5-1 第５回の内容

（人工知能の基本）

URL: https://www.kkaneko.jp/db/mi/index.html

金子邦彦

第５回の内容

- **プロダクションシステム**

  知識と**推論**をコンピュータで.

【次回に向けての準備学習】
インターネットや教科書などを使い「**述語**」について調べてみる

5-2 総当たりでの変数の値の変化

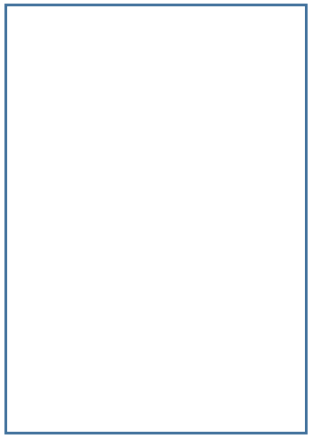
（人工知能の基本）

URL: https://www.kkaneko.jp/db/mi/index.html

金子邦彦

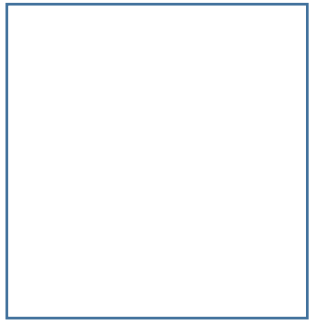
２つの水差し

- 水差し①　大きさ**4**
- 水差し②　大きさ**3**



水差し①

水の量：変数 x



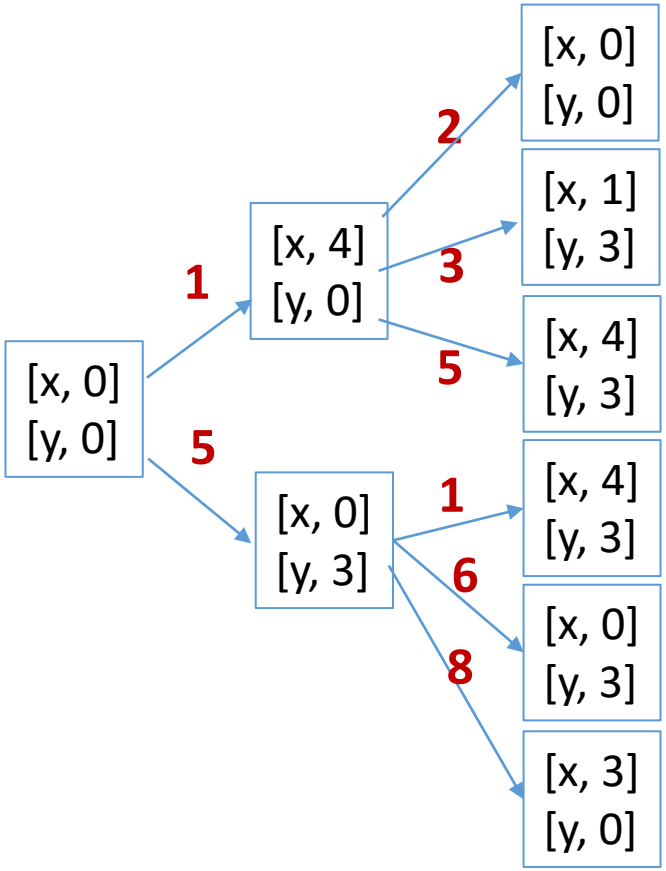
水差し②

水の量：変数 y

総当たり

- **総当たり**では，すべての経路（**パス**）を試す

```
(1, 2) 0 0
(1, 3) 1 3
(1, 5) 4 3
(5, 1) 4 3
(5, 6) 0 3
(5, 8) 3 0
```



パス長２の経路（パス）をすべて試す

パス：（１，２）など

最終状態：０　０などは x, y の値を表示

5-3 知識表現の例

（人工知能の基本）

URL: https://www.kkaneko.jp/db/mi/index.html

金子邦彦

ここで行うこと

- 次の知識を，コンピュータで扱う

  変数 **x** の値は **0**

  変数 **y** の値は **0**

```
1  m = [['x', 0], ['y', 0]]
2  print(m)
```

```
[['x', 0], ['y', 0]]
```

