

mi-7. Prolog

人工知能を演習で学ぶシリーズ（7）

<https://www.kkaneko.jp/dblab/intro/m/index.html>

金子邦彦



7-1 人工知能デモサイト

パソコン演習

資料

<https://www.kkaneko.jp/a/m.html>

ケイ かねこ ドット ジエイ ピー

スラッシュ

エイ

スラッシュ

エム ドット エイチ ティー エム エル

① 「7-1. 人工知能デモサイト」のところのリンクをクリック

[[サイトマップ](#)へ], サイト内検索:

Google

授業「人工知能」サポートページ

担当者： 金子邦彦

第7回授業

7-1. 人工知能デモサイト

- <https://tenso.rs/demos/fast-neural-style>

自分で画像をアップロードして、画像のスタイル変換を試してみなさい。

7-2. 述語

- <https://www.tutorialspoint.com/codingground.htm>

説明資料: [第7回授業資料 \[PDF版\]](#), [\[パワーポイント版\]](#),

画像のスタイル変換

画像の種類

画像のアップロード



スタイルの種類

<https://tenso.rs/demos/fast-neural-style>

7-2 述語

述語での知識表現

事実

ali は **male** (男性) である。
veli は **male** (男性) である
zeyn は **female** (女性) である。

	male	female	} 属性名
ali	true	false	
veli	true	false	
zeyn	false	true	

「述語」というとき,

属性値は true か false のみ

true: そうである

false: そうでない. 知らない

述語での知識表現

事実

ali は anne の parent (親) である。
ali は amet の parent (親) である。
zeyn は anne の parent (親) である。
zeyn は amet の parent (親) である

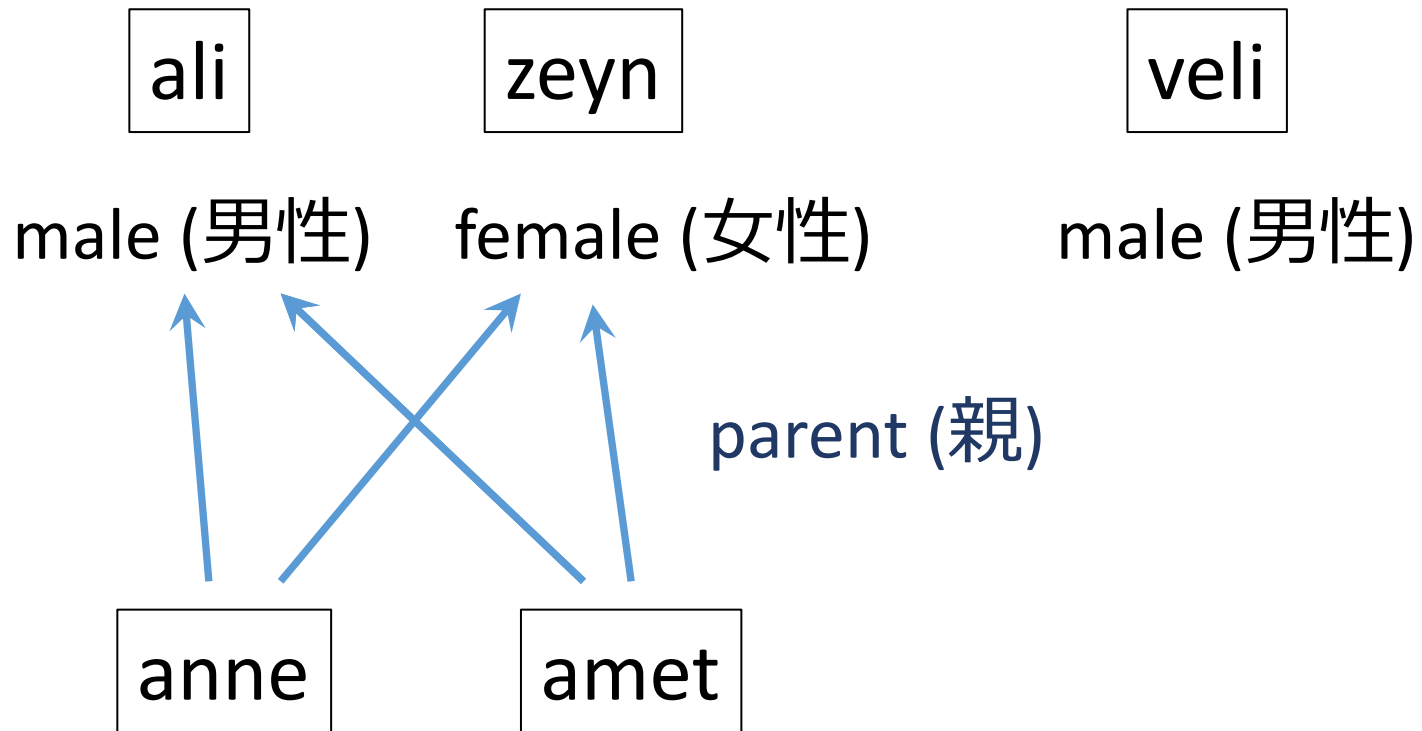
		parent	属性名
ali	anne	true	
ali	amet	true	
zeyn	anne	true	
zeyn	amet	true	

「述語」というとき,

true: そうである

属性値は true か false のみ

false: そうでない. 知らない



述語

事実

ali は **male** (男性) である.

veli は **male** (男性) である

zeyn は **female** (女性) である.



male(ali).

male(veli).

female(zeyn).

