


13. 述語、Prolog

(人工知能)

URL: <https://www.kkaneko.jp/ai/mi/index.html>

金子邦彦



- 
- ① 理論と実践のバランス。オンラインツール SWISH を活用したインタラクティブな学習
 - ② 述語論理の基本概念（事実、規則、推論）
 - ③ 述語論理を SWISH での Prologプログラミングで実践。
 - ④ 述語論理の基本原則と応用方法の理解、AIへの理解の深化、論理的思考力の向上、問題解決能力の向上

13-1 述語

述語と事実



事実は述語とその引数の組み合わせで表現

(単項の述語)

john は男である

male(**john**).

john という項が male という述語に関係している

(二項の述語)

mike は **john** の親である

parent(**mike**, **john**).

mike, john という二つの項が parent という述語に関係している

事実

ali は **male** (男性) である.
zeyn は **female** (女性) である
anne は **female** (女性) である.



male(**ali**).
female(**zeyn**).
female(**anne**).

述語

事実

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



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

述語

述語と規則



- **述語論理**では、**述語**は**事実**または**規則**を**表現**するのに用いられる。
- 規則は、**事実**に**含意関係**を追加することで**表現**される。
(この資料では、**含意関係**のために「:-」を用いる)

「Xが温かければ、Xを好む」という規則

like(X) :- hot(X)

規則

Xが何らかのYを学んでいれば、Xは学生である

Xが運動をし、良い食事を摂っていれば、Xは健康である



student(**X**) :- studying(**X**, **Y**).

healthy(**X**) :- exercises(**X**), eatsWell(**X**)

述語と含意関係

規則

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



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

述語と含意関係

規則

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



happy(X) :- human(X).

次の**規則**を上と同じ書き方で書きなさい

規則

X が **human** (人間) ならば
X は **animal** (動物) である

animal(X) :- human(X).

13-2 述語での推論

推論の例

事実

```
parent(zeyn, anne).
```

規則

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



```
child(zeyn, anne).
```

推論により得られた新しい**事実**

述語論理の基礎：パターンマッチング



- 与えられた情報が既知の事実や規則と一致するかを確認

例：

「**male(john)**」 事実

「**male(X)**」 男性が誰かを知りたい（問い合わせ）

「**X=john**」 事実と問い合わせのパターンマッチング結果

- **パターンマッチングと推論の有用性**

述語は、パターンマッチングによる推論を行い、問題解決に役立てる。

- **事実と規則**

述語を用いると、事実や規則をコンピュータが理解できる形式で扱うことができるようになる。そして、コンピュータでの推論の実行ができるようになる。

- **抽象化**

述語は具体的な項だけでなく、「like(X) :- hot(X)」のように、変数を用いることができる。一つの規則をさまざまな状況に適用できるようになる。

述語と推論は、自然言語処理、データベース管理、エキスパートシステムなど、様々な分野で応用されている

13-3 Prolog

Prolog の特徴

- **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**問い合わせ**に**答える能力**を持つ

推論の例



• 事実の例

human(hanako).

花子は, 人間

human(taro).

太郎は, 人間

• 規則の例

think(X) :- human(X).

人間は, 必ず考える



think(hanako).

think(taro).

推論により得られる事実

Prolog プログラムの例



```
human(hanako).
```

```
human(taro).
```

```
think(X) :- human(X).
```

《Prologプログラム》

- ・「hanako」と「taro」が「人間」という**事実**
 - ・「人間」であれば「**思考する**」という**規則**
- これにより、**誰が「思考する」かを推論**できる