すべてを m に格納する Python 言語のプログラム

ソースコードと実行結果

```
1  m = [['x', 0], ['y', 0]]
2  print(m)
```

ソースコード（Python 言語）

```
[['x', 0], ['y', 0]]
```

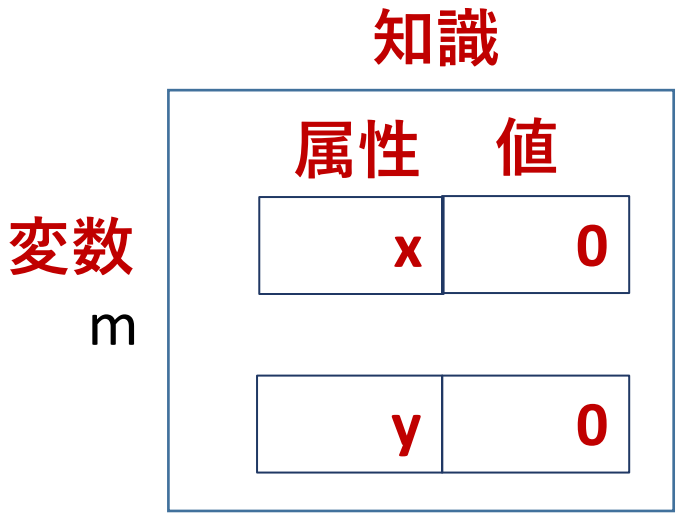
実行結果

すべてを m に格納する Python 言語のプログラム

GDB online を使用

https://www.onlinegdb.com/

※ プログラム実行開始は「Run」，停止は「Stop」




```
1  m = [['x', 0], ['y', 0]]
2  print(m)
```

```
[['x', 0], ['y', 0]]
```

すべてを m に格納する Python 言語のプログラム

5-4 知識表現の例②

（人工知能の基本）

URL: https://www.kkaneko.jp/db/mi/index.html

金子邦彦

知識

属性	値
X	100
Y	-50

属性	値
taimou	True
nikusyoku	True

```
1  m = [['X', 100], ['Y', -50]]
2  print(m)
```

```
[['X', 100], ['Y', -50]]
```

```
1  m = [['taimou', True], ['nikusyoku', True]]
2  print(m)
```

```
[['taimou', True], ['nikusyoku', True]]
```

Python 言語のプログラム

演習問題

- 属性 **taimou** の値が **True**，属性 **nikusyoku** の値が **True** であることを，Python 言語のプログラムで次のように書くことができる

[['**taimou**', **True**], ['**nikusyoku**', **True**]]

