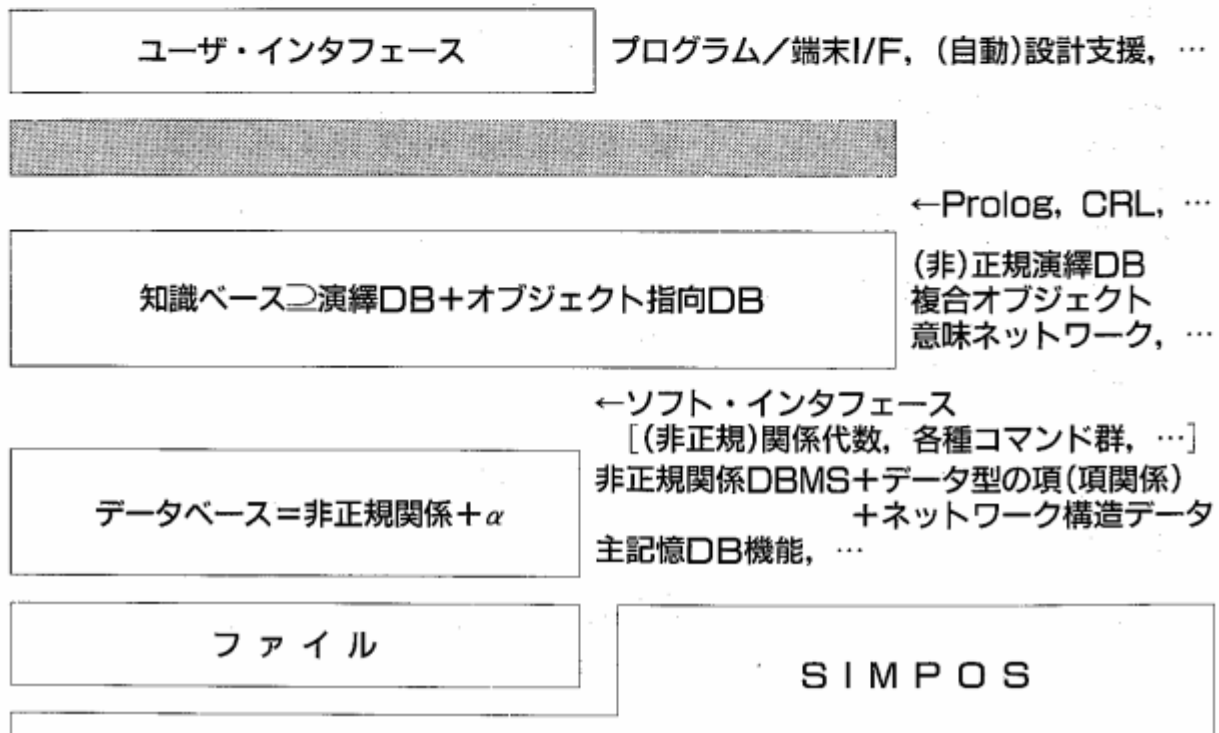
- 知識応用システム向きのDBMS/KBMS (基本ソフトウェア)
自然言語処理, 定理証明支援, 各種エキスパート・システムなど
大容量のDB/KBの構築が可能 (たとえば電子化辞書)
- 知識ベースの演繹DB+オブジェクト指向DB
ワーキング・グループ (1988.4—), 第一回国際会議 (1989.12.4—6, 京都)
非正規関係, 複合オブジェクトは「オブジェクト」のサブクラス
言語=Prolog, CRL, ...
- データ・モデル=非正規関係モデル 構造表現と処理効率
言語 (非正規) 関係代数, 種々のコマンド群
意味論 行 (アン) ネスト演算から独立
- 高水準データ・モデル (基本モデル)
データ型としての項=項関係, ネットワーク型構造データの格納
- 利用者インタフェース
プログラム ソフトI/F (各種応用に合わせ定義可能)
端末利用者 グラフィック機能, マルチ・ウインドウ, ...
(自動)設計支援 意味制約からスキーマ生成

研究開発状況と計画

- Kappa-I (試作システム) 1985.9—1987.8
非正規関係DBMS+データ型としての項 (=項関係)+ネットワーク型構造データ
評価 電子化辞書格納, 効率的に動作, β -テスト中
- Kappa-II (全体システム) 1987.4—1989.3
Kappa-Iの効率化 (CPU, 主記憶) と機能強化 (主記憶DB機能)
端末ユーザ・インタフェースの実装
知識ベース機能の検討, 強化
Prolog, CRLによる演繹データベース機能
意味ネットワークのサポート
演繹+オブジェクト指向データベースWG (1988.4—)
- Kappa-IIp 1988.12—
PIM上にKappa-IIを移植
サブモジュールの並列アルゴリズム検討
- Kappa-III 1988.12—
知識ベース \supset 演繹DB+オブジェクト指向DB
知識ベース層の拡充 より複雑な構造データの扱い (CRL \rightarrow CRL')



