

『論理と形式化』資料 No.3
一階述語論理の意味論
亀山幸義 (kam[at]cs.tsukuba.ac.jp)

1 一階述語論理の解釈の例 (言語 L_1 に対して)

意味論 (semantics) は、論理式に意味を与えるものである。

論理式に意味を与える「手段」を、解釈 (interpretation) と呼ぶ。「論理式 A をこのように解釈すると、真 (あるいは偽) である」といったことを決めるものである。あるいは、「この解釈のもとで、論理式 A は、真 (あるいは偽) である」という。

この資料では、わかりやすさのため、意味論を与える対象を、一般の言語ではなく、特定の言語 L_1 に限定する。

言語 L_1 :

- 定数は 0 のみ。
- 関数記号は、 s (1 引数), $+$, $*$ (2 引数) の 3 つ。
- 述語記号 $=$ (2 引数) のみ。

この言語に対する 3 つの解釈 I^1, I^2, I^3 を与える。

言語 L_1 に対する解釈 I_ρ^1 :

- 領域 D : 自然数の集合 $\mathbb{N}$ とする。
- 付値 $\rho: \text{Var} \rightarrow D$: それぞれの変数に、 D の要素を対応付ける。ここで、 Var はすべての変数の集合である。 ρ の例として、すべての $x \in \text{Var}$ に対して $0 \in D$ を対応付けるものを、ここでは選ぶ。すなわち、 $\rho(x) = 0$ である。
- 定数の解釈: 定数 0 を、 D の要素としての 0 に対応付ける。
- 関数記号の解釈: s は、自然数上の「1 を加える」関数、 $+$, $*$ はそれぞれ、足し算、かけ算とする。
- 述語記号の解釈: $=$ の解釈は、 D の要素として等しいとき真で、そうでないとき偽とする。

I_ρ^1 のもとでの論理式 F の真理値を、 $I_\rho^1[F]$ とあらわす。 ρ があきらかなときは、 $I^1[F]$ と書くこともある。また、 ρ のもとでの項 t の値を、 $\rho(t)$ とあらわす。

この解釈のもとで以下が成立する。

例 1. 解釈 I_ρ^1 のもとで、論理式 $x + y = x * s(s(0))$ は真であり、 $x + y = s(x) * s(s(0))$ は偽である。

例 2. 解釈 I_ρ^1 のもとで、論理式 $(x + y = x * s(s(0))) \vee (x + y = s(x) * s(s(0)))$ は真であり、 $(x + y = x * s(s(0))) \wedge (x + y = s(x) * s(s(0)))$ は偽である。

例 3. 解釈 I_ρ^1 のもとで、論理式 $\forall x. \forall y. (x + y = x * s(s(0)))$ は偽であり、 $\exists x. \exists y. (x + y = x * s(s(0)))$ は真である。

例 4. 解釈 I^1 のもとで、論理式 $\forall x. \exists y. (x + y = x * s(s(0)))$ は真であり、 $\exists x. \forall y. (x + y = x * s(s(0)))$ は偽である。

上記の解釈の計算をひとつひとつ見て行こう。

例 1. 解釈 I_ρ^1 のもとで、論理式 $x + y = x * s(s(0))$ は真であり、 $x + y = s(x) * s(s(0))$ は偽である。

理由: 以下のように計算できるから。 $\rho(x + y) = \rho(x) + \rho(y) = 0 + 0 = 0$. $\rho(x * s(s(0))) = \rho(x) * \rho(s(s(0))) = 0 * (1 + \rho(s(0))) = 0$. $\rho(x + y) = \rho(x * s(s(0)))$ なので、 $I_\rho^1[x + y = x * s(s(0))]$ は真である。

同様に、 $\rho(x + y) \neq \rho(s(x) * s(s(0)))$ なので、 $I_\rho^1[x + y = s(x) * s(s(0))]$ は偽である。

例 2. 解釈 I_ρ^1 のもとで、論理式 $(x+y = x*s(s(0))) \vee (x+y = s(x)*s(s(0)))$ は真であり、 $(x+y = x*s(s(0))) \wedge (x+y = s(x)*s(s(0)))$ は偽である。