属性 **akarui** の値が **True**，属性 **hima** の値が **True** であることを，同じ書き方で書きなさい

5-5 プロダクションシステム①

（人工知能の基本）

URL: https://www.kkaneko.jp/db/mi/index.html

金子邦彦

プロダクションシステムの特質

① **総当たり**において，**パス**が**複数あるとき**，どの**パス**であっても，**最終状態**が**同じ**という場合がある

このとき，**探索**は**考えない**．しかし，

最終状態を得ることには意味がある

② **新しい属性**を生成するための**ルール**の書き方

（②は次の資料で説明）

ルールの例

1. (x, y | x > y) → y, x
2. (x, y | x < 0) → 0, y

条件

x, y の値をどう変化させるか

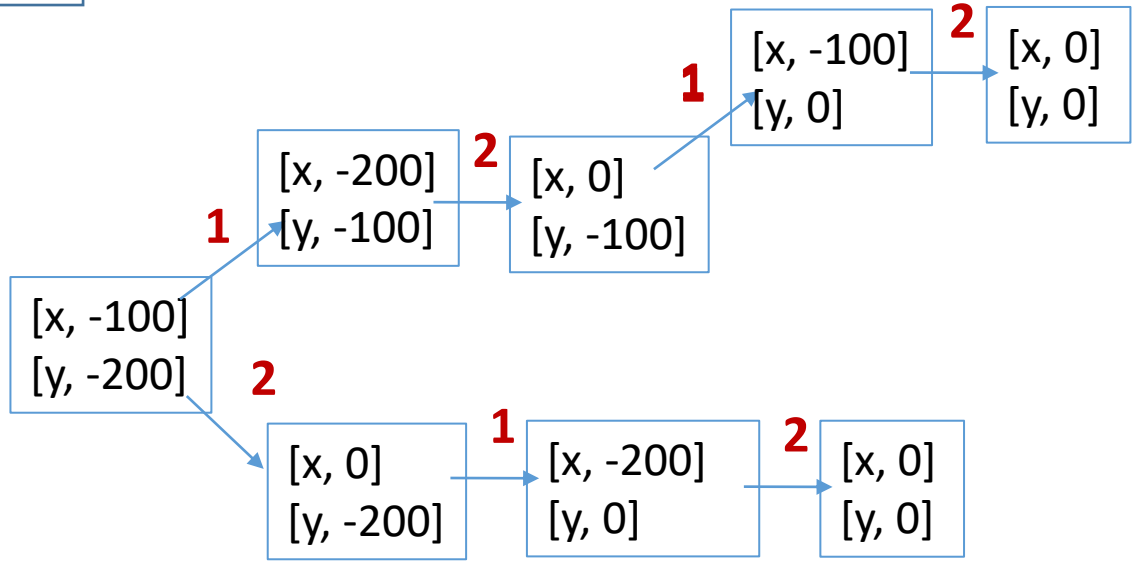
ルール

1. (x, y | x > y) → y, x
2. (x, y | x < 0) → 0, y

ルール

X	-100
Y	-200

スタート



最終状態が同じ

まとめ

① **総当たり**において，**パス**が**複数あるとき**，どの**パス**であっても，**最終状態**が**同じ**という場合がある

このとき，**探索**は**考えない**．しかし，

最終状態を得ることには意味がある

演習問題

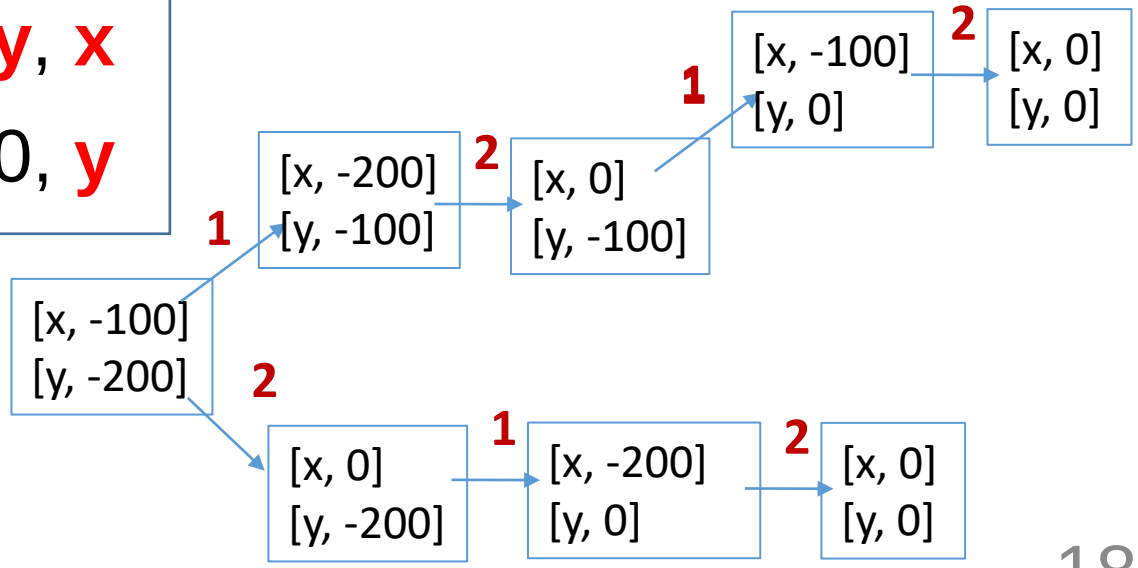
- **x** の値が **-100**, **y** の値が **-200** のとき，**最終状態**は？

　　答．　**x** の値は **0**，**y** の値は **0**

では，**x** の値が **-100**, **y** の値が **-50** のとき，最終状態は？

1. (**x, y** | **x** > **y**) → **y**, **x**
2. (**x, y** | **x** < 0) → 0, **y**

