

スーパーサイエンスハイスクール
人工知能プログラミング
 AI programming
<http://www.wada.elcom.nitech.ac.jp/~inuzuka/SSHAprogramming/>

講師 名古屋工業大学
 犬塚信博
 中野智文
 助手 TA 近藤真一

今日の講義と演習のねらい

- 人工知能の基礎技術である論理と推論の働きについて理解する。
- 論理プログラムの基本を体験する。

キーワード

- 人工知能
- 論理、推論、知識
- 論理プログラム、Prolog

副作用

- 論理的思考を訓練する。



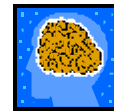
今日の予定



- 午前**
- 人工知能と論理の講義・演習
 - 論理、推論、正しい推論
 - Prologの入門
 - Prologをコンピュータに入れる、推論させる
- 午後**
- 人工知能の研究
 - データからの知識発見、帰納学習
 - Prologの演習
 - Prologを使って知識を書く、問題を解く

人工知能 (Artificial Intelligence)

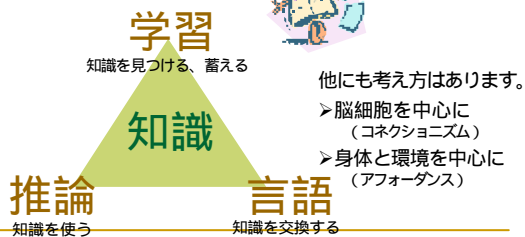
- 機械的に人の持つ知能を実現する技術 (またはそうした技術の研究)



- 知能
 - ことばを話す、理解する、議論する。
 - 知識を使って考える (推論する、計画を立てる)
 - 学習する、科学する、技術のみがく。
 - 見てわかる。聞いてわかる。回りの状況を理解する。

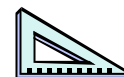
人工知能の考え方

- 知識を中心に考える。



命題 (propositions)

- 真偽のはっきりとわかる断定された文。
 - 平行四辺形の向かい合う二辺の長さは等しい。
 - 今朝、朝食をとった。
 - 素数は無限個存在する。



- 命題でない文

- おはよう。
- 今日は何日ですか？
- 毎日朝食をとりなさい。
- 神は存在する。
(もし、神が何であるかはっきり定義してないならば)

推論 (inference, reasoning)

分かっていること (命題) から、新しい知識 (命題) を導き出すこと。



- 演繹推論 確実に結論できる仕方新しいことを導く。
- 帰納推論 個別の事実を総合して、一般法則を導く。
- 類推 分かっていることとの類似に注目して、新しい知識を導く。

いつも安心して使えるのは、演繹推論のみ。

推論 (演繹推論) の例 1



- (1) 小泉首相は男性あるいは女性のどちらかである。
 (2) 小泉首相は女性ではない。

仮定
(前提)

- (3) 小泉首相は男性である。

結論

推論 (演繹推論) の例 1

- (1) P または Q
 (2) Q でない

- (3) P

P = 小泉首相は男性である
 Q = 小泉首相は女性である

推論 (演繹推論) の例 2

- (4) エンジンがかかるならば、燃料は入っている。
 (5) エンジンがかからない。

- (6) 燃料が入っていない。



推論 (演繹推論) の例 2

- (1) P ならば Q
 (2) Q でない



- (3) P でない

P = 燃料がはいっている
 Q = エンジンがかかる

命題論理 (propositional logic)

- 命題を組合せて命題を作ることができる。
- 命題の組合せの正しさで、推論の正しさを考える論理を命題論理という。

命題の組合せ方

Pでない	PまたはQ	PかつQ	PならばQ
$\neg P$	$P \vee Q$	$P \wedge Q$	$P \supset Q$
否定	論理和(連言)	論理積(宣言)	含意

組合わせた命題の真偽

- 命題を組合わせてできた命題の真偽は、元の命題の真偽で決まる。
- $P \wedge Q$ (PかつQ) : PとQが両方真であるとき、真。

P	Q	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$\neg P$
真	真	真	真	真	偽
真	偽	偽	真	偽	真
偽	真	偽	真	真	偽
偽	偽	偽	偽	真	真

この表を真理値表という



正しい推論

- 仮定が正しいならば、導かれた結論がいつも正しい形式の推論を**正しい推論**という。

A君は野球部員である。
A君は理系クラスである。



P	Q	$P \rightarrow Q$
真	真	真
真	偽	偽
偽	真	真
偽	偽	真

A君は野球部員で、理系クラスである。

$\frac{P}{Q}$
 $\frac{P \rightarrow Q}{P \rightarrow Q}$

- 推論の正しさは、形式でのみ決まる。
- 話の内容には関係ない。

正しい推論の例

小泉首相は男性あるいは女性のどちらかである。

小泉首相は女性ではない。

小泉首相は男性である。

結論	仮定 1	仮定 2
$P \rightarrow Q$	$P \rightarrow Q$	$\neg Q$
真	真	偽
真	偽	真
偽	真	偽
偽	偽	真

