

題名	知識ベース管理システム：Kappa
目的	高度な知識情報処理システムの基盤となる大量かつ複雑な知識を統合管理するシステムである。電子化辞書、数学の知識などを蓄積して、自然言語処理、定理証明処理などの広範な応用に利用可能である。
概要及び特徴	<p>Kappa は逐次型推論マシン PSI 上で稼働し、以下の特徴を持つ：</p> <ol style="list-style-type: none"> 1) 非正規関係 (入れ子) の構造の知識を素直に、表現、蓄積できる。 2) 概念ネットワークとして、シソーラスや意味ネットワークが蓄積できる。 3) データ型の一つに項 (ターム) があり、単一化により検索が行える。 4) 具体的応用 (自然言語処理、定理証明処理) を設計に反映している。 5) 大量 (数十万件) の単語辞書や概念辞書を効率的に処理できる。 6) 高機能端末インターフェースと、構造エディタに基づくユーザ・インターフェースを持つ。前者は、互いに連動した複数個の表現形式を持ち、それらの間に制約を設定できる。 7) 対象指向論理型言語 ESP で記述された最大級規模の基本応用ソフトウェアである。
構成	<p style="text-align: center;">(* はデモ中)</p>

国語辞書の操作・編集：1,4,5,6

Kappa に蓄積された 6 万件の国語辞書から単語を検索し、その内容を編集します。

最初にテーブルの構造を御覧に入れます。非正規関係のテーブルの定義を構造エディタを使ったツールから簡単に操作できます。次に、“雨”という語を語尾に持つ言葉を後方一致検索で見つけます。その中で、“五月雨”という単語の内容を詳しく見ます。そして、テキスト形式シートによる編集の簡易性を御覧に入れるために、まず“五月雨”をテキスト形式シートで編集します。さらに、テーブル形式シートとテキスト形式シートとが連動して有効に機能する場面を御覧に入れます。

```

semacs_51 ==> String Node...
Primitive_table prtte has
  structure
    [ atr_a . atr_d [ list . atr_d1 . atr_d2 ] ] ;
  attributes
    atr_a := #access ( index ) , value ( not_nil ) , unique ( table_key ) ;
end .
    
```

Area window

単語	科	生息地	産卵 場所	産卵 季節	色	体長	食用
さけ	さけ	北海	川	秋		70cm	肉 卵
ます	さけ	北緯40度以北	川		褐色 茶の小点 腹は銀白色		食用

連動する

さけは、体長70センチほどの、さけ科の魚、北海に住み、成長すると、秋産卵のため川をのぼる。肉も卵も食用。「しゃけ」ともいう。ますは、さけ科の海魚。だいたい北緯40以北にすむ。褐色で、茶色の小点が散在し、腹は銀白色。海で成長し、川をさかのぼって産卵する。食用。

70cm・200g

20-22・200g

概念辞書の編集 : 2,4,5,6 関連デモ

概念辞書は概念間の関係を記述した辞書です。概念辞書は機械翻訳や知的情報検索などに役立つことが知られています。Kappaに蓄積された概念辞書は、概念ネットワーク形式シートにより容易にアクセスできます。ここでは、単語辞書と組み合わせて次の三つのデモ

- 概念の検索・追加
- 三段論法による基本的な常識の検索
- 多重継承により生じる矛盾の回避

の例を御覧にいたします。

FGCS: '88 Kappa Concept Network Demonstration

Living life :breaths increase :breed intelligence	is_a	Animal life :.... increase intelligence	is_a	Human life :.... increase intelligence :high	is_a	Joiner life :.... increase intelligence
--	------	---	------	---	------	---

Question Menu

Question :
Does Joiner breathe ?

Answer :
Joiner<Human<Animal<Living
Living -> breathe
So Joiner -> breathe

FGCS: '88 Kappa Concept Network Demonstration

Living life :breaths increase :breed intelligence	is_a	Animal life :.... increase intelligence	is_a	Human life :.... increase intelligence :high	is_a	Joiner life :.... increase intelligence
--	------	---	------	---	------	---

Question Menu

Question :
Does PineApple breed ?

Answer :
PineApple<TropicalPlant<Plant<Living
Living -> breed
So PineApple -> breed

PineApple<Crop<Product<Matter
Matter -> not breed
So PineApple -> not breed

Both answers are contradictory to each other.

Matter life :not breathe increase :not breed intelligence :not have	is_a	Product life :.... increase intelligence	is_a	Crop life :.... increase intelligence
---	------	--	------	---

デ モ 内 容 (3/3)

項(ターム)の蓄積・単一化検索:3,4 関連デモ

Kappa が定理証明の分野でどのように有効に機能するかを御覧にいたします。デモでは最初に、ルールの集合、ルール中の演算子、演算子の優先順序を与えて、それから生成しうるすべてのルールを、冗長なルールを除きつつ、生成していきます。そして、その過程において、知識ベースの内容がどのように変化(追加、検索、削除)していくかを御覧にいたします。

Kappa Term Processing System - Version 1.0	
<pre> initial data (1) 0=A (2) A=A*0 (3) A*B=C*A B=0 </pre>	<pre> operator data 0 * (* 1) 0 * 1 * - 0 * (*) </pre>
<pre> rule data No rules </pre>	<pre> operator data 0 / 0 - / 1 - / 2 (lexicographic ordered) left to right </pre>
<pre> Kappa Term Processing System - Version 1.0 initial data No operators </pre>	<pre> operator data read initialize acquisition write ID automatic next equation list ID interactive select trace prove new ordering no trace reduce a term new symbol describe current equation convert save delete exit resurgence critical pairs </pre>
<pre> rule data (1) 0=A (2) A=A*0 (3) A*B=C*A B=0 (4) A*(A*B) -> B (7) A*0 -> A (8) -(A) -> A (9) A*(A) -> 0 (10) 0 -> 0 (11) A*(A+B) -> B (15) -(A*B) -> -(B*(A)) </pre>	<pre> operator data delete (0) - 0 * A -> A A * (A * B) -> B ... (11) A * (B * (A * B)) -> 0 ... (12) B * (A * B) -> A ... (13) delete (12) A * (B * (A * B)) -> 0 -(A * B) -> A ... (14) -(B * A) -> A * (B) ... (15) delete (13) A * (B * A) -> B delete (14) -(A * B) -> A </pre> <p>Each - Bndix is terminated. your system is COMPLETE.</p>