SWISH を使用

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

Prolog での問い合わせの例①



コンピュータが返す
答え

人間は誰ですか？

```
human(X).
```

hanako taro

```
X = hanako
```

```
X = taro
```

SWISH を使用

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

Prolog での問い合わせの例②



コンピュータが返す
答え

考えるのは誰ですか？

hanako taro

```
think (X).
```

```
X = hanako
```

```
X = taro
```

SWISH を使用

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

述語の true と false

- 述語の属性値は true, false がある

true: そうである

false: そうでない. 知らない

- **john** が人間であるとき

human(**john**). **true**

panda(**john**). **false**



述語

属性値

Prolog での問い合わせの例③



taro は考えますか？

```
think(taro).
```

コンピュータが返す
答え

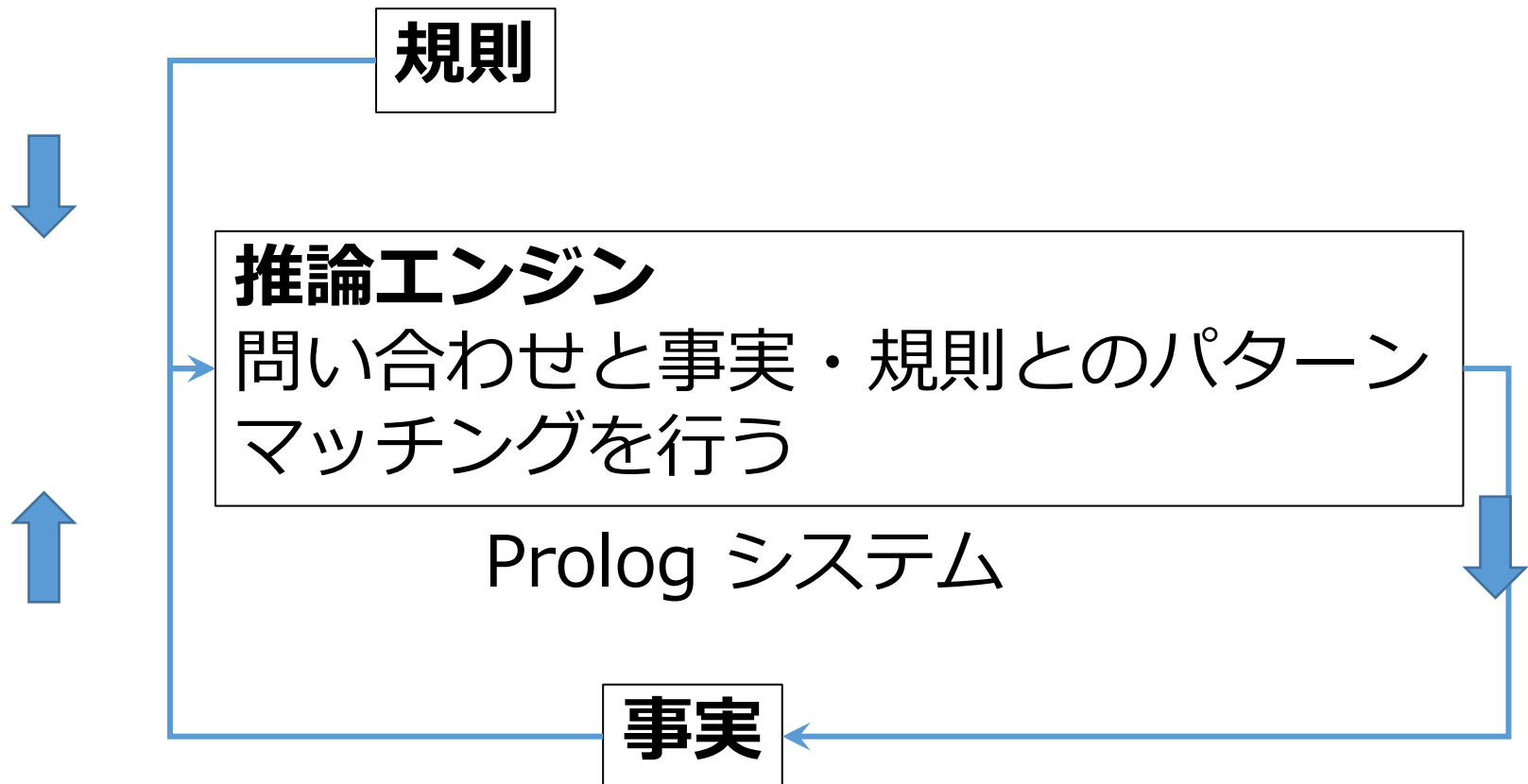
```
true
```

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

Prolog の仕組み



- **推論エンジン**は、Prologシステムの中核をなす部分
- 与えられた**問い合わせ**に対して、**事実**と**規則**を用いて**推論**を行う機能を持つ