述語

述語

事実

ali は **anne** の **parent** (親) である。
ali は **amet** の **parent** (親) である。
zeyn は **anne** の **parent** (親) である。
zeyn は **amet** の **parent** (親) である



parent(ali, anne).
parent(ali, amet).
parent(zeyn, anne).
parent(zeyn, amet).

2つを
カンマで区切る

述語

述語

parent(taro, hanako).

parent(hanako, taro).

この2つの**述語**は、意味は全く違う

述語

ルール (規則)

X が **Y** の parent (親) ならば,
Y は **X** の child (子供) である



child(Y, X) \Leftarrow parent(X, Y).

述語

X, Y は変数

述語の種類, 書き方

- 事実

```
male(ali).
```

```
parent(ali, anne).
```

関係性を書くこともできる

半角かっこで囲む. 末尾はピリオド

- ルール (規則)

```
child(Y, X) :- parent(X, Y).
```

半角かっこで囲む. 末尾はピリオド
左向き矢印の代わりに「:-」

この授業では, 変数は大文字. 残りはすべて小文字

演習問題①

次の**事実**、**ルール（規則）**を**述語**で書きなさい

事実

taro は **human** (人間) である.

ルール
(規則)

X が **human** (人間) ならば,
X は **happy** (幸せ) である.

7-3 Prolog

Prolog とは

- **事実**や**規則（ルール）**を**述語**の形で書く.
- **述語**が, コンピュータの**プログラム**になる.
- コンピュータが様々な**質問**に答える能力を持つ

質問の例

コンピュータの答え

male(ali).

• • • true

male(zeyn).

• • • false

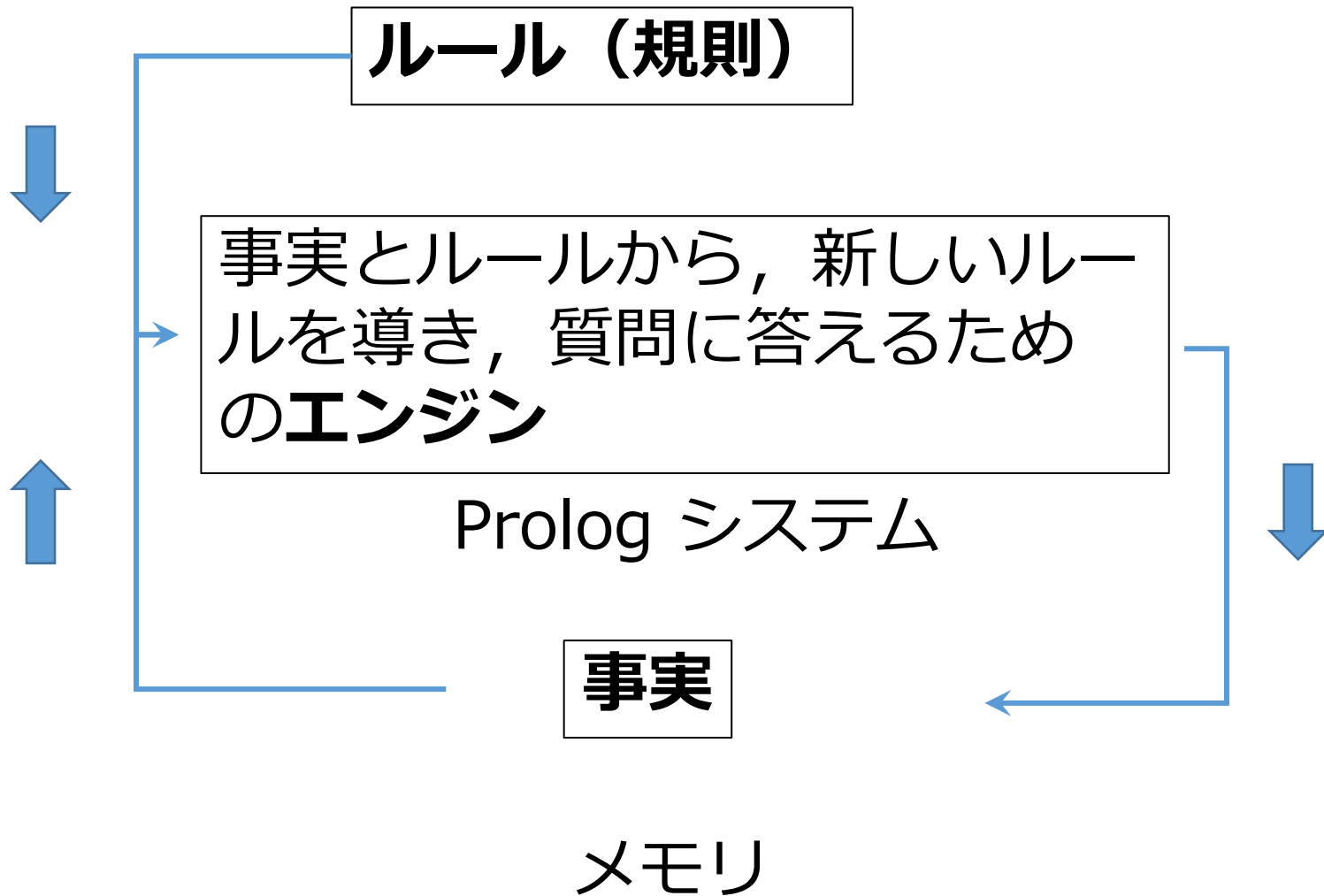
parent(ali, anne).