ルール



5-6 プロダクションシステム②

（人工知能の基本）

URL: https://www.kkaneko.jp/db/mi/index.html

金子邦彦

プロダクションシステムの特質

① **総当たり**において，**パス**が**複数あるとき**，どの**パス**であっても，**最終状態**が**同じ**という場合がある

このとき，**探索**は**考えない**．しかし，

最終状態を得ることには意味がある

（①は次の資料で説明）

② **新しい属性**を生成するための**ルール**の書き方

ルールの例

1. (**体毛, 種類, 肉食| 体毛 = ‘ある’**) → **体毛**, **‘哺乳類’, 肉食**

体毛が「ある」ならば，種類は「哺乳類」である

2. (**体毛, 種類, 肉食| 種類 = ‘哺乳類’ and ‘肉食’ = する**) → **体毛**, **‘肉食動物’, 肉食**

種類が「哺乳類」であり，肉食が「する」ならば，種類は「肉食動物」である

1. (**体毛, 種類, 肉食| 体毛 = ‘ある’**) → **体毛, ‘哺乳類’, 肉食**

2. (**体毛, 種類, 肉食| 種類 = ‘哺乳類’ and ‘肉食’ = する**) → **体毛, ‘肉食動物’, 肉食**

スタート

[体毛, ある]
[種類, 不明]
[肉食, する]

1 →

[体毛, ある]
[種類, **哺乳類**]
[肉食, する]

2 →

[体毛, ある]
[種類, **肉食動物**]
[肉食, する]

最終状態

ここで行うこと

- **不明**のときには，**属性は作らない**ことにする

  種類が不明のとき，「種類」という名前の属性は作らない

ルールの別の書き方

1. **‘体毛’ = ‘ある’** → [‘種類’, **‘哺乳類’**]
2. ‘**種類’ = ‘哺乳類’ and ‘肉食’ = ‘する’**→ [‘種類’, **‘肉食動物’**]

条件

属性の値の変化，新しく生成される属性

※ 属性が無ければ作る．あれば値を変化させる

スタート

[‘体毛’, ‘ある’]
[‘肉食’, ‘する’]

→ 1 →

[‘体毛’, ‘ある’]
[‘種類’, **‘哺乳類’**]
[‘肉食’, ‘する’]

→ 2 →

[‘体毛’, ‘ある’]
[‘種類’, **‘肉食動物’**]
[‘肉食’, ‘する’]

最終状態

まとめ

② **新しい属性**を生成するための**ルール**の書き方

属性が無ければ作る．あれば値を変化させる

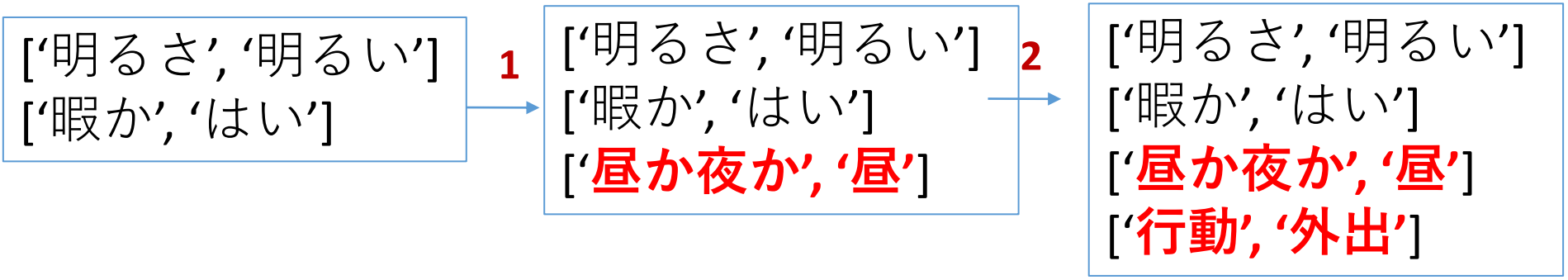
演習問題

- 次の２つのルールがある

1. **‘明るさ’ = ‘明るい’ → [‘昼か夜か’, ‘昼’]**
2. **‘昼か夜か’ = ‘昼’** and **‘暇か’ = ‘はい’**
   → [‘行動’, ‘外出’]