Prolog プログラム



- Prolog プログラムは, **述語**で構成される.
- **事実, 規則**の2種
- **問い合わせ**に回答する**機能**がある

演習 1

Prologの基本機能：事実の定義と問い合わせの実践

ページ 25 ~ 33

【トピックス】

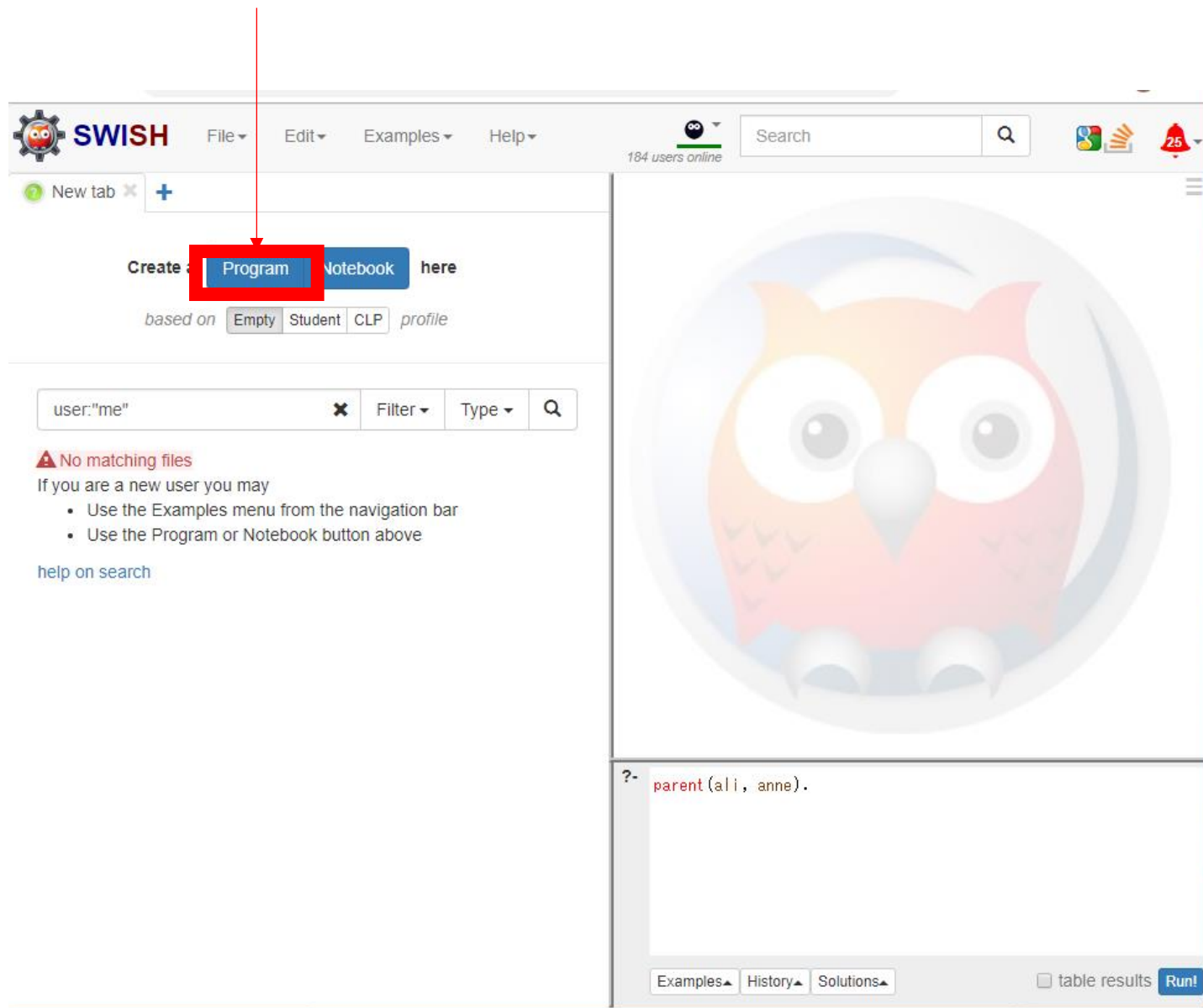
- swish の利用
- Prolog
- 述語
- 事実
- 規則
- 述語での変数

① swish のページを Web ブラウザで開く

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

このサイトは、オンラインで Prolog の体験、
学習ができる

② Prolog の画面に変わる. 「Program」をクリック.



Program +

Your Prolog rules and facts go here ...

事実, 規則
の編集

答えが表示される

?- parent(ali, anne).

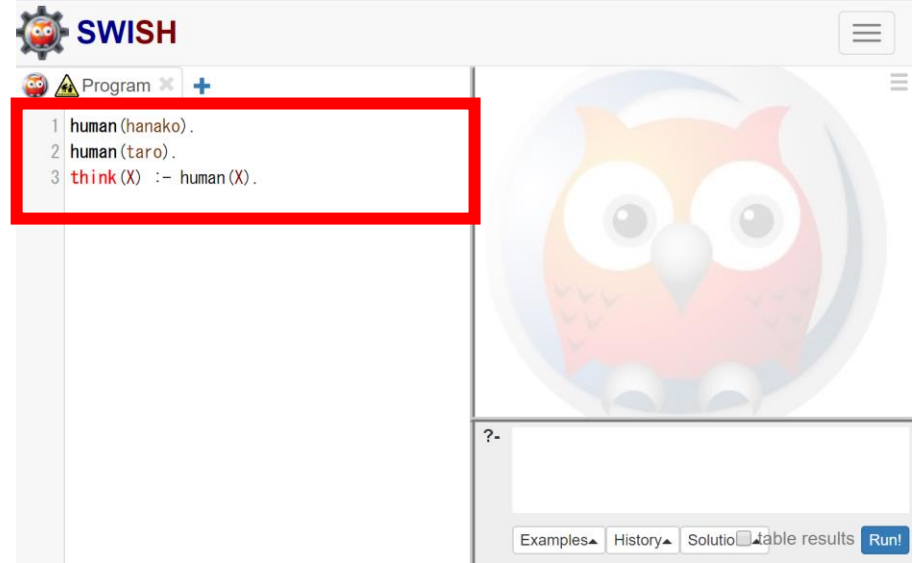
問い合わせをコンピュータに
与える

事実, 規則
をコンピュータに
読み込ませるボタン

Run!