• • • true

child(anne, ali).

• • • true

Prolog の仕組み



① 「7-2. 述語」のところのリンクをクリック

第7回授業

7-1. 人工知能デモサイト

- <https://tenso.rs/demos/fast-neural-style>

自分で画像をアップロードして、画像のスタイル変換を試してみなさい。

7-2. 述語

- <https://swish.swi-prolog.org/>

- プログラム 1つめ

```
male(ali).
male(veli).
female(zeyn).
parent(ali, anne).
parent(ali, amet).
parent(zeyn, anne).
child(Y,X) :- parent(X,Y).
```

② Prolog の画面に変わる. 「Program」をクリック.

SWISH File Edit Examples Help 184 users online Search

Create **Program** Notebook here
based on Empty Student CLP profile

user:"me" Filter Type

⚠ No matching files
If you are a new user you may

- Use the Examples menu from the navigation bar
- Use the Program or Notebook button above

[help on search](#)

?- parent(ali, anne).

Examples History Solutions table results Run!

Your Prolog rules and facts go here ...

事実, 記録 (ルール)
の編集

答えが表示される

?- parent(ali, anne).

質問をコンピュータに
与える

実行履歴 (History) 結果 (Results)

事実, 記録 (ルール)
をコンピュータに
読み込ませるボタン

② 「プログラム 1つめ」

第7回授業

7-1. 人工知能デモサイト

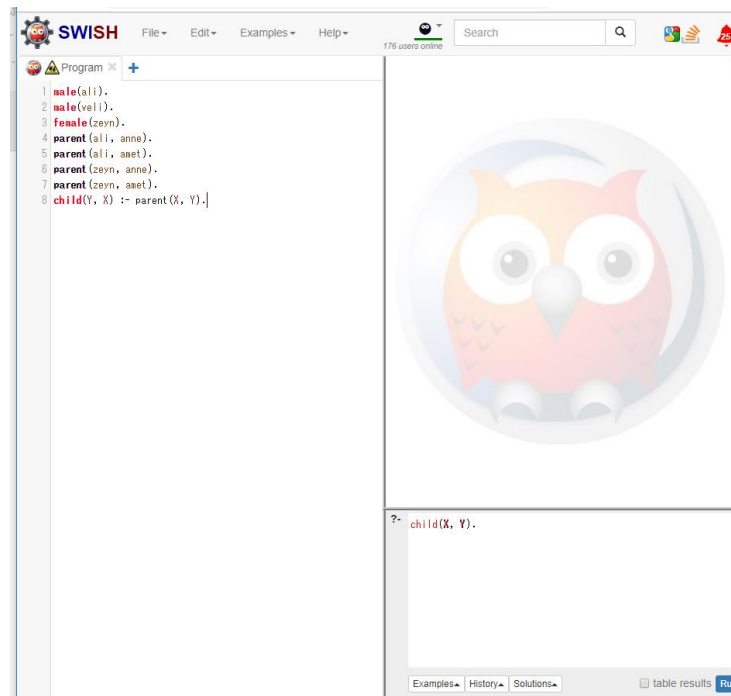
- <https://tenso.rs/demos/fast-neural-style>

