

知識表現言語 QUIXOTE

概要

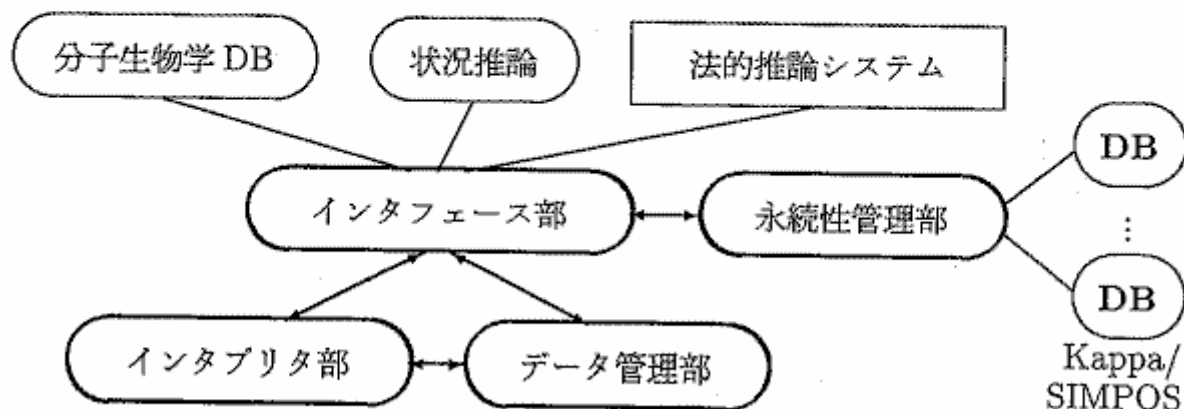
知識情報処理の基盤技術である、知識表現、推論などの基本機能を提供する言語処理系であり、データベースエンジンとして拡張非正規関係データベース管理システム Kappa を利用した、統合型知識ベース管理システム構築のための機能を提供する。

特徴

論理型言語や演繹データベースに、オブジェクト指向概念、包摂制約、モジュール階層などを導入して拡張した、演繹オブジェクト指向データベース (DOOD) のための言語であり、以下の特徴を持つ。

- オブジェクト識別性: オブジェクトの属性から定義される拡張項記述 (オブジェクト項) のオブジェクト識別子としての利用
- 包摂制約: オブジェクト項間の上位 - 下位概念を表す包摂関係を用いた制約記述・処理
- 属性継承: 包摂制約に基づくオブジェクト間の属性継承とオブジェクト識別性に基づく例外の処理
- モジュール: モジュール概念の導入による、知識ベースの階層化
- ルール継承: モジュール間の関係に基づくルールの継承と、ルールの性質に基づく例外の処理
- 条件付き問合せ: 知識ベース定義への追加条件を伴う問合せと条件つき回答

システム構成



デモの構成

QUIXOTE は演繹オブジェクト指向データベースのために知識表現言語であるだけでなく、制約論理型言語、データベース・プログラミング言語、状況プログラミング言語などの多くの特徴を備えている。

QUIXOTE の基本的な特徴とその有効性を、以下の4種類のデモで示す。

- *QUIXOTE* の基本的な特徴
- 状況推論への応用
- 分子生物学データベースへの応用
- 法的推論システム TRIAL

QUIXOTE の基本的な特徴

オブジェクト

QUIXOTE の中心となる概念はオブジェクトである。オブジェクトは、オブジェクト識別子 (OID) と属性 (プロパティ) の集合とからなる。OID は複合オブジェクトで表現する。たとえば

apple, *apple*[*color* = *red*], *apple*[*color* = *red*, *area* = *nagano*]

は OID であり、前者は *apple* という概念を、後2つは固有の属性を持った *apple* の部分概念を表している。非固有の属性は

apple/[*color* = *red*]
apple[*color* = *red*]/[*area* = {*nagano*, *aomori*}, *weight* = *heavy*]

のように “/” の右に書く。

オブジェクトはルールによって内包的に定義することもできる。

path[*from* = *X*, *to* = *Y*] \Leftarrow *arc*[*from* = *X*, *to* = *Y*]
path[*from* = *X*, *to* = *Y*] \Leftarrow *arc*[*from* = *X*, *to* = *Z*], *path*[*from* = *Z*, *to* = *Y*]