$\frac{P \rightarrow Q}{\neg Q}$
P

確かに、仮定1と仮定2が正しいときには結論が常に正しい。

正しい推論の例

エンジンがかかるならば、燃料が入っている。

エンジンがかからない。

燃料が入っていない。

結論	仮定 1	仮定 2
$P \rightarrow Q$	$P \rightarrow Q$	$\neg Q$
真	真	偽
真	偽	真
偽	真	偽
偽	偽	真

$\frac{P \rightarrow Q}{\neg P}$
 $\neg Q$

仮定1と仮定2が真でも、結論が真でない場合がある。

注意 推論の正しさは、実際に書く命題が正しいかどうかとは無関係。

練習 次の推論は正しいかどうか、確かめなさい。

- 雨が降る日はA先生は青いネクタイをしてくる。
A先生は青いネクタイをしていない。
だから、今日は雨は降らないだろう。



- レポートを提出するか、試験が70点以上であれば合格である。
レポートを提出した。
だから、合格である。



公理と理論



- 推論で分かることを増やす。
- 最初にあった分かっていること
= 文句なしに認められること
= 公理あるいは理論
axiom theory

例 ユークリッドの幾何の公理
ニュートンの力学の理論

論理と理論

論理と理論は見た目が似ていますが、全く違います。

- **論理** = 分かっていることから、新しいことを導くための**考え方**。
logic
- **理論** = ものごと (たとえば、経済、物理)を**説明**するため、
theory はじめに皆で認めることにした事実 (仮説)

十分強力な論理があれば、理論からその帰結を知ることができる。

命題論理の限界

- 命題論理、命題の組合せでのみ結論を導き出しますが、それではつぎの推論は扱えません。

すべての人間は死ぬ。	P
ソクラテスは人間である。	Q
ソクラテスは死ぬ。	R

どの命題も基本命題の組合せでできているわけではない。

命題論理の限界

だれでも人間は死ぬ。
ソクラテスは人間である。
ソクラテスは死ぬ。



これを扱うために、命題を主語と述語に分解して扱う
= 述語論理 (predicate logic)

ソクラテスは人間である
人間(ソクラテス)

述語論理

- 述語論理では、変数を使います。
 x は人間である = 人間 (x)
 - 人間は死ぬ
= あるものは人間であるならば、そのものは死ぬ
= 「 x が人間ならば、 x は死ぬ」がどんな x にも成立つ。
- これを、 x 、(人間 (x) 死ぬ (x)) と書きます。
 x は、 x に何を代入しても真であることを表します。
= 全称記号といいます。

述語論理

- 述語論理では、変数を使った論理のために2種類の記号を使います。
- 全称記号 (allのAの逆)
 x . $P(x)$: どんな x に対しても $P(x)$ は真である。
- 存在記号 (existのEの逆)
 x . $P(x)$: $P(x)$ が真であるような x が存在する。

例 どんな人間も死ぬ x . (Human(x) Mortal(x))
死なない人間がいる x . (Human(x) \neg Mortal(x))

練習 述語論理で表す

次の命題の意味が分かりますか？
ただし、Human(x): x は人間、
Like(x , y): x は y を好む、とします。

- x . {Human(x) y . [Human(y) Like(x , y)]}
- \neg { y . [Human(y) (x . Human(x) Like(x , y))]}

述語論理での推論

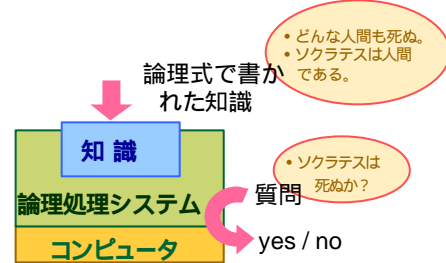
- 述語論理の推論の正しさは、真理値表では確かめられませんが、次の形式は正しい推論です。

すべての人間は死ぬ。 ソクラテスは人間である。	すべての動物は死ぬ。 すべての人間は動物である。
ソクラテスは死ぬ	すべての人間は死ぬ。

$$\frac{x. p(x) \quad q(x) \quad p(S)}{q(S)}$$

$$\frac{x. p(x) \quad q(x) \quad x. r(x) \quad p(x)}{x. r(x) \quad q(x)}$$

論理プログラム (logic programs)



- 論理処理システムとして、Prologを用います。

論理プログラムの形式

Prolog : Programming in logic

述語論理の命題をコンピュータで扱うことができるようにしたシステム。

今日の演習ではSWI-Prologを使います。

SWI-Prolog :

