# 情報論理学 (第６回)

基礎工学部情報科学科　泉 泰介

# 今回の内容



- 述語論理の形式的表現と意味論
  - 記号列としての述語論理の定義
  - 変数と限定作用素の取り扱い，変数への代入
  - 述語論理の解釈，充足可能性，恒真性

- 今回は論理式の意味論的な取り扱いにふたたび戻るので，頭の切り替えを
  - ただし定義等は演繹的な取り扱いと共通する
  - 述語論理の公理系と演繹的な取り扱いは授業後半(中川先生担当)で検討する

# Part1：述語論理の定義

# 述語論理の記号集合

- 以下に示す6つの記号集合からなる
  1. 対象定数(individual constant)：$C = \{a, b, c, d, \dots, a_1, a_2, \dots\}$
  2. 対象変数(individual variable)：$X = \{x, y, \dots, x_1, x_2, \dots\}$
  3. 関数記号(function symbol)：$F = \{f, g, \dots, f_1, f_2, \dots\}$
  4. 述語記号(predicate symbol)：$P = \{p, q, \dots, p_1, p_2, \dots\}$
  5. 論理記号：$\neg, \rightarrow, \forall, \exists$
  6. 補助記号：(, ), “,“（カッコとコンマ)

- 例との対応
  - $\forall x\, \forall y\, \left(\exists z\, p(x, y, z) \rightarrow q\big(f(x), g(z)\big)\right)$
  - $\forall x\, (\exists y\, f(x) = y + 3)$

# 述語論理の記号集合

- 以下に示す6つの記号集合からなる
  1. 対象定数(individual constant)：$C = \{a, b, c, d, \dots, a_1, a_2, \dots\}$
  2. 対象変数(individual variable)：$X = \{x, y, \dots, x_1, x_2, \dots\}$
  3. 関数記号(function symbol)：$F = \{f, g, \dots, f_1, f_2, \dots\}$
  4. 述語記号(predicate symbol)：$P = \{p, q, \dots, p_1, p_2, \dots\}$
  5. 論理記号：$\neg, \rightarrow, \forall$
  6. 補助記号：(, ), “,“ (カッコとコンマ)

- 例との対応
  - $\forall x\ \forall y \left(\exists z\ p(x, y, z) \rightarrow q(f(x), c)\right)$
  - $\forall x\ (\exists y\ f(x) = y + 3)$

算術的な陳述を取り扱うときは
関数記号，述語記号として二項演算が
(ある種の別表記として)用いられる
ときもある

厳密に記述するなら$\forall X \left(\exists y = \left(f(x), +(y, 3)\right)\right)$

# 論理式の定義



**定義($C, X, F$上の項)**

以下に該当するものを$C, X, F$上の(term)($C, F$が明らかなときは単に項)と呼ぶ

a.　$C$中の対象定数あるいは$X$中の対象変数

b.　$f \in F$を$n$引数の関数記号，$t_1, t_2, \ldots t_n$を項とすると，$f(t_1, t_2, \ldots, t_n)$は項

**定義($C, X, F, P$上の論理式集合)**

以下に該当するもの全てからなる集合を$C, X, F, P$上の論理式集合$L(C, X, F, P)$と呼ぶ

a.　$p \in P$を$n$引数の述語記号，$t_1, t_2, \ldots t_n$を$C, X, F$上の項とするとき，$p(t_1, t_2, \ldots, t_n)$

b.　2つの論理式$A, B$に対する，$(A) \rightarrow (B)$ および $\neg(A)$

c.　論理式$A$,対象変数$x$に対する，$\forall x(A)$

- 上の定義a.に該当する論理式は原子論理式(atomic formula)と呼ばれる
- 論理式集合$L(C, X, F, P)$は**言語**と呼ばれる場合もある