包摂関係と属性継承

OID 間には包摂関係 \sqsubseteq が定義できる。

apple \sqsubseteq *fruit*, *fruit* \sqsubseteq *plant*

この拡張として *apple*[*color* = *red*] \sqsubseteq *apple* も定義できる。属性は包摂関係とその拡張の集合間の Hoare 順序 \sqsubseteq_H による制約として表現できる。

apple/[*family* \rightarrow *rose*, *area* \leftarrow {*nagano*, *aomori*}]
 \iff *apple*{*apple.family* \sqsubseteq *rose*, *apple.area* \sqsupseteq_H {*nagano*, *aomori*}

Quixote

包摂関係によって OID は束となり、この順序にしたがって属性継承が定義される。

$$\begin{aligned} & \text{apple}/[\text{color} = \text{red}, \text{family} \rightarrow \text{rose}] \\ & \implies \text{apple}[\text{color} = \text{green}]/[\text{family} \rightarrow \text{rose}, \text{color} = \text{green}] \end{aligned}$$

例外は固有属性によって生じ、多重継承は制約のマージとして処理される。

モジュールとルール継承

ルールの集合をモジュールとして定義できる。

$$\begin{aligned} \text{west} &:: \text{cider}/[\text{source} = \text{apple}] \\ \text{usa} &:: \text{cider}/[\text{alcohol} = \text{no}] \\ \text{uk} &:: \text{cider}/[\text{alcohol} = \text{yes}] \end{aligned}$$

モジュール間にはサブモジュール関係 \sqsubseteq_S が定義でき、ルール継承が行われる。

$$\left. \begin{array}{l} \text{usa} \sqsubseteq_S \text{west} \\ \text{uk} \sqsubseteq_S \text{west} \end{array} \right\} \implies \left\{ \begin{array}{l} \text{usa} :: \text{cider}/[\text{source} = \text{apple}, \text{alcohol} = \text{no}] \\ \text{uk} :: \text{cider}/[\text{source} = \text{apple}, \text{alcohol} = \text{yes}] \\ \text{japan} :: \text{cider}/[\text{source} = \text{soda_pop}] \end{array} \right.$$

ルール継承に関して例外と多重継承も定義できる。パラメータつきモジュールによって動的にモジュールを生成することもできる。

問合せと解

QUIXOTE プログラム (データベース) は、制御情報、包摂関係定義、サブモジュール関係定義、ルール定義の4つの部分から構成されている。以下の問合せが可能である。

- 問合せ時にデータベースに新たな内容を条件として付加できる。

$$\text{if } \text{west} :: \text{cider}/[\text{process} = \text{ferment}] \text{ then } ?\text{-uk}:\text{cider}/[\text{process} = X]$$

- 解は属性に関する仮説つきで返される。

$$?\text{-japan}:\text{cider}/[\text{alcohol} = \text{yes}] \Rightarrow \text{if } \text{japan}:\text{cider}.\text{alcohol} = \text{yes} \text{ then } \text{yes}$$

- 必要ならば解の導出過程を説明として返すこともできる。

データベースに対する更新は、問合せ時の他にプログラム中でも指定ができ、入れ子トランザクションで実行される。QUIXOTE オブジェクトはすべて永続性を持っており、Kappa データベースまたはファイルに格納される。

状況推論への応用

対象とする問題:

自然言語理解における、信念の扱い、テンス・アスペクトなどの時間的情報の扱いにおいては、状況(文脈)に依存した知識の表現と、推論機能が必要とされる。状況推論のために必要となる機構として以下のものがある。

- 環境に依存するパラメータの記述と利用
- 環境に依存した推論規則の適用
- 暗黙的なパラメータの明示化

QUIXOTE の利点:

上の機構は、QUIXOTE の基本的な機能により自然に実現できる。

- モジュールによる状況の表現

$$s :: o \iff s \models \sigma$$

- オブジェクト項とその属性による、パラメータの表現

$$o[\text{rel}/xxx, \text{agent}/yyy, \dots] \iff \langle \text{rel}, \text{agent}, \dots \rangle$$

- モジュールを伴うルール記述と、モジュール間関係の設定による、環境に依存した推論規則の適用

$$s_0 :: o_0 \leftarrow s_1 :: o_1, s_2 :: o_2, \dots \iff s_0 \models \sigma_0 \leftarrow s_1 \models \sigma_1, s_2 \models \sigma_2, \dots$$