- 次を確認しなさい

スタート



最終状態

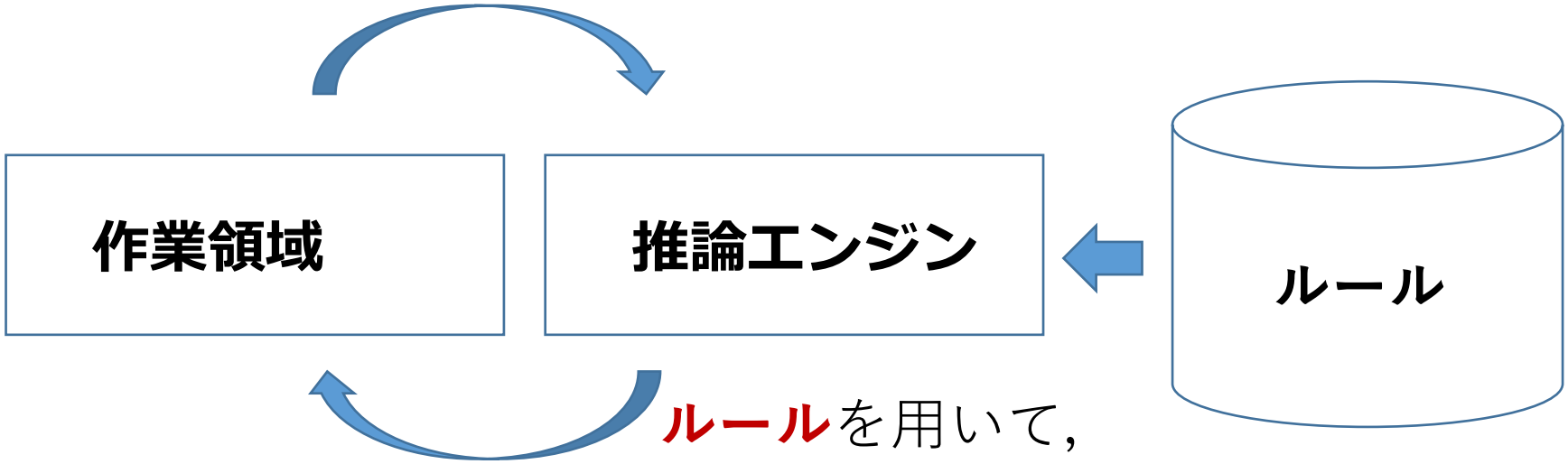
5-7 プロダクションシステムの仕組み

（人工知能の基本）

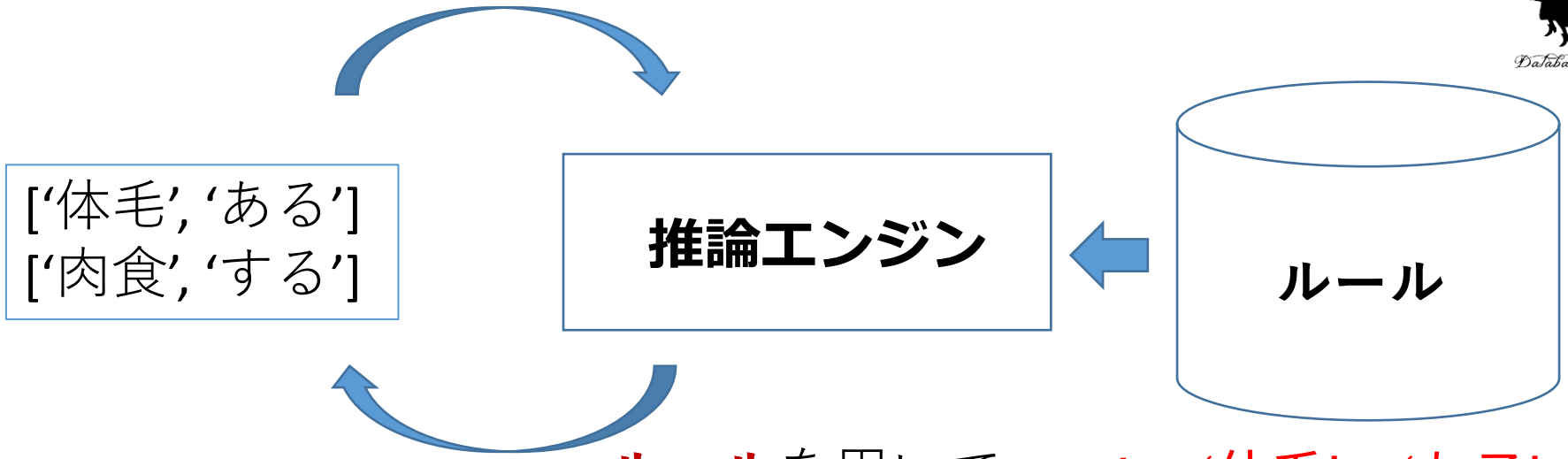
URL: https://www.kkaneko.jp/db/mi/index.html