③ Prolog プログラムの準備





```
human(hanako).  
human(taro).  
think(X) :- human(X).
```



Prolog プログラム

編集画面を使用

- ④ 問い合わせ「**human(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)

 `human(X).`   

X = hanako

X = taro





答え

?- `human(X).` 問い合わせ

Examples▲ History▲ Solutions▲ results

Run!

- ⑤ 問い合わせ「**think(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)

 *think(X).*   

X = hanako
X = taro

答え

?- think (X).

問い合わせ

Examples▲ History▲ Solution ☐ table results **Run!**


- ⑥ 問い合わせ「**think(taro).**」を入れ, 「Run」ボタン
think カッコ taro カッコ ピリオド



「true」を確認

問い合わせを，1度に 2個書くことはできない

?- 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).

問い合わせを書くたびに，
前の問い合わせは消す。

Prolog を使ってみた

- Prolog では, **問い合わせ**に対し
事実との照合や
推論を行い

問い合わせでは,

値 **false**, **true**を回答したり,

変数値 **X = hanako**, **X = taro** を回答する能力を持つ

13-4 Prolog での true, false を得る問い合わせ

演習 2

Prolog 中の事実を問い合わせて, True, False を得る

ページ 36 ~ 44

【トピックス】

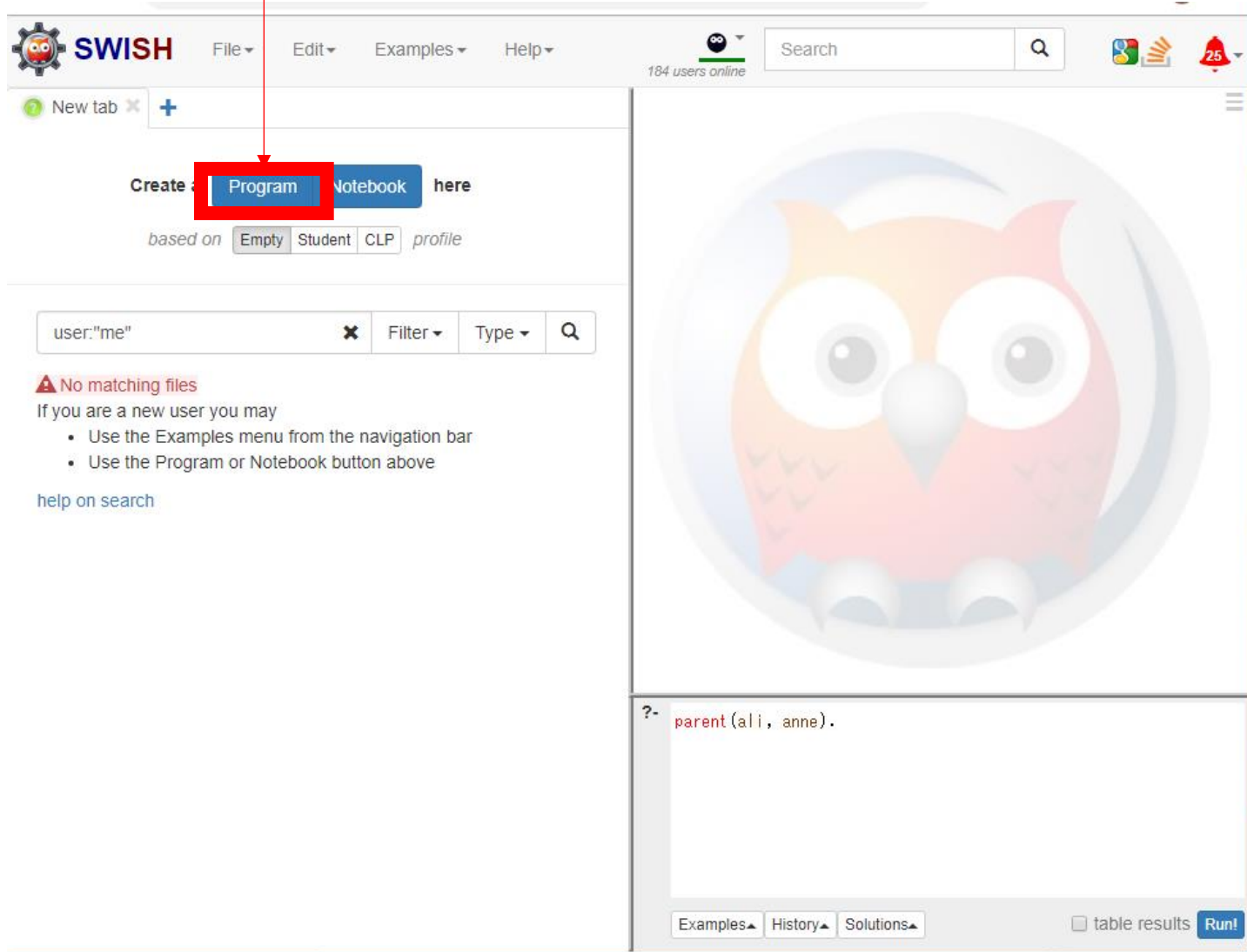
- swish の利用
- Prolog
- 推論
- True
- False

swish のページを Web ブラウザで開く

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

このサイトは, オンラインで Prolog の体験,
学習ができる

Prolog の画面に変わる. 「Program」をクリック.



