関数・論理型プログラミング実験
第12回

江口 慎悟
押川 広樹
塚田 武志

講義のサポートページ

http://www.kb.is.s.u-tokyo.ac.jp/~tsukada/cgi-bin/m/

- 講義資料等が用意される
- レポートの提出先
- 利用にはアカウントが必要

- 名前/学籍番号/希望アカウント名をメールを
  tsukada@kb.is.s.u-tokyo.ac.jp
  までメールしてください。
  - 件名は「FL/LP実験アカウント申請」
  - アカウント名/パスワードを返信
  - PC からのメールを受け取れるように

論理型プログラミング（全３回）

第10回　Prologの使い方
- Prologを使ってみよう

第11回　手続き的側面
- 評価メカニズム

第12回　論理的側面
- 完全性・健全性
- 否定と閉世界仮説

今日の内容

- Prolog の論理的側面
  - Prolog プログラムの論理的解釈
  - 否定と閉世界仮説
  - SLD導出の完全性・健全性
  - 実際の処理系との差異

Prologプログラムの論理的解釈

プログラムと問い合わせの意味

否定を扱うには

Prologの論理的解釈

- 問い合わせは（一階述語論理の）命題
  - `?- male(koji).` という問い合わせは `male(koji)` が証明できるかどうかを聞いている
- プログラムは命題の証明に使える仮定

```
male(koji).
parent(kobo, koji).
father(X, Y) :- parent(X, Y), male(Y).
```

  - このプログラムは三つの仮定を記述している

問い合わせの意味

- 複数の問い合わせは 論理積 を取る
  - 例： `?- parent(kobo, koji), male(koji).`
    ⇒ $\mathrm{parent}(\mathrm{kobo}, \mathrm{koji}) \wedge \mathrm{male}(\mathrm{koji})$

- 変数は存在量化する
  - 例： `?- parent(kobo, X).`
    ⇒ $\exists X.\, \mathrm{parent}(\mathrm{kobo}, X)$

  - 例： `?- parent(kobo, X), male(X).`
    ⇒ $\exists X.\, \mathrm{parent}(\mathrm{kobo}, X) \wedge \mathrm{male}(X)$

プログラムの意味

- 変数は全称量化する
  - 例： `positive(s(X)).`
    $\Rightarrow \forall X.\, \mathrm{positive}(\mathrm{s}(X))$

- ルールは「含意（ならば）」に対応
  - 例： `father(X,Y) :- parent(X,Y), male(Y).`
    $\Rightarrow \forall X \forall Y.\, \mathrm{parent}(X, Y) \wedge \mathrm{male}(Y) \rightarrow \mathrm{faher}(X, Y)$

- プログラムはすべてのルール・事実の論理積

例

- プログラム

```
male(koji).
parent(kobo, koji).
father(X, Y) :- parent(X, Y), male(Y).
```

  - 次の三つの仮定から成る
    1. $\mathrm{male}(\mathrm{koji})$
    2. $\mathrm{parent}(\mathrm{kobo}, \mathrm{koji})$
    3. $\forall X \forall Y . \mathrm{parent}(X, Y) \land \mathrm{male}(Y) \rightarrow \mathrm{father}(X, Y)$

- 問い合わせ

```
?- father(kobo, X).
```

  - $\exists X . \mathrm{father}(\mathrm{kobo}, X)$

証明してみよう

1. $\mathrm{male}(\mathrm{koji})$
2. $\mathrm{parent}(\mathrm{kobo}, \mathrm{koji})$
3. $\forall X \forall Y . \mathrm{parent}(X, Y) \wedge \mathrm{male}(Y) \rightarrow \mathrm{father}(X, Y)$

i) 仮定 2 から

$$\mathrm{parent}(\mathrm{kobo}, \mathrm{koji})$$

ii) さらに仮定 1 から

$$\mathrm{parent}(\mathrm{kobo}, \mathrm{koji}) \wedge \mathrm{male}(\mathrm{koji})$$

iii) 仮定 3 の $X = \mathrm{kobo}, Y = \mathrm{koji}$ の場合から

$$\mathrm{father}(\mathrm{kobo},\mathrm{koji})$$

従って $\exists X . \mathrm{father}(\mathrm{kobo}, X)$

証明してみよう

1. $\mathrm{male}(\mathrm{koji})$
2. $\mathrm{parent}(\mathrm{kobo}, \mathrm{koji})$
3. $\forall X \forall Y . \mathrm{parent}(X, Y) \wedge \mathrm{male}(Y) \rightarrow \mathrm{father}(X, Y)$