# 述語論理における略記法



- ∨,∧, ↔等の記号の導入は命題論理と同様
  - $P \vee Q$ ：$(\neg P) \to Q$ の略記
  - $P \wedge Q$ ： $\neg(P \to \neg Q)$の略記
  - $P \leftrightarrow Q$ ：$(P \to Q) \wedge (Q \to P)$の略記
- 記号∃も同様に略記法として定義される
  - $\exists x(A)$ ：$\neg\forall x(\mathrm{A})$の略記
  - ∀, ∃はいろいろな呼び名がある(右参照)
- カッコの省略も命題論理のときと同様
  - ∀, ∃の優先順位は¬と同様(→,∧,∨よりも高い)

| ∃ | ∀ |
|---|---|
| 存在記号 | 全称記号 |
| 存在作用素 | 全称作用素 |
| 存在量化子 | 普遍量化子 |

(英語だと∃はexistential quantifier, ∀はuniversall quantifier，両方合わせてquantifierと呼ぶ)

# 束縛変数と自由変数



- 限定作用素∀, ∃は引き続いて現れる変数を**束縛**する
  - $p(x)$を1引数の述語としたとき，論理式$\forall x\ p(x)$ や $\exists x\ p(x)$において，変数$x$はこれら論理式の外側からは操作できない →束縛変数(bound variable)
  - $q(x,y)$を2引数の述語としたとき，論理式$\forall x\ q(x,y)$や論理式$\exists x\ q(x,y)$において，$y$は論理式の外側から操作できる →自由変数(free variable)
- 例：$C$を自然数すべての集合として，×, =を標準的な算術演算(の関数)として定義しているとする)
  - 論理式$\exists x(x \times x = y)$において$x$は束縛変数，$y$は自由変数
    - この論理式は（記号に対する自然な意味づけのもとで)「$y$は平方数である」という述語のように働く
  - 論理式$\forall x \forall y(x + y = y + x)$において$x$,$y$はいずれも束縛変数
    - この論理式は真理値が確定する命題のように働く

# 束縛変数と自由変数



- ある変数(記号)が束縛されているか自由であるかは必ずしも一通りに決まらない

$$\exists x\big((x \times x = y) \land \exists y(y + y = z)\big)$$

$y$のスコープ

$y$のスコープ

- この式においては$y$と$y$はある意味で別物
  - プログラミング言語のローカル変数とスコープをイメージするとわかりよいかもしれない

- このようなケースを対処するため，ある変数記号$y$についてその論理式中の特定の位置を「$y$の出現(occurrence)」と呼ぶ
  - 束縛/自由は，より厳密には変数の出現に対して定められる

# 束縛された出現と自由な出現



**定義(束縛された出現・自由な出現)**

a. 論理式Aが原子論理式のとき，$A$中の変数のすべての出現は自由

b. 論理式Aが$B \rightarrow C$または$\neg C$の形のとき，$B$または$C$における自由な出現は$A$における自由な出現

c. 論理式$A$が$\forall z(B)$の形のとき，$z$の出現はすべて束縛されている．$B$における変数$z$以外の出現については，それが自由な出現であれば$A$でも自由であり，束縛された出現であれば$A$でも束縛されている

- 自由な変数=自由な出現を持つ変数
- 束縛された変数=束縛された出現を持つ変数
- 自由な出現と束縛された出現両方がある変数は，**自由かつ束縛されている**

- 自由な出現を一つも含まない論理式を閉論理式(closed formula)と呼ぶ

# 代入



- 論理式中の自由な変数は(文字通り)自由に他の項で置き換えることができる
  - 論理式$T$において変数$x$が自由変数であることを明記するために$T(x)$と記述する
  - $T(x)$における$x$の自由な出現を項$t$で置き換える操作を($T(x)$における)$t$の$x$への代入と呼び，得られる論理式を$T(t)$ or $T^{x \leftarrow t}$で表す
  - 複数の代入を同時に行うときもある．その場合，$\Theta: x \leftarrow t_1, y \leftarrow t_2$として，$T^{\Theta}$,$T(\Theta)$のように書く場合もある
- 代入操作はいつも安全に行えるわけではない
  - $t$中に$T$中で束縛されている変数が出現するときには注意が必要
  - 例：$\exists z(x \times z = y)$ に対して代入$x \leftarrow z \times z$を考える
    - 元の式の意味は「$y$は$x$の倍数」で，代入の意図は「$y$は$z \times z$の倍数」
    - 代入後の式は「$y$は立方数」　(いずれも記号に対する自然な意味づけのもとで)