SWISH

File Edit Examples Help

184 users online

Search

New tab +

Create **Program** Notebook here

based on Empty Student CLP profile

user:"me" x Filter Type Q

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).

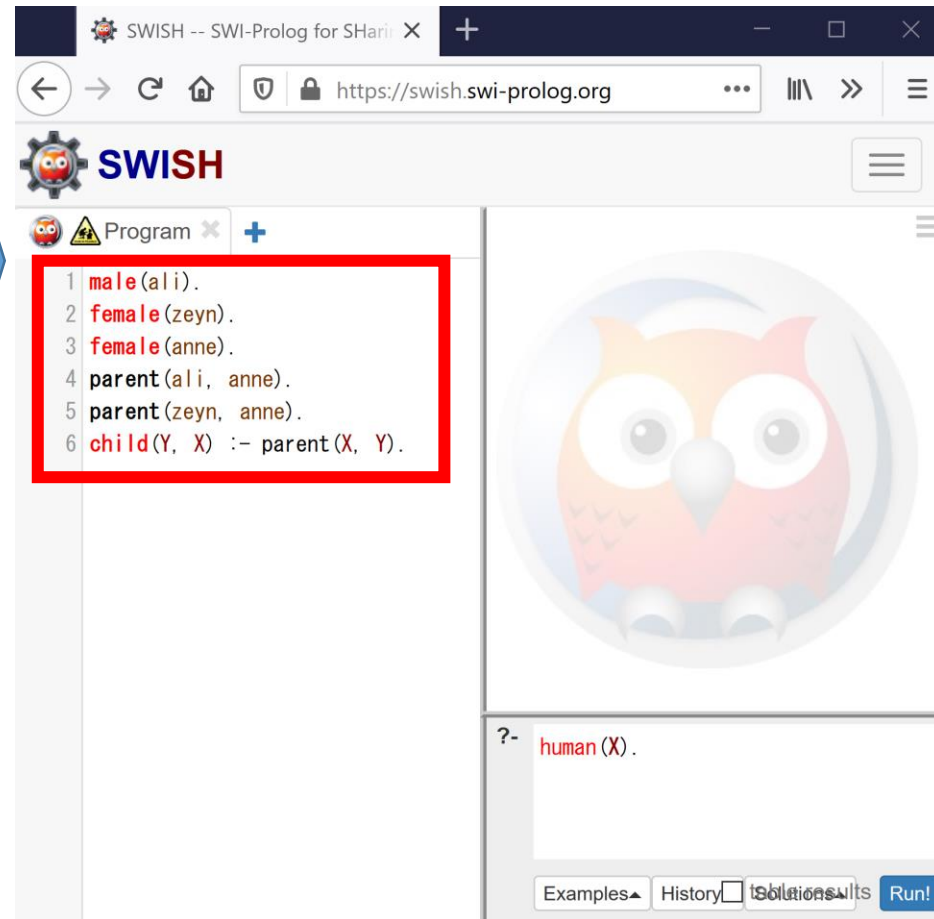
Examples History Solutions

table results Run!

Prolog プログラムの準備

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

Prolog プログラム



編集画面を使用

問い合わせ「male(ali).」を入れ, 「Run」ボタン
male カッコ ali カッコ ピリオド Enterキー

The image shows a screenshot of the SWISH programming environment. The main window displays a query `male(ali).` in the input area, which is highlighted with a red box and labeled 「問い合わせ」 (Query). The result area shows `true`, highlighted with a red box and labeled 「答え」 (Answer). Below the main window, there is a smaller inset window showing the same query and result, with the `true` result highlighted by a red box. At the bottom right of the inset window, the `Run!` button is highlighted with a red box. The text 「true」を確認 (Check 'true') is written below the main window.