自分で画像をアップロードして、画像のスタイル変換を計

7-2. 述語

- <https://swish.swi-prolog.org/>
- プログラム 1つめ

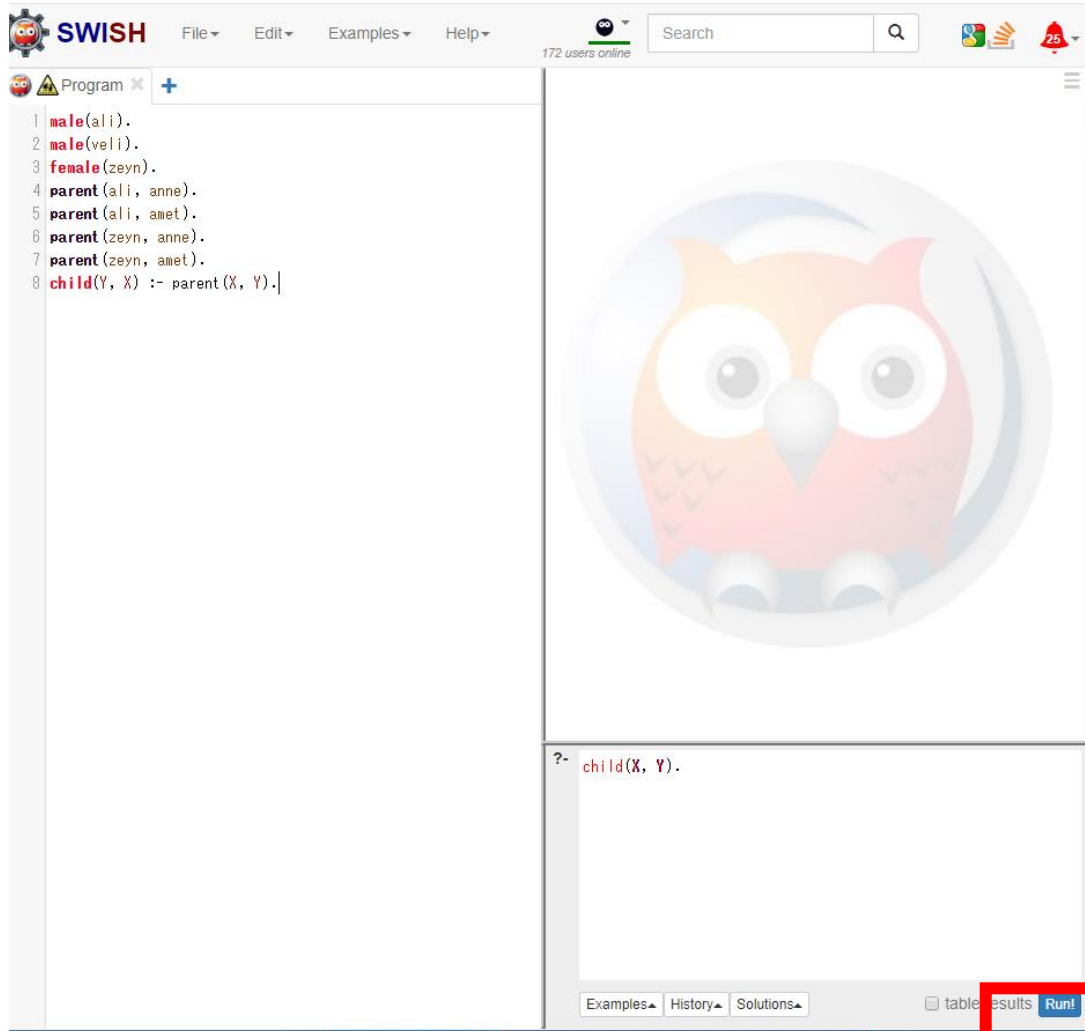
```
male(ali).
male(veli).
female(zevn).
parent(ali, anne).
parent(ali, amet).
parent(zevn, anne).
parent(zevn, amet).
child(Y, X) :- parent(X, Y).
```



マウスの右クリックメニュー
で「貼り付け」が便利

コピーしたい範囲をマウスで選び、
マウスの右クリックメニュー
で「コピー」が便利

③ 「Run」 をクリックして, 少し待つ.



The screenshot shows the SWISH Prolog IDE interface. The top menu bar includes "File", "Edit", "Examples", and "Help". A search bar is located on the right. The main editor area contains the following Prolog code:

```
1 male(ali).  
2 male(veli).  
3 female(zeyn).  
4 parent(ali, anne).  
5 parent(ali, amet).  
6 parent(zeyn, anne).  
7 parent(zeyn, amet).  
8 child(Y, X) :- parent(X, Y).
```

The right side of the interface displays a large owl logo. Below the logo, a query window shows the query: `?- child(X, Y).` At the bottom right, the "Run!" button is highlighted with a red box.

質問「male(ali).」を入れ, 「Run」ボタン
male かつこ ali かつこ ピリオド Enterキー

The image shows a Prolog IDE interface. On the left, a large box highlights the query and result. The query is `male(ali).` and the result is `true`. On the right, a smaller window shows the same query and result, with a `Run!` button highlighted. The background features a large owl logo.