# 代入



- 安全な代入のためには，$T$の束縛変数を事前に別の記号に置き換えする(実際には，この操作も含めて「代入」と定義される)
- 前述の例だと，$\exists z(x \times z = y)$ に対してまず$\exists u(x \times u = y)$と変数を置き換えてから$x \leftarrow z \times z$と代入すると安全に代入結果$\exists u(z \times z \times u = y)$が得られる
- 束縛変数を置き換える記号は($t$にも$T$にも出現しないものを)任意に選んでよい
  - この意味で，代入後の式は一意には決まらない
  - 選ぶルールを事前に何か決めておけば(e.g., 変数記号に通し番号を打っておいて，使われていない最小の番号のものを選ぶ)一意に決まる
- 項$t$，論理式$T(x)$に対して，$t$中の変数の出現が$T(t)$ですべて自由であるとき(＝置き換え不要で代入できるとき)$\boldsymbol{t}$**は**$\boldsymbol{T(x)}$**の**$\boldsymbol{x}$**に対して自由**であるという

# 代入の例(問3.1.1)



- $T(x) = \left(\forall x\, r(x) \to p(x)\right) \to \forall y \neg \left(r(y) \to q(x,y)\right)$とする.
- $t_1 = f(x,y)$, $t_2 = g(x)$, $t_3 = h(a,v)$とする

1. $t_1$は$T(x)$の$x$に対して自由か？ 自由ならば代入結果$T(t_1)$を求めよ
2. $t_2$, $t_3$に関してはどうか？

# 代入の例(問3.1.1)



- $T(x) = \left(\forall x\, r(x) \to p(x)\right) \to \forall y \neg \left(r(y) \to q(x,y)\right)$とする.
- $t_1 = f(x,y)$, $t_2 = g(x)$, $t_3 = h(a,v)$とする

- $t_1$：$x$に対して自由ではない(後半の$\forall y \neg \left(r(y) \to q(x,y)\right)$は$y$が束縛されており，$t_1$は変数$y$を含んでいる)
- $t_2$：自由．代入結果は$T(t_2) = \left(\forall x\, r(x) \to p\left(g(x)\right)\right) \to \forall y \neg \left(r(y) \to q(g(x),y)\right)$
- $t_3$：自由．代入結果は$T(t_3) = \left(\forall x\, r(x) \to p\left(h(a,v)\right)\right) \to \forall y \neg \left(r(y) \to q(h(a,v),y)\right)$

# Part2：述語論理の解釈と恒真性

# 対象領域と解釈

- 述語論理の論理式に意味を与えるには登場する記号群を順次具体化していく(性質を定義づける)
  - 変数や定数記号に対する「取りえる値」の集合$D$をまず対象領域(domain)として定める
  - 定数記号$a \in C$に対しては$D$中の要素1つが対応付けられる
  - ($n$変数)関数記号$f \in F$に対しては，$D^n \rightarrow D$という写像を対応付ける
  - ($n$変数)述語記号$p \in P$に対しては，$D^n \rightarrow \{真,偽\}$という写像を対応付ける

これらをすべての記号に対して定めたものを述語記号の**解釈**と定義する

- 対象領域$D$とその上の解釈$I$の対$(D, I)$を構造(structure)という
- 実際には(対象の)論理式に出てくる変数だけを意味づけて解釈とすることもある

# 解釈の例



- $\forall x\, p(f(x,a),b) \vee q(x)$ に対する解釈例
- その1：$D$として無理数全体を取る
  - $a$を$\sqrt{5}$とする．$b$を$\sqrt{11} \times \sqrt{67}$とする
  - $f(u,v)$を$u + v < \pi$のとき$v$，そうでないとき$v^2 + u$を表す
  - $p(u,v)$は$u^2 > v$を表す述語，$q(u)$は$10 < u < 13$を表す述語とする
- その2：$D = \{A, B\}$
  - $a$は$A$，$b$は$B$とする
  - $f(u,v)$は$u = v = A$ならば$A$，それ以外は$B$
  - $p(u,v)$は$u \neq v$ならば真，$u = v$ならば偽
  - $q(u)$は$u = A$ならば真，$u = B$ならば偽