「true」を確認

エラーメッセージが出ることがある



```
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

問い合わせ「**male(anne).**」を入れ, 「**Run**」ボタン
male かつこ anne かつこ ピリオド Enterキー



male(anne).

false

?-

ma l e (anne) .

「false」を確認

問い合わせ 「 `parent(ali, anne).` 」 を入れ, 「Run」
ボタン



`parent(ali, anne).`

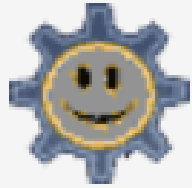
true

?-

`parent(ali, anne).`

「true」を確認

問い合わせ 「child(anne, ali).」 を入れ, 「Run」 ボタン



child(anne, ali).

true

?-

child(anne, ali).

「true」を確認

確認クイズ（余裕のある人向け）

問い合わせの答えは何になるか，空欄の true または false を考えてみよう



問い合わせ	答え
female(anne).	
parent(zeyn, ali)	
child(zeyn, ali).	
child(ali, anne).	

そして Prolog で実際に実行して確かめてみよう

Prolog における true, false の意味

- **true** 「正しい」
- **False** 「正しくない」ではなく「**不明**」を意味する

例

- male(john). に対して **true**
john は男である
 - male(anne). に対して **false**
anne は男である事実が知られていない、証明できない（不明）。
- ※ 「anneが男ではない」という意味ではないので注意

13-5 Prolog で値を得る問い合わせ

演習 3

Prolog 中の事実を問い合わせ、True, False を得る

ページ 48 ~ 57

【トピックス】

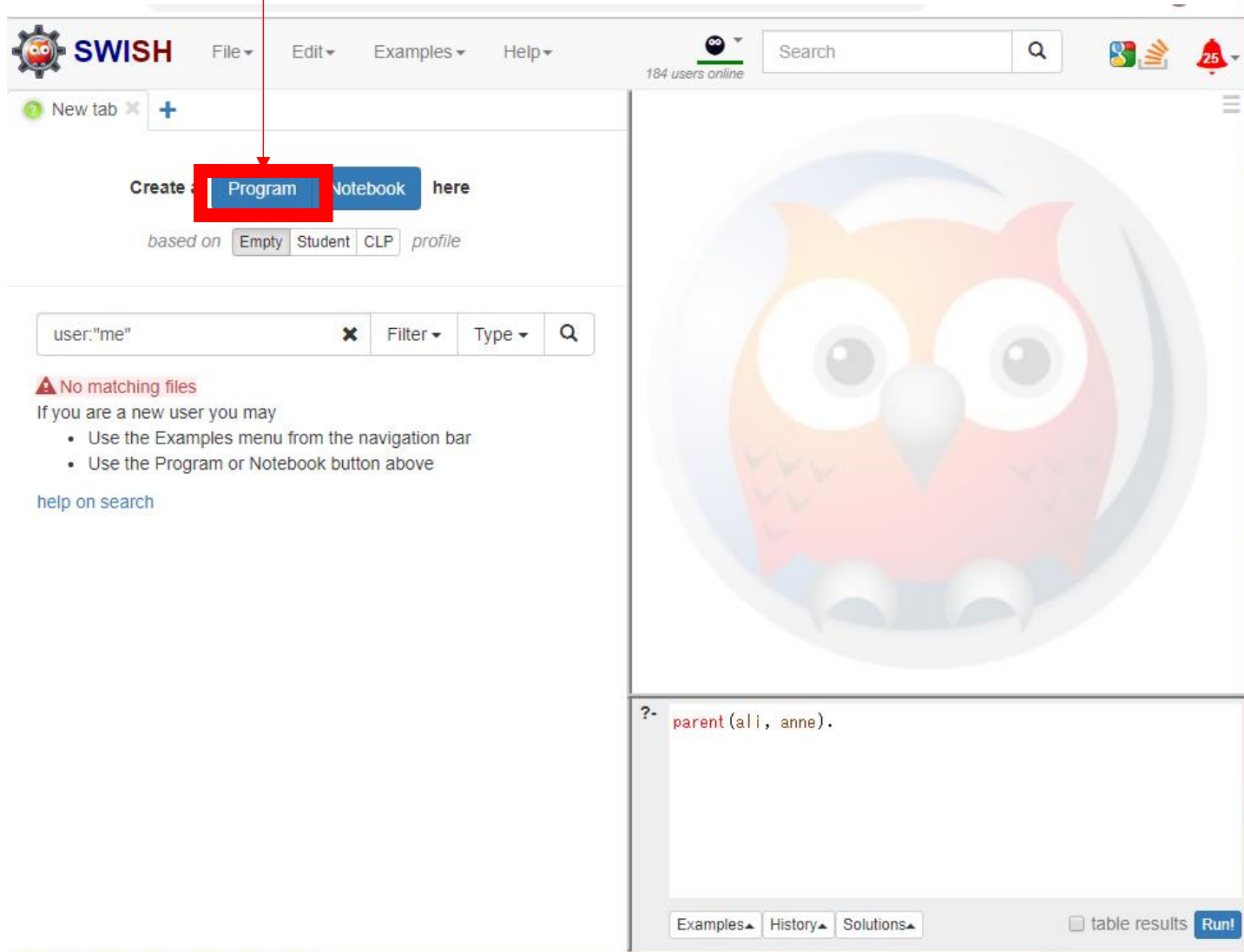
- swish の利用
- Prolog
- 推論
- Prolog の変数

swish のページを Web ブラウザで開く

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

このサイトは, オンラインで Prolog の体験,
学習ができる

Prolog の画面に変わる. 「Program」をクリック.



SWISH

File Edit Examples Help

184 users online

Search

New tab +

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).

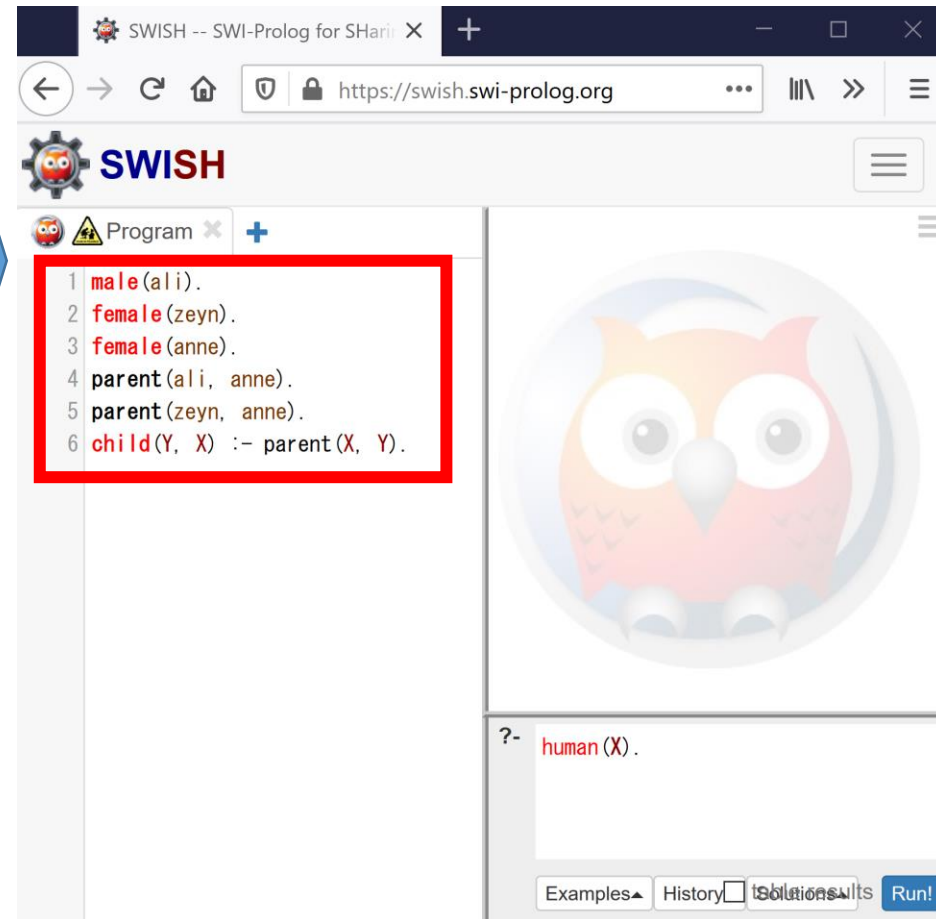
Examples History Solutions

table results Run!

「プログラム」の準備