理由: 以下のように計算できるから。 $I_\rho^1[x+y = x*s(s(0))]$ は真であり、 $I_\rho^1[(x+y = s(x)*s(s(0)))]$ は偽であるので、 $\vee$ の真理値表により、 $I_\rho^1[(x+y = x*s(s(0))) \vee (x+y = s(x)*s(s(0)))]$ は真である。

例 3. 解釈 I_ρ^1 のもとで、論理式 $\forall x.\forall y.(x+y = x*s(s(0)))$ は偽であり、 $\exists x.\exists y.(x+y = x*s(s(0)))$ は真である。

理由: 以下のように計算できるから。 $I_\rho^1[\forall x.\forall y.(x+y = x*s(s(0)))]$ が真であることは、すべての $u, v \in \mathbb{N}$ に対して、 $I_{\rho[x \mapsto u, y \mapsto v]}^1(x+y = x*s(s(0)))$ が真であること。よって、 $\rho[x \mapsto u, y \mapsto v](x+y) = \rho[x \mapsto u, y \mapsto v](x*s(s(0)))$ が真であること、となる。この左右両辺を計算すると、 $u+v = 2u$ となるが、これはすべての自然数 u, v に対して成立するわけではないので、偽である。よって、 $I_\rho^1[\forall x.\forall y.(x+y = x*s(s(0)))]$ は偽である。

ここで $\rho[x \mapsto u, y \mapsto v]$ というのは、 ρ とほとんど同じ付値であるが、 x には u を、 y には v を対応付けたもの。

同様に、 $I_\rho^1[\exists x.\exists y.(x+y = x*s(s(0)))]$ を計算すると、ある $u, v \in \mathbb{N}$ に対して、 $I_{\rho[x \mapsto u, y \mapsto v]}^1(x+y = x*s(s(0)))$ が真であること。よって、 $\rho[x \mapsto u, y \mapsto v](x+y) = \rho[x \mapsto u, y \mapsto v](x*s(s(0)))$ が真であること、となる。

この左右両辺を計算すると、 $u+v = 2u$ となるが、これは、ある自然数 u, v に対して成立するので ($u = v = 0$ とすればよい)、真である。よって、 $I_\rho^1[\exists x.\exists y.(x+y = x*s(s(0)))]$ は真である。

例 4-1. 解釈 I^1 のもとで、論理式 $\forall x.\exists y.(x+y = x*s(s(0)))$ は真である。

理由: 例 3 と同様に計算すると、結局、「どんな自然数 u に対しても、ある自然数 v があって、 $u+v = 2u$ である」ということが成立するかどうか、ということに帰着され、これは $v = u$ と置けば成立するので、真である。

例 4-2. 解釈 I^1 のもとで、論理式 $\exists x.\forall y.(x+y = x*s(s(0)))$ は偽である。

理由: 例 3 と同様に計算すると、結局、「ある自然数 u があって、どんな自然数 v に対しても、 $u+v = 2u$ である」ということが成立するかどうか、ということに帰着され、そのような自然数は存在しないので、真である。

別の解釈も与えてみよう。

言語 L_1 に対する解釈 I^2 :

- 領域 D : 自然数を要素とする $(2,2)$ -正方行列の集合。
- 付値 ρ : すべての変数に、ゼロ行列を対応付ける。
- 定数の解釈: 定数 0 を、ゼロ行列に対応付ける。
- 関数記号の解釈: s は、「単位行列を加える」関数とする。 $+$, $*$ はそれぞれ、行列としての足し算とかけ算である。

例 11: この解釈のもとで、 $\forall x.\forall y.(x = y \supset s(x) = s(y))$ という論理式は真である。

例 12: この解釈のもとで、 $\forall x.\forall y.(x * y = y * x)$ という論理式は偽である。

例 13: この解釈のもとで、 $\forall x.\forall y.(x * y = 0 \supset (x = 0 \vee y = 0))$ という論理式は偽である。

2 一階述語論理の解釈の方法 (言語 L_1 に対して)

解釈 I_ρ が与えられると、原子論理式の意味は定まる。これを拡張して、一般の論理式の意味を定めよう。なお、 I における領域を D とする。

言語 L_1 に対する意味論 (まずは命題論理部分に対して):

