知能プログラミング演習 I

世木　博久　　加藤　昇平
（情報工学科・知能系）
{seki, kato.shohei} @nitech.ac.jp

本日の内容

- 宣言的プログラミング言語
  手続き型言語との違い
  記号処理，人工知能の問題解決向き言語
  論理型プログラミング言語Prolog
- 演習で使う処理系 ⇒ SWI-Prolog
- 演習の進め方，計算機環境(@計算機室)

本日の内容：知能プログラミング演習 I・イントロダクション・演習の進め方

- 宣言的プログラミング言語とは?
  手続き型言語(C, C++など)との違い
  関数型言語(Lispなど)，論理型言語(Prologなど)
  記号処理・人工知能の問題解決向き言語

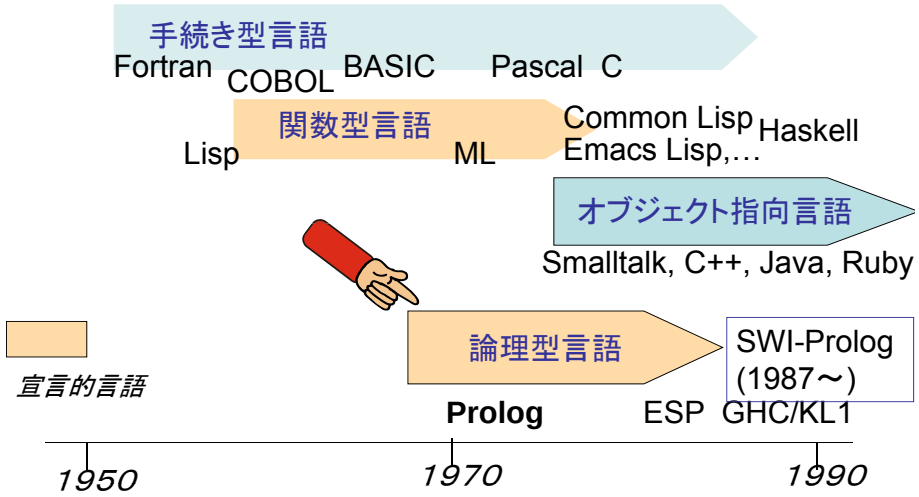
  LispなどのAI言語
  ☞ リスト処理などの便利な機能を持つ汎用言語

- 演習で使う処理系 ⇒ SWI-Prolog
  演習の進め方
  計算機環境(@計算機室)

演習の目的

宣言的プログラミング言語Prologの基本的事項を習得すること．また，それを用いてアルゴリズムとデータ構造を自分で設計し，問題解決できること．

プログラミング言語の流れ

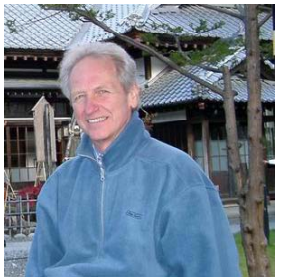


Prolog: **Prog**ramming in **Log**ic

- 論理に基づくプログラミング言語

  R. Kowalski, “*Predicate Logic as Programming Language*”, IFIP, 1974.

関連する授業：
「知識表現と推論」
「数理論理学」
データベース（SQL, 関係論理）など



[注] 同じ頃，*Alain Colmerauer* (マルセイユ，仏)も提案．

宣言的知識表現　vs　手続き的知識表現

論理表現：宣言的(what)

$$human(bob)$$
$$human(alan)$$
$$\forall X\ \{human(X) \rightarrow mortal(X)\}$$
$$\forall X\ \{dog(X) \rightarrow mortal(X)\}$$
$$\ldots$$

手続き的表現(how)

```
procedure human(x)
if (x=bob) or (x=alan) then
        return true
else    return false

procedure mortal(x)
if human(x)  then return true
else if dog(x) then return true
else return false
```

[注]mortal（形）いつかは死ぬ

宣言的知識表現 = Prologプログラム

論理による表現：宣言的

$$human(bob)$$
$$human(alan)$$
$$\forall X\ \{human(X) \rightarrow mortal(X)\}$$
$$\forall X\ \{dog(X) \rightarrow mortal(X)\}$$
$$\ldots$$

Prologプログラム

```
human(bob).
human(alan).
mortal(X) :-  human(X).
mortal(X) :-  dog(X).
. .
```

[注]:-　は　←　(if)の意味

“論理表現＝プログラム”
→　可読性が高い・理解が容易・バグが入りにくい
　　修正やメンテナンスが容易→生産性が高い

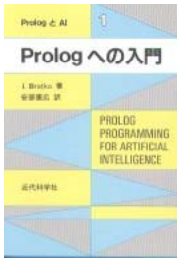
Prologプログラム：実行例

```
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 5.6.64)
Copyright (c) 1990-2008 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

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

1 ?- [intro].    %  プログラムの読み込み
intro.pl compiled, 0.00 sec, 1,100 bytes.
Yes              %  コンパイル成功
2 ?- human(bob).   % bobはhuman?
Yes
3 ?- human(X).   % humanはだれ？
X = bob ;          % ; (セミコロンを入力)  他の解は？
X = alan ;
No               % 他の解はない
```

```
% プログラム   intro.pl

human(bob).
human(alan).
mortal(X) :-  human(X).
mortal(X) :-  dog(X).
```

知能プログラミング演習Ⅰ：進め方とスケジュール

【演習の進め方】
毎回出題される演習問題についてレポート
(プログラム・実行結果・説明・考察)を提出



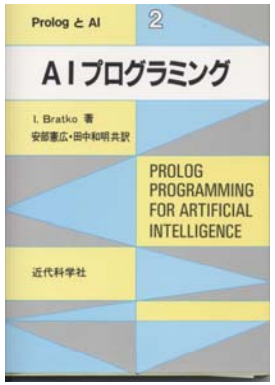
- [第1ステージ：基礎]　Prolog習得
  教科書：「Prologへの入門」
  （4～6月）
- [第2ステージ：応用]　AIアルゴリズムの実装
- [最終ステージ：最終]　自由課題

(以前の)参考書：AIプログラミング

現在出版社品切れ．
図書館にある．

演習のテキストは，次の本の前半の日本語訳である．

I. Bratko, *“Prolog Programming for Artificial Intelligence”*



パターンマッチング，構造体データ，
自動後戻り（バックトラッキング）

探索・問題解決，制約充足
推論システムシェル（不確定な推論）
機械学習，ゲーム，…

他のシステムとの連携：Packages

Standard packages

- With *standard* packages, we refer to add-ons that are part of the normal SWI-Prolog binary distributions and compiled by default if you compile and install from the sources. These packages are:

- PlDoc - Documentation framework
- PlUnit - Unit testing framework
- clib - TCP/IP sockets, Processes, CGI, MIME, ...
- cpp - C++ interface
- JPL - Java interface
- SGML - SGML, HTML, XML handling
- RDF - RDF-handling with online demo
- SemWeb - Store and query RDF/RDFS
- nlp - Natural Language Processing primitives
- http - (Embedded) HTTP protocol support
- ssl - Secure Socket Layer interface (OpenSSL)
- ODBC - ODBC access to databases
- table - Access records in a file

Prolog ⇔ Java 双方向インターフェースをサポート

各種DB に対応しSQLをサポート