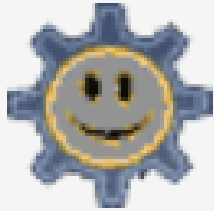
```
male(ali).  
female(zeyn).  
female(anne).  
parent(ali, anne).  
parent(zeyn, anne).  
child(Y, X) :- parent(X, Y).
```

Prolog プログラム



編集画面を使用

問い合わせ「male(X).」を入れ, 「Run」ボタン



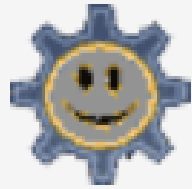
male(X).

X = ali

?-

male(**X**).

問い合わせ「**female(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)



female(X).

X = zeyn

X = anne

?-

female(X).

問い合わせ「**parent**(X, Y).」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)



parent(X, Y).

X = ali,

Y = anne

X = zeyn,

Y = anne

?-

parent (X, Y) .

問い合わせ「**child(X, Y).**」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

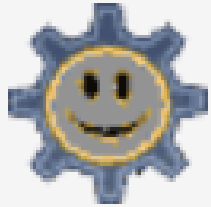
X = anne,

Y = zeyn

?-

child(X, Y).

問い合わせ 「parent(**ali**, Y).」 を入れ, 「Run」 ボタン



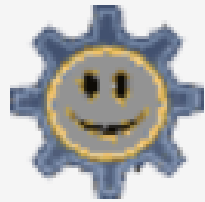
parent(ali, Y).

Y = anne

?-

parent (ali, Y).

問い合わせ 「`child(ali, Y).`」 を入れ, 「Run」 ボタン



`child(ali, Y).`

false

?-

`child(ali, Y).`

false は「不明」.

確認クイズ (余裕のある人向け)

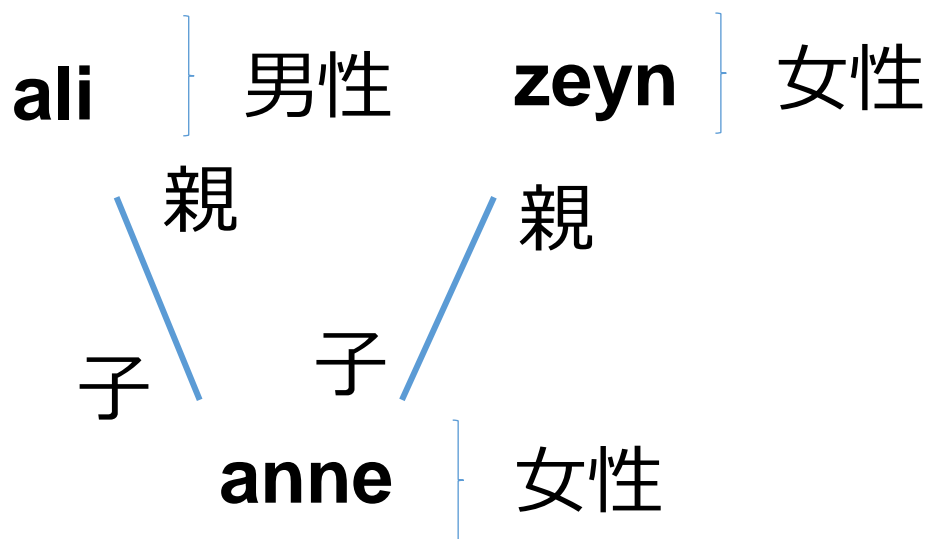
それぞれの問い合わせの答えは何か考えてみよう.
そして, 実際に Prolog で動かして確かめてみよう

1. `parent(zeyn, Y).`

- ① false ② true ③ `Y = anne` ④ `Y = ali`

2. `child(zeyn, Y).`

- ① false ② true ③ `Y = anne` ④ `Y = ali`



全体まとめ

- **述語論理**は、**述語とその引数の組み合わせ**で表現される**事実や規則の体系**である。**規則は、事実と含意関係を組み合わせ**て表現される。

「male(john)」は「**Johnは男性である**」という**事実**

「like(X) :- hot(X)」は「**xが温かければ、xを好む**」という**規則**