金子邦彦

プロダクションシステムの仕組み



ルールを用いて，
作業領域内の知識
を**変化**させる**推論**
を行う



スタート

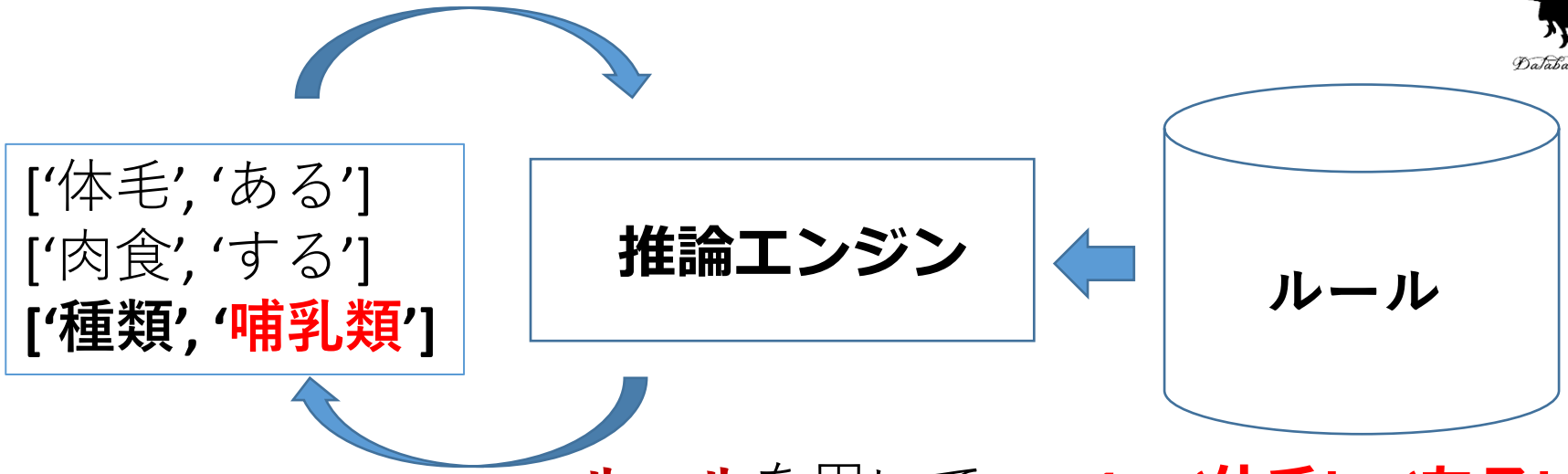
ルールを用いて，
作業領域内の知識
を**変化**させる**推論**
を行う

1. '体毛' = 'ある'

→ ['種類', '哺乳類']

2. '種類' = '哺乳類'
and '肉食' = 'する'

→ ['種類', '肉食動物']