- 制約による暗黙的なパラメータの明示化

$$s :: o \leftarrow \dots \parallel C \iff s' \models \langle \sigma, C \rangle$$

- 仮説付き問合せによる、環境情報や仮説的情報の付加

デモ内容:

- 見方 (perspective) の定式化による時間情報の推論

テンス・アスペクトは同じ状況に対する異なった見方であるという前提に立ち、個々の単語が持つ時間特性を集積して、語句全体の時間的意味を導く。例として、複数の意味をとりうる日本語の「ている」を取り上げ、それらの曖昧性を明示する推論を行なう(以下は「着ている」の解釈例)。

(進行) 着る最中である。(結果状態) 着た状態である。(経験) 着たことがある。

- 信念と知識に依存した推論

ある同じ推論規則を適用しても背景にある隠れた信念・知識により、異なった推論結果を導くことがある。ここでは次の例文で、各々の発話者の知識に基づく推論を考える。

If Bizet and Verdi are compatriots, then Bizet is Italian.

If Bizet and Verdi are compatriots, then Verdi is French.

分子生物学への応用

対象とする問題:

分子生物学データは増加の一途であり、自動的な解析の方法を考案することと、そのために可能な限りの知識を自動的に利用できる形で表現することが必要とされている。表現すべき知識としては、タンパク質に関するデータベースという観点からは以下のものが挙げられる。

- タンパク質の機能の記述
- 公共データベースの統合的表現
- 未整理データの暫定表現

QUIXOTE の利点:

QUIXOTE の基本機能は分子生物学の知識表現にさまざまな利点を提供する。下記のデモ内容以外に、以下のような利点が考えられる。

- モジュールによる物質 / 種固有な知識の記述
- モジュールによる各公共データベース固有な知識の記述
- モジュールによる未整理矛盾データの隔離

デモ内容:

● ProSite の表現

機能モチーフのひとつ、zinc finger を例に、QUIXOTE を使った場合とテーブルで格納した場合との表現の違い、使われ方の違いを示す。

- スキーマの柔軟性
- 質問処理能力

● シトクロームの機能

代謝反応の一部であるシトクロームの電子輸送の機能を例に、QUIXOTE のモジュールの効果とオブジェクト間関係の利用法の一例を示す。

- モジュールによる回答の詳細化レベルの設定
- オブジェクト階層によるシソーラス表現

● クエン酸回路の表現

同じく代謝反応の一部であるクエン酸回路と、大腸菌などにみられるグリオキシル酸回路の差異を例に、代謝経路の表現・識別方法について述べる。

- ループを含むパスの表現

法的推論実験システム TRIAL

対象とする問題:

法律分野への知識ベース / 知識表現技術の応用として、法的推論の問題を取り上げる。法的推論は、事実認定、法令解釈、および条文適用の三つの基本的な要素から構成されるが、事実認定は現在の技術水準では取り扱いが難しいため、法令解釈、特に、類似した既存の判例を準用する類推解釈をいかに実現するかが現在の課題である。

QUIXOTE 上の法的推論実験システム TRIAL は、条文適用に加えて、類推解釈を実行することを目指したシステムである。すなわち、新規の事案が与えられると、それに類似した既存の判例を利用して可能な判断を導出し、そしてその判断に至った論証を構築する。

QUIXOTE を用いた利点:

TRIAL では、類推解釈を図式

$$\langle \text{類推解釈} \rangle = \langle \text{概念緩和} \rangle + \langle \text{制約解消} \rangle + \langle \text{条件放棄} \rangle$$

により実現しているが、その実現のためには以下の *QUIXOTE* の提供する機能が有効である:

- 型階層により、法的概念の階層構造に基づく概念緩和が容易に扱える。
- 条件付き解により、条件を伴った判断を得ることができる。また、これにより不足情報の補充が行える。

さらに、法的推論の思考実験のための以下の二つの機能が容易に提供できる:

- 仮説付き問合せおよびモジュール階層により、思考実験のための仮説的な問合せが柔軟に行える。
- 解の説明機能により、推論結果の検証のための論証が構築できる。

デモ概要:

専門家または準専門家が、TRIAL を利用しながら目標とする論証を構築していく過程を、労災認定(「過労死」)の例題を通して紹介する。

- TRIAL への仮説的な問合せにより、求める判断を得る。
- TRIAL の示した論証について抽象化の程度が適切であるか否かを検証する。