

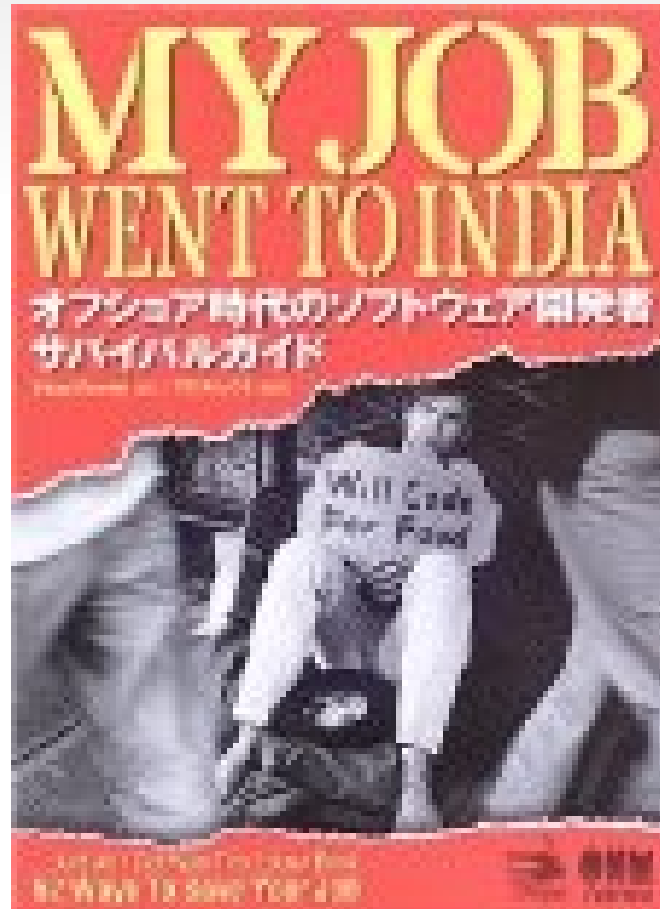
春なので
AI を込めて
Prolog 宣伝プレゼンを作
ってきました

おがさわらなるひこ

naruoga at gmail, twitter, wassr, hatena

@ Genesis Lightning Talks #13, Feb25th 2009

"My Job Went To India" 目く



Chad Flower / でびあんぐる
オーム社
ISBN: 4274066592

「プログラマとして
生き残りたいければ

いろいろな
パラダイム

の言語を学べ」

C++ と

Java と

Perl と

Ruby と

Python と

PHP と.....

**それ全部
同じパラダイム
(手続き型言語)
ですから！**

違う
パラダイム
?

関数型言語とか
ちよつと
流行ってるよね

Lisp/Scheme

とか

Haskell

とか

Ocaml

とか

でも
忘れないで

もう一つの
プログラミング言語
パラダイム

論理型言語！

1980年代 日本を席卷した 「第五世代 コンピュータ」

みんな忘れたがってると思うが。

その中心に
あったのが
「人工知能 (AI)
言語
Prolog」

今はほとんど
忘れ去られて
いるけれども

俺は
Prologって
奴が
大好きなんだ！

とゆことで
その魅力を
愛 (AI) を以って
語ります。

Prolog とは

おフランス
生まれの
小粋な言語

PLOgramming In LOGIc の略

ホントはフランス語なんだけどそんなん知らんわ。

Prolog に
おける
プログラム
とは

**「論理」で
表現された
「知識」**

**「知識」に
問い合わせを
して
答えをもらうのが
Prolog の「実行」**

例

男 (なるひこ).

男 (まさと).

女 (あつこ).

親 (まさと, なるひこ).

親 (あつこ, なるひこ).

人間 (X) :- 男 (X).

人間 (X) :- 女 (X).

父 (X, Y) :- 親 (X, Y), 男 (X).

母 (X, Y) :- 親 (X, Y), 女 (X).

誰かさんの家族構成がモロバレなのは気にしないでください。

なんじゃ
こりゃ？

順を追って
説明
しましょう。

基本概念 1

述語 (predicate)

男（なるひこ）



「なるひこ」は
「男」である

と読む

一般的には

$$p(a_1, a_2, \dots, a_n)$$

は

「 a_1, a_2, \dots, a_n は

p (という関係)」

と読める

**Prolog では
すべての論理を
述語で
表現するので
「述語論理言語」**

詳しくは
コンピュータ
言語
意味論とか
勉強してね

Robert Harper: [Practical Foundations for Programming Languages](#). とか

基本概念 2

事实 (fact)

述語にピリオドを
打てば
それが事実。

男(なるひこ).

「なるひこは男、
というのは
(Prolog 内知識では)
事実」
という意味

基本概念 3

規則 (rule)

$p_0 :- p_1, p_2, \dots, p_n.$

は

「 p_1, p_2, \dots, p_n が
すべて真であれば
 p_0 も真」という意味

人間 (X) :- 男 (X).

.....