ルール１による変化

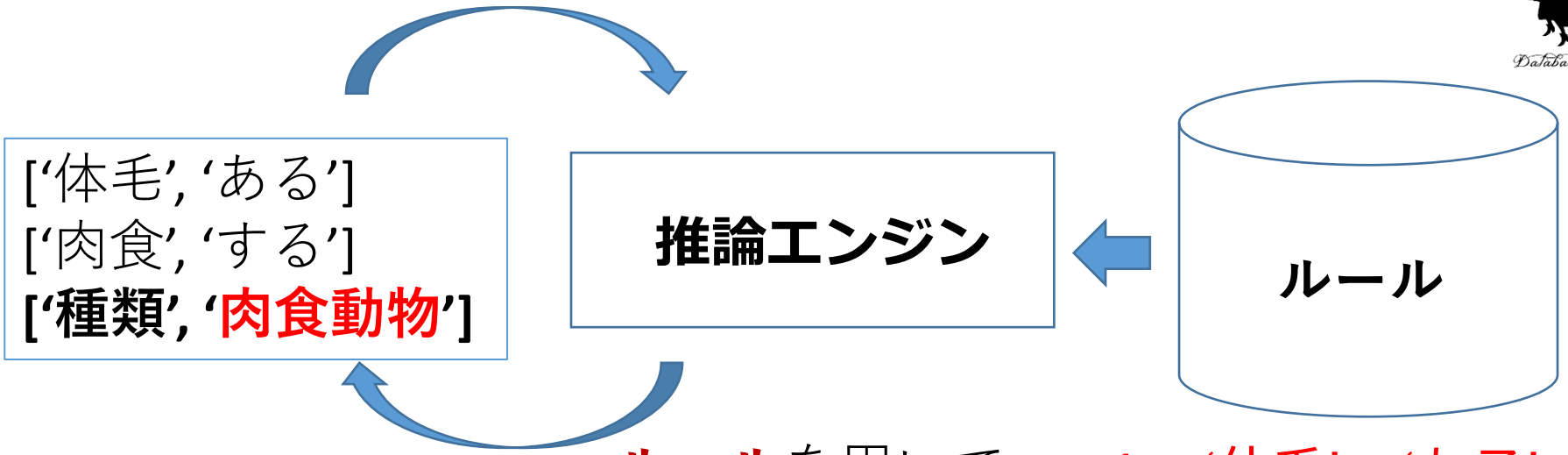
ルールを用いて，**作業領域内の知識**を**変化**させる**推論**を行う

1. **'体毛' = 'ある'**

→ **['種類', '哺乳類']**

2. '種類' = '哺乳類' and '肉食' = 'する'

→ ['種類', '肉食動物']



ルール２による変化

ルールを用いて，**作業領域内の知識**を**変化**させる**推論**を行う

1. '体毛' = 'ある'

→ ['種類', '哺乳類']

2. **'種類' = '哺乳類' and '肉食' = 'する'**

→ **['種類', '肉食動物']**

推論エンジンの仕組み

(1) 作業領域と**ルール**を調べる

　　ルールの**条件**により使える**ルール**を探す

(2) (1) で探した**条件**について，そこで定められた

　　ルールの通りに，**作業領域**を書き換える

　　＝**推論**

(3) 以上を繰り返し，**使えるルール**が**見つからなくなったら**，終了.

　　結論を得る

使える**ルール**が複数ある場合.

その中で，どの**ルール**を**最初に選ぶか**，さまざまなバリエーションを考えることが可能.

- 順序：　ルール番号の小さいものを優先
- 優先順位：　ルールごとに前もって優先順を決めておく
- 最新：　新しく追加されたものを優先
- 複雑さ：　複雑なルールを優先

プロダクションシステム　まとめ

① 作業領域

- 作業領域には**知識**を置く。作業領域の**知識**は、**変化する**ものである。
- **知識**は、次のような形で書くことができる。

['体毛', 'ある']

['肉食', 'する']

② 推論エンジン

- **ルール**を用いて作業領域内の**知識**を変化させるという推論を行う

③ ルール

- **ルール**は既存の知識から新しい**知識**を生み出したり、**知識**を変化させるためのルール**である。**
- **ルールは**、次のように書くことができる。

'体毛' = 'ある'→ ['種類', '哺乳類']

'種類' = '哺乳類' and '肉食' = 'する' → ['種類', '肉食動物'

5-8 人工知能の種類

（人工知能の基本）

URL: https://www.kkaneko.jp/db/mi/index.html

金子邦彦

人工知能システムの種類

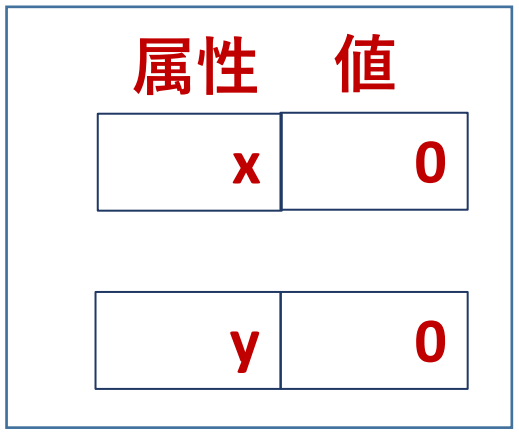
（１）知識や知的能力をコンピュータに組み込んでおく

最初から知的である

１．状態空間表現，ルール
探索

２．作業領域内の知識，ルール
推論

３．述語を用いた知識表現
推論，マッチング



（２）学習能力をコンピュータに組み込んでおき，あとでデータを与えて学習させる

学習能力がある