システムの全体構成



非正規関係の意味論

- 行ネストによる変換

$$\begin{array}{l}
 a/\{c_1, c_2\} * b/c_4 \\
 a/c_3 * b/\{c_4, c_5\}
 \end{array}
 \begin{array}{l}
 \xrightarrow{a} a/c_1 * b/c_4 \\
 \xrightarrow{a} a/c_2 * b/c_4 \\
 \xrightarrow{a} a/c_3 * b/\{c_4, c_5\} \\
 \xrightarrow{a} a/\{c_1, c_2, c_3\} * b/c_4 \\
 \xrightarrow{a} a/c_3 * b/c_5
 \end{array}
 \begin{array}{l}
 \xleftrightarrow{b} a/c_1 * b/c_4 \\
 \xleftrightarrow{b} a/c_2 * b/c_4 \\
 \xleftrightarrow{b} a/c_3 * b/c_4 \\
 \xleftrightarrow{a} a/c_3 * b/c_5
 \end{array}$$

- 行ネストからの意味の独立

$$\left. \begin{array}{l}
 a/c_1 * b/\{c_2, c_3\} \\
 a/c_1 * b/\{c_4, c_5\}
 \end{array} \right\}
 \begin{array}{l}
 \rightarrow a/c_1 * b/\{c_2, c_3, c_4, c_5\} \\
 \Rightarrow \text{ネスト構造を意識しない質問を可能にする:} \\
 \quad ? - a/\{c_1, c_3\} * b/X \Rightarrow X=c_4 \\
 \rightarrow a/c_1 * b/\{\{c_2, c_3\}, \{c_4, c_5\}\} \\
 \Rightarrow \text{非正規関係の正規形に自動変換}
 \end{array}$$

DB層のソフト・インタフェース

- インタフェース言語/コマンドをユーザ定義可能
保守機能(構造エディタ), デバッグ機能, 管理機能, 実行機能
- システム提供インタフェース
非正規関係代数, ユーティリティ・コマンド, ...
- 非正規関係代数
上記意味論を反映
(ネスト, 射影) × (行, 列), 集合演算, 置換
- 集合演算とスキーマ
非正規関係 R_1, R_2 , 集合演算 \diamond , ネスト属性列 $a = (a_1, \dots, a_n)$ に対して

$$\begin{aligned}
 R_1 \diamond R_2 &= \\
 R_1' &:= \text{unnest}(R_1, a^{-1}), \\
 R_2' &:= \text{unnest}(R_2, a^{-1}), \\
 R' &:= R_1' \diamond R_2', \\
 R &:= \text{nest}(R', a)
 \end{aligned}$$
「キー」の保存 \Rightarrow Verso 集合演算 (実装が容易)
- 拡張非正規関係代数
集合とリストを導入

プログラムとモデル

CRLプログラム：親/{"薫"} * 子供/{"孝", "蘭"}.
 親/{"徹", "夏"} * 子供/{"薫"}.
 先祖/X * 子孫/Y ← 親/X * 子供/Y.
 先祖/X * 子孫/Y ← 親/X * 子供/Z, 先祖/Z * 子孫/Y.

$$p(t_1, \dots, t_n) \Rightarrow p(\$1/t_1 * \dots * \$n/t_n)$$

NAVの意味論： 親/{"薫"} * 子供/{"孝", "蘭"}
 $\Rightarrow \{ \langle \rangle \begin{array}{l} \text{親} \text{---} \text{"薫"}, \\ \text{子供} \text{---} \text{"孝"} \end{array}, \langle \rangle \begin{array}{l} \text{親} \text{---} \text{"薫"}, \\ \text{子供} \text{---} \text{"蘭"} \end{array} \}$

プログラムの意味論：

$$\begin{aligned} & \text{先祖/X * 子孫/Y} \leftarrow \text{親/X * 子供/Z, 先祖/Z * 子孫/Y} \\ & \Rightarrow \iota(\text{親/X * 子供/Z}) \subseteq M \wedge \iota(\text{先祖/Z * 子孫/Y}) \subseteq M \\ & \supset \iota(\text{先祖/X * 子孫/Y}) \subseteq M \end{aligned}$$

モデル：

$\{ \text{pc("薫", "孝"), pc("薫", "蘭"), pc("徹", "薫"), pc("夏", "薫"),$
 $\text{ad("薫", "孝"), ad("薫", "蘭"), ad("徹", "薫"), ad("夏", "薫"),$
 $\text{ad("徹", "孝"), ad("徹", "蘭"), ad("夏", "孝"), ad("夏", "蘭")} \}$.
 ただし $\text{pc}(X, Y) = (\{ \langle \rangle, \text{親}, \text{子供} \}, \{ (\text{親}, X), (\text{子供}, Y) \})$,
 $\text{ad}(X, Y) = (\{ \langle \rangle, \text{先祖}, \text{子孫} \}, \{ (\text{先祖}, X), (\text{子孫}, Y) \})$.