母 (X,Y) :- 親 (X,Y), 女 (X).

「Xが男なら、
Xは人間」

「XがYの親で、Xが
女なら、XはYの母」

ほしいじゃま
この
「プログラム」
を
実行して
みましょう

失敗が
怖いので
弱気に
結果だけ
コピペします (^_^;)

```
naruhiko@naruhiko-ubuntu-s101:~/Prolog$ cat human.pl
```

```
男 (なるひこ).
```

```
男 (まさと).
```

```
女 (あつこ).
```

```
親 (まさと, なるひこ).
```

```
親 (あつこ, なるひこ).
```

```
人間 (X) :- 男 (X).
```

```
人間 (X) :- 女 (X).
```

```
父 (X,Y) :- 親 (X,Y), 男 (X).
```

```
母 (X,Y) :- 親 (X,Y), 女 (X).
```

```
naruhiko@naruhiko-ubuntu-s101:~/Prolog$ prolog
```

```
Welcome to SWI-Prolog (Multi-threaded, ...
```

```
...
```

```
For help, use ?- help(Topic). or ?- apropos(Word).
```

```
?- [human]. ← "?-" は質問受付。[] はプログラムのロード。
```

```
% human compiled 0.00 sec, 1,964 bytes
```

```
true.
```


?- 男 (なるひこ). ← 単純な事実の確認

true. ← 真だったら true と返ってくる (偽だったら false)

?- 人間 (Who). ← 人間なのはだれ? という質問

Who = 'なるひこ' ; ← ";" は、別の答えはないの、の意味

Who = 'まさと' ;

Who = 'あつこ'.

?- 父 (まさと, なるひこ).

true.

?- 父 (あつこ, なるひこ).

false.

?- 父 (Who, なるひこ).

Who = 'まさと' ;

false.

?- 父 (まさと, Who). ← 逆の質問もできてしまう

Who = 'なるひこ'.

事実と
規則からなる
「知識」から
答えを探索して
くれるのが
ポイント

Prolog は
知識表現が
できるって
ことは
分かった

普通の
プログラミングって
できないの？

否！

今までの
Prolog の解釈は
「宣言的」

それに対して
「手続的」
解釈も
可能

$p_0 := p_1, p_2, \dots, p_n$

「 p_1, p_2, \dots, p_n が
すべて真であれば

p_0 も真」:

宣言的解釈

$p_0 := p_1, p_2, \dots, p_n.$

「 p_0 という処理は

p_1, p_2, \dots, p_n を

実行すること」:

手続的解釈

宣言的に
考えた方が
Prolog らしい
ですが
手続的な
処理も書けます。

ところで

Prolog では
変数への
再代入は
できません。

```
?- X is 1 + 2.
```

```
X = 3.
```

```
?- X is X + 1.
```

```
ERROR: is/2: Arguments are not sufficiently  
instantiated.
```

**「 $X + 1 = X$ 」は
論理的に
成り立たない！**

Q:

じゃ、ループって
どうやって
書くの？

A:
再帰か
バックトラック
です。

再帰は
他の言語でも
使えるから
とりあえず
飛ばします。

バックトラック
とは？

失敗したときに
次の答えを
探しに行く
しくみ

?- 人間 (Who) . ← 人間なのはだれ? という質問

Who = 'なるひこ' ; ← ";" は、別の答えはないの、の意味

Who = 'まさと' ;

Who = 'あつこ' .

これもバックトラック
";" の入力で
答えを失敗させて
次の答えを探させる

「失敗駆動ループ」

探索の最後に
無理矢理失敗させて
バックトラックで
全ての解を得る

```
?- 人間 (X), write(X), nl, fail.
```

```
なるひこ
```

```
まさと
```

```
あつこ
```

```
false.
```

さっきの例を
自動でやると
こんな感じ

ここはすごく
Prolog らしくて
面白いんだけど
時間の都合で
割愛 (;_ ;)

その他の 特徴

**「超強力な
リスト処理」**

Lisp に
負けるな！

「自己増殖機能」

自分自身で
知識や
事実を
(=プログラム)を
拡張して
いける

文字列で
コード作って
eval とか
クソだぜクソ！
(下品でごめん)

そんなわけで
エレガントかつ
キャッチーな機能
たっぷりな
超高水準言語
Prolog。

あなたの
言語コレクションに
お加え
ください。

**interest(prolog).
love(Someone,
 Something) :-
 engineer(Someone),
 interest(Something).**

?- love(you, prolog).

ご清聴
ありがとうございます
ございました。

参考 URL

- SWI-Prolog
 - GPL の Prolog 処理系
 - 様々なプラットフォームに対応
 - Debian 系 Linux なら `apt-get install swi-prolog` でインストールできるよ
 - Linux 版は X インタフェースやシェル呼出しも可
 - ドキュメントも充実
- お気楽 Prolog プログラミング入門
 - 日本語で書かれたチュートリアルとしてはかなりまとまっていると思います。オススメ。