答え

質問

「true」を確認

エラーメッセージが出ることがあります



```
mail(ali).
```

```
procedure `mail(A)' does not exist
```

```
?-
```


```
mail(ali).
```

正しくは「male」

間違って「mail」と書いてしまった

質問を，1度に2個書くのはやめよう

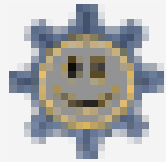
```
?- male(ali).  
ma/e(zeyn).|
```



```
male(ali). male(zeyn).  
Cannot run query due to a syntax error (check query window)  
?- male(ali).  
Syntax error: Operator expected  
ma/e(zeyn).
```

質問を書くたびに，
前の質問は消すこと。

質問「male(zeyn).」を入れ, 「Run」ボタン
male かつこ zeyn かつこ ピリオド Enterキー



```
male(zeyn).
```

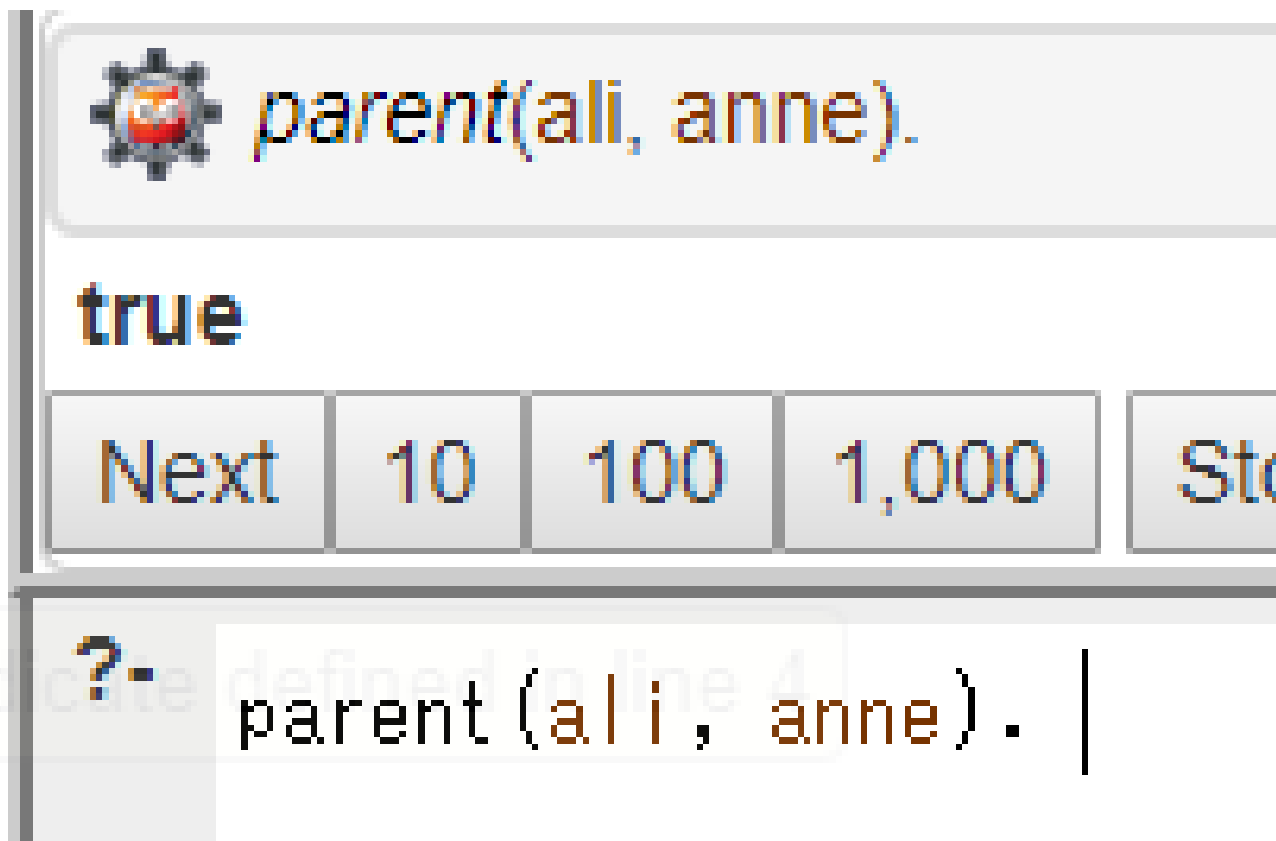
```
false
```



```
male(zeyn).
```

「false」を確認

質問 「parent(ali, anne).」 を入れ, 「Run」 ボタン



The screenshot shows a Prolog IDE interface. At the top, a query box contains a gear icon and the text `parent(ali, anne).`. Below this, the result `true` is displayed. A row of buttons is visible, including `Next`, `10`, `100`, `1,000`, and `Stop`. Below the buttons, a code editor shows the query `?- parent(ali, anne).` with a cursor at the end of the line. A tooltip above the cursor indicates that the predicate is defined in line 4.