$\{ \langle \rangle \begin{array}{l} \text{親} \text{---} \text{"薫"}, \\ \text{子供} \text{---} \text{"孝"} \end{array}, \langle \rangle \begin{array}{l} \text{親} \text{---} \text{"薫"}, \\ \text{子供} \text{---} \text{"蘭"} \end{array}, \langle \rangle \begin{array}{l} \text{親} \text{---} \text{"徹"}, \\ \text{子供} \text{---} \text{"薫"} \end{array}, \dots \\ \langle \rangle \begin{array}{l} \text{先祖} \text{---} \text{"薫"}, \\ \text{子孫} \text{---} \text{"孝"} \end{array}, \langle \rangle \begin{array}{l} \text{先祖} \text{---} \text{"薫"}, \\ \text{子孫} \text{---} \text{"蘭"} \end{array}, \dots \}$

手続き的意味論と集合の扱い

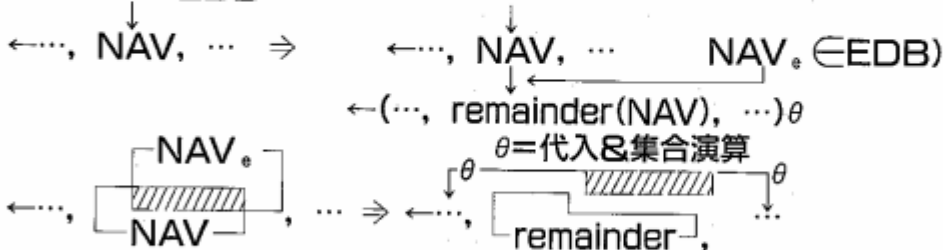
● NAVの単一化

PTTのマージ $t_1 \cup t_2 = (T_1 \cup T_2, f_1 \cup f_2)$, (ただし $f_1 \cup f_2$ が関数として定義可能なとき)
 $s \cup s' = \{ t \cup t' \mid t \in s, t' \in s', t \cup t' \text{ が定義される} \}$
 $\iota(\text{uni}(p, q, \theta)) = \iota(p\theta) \cup \iota(q\theta)$

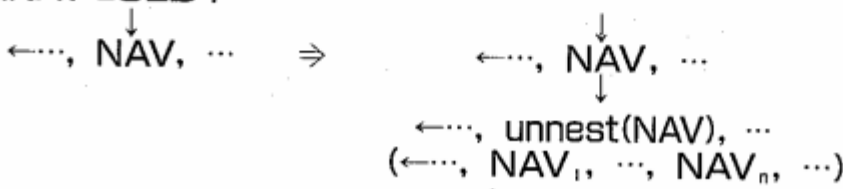
● 演繹データベース (EDB) の枠組：

$\text{DDB} = \text{IDB} \cup \text{EDB}$, $\text{IDB} \cap \text{EDB} = \emptyset$, 基礎NAV $\in \text{EDB}$, 導出NAV $\in \text{IDB}$

● 基礎NAVとEDB



● 導出NAVとSLD木



CRLとHCT/R

- HCT/R (Horn Clause Transformation by Restrictor)

PHI (分散知識ベース・システム) で開発されたアルゴリズム

制約子を用いて, 上昇評価時に下降評価をシミュレート

“マジック集合” はHCT/Rの具体例として捉える

- CRLデータベース

EDB: 親/|“蕭”| * 子供/|“孝”, “蘭”|.

親/|“徹”, “夏”| * 子供/|“蕭”|.

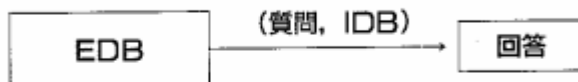
IDB: 先祖/X * 子孫/Y ← 親/X * 子供/Y.

先祖/X * 子孫/Y ← 親/X * 子供/Z, 先祖/Z * 子孫/Y.

質問: ←先祖/|“徹”, “夏”| * 子孫/X.

回答: X = |“孝”, “蕭”|.

- 質問処理



質問: ←先祖/“徹” * 子孫/X, 先祖/“夏” * 子孫/X.

IDB: 先祖/X * 子孫/Y ← 先祖@/X, 親/X * 子供/Y.

先祖/X * 子孫/Y ← 先祖@/X, 親/X * 子供/Z, 先祖/Z * 子孫/Y.

先祖@/“徹”.

先祖@/“夏”.

先祖@/Z ← 先祖@/X, 親/X * 子供/Z.

今後の計画

- 知識ベース ⇨ 演繹DB + オブジェクト指向DB
ワーキング・グループ(DOO-WG) (1988. 4-)
- 自動設計
多値従属性 → スキーマ生成 (非正規関係の正規形)
構造データと意味制約の統一的記述
意味制約の演繹への利用
単一化の制限による深いネストの実現
- 構造の複雑化
深いネスト, 複合オブジェクト, 「オブジェクト」
- 意味の扱い
part_of (複合オブジェクト), is_a (継承), ...
意味モデルの反映
- 意味ネットワーク
明確な応用の存在 (とくに自然言語処理)
サブクラスに対する形式的アプローチ (言語, 推論)
複合オブジェクトを包摂