オランダ アムステルダム大学が開発した無料のシステム。

Prologの記法 (1)

$x. (\text{Human}(x) \quad \text{Mortal}(x))$

をprologで書くと

$\text{mortal}(X) :- \text{human}(X).$

- 「 $\text{}$ 」は、逆の「 $\text{}$ 」の意味で「 $:-$ 」と書く。
- 全称記号は書かない。(変数にはすべて全称記号がついていると思う)
- 述語は小文字で始める。
- 変数は大文字で書く。
- 最後はピリオド。

Prologの記法 (2)

一般にprologは次の形式の命題を許す。これを論理プログラムでは「節(せつ)」と呼びます。

$p(t_1, \dots, t_n).$

$p(t_1, \dots, t_n) :- q_1(\dots), q_2(\dots), \dots, q_m(\dots).$

- t_i は定数 (定項) (ex. socrates) か、変数 (ex. X)。
- 1つ目の形式は**事実**、2つ目は**規則** (ルール) といいます。
- 規則の右側 (条件) は「かつ」で繋がっている。
- $\neg A$ は $\text{not}(A)$ と書く。他の接辞は基本的に使わない。

Prologの節

事実 \sim である。

$\text{human}(\text{socrates}).$ ソクラテスは人間である。
 $\text{parent}(\text{taro}, \text{ichiro}).$ タロウは一郎の親である。

ルール \sim ならば \sim である。

\sim であるには \sim である必要がある。

$\text{father}(X, Y) :- \text{parent}(X, Y), \text{male}(X).$

XがYの親であり、Xが男性ならば、XはYの父親である。

XはYの父親であるには、XがYの親で、Xが男性である必要がある。

Prologで推論する例

仮定の命題

```
mortal(X) :- human(X).  
human(socrates).
```

質問

```
?- mortal(socrates).
```

Prologで推論する例 (つづき)

- 質問をルールや事実とマッチングしてゆきます。

```
質問      ?- mortal(socrates).  
           ↕ マッチング  
           mortal(X) :- human(X).
```

- mortal(X)の条件はhuman(X)なので、これを次に質問します。
- マッチングでX=socrates になったので、新しい質問は

```
新たな質問 ?- human(socrates).  
            ↕ マッチング  
            human(socrates).
```

Prologをコンピュータにインストールする

1. SWI-Prologのサイトをインターネットで検索する。
2. SWI-Prologのファイルをダウンロードする。
3. パソコンにインストールする。



プログラミングの基本ステップ

1. 方法 (アルゴリズム、理論) の設計
2. コーディング (プログラムを書く)
3. エディット (編集) edit
4. 実行 run
5. デバッグ (誤り探し) debug
1 ~ 3に戻る



Prologを使ってみる (1)

1. 理論となる節 (命題) をファイルに書く。
2. ファイルを読み込む。
3. 質問の命題を入れて見る。
4. 上手くいかなければ、もう一度エディット



Prologを使ってみる (2)

1. 理論となる節 (命題) をファイルに書く。
 - i. File > New からファイルを開く。適切な名前を付ける。 file.pl
 - ii. ファイルに節 (命題) を書く。
 - iii. セーブする。エディタの Save buffer



Prologを使ってみる (3)

2. ファイルを読み込む。

iv. File > Consult

読み込むと、節 (命題) は仮定された命題となります。

3. 質問の命題を入れて見る。

?- mortal(socrates).

仮定された知識から推論された答えを出します。

Prologを使ってみる (4)

4. 上手くいかなければ、もう一度エディット

v. File > Edit から、 で付けた名前のファイル呼び出す。

vi. から繰り返し。



練習 Prologで推論してみる

質問に答えられるよう、 事実をPrologの節として与えよ。
明示的に書かれていないことも必要なら加えよ。

事実

- タロウはヨウコにセーターを贈った。
- タカシはコーラを買った。



質問

- ヨウコは洋服を持っていますか？
- 飲み物を持っているのは誰ですか？

練習のヒント

事実

- タロウはヨウコにセーターを贈った。 (gave)
- タカシはコーラを買った。 (bought)

書いてないけど必要なこと

- AがBにCを贈ったら、BがCを持っている。 (have)
- AがBを買ったら、AはBを持っている。
- セーターは洋服である。
- コーラは飲みものである。
- AがBであり (isa)、CがAを持っていれば、CはBを持っている。

Prologを使った数値計算

計算結果を出すには「X is 式」で書きます。

?- X is 2*4+6 / (1+2).

?- X is sin(pi / 6).

不等式も書けます。

?- 2+3 >= 1.