i) 仮定 2 から

$$\mathrm{parent}(\mathrm{kobo}, \mathrm{koji})$$

ii) さらに仮定 1 から

$$\mathrm{parent}(\mathrm{kobo}, \mathrm{koji}) \wedge \mathrm{male}(\mathrm{koji})$$

iii) 仮定 3 の $X = \mathrm{kobo}, Y = \mathrm{koji}$ の場合から

$$\mathrm{father}(\mathrm{kobo},\mathrm{koji})$$

従って $\exists X . \mathrm{father}(\mathrm{kobo}, X)$

証明

証明してみよう

1. male(koji)
2. parent(kobo, koji)
3. $\forall X \forall Y . \mathrm{parent}(X, Y) \wedge \mathrm{male}(Y) \rightarrow \mathrm{father}(X, Y)$

i) 仮定 2 から

$$\mathrm{parent}(\mathrm{kobo}, \mathrm{koji})$$

ii) さらに仮定 1 から

$$\mathrm{parent}(\mathrm{kobo}, \mathrm{koji}) \wedge \mathrm{male}(\mathrm{koji})$$

iii) 仮定 3 の $X = \mathrm{kobo}, Y = \mathrm{koji}$ の場合から

$$\mathrm{father}(\mathrm{kobo},\mathrm{koji})$$

従って $\exists X . \mathrm{father}(\mathrm{kobo}, X)$

Prologプログラムの論理的解釈

プログラムと問合せの意味

否定を扱うには

復習：失敗による否定

- Prolog における否定は「失敗による否定」
  - A の導出を試みて失敗したら、¬A を導出する

否定の記号は Prolog では \+

- 例

```
male(koji).
parent(kobo, koji).
father(X, Y) :- parent(X, Y), male(Y).
```

  - female(koji) は導出できない。
    よって ¬female(koji) が導出できる

さきほどの解釈の問題点

o 否定的な命題を証明することはできない

```
male(koji).
parent(kobo, koji).
father(X, Y) :- parent(X, Y), male(Y).
```

1. $\mathrm{male}(\mathrm{koji})$
2. $\mathrm{parent}(\mathrm{kobo}, \mathrm{koji})$
3. $\forall X \forall Y . \mathrm{parent}(X, Y) \land \mathrm{male}(Y) \rightarrow \mathrm{father}(X, Y)$

閉世界仮説

- 「書かれていない命題は偽」とする仮定
  - 例

```
male(koji).
parent(kobo, koji).
father(X, Y) :- parent(X, Y), male(Y).
```

    - $\neg \mathrm{parent}(\mathrm{koji}, \mathrm{kobo})$
    - $\forall X. \neg \mathrm{female}(X)$
- Prolog の否定の振舞いは、閉世界仮説でおおむね説明できる

もう少し複雑な例

```
male(kobo).
male(koji).
female(sanae).
parent(kobo, koji).
parent(kobo, sanae).
father(X, Y) :- parent(X, Y), male(Y).
```

もう少し複雑な例

```
male(kobo).
male(koji).
female(sanae).
parent(kobo, koji).
parent(kobo, sanae).
father(X, Y) :- parent(X, Y), male(Y).
```

$$\forall X.\left(male(X) \leftrightarrow (X = kobo) \vee (X = koji)\right)$$

もう少し複雑な例

```
male(kobo).
male(koji).
female(sanae).
parent(kobo, koji).
parent(kobo, sanae).
father(X, Y) :- parent(X, Y), male(Y).
```

$$\forall X.\left(\mathrm{male}(X) \leftrightarrow (X = \mathrm{kobo}) \vee (X = \mathrm{koji})\right)$$

$$\forall X.\left(\mathrm{female}(X) \leftrightarrow (X = \mathrm{sanae})\right)$$

もう少し複雑な例

```
male(kobo).
male(koji).
female(sanae).
parent(kobo, koji).
parent(kobo, sanae).
father(X, Y) :- parent(X, Y), male(Y).
```

$$\forall X.\left(\mathrm{male}(X) \leftrightarrow (X = \mathrm{kobo}) \vee (X = \mathrm{koji})\right)$$

$$\forall X.\left(\mathrm{female}(X) \leftrightarrow (X = \mathrm{sanae})\right)$$

$$\forall X \forall Y.\left(\begin{array}{c}\mathrm{parent}(X, Y) \leftrightarrow \\ (X = \mathrm{kobo}) \wedge (Y = \mathrm{koji} \ \vee Y = \mathrm{sanae})\end{array}\right)$$