「true」を確認

質問 「child(anne, ali).」 を入れ, 「Run」 ボタン

The screenshot shows a Prolog environment with a query and its result. The query is `child(anne, ali).` and the result is `true`. Below the result, there are buttons for `Next`, `10`, `100`, `1,000`, and `Stop`.

```
true
```

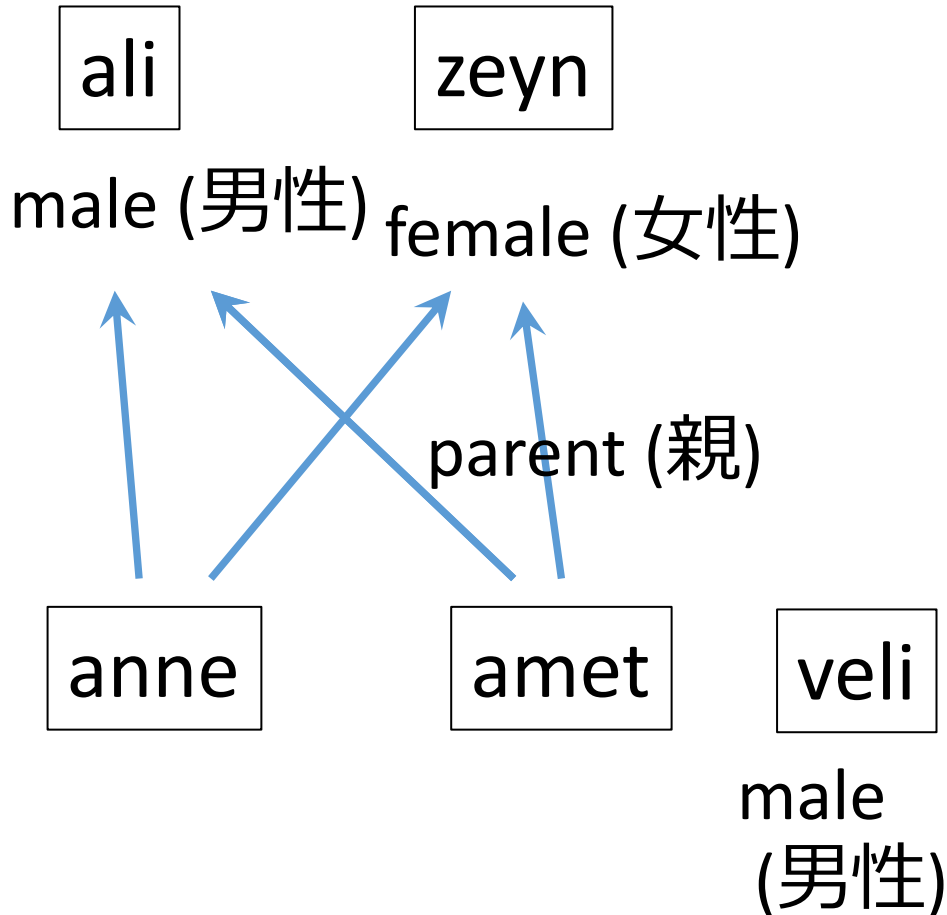
Next	10	100	1,000	Stop
------	----	-----	-------	------

```
?- child(anne, ali).
```

「false」を確認

演習問題②

質問の答えは何になるか，空欄に記入しなさい



質問	答え
female(veli).	
male(anne).	
child(veli, ali).	
child(amet, ali).	

Prolog での質問の答え

- **true**

事実や規則（ルール）での裏付けがあるとき

- **false**

「**不明**」の意味

```
male(anne).  
false  
?- male(anne).
```

false は、「男ではない」とは
言っていない

「男である裏付けがない」
(不明) の意味.

7-4 Prolog で答えの全探索

全探索

- 質問に書いた答えをすべて求めること

質問の例

コンピュータの答え

male(X). • • • X = ali, veli

child(X, Y). • • • (X, Y) =

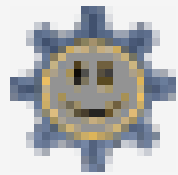
(anne, ali),

(amet, ali),

(anne, zeyn),

(amet, zeyn),

質問「male(X).」を入れ, 「Run」ボタン
そして「Next」ボタンをクリック
(答えが増えなくなるまで)



```
male(X).
```

```
X = ali
```

```
X = veli
```



```
male(X).
```

質問「`child(X).`」を入れ, 「Run」ボタン
そして「Next」ボタンをクリック
(答えが増えなくなるまで)

```
X = anne,
```

```
Y = ali
```

```
X = amet,
```

```
Y = ali
```

```
X = anne,
```

```
Y = zeyn
```

```
X = amet,
```

```
Y = zeyn
```

```
?- child(X,Y).
```

演習問題③

それぞれの質問の答えは何か。

空欄に書きなさい。

X は

--

female(X).