数値を計算する関数を論理的に書く

関数を定義してみます。

$$y = 2x^2 + 5$$

f(X,Y) :- Y is 2*X*X+5

漸化式のような定義 (再帰的定義)

$$1! = 1$$

$$n! = (n - 1)! * n$$

fact(1,1).

fact(X,Y) :- X1 is X-1, fact(X1,Y1), Y is Y1*X.



練習課題 1: 家族

家族に関する述語についての規則 (定義) を与えてください。

父、母、祖父、おば

`father(X,Y)`, `mother`, `grandFather`, `aunt`

兄弟、姉妹、いとこ、一人っ子

`brother(X,Y)`, `sister`, `cousin`, `anOnlyChild(X)`

祖先、子孫

`ancestor(X,Y)`, `offspring(X,Y)`

家族 : 与えられている事実

■ 親子関係

□ `parent(taro, hanako)`.

タロウはハナコの親である。

■ 性別

□ `male(taro)`.

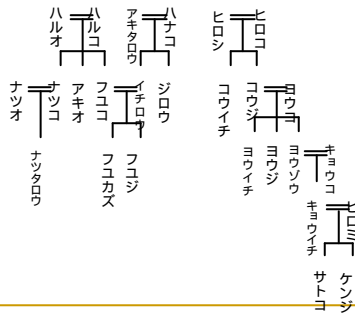
タロウは男である。

□ `female(hanako)`.

ハナコは女である。

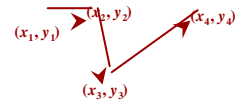
練習課題はこれらの事実で決まる関係です。

家族のデータ



練習課題 2: 矢印の連結

点と点を矢印で結んだ図を考える。



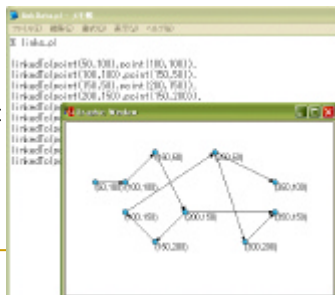
矢印で、2つの点 (x, y) と (x', y') が結ばれていることを次のように書くとする。

`linkedTo(point(x,y),point(x',y'))`.

描画のシステム

矢印を画面に出すプログラムを用意しています。

- `graphic.pl` を consult する。
- `linkedTo` の関係のデータを consult する。
- `?-drawArrows` で描画する。



課題

- 次の述語を定義してください。
点 (x, y) から矢印を通して点 (x', y') へ達することができるとき真となる述語
`canReach(point(x,y),point(x',y'))`.
点 (x, y) から矢印を通して右の点 (x', y') へ一度も左に戻らずに達することができるとき真となる述語
`doesNotBack(point(x,y),point(x',y'))`.

練習課題 3

次の述語で図形が座標上にあることを表します。

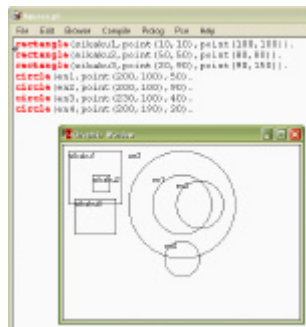
- 各辺が座標軸に平行な長方形 (矩形)
(左上の点と右下の点で表す)
`rectangle(shikaku1, point(x,y), point(x',y'))`.
- 円 (中心点と半径で表す)
`circle(en1, point(x,y), r)`.

課題

- 次の述語を定義してください。
2つの長方形、2つの円が重なりがあるときに真となる述語。
`intersect(shikaku1,shikaku2)`.
`intersect(en1,en2)`.
ある長方形 (円) が別の長方形 (円) に含まれるとき真となる述語。
`included(shikaku1, shikaku2)`.
`included(en1,en2)`.

図形の描画

- `graphic.pl`で描画できます。
- `graphic.pl`と図形データをconsult。
- `?-draw.`で描画できます。



練習課題 4

- 文字を線分の集まりで表現します。
- 文字 `moji1` を構成する線分が座標平面上にあることを次のように表す。
`line(moji1, point(x, y), point(x',y'))`.
たとえばしならば次のようになるでしょう。
`line(moji2, point(50, 50), point(50,150))`.
`line(moji2, point(50, 150), point(150,150))`.

課題

- いろいろな文字を認識する述語を作りたい。
- たとえば、`?-letterA(moji1)` で、`moji1` がAと読めるときに真となるようにしたい。
- 4種類の文字データを用意しました。

`letterData1` : L, T, X,

`letterData2` : A, E, W, Z,

`letterData3` : A, B, C, D, E,

`letterData4` : A~Zの26文字。

文字データの描画

- 文字も`graphic.pl`で描画できます。
- `?-draw.`で描画。



課題のヒント

次のような述語を用意するとよいかもしれません。

- 線分が鉛直である、水平である。
- 線分が直角に接続している。
- 線分がなめらかに (鈍角で)あるいは鋭く(鋭角で)接続している。