もう少し複雑な例

```
male(kobo).
male(koji).
female(sanae).
parent(kobo, koji).
parent(kobo, sanae).
father(X, Y) :- parent(X, Y), male(Y).
```

$$\forall X.\left(\mathrm{male}(X) \leftrightarrow (X = \mathrm{kobo}) \vee (X = \mathrm{koji})\right)$$

$$\forall X.\left(\mathrm{female}(X) \leftrightarrow (X = \mathrm{sanae})\right)$$

$$\forall X \forall Y.\left(\begin{array}{c}\mathrm{parent}(X, Y) \leftrightarrow \\ (X = \mathrm{kobo}) \wedge (Y = \mathrm{koji} \ \vee Y = \mathrm{sanae})\end{array}\right)$$

$$\forall X \forall Y.\left(\mathrm{father}(X, Y) \leftrightarrow \mathrm{parent}(X, Y) \wedge \mathrm{male}(Y)\right)$$

手続きと論理の関係

SLD導出の完全性・健全性

Prologと論理の差

手続きと論理と

- 前回： Prolog の処理系の動作原理
  - Prolog の手続き的側面
- 今回： Prolog プログラムの論理的解釈
  - Prolog の論理的側面
- この二つの関係はどうなっているのか？

手続きと論理と

- 前回： Prolog の処理系の動作原理
  - Prolog の手続き的側面
- 今回： Prolog プログラムの論理的解釈
  - Prolog の論理的側面
- この二つの関係はどうなっているのか？
  - 原理的には二つは一致
  - しかし、実装上の都合で、異なる側面も

手続きと論理の関係

SLD導出の完全性・健全性

Prologと論理の差

SLD導出の完全性・健全性

o プログラム P と問合せ Q について以下は同値

- P を使って SLD導出によって Q が導出できる
  （Prolog プログラムの手続き的解釈）
- P の論理的解釈から Q の論理的解釈が証明できる
  （Prolog プログラムの論理的解釈）

o ゆえに手続き的解釈と論理的解釈は一致する

※ ただし、ここでいう SLD 導出では、
「ルールは上が優先」という規則は採用しない

手続きと論理の関係

SLD導出の完全性・健全性

Prologと論理の差

Prologと論理の差

- ルールとゴールの優先順位
- 単一化の出現チェック
- 変数のある命題の否定
- カットの有無
  - 論理に（Prolog的な）カットはない

ルールとゴールの優先順位

- Prolog
  - 上のルールが優先
  - 左から右の順序でゴールを解決
- 論理
  - ルールに優先順位を考えない
    - いつでも、どのルールでも適用可能
  - ゴールの解決の順序も任意

例

- 下のプログラムは、
  - Prolog 処理系に与えたときの挙動が異なる
    - 上は無限ループに陥る
  - 論理的解釈は同じ

```
ancestor(X,Y) :- ancestor(Z,Y), parent(X,Z).
ancestor(X,Y) :- parent(X,Y).
```

```
ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(X,Z), ancestor(Z,Y).
```

単一化の出現チェック

- 多くの Prolog 処理系では、単一化において「出現チェック」を省いている
  - パフォーマンスのため

  ※ 出現チェック：「X = t」という制約は、
  X に t を入れる代入が最汎単一化子になる。
  ただし t 中に X が出現してはならない

- 出現チェックをさぼると、
  証明できないのに導出される命題が生じる

例

```
add(z, Y, Y).
add(s(X), Y, s(Z)) :- add(X, Y, Z).

p :- add(s(z), X, X).
```

```
?- p.
true.
```

例

```
add(z, Y, Y).
add(s(X), Y, s(Z)) :- add(X, Y, Z).

p :- add(s(z), X, X).
```

```
?- p.
true.
?- add(s(z), X, X).
X = s(X).
```

変数のある命題の否定

- 「失敗による否定」は
  変数のない命題にだけ使える
  - 論理的に正しい「失敗による否定」の規則：
    変数を持つ命題の否定は計算せずに、
    別のゴールの導出を先に行う。
    変数が具体化されるのを待って、否定を計算する