X, Y は

X

Y

X	Y

parent(X, Y).

7-5 演習問題

- 事実

taimou(cat). 猫に体毛 あり

nikusyoku(cat). 猫は肉食

- 規則 (ルール)

honyu(X) :- taimou(X).

体毛があれば哺乳類

nikusyokudoubutu(X) :- honyu(X), nikusyoku(X).

哺乳類で, 肉食であれば,

肉食動物である

(架空の例)

```
1 taimou(cat).  
2 nikusyoku(cat).  
3 honyu(X) :- taimou(X).  
4 nikusyokudoubutu(X) :- honyu(X), nikusyoku(X).  
5
```

honyu(X) の結果は何か

```
X = cat
```

```
?
```

```
honyu(X).
```


nikusyokudoubutu(X) の結果は何か

```
X = cat
```

```
?
```

```
nikusyokudoubutu(X).
```

- 次の2つの事実を書き加えたとする。
(2行目と4行目)

```
1 taimou(cat).  
2 taimou(dog).  
3 nikusyoku(cat).  
4 nikusyoku(dog).  
5 honyu(X) :- taimou(X).  
6 nikusyokudoubutu(X) :- honyu(X), nikusyoku(X).  
7
```

1. honyu(X). の答えをすべて書きなさい。
2. nikusyokudoubutu(X). の答えをすべて書きなさい。

演習問題①

次の**事実**, **ルール (規則)** を**述語**で書きなさい

事実

taro は **human** (人間) である.

ルール
(規則)

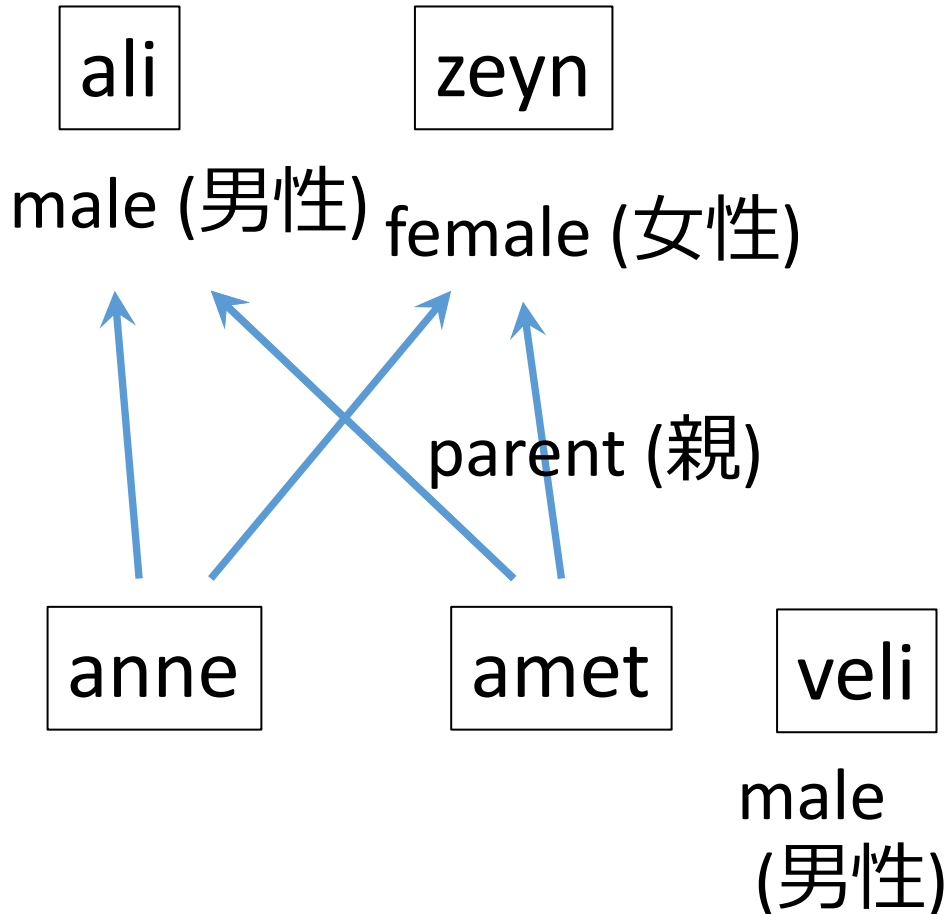
X が **human** (人間) ならば,
X は **happy** (幸せ) である.

human(taro).

happy(X) :- human(X).

演習問題②

質問の答えは何になるか，空欄に記入しなさい



質問	答え
female(veli).	false
male(anne).	false
child(veli, ali).	false
child(amet, ali).	true

演習問題③

それぞれの質問の答えは何か。

空欄に書きなさい。

female(X).