# 閉論理式の意味づけ



- 論理式$A$に対して領域空間$D$のもとでの解釈$I$を与える
  - 論理式$A$が自由な変数を含む場合(=閉論理式でない場合)，それらに対して$D$中の値の割り当ての自由度がある
  - $A$に含まれる自由変数への$D$中の要素の割り当て$g$を考えて，$I, g$を固定すると$A$の真偽が確定する
- 命題論理のときと同様に$[\![A]\!]^{I,g}$で$I, g$のもとでの$A$の真偽を表す
  - $A$が閉論理式（＝自由な変数を含まない）ときは$g$を省略して$[\![A]\!]^{I}$と書く

# 論理式の意味づけ



- 形式的には$[\![A]\!]^{I,g}$は以下のような方法で確定する
- 前処理
  - $g$に従って$A$中の自由な変数の値を確定させる

原子論理式：$p \in P$を$n$引数の述語記号，$t_1, t_2, \dots t_n$を$C, X, F$上の項とするとき，$p(t_1, t_2, \dots, t_n)$

- 以下のように再帰的に評価する
  1. $A$が原子論理式ならば，$I$に従って真偽値を確定する
  2. $A = \neg B$ならば$B$の真偽を評価し，それを反転
  3. $A = B \rightarrow C$ならば$B, C$の真偽を評価し，$\rightarrow$の真理値表に従って$A$の真偽を確定
  4. $A = \forall x\ B$ のとき，すべての$d \in D$に対して$B^{x \leftarrow d}$が真ならば$A$は真，そうでないなら偽

前処理により$B, C$も自由な変数を持たない閉論理式であることが保証される
(ので，真偽値は必ず確定する)

前処理により，$B$は自由変数として$x$だけを持つことが保証される

# 充足可能性



- 解釈$I$が$g$について論理式$A$を充足(satisfy)　$\Leftrightarrow$ $[\![A]\!]^{I,g}$が真
  - $\vDash_{I,g} A$と表す
- 解釈$I$が論理式$A$を充足(satisfy)する $\Leftrightarrow$どのような$g$についても $[\![A]\!]^{I,g}$が真
  - $\vDash_{I} A$と表す
  - $A$が閉論理式のときは単に$[\![A]\!]^{I}$が真であることの言い換えになる
- $A$を充足する解釈$I$が存在するとき，$A$は充足可能(satisfiable)といい，充足可能でない論理式を充足不能(unsatisfiable)という
  - <u>$A$が閉論理式でないとき「$A$が充足不能$=$ $A$が常に偽」ではないことに注意が必要</u>

# 論理式の閉包



- $g$なしで充足性を定義するために，閉じていない論理式を以下の操作で閉論理式に置き換えることもある

**定義(論理式の閉包)**

$f(x_1, x_2, \dots x_n)$を$x_1, x_2, \dots x_n$が自由変数であるような論理式とするとき，
$\forall x_1 \forall x_2, \dots, \forall x_n\ f(x_1, x_2, \dots x_n)$ を$f$の**閉包**(closure)と呼ぶ

- 閉包を考えるとすべての論理式は閉論理式になり，「論理式が充足可能=真になるような解釈」と命題論理同様の定義が成り立つ

# 恒真性とモデル



- 論理式$A$が恒真　$\Leftrightarrow$　いかなる$I, g$に対しても$[\![A]\!]^{I,g}$が真
  　$\Leftrightarrow$　$A$の閉包$A'$について，いかなる$I$も$[\![A']\!]^{I}$を真にする

- $\Gamma$を論理式の集合としたとき，$\Gamma$中のすべての論理式を充足する解釈$I$を$\Gamma$のモデルと呼ぶ

- $\Gamma$の任意のモデル$I$が$\vDash_I A$を満たすことを$\Gamma \vDash A$と書く
  - 命題論理のときと同じ

# 恒真性の判定



- 命題論理のときの授業に倣うと，意味論の次に恒真性の判定を検討したい
- ...が，実は以下のような事実が知られている

**定理**

(1階)述語論理式の恒真性を有限時間で判定するアルゴリズムは存在しない

- 命題論理のように完全な形で判定することはできないが「答えが恒真であるときだけ正しい答えを返す」という半判定アルゴリズムは作ることはできる
  - 標準形の話や導出原理の述語論理への拡張は後半の授業で説明される