- Prolog の処理系では、変数の有無を気にせず、
  「失敗による否定」規則を使おうとする

例

```
male(kobo).
male(koji).
female(sanae).
parent(kobo, koji).
parent(kobo, sanae).
mother(X,Y) :- \+male(Y), parent(X, Y).
```

```
?- \+male(Y).
false.
?- mother(kobo, Y).
false.
?- mother(kobo, sanae).
true.
```

例

```
male(kobo).
male(koji).
female(sanae).
parent(kobo, koji).
parent(kobo, sanae).
mother(X,Y) :- \+male(Y), parent(X, Y).
```

parent(X,Y), \+male(Y)
なら期待通りの動作

```
?- \+male(Y).
false.
?- mother(kobo, Y).
false.
?- mother(kobo, sanae).
true.
```

例題

理解の確認をするための課題です
課題提出システム上での提出の必要はありません
例題を解きTAに見せることで出席とします
分からないことがあったら、積極的に質問しましょう

例題

- 次を意味するプログラムを書け
  1. $\forall X.\,\mathrm{sub}(X, \mathrm{z}, X).$
  2. $\forall X \forall Y \forall Z.\,\mathrm{sub}(X, Y, Z) \rightarrow \mathrm{sub}(\mathrm{s}(X), \mathrm{s}(Y), Z).$

- 以下のそれぞれの論理式に相当する問い合わせをせよ
  - $\exists X.\,\mathrm{sub}(\mathrm{s}\left(\mathrm{s}\left(\mathrm{s}(\mathrm{z})\right)\right), \mathrm{s}(\mathrm{z}), X)$
  - $\exists X.\,\mathrm{sub}(\mathrm{z}, \mathrm{s}(\mathrm{z}), X)$

レポート課題12

締切：2018/7/17 13:00(JST)

問 1

o 次のプログラムを考える

```
eq(a, b).
eq(c, b).
eq(X, Z) :- eq(X, Y), eq(Y, Z).
eq(X, Y) :- eq(Y, X).
```

- 論理的解釈では eq(a,c) が true であることを示せ
- Prolog 処理系で ?- eq(a,c). を
  問い合わせるとどうなるか。それはなぜか
- 処理系をどう工夫すれば、この差が埋まるか

参考：プログラムの論理的解釈

```
eq(a, b).
eq(c, b).
eq(X, Z) :- eq(X, Y), eq(Y, Z).
eq(X, Y) :- eq(Y, X).
```

1. $\mathrm{eq}(\mathrm{a}, \mathrm{b})$
2. $\mathrm{eq}(\mathrm{c}, \mathrm{b})$
3. $\forall X \forall Y \forall Z.\ \mathrm{eq}(X, Y) \land \mathrm{eq}(Y, Z) \rightarrow \mathrm{eq}(X, Z)$
4. $\forall X \forall Y.\, \mathrm{eq}(Y, X) \rightarrow \mathrm{eq}(X, Y)$

eq は部分同値関係（partial equivalence relation）

問２

o 次のプログラムを考える

```
test :- q(X, X).
q(X, f(X)).
```

- 論理的解釈では ?- test. の問合せの結果はどうなると考えられるか

- 実際に ?- test. を Prolog処理系に問い合わせるとどうなるか。どうしてそうなるのか。

  - ヒント：?- q(X, X). を問い合わせるとどうなるか

問３

o 以下のプログラムと問合せを考える。

```
p(a).
q(b).
```

```
?- \+p(X), q(X).
```

- 論理的な解釈から期待される結果は何か
- Prolog 処理系に実際に問い合わせるとどうなるか。なぜそのような結果になるか。
- 論理的解釈と Prolog処理系の応答が一致するように問合せを書き換えよ

参考：プログラムの論理的解釈

```
p(a).
q(b).
```

1. $\forall X.(p(X) \leftrightarrow X = a)$
2. $\forall X.(q(X) \leftrightarrow X = b)$

発展 1

o 次のプログラムと問合せを考える。

```
r(a) :- p(a).
r(a) :- \+p(a).
p(X) :- p(f(X)).
```

```
?- r(a).
```

- r(a) がプログラムの論理的帰結であることを示せ
- Prolog処理系はこの問合せにどう答えるか。
  それはなぜか。

発展 2

o 論理的意味に対して完全かつ健全な
Prolog ライクな論理型言語を実装せよ

- すべての解代入が、いずれ出力される
  - 同じ解代入を何度出力してもよい
  - すべての解代入を出力しても、停止する必要はない
- 構文解析器を準備する必要はない
- カットや否定はなくてよい