X は

zeyn

X, Y は

X

Y

ali	anne
ali	amet
zeyn	anne
zeyn	amet

parent(X, Y).

述語の種類, 書き方

- 事実

```
male(ali).
```

```
parent(ali, anne).
```

関係性を書くこともできる

半角かっこで囲む. 末尾はピリオド

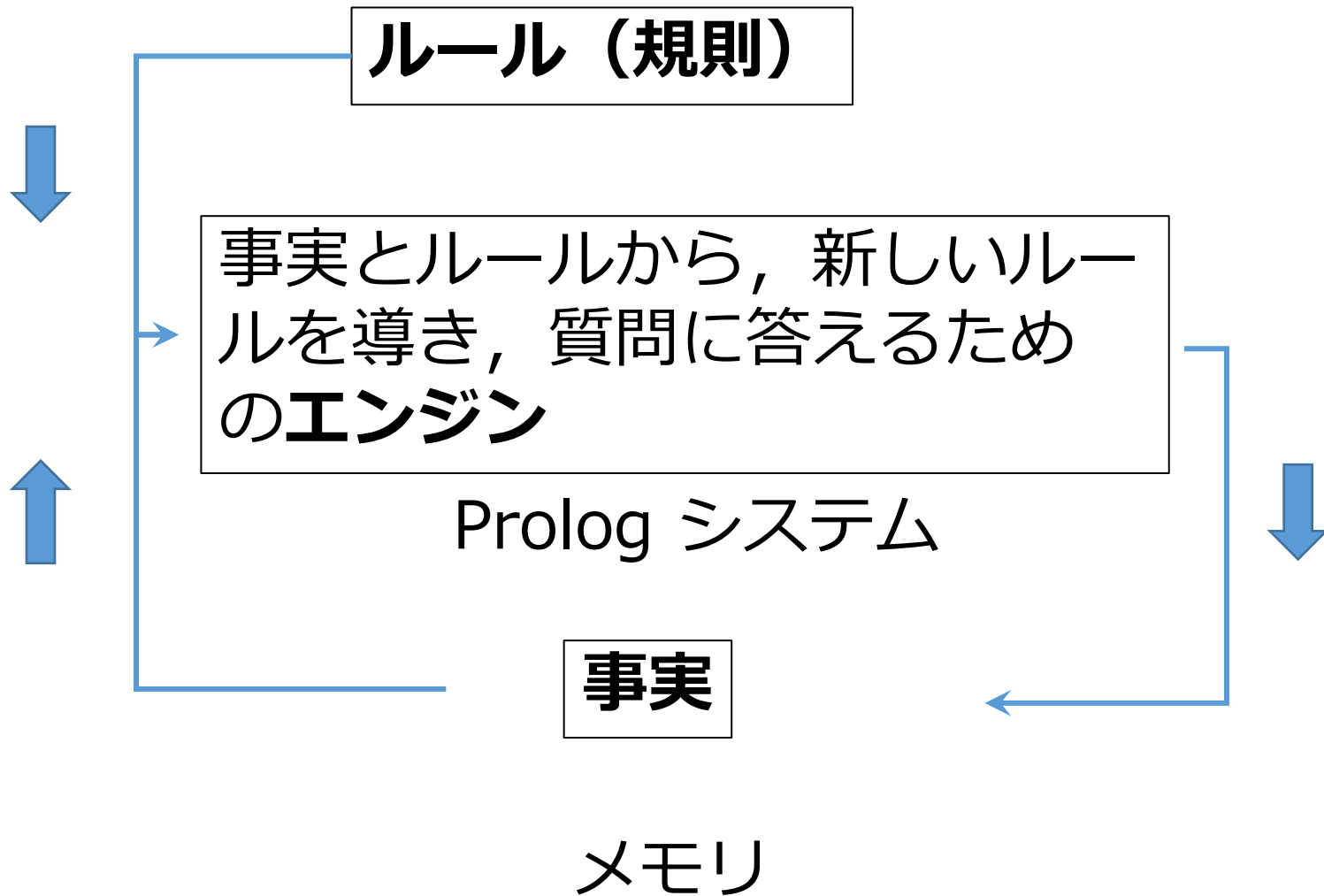
- ルール (規則)

```
child(Y, X) :- parent(X, Y).
```

半角かっこで囲む. 末尾はピリオド
左向き矢印の代わりに「:-」

この授業では, 変数は大文字. 残りはすべて小文字

Prolog の仕組み



推論の例

- 事実

```
male(ali).  
parent(ali, anne).
```

- ルール（規則）

```
child(Y, X) :- parent(X, Y).
```

これら**事実**と、**ルール（規則）**から導かれる
新しい事実

```
child(anne, ali).
```