- **Prolog**というプログラミング言語では、**述語を使用してコンピュータプログラムを記述**する。Prologプログラムは、**問い合わせに対して推論に基づいた回答を提供**する。

Prologプログラムの例

male(ali).

female(zeyn).

female(anne).

parent(ali, anne).

parent(zeyn, anne).

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

授業の学ぶ意義と満足感



- ① 論理的思考力の向上：複雑な関係を簡潔に表現する能力。例：「xが人間ならば、xは考える」を `think(X) :- human(X).` と表現
- ② AIの知識表現の基礎を習得：問題解決能力の獲得。事実と規則から新しい結論を導く推論の理解。例： `parent(ali, anne).` から `child(anne, ali).` を推論
- ③ 推論の仕組みを理解する達成感：事実と規則に基づく問い合わせ応答の仕組み。
例： `human(X).` への回答
- ④ 実践的スキルの習得：SWISHでのAIプログラム作成と実行体験、簡単なAIシステムを自ら実行、最新のAI技術への理解を深める充実感

さらに練習したい人向けの
追加演習

追加演習 1. 考えてみよう

次の**事実**, **規則**の Prolog プログラムは何か？

事実

taro は **human** (人間) である.

規則

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

演習問題 1 の回答例

次の**事実**， **規則**の Prolog プログラムは何か？

事実

taro は **human** (人間) である.

規則

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

解答例

human(taro).
happy(X) :- human(X).

追加演習 2 : 動物の分類

1. 事実の定義:

taimou(cat).

猫に体毛あり

nikusyoku(cat).

猫は肉食

2. 規則の定義:

honyu(X) :- taimou(X).

体毛があれば哺乳類

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

哺乳類で肉食なら肉食動物

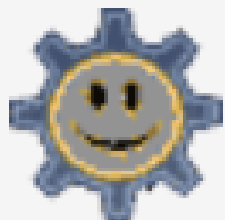
3. 問い合わせ例:

honyu(cat).

nikusyokudoubutu(X).


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

honyu(X) の結果



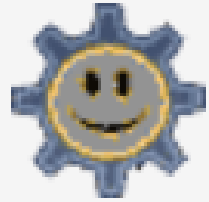
honyu(cat).

true

?-

honyu(cat).

nikusyokudoubutu(X) の結果は何か



nikusyokudoubutu(X).

X = cat

?-

nikusyokudoubutu(**X**).