解釈 I と付値 ρ のもとでの論理式 F の意味 $I_\rho[F]$ を次のように定義する。

- 原子論理式 $\perp$ の意味: $I_\rho[\perp]$ は偽である。
- 原子論理式 $t = s$ の意味 $I_\rho[t = s]$: $\rho(t)$ と $\rho(s)$ が D の要素として等しいとき真で、そうでないとき偽である。
- $I_\rho[(F \wedge G)]$ は、 $I_\rho[F]$ と $I_\rho[G]$ の両方が真のとき、真であり、それ以外の場合、偽である。
- $I_\rho[F \vee G]$ の意味は、 $I_\rho[F]$ と $I_\rho[G]$ の1つ以上が真のとき、真であり、それ以外の場合、偽である。
- $I_\rho[F \supset G]$ の意味は、 $I_\rho[F]$ が偽か、 $I_\rho[G]$ が真のとき、真であり、それ以外の場合、偽である。
- $I_\rho[\neg F]$ の意味は、 $I_\rho[F]$ が偽のとき、真であり、それ以外の場合、偽である。

ここまでは、命題論理の真理値表と同じである。

例: $\rho(x) = 13, \rho(y) = 15$ のとき、 $I_\rho[(x + y = s(x) * s(s(0))) \wedge (y = s(s(x)))]$ は真である。

さて、一番の難関である。 $\forall$ と $\exists$ の意味を与える準備として、付値の拡張 $\rho[x \mapsto u]$ を定義する。ここで、 x は変数であり、 u は領域 D の要素である。

$$\rho[x \mapsto u](y) = \begin{cases} u & \text{if } y = x \\ \rho(y) & \text{if } y \neq x \end{cases}$$

例: $\rho(x) = 13, \rho(y) = 15$ のとき、 $\rho[x \mapsto 20](x) = 20$ であり、 $\rho[x \mapsto 20](y) = 15$ である。

言語 L_1 に対する意味論 (述語論理部分に対して): 述語論理部分 (限量子) の意味論を次のように与える。

- $I_\rho[\forall x F]$ が真となるのは、 D の任意の要素 u に対して、 $I_{\rho[x \mapsto u]}[F]$ が真のときであり、それ以外の場合、偽である。
- $I_\rho[\exists x F]$ が真となるのは、 D のある要素 u に対して、 $I_{\rho[x \mapsto u]}[F]$ が真のときであり、それ以外の場合、偽である。

ここは難しいので、例をじっくり見てほしい。

例. L_1 に対する解釈 I^1 による意味論

$\forall$ と $\exists$ を含む、いろいろな論理式を解釈してみよう。ただし、 $\rho(x) = 13, \rho(y) = 15$ とする。

- $I^1_\rho[\exists z x + y = z * s(s(0))]$ は、 $I^1_{\rho[z \mapsto u]}[x + y = z * s(s(0))]$ が真になる $u \in D$ が1つでも存在すれば、真である。
ここで、 $u = 14$ とすると、 $\rho[z \mapsto u]$ のもとで、 $x + y$ は28であり、 $z * s(s(0))$ は28となるので、 $I^1_{\rho[z \mapsto u]}[x + y = z * s(s(0))]$ は真である。よって、 $I^1_\rho[\exists z x + y = z * s(s(0))]$ は真である。
- $I^1_\rho[\forall x \exists y s(x) = y]$ は、 $I^1_{\rho[x \mapsto u]}[\exists y s(x) = y]$ がすべての $u \in D$ に対して真となれば、真である。これは、 $I^1_{\rho[x \mapsto u][y \mapsto v]}[s(x) = y]$ がある $v \in D$ に対して真となれば、真である。
ここで、 v として $u + 1$ を取ると、 $s(x) = y$ は、 $u + 1 = u + 1$ となって真である。よって、そのような v を (どんな u に対しても) 取ることができる。よって、 $I^1_\rho[\forall x \exists y s(x) = y]$ は真である。
- $I^1_\rho[\exists y \forall x s(x) = y]$ は、 $I^1_{\rho[y \mapsto v]}[\forall x s(x) = y]$ がある $v \in D$ に対して真となれば、真である。
これは、 $I^1_{\rho[y \mapsto v][x \mapsto u]}[s(x) = y]$ がすべての $u \in D$ に対して真となれば、真である。

ここで、 y は、付値 $\rho[y \mapsto v][x \mapsto u]$ のもとで、 v という値である。 $s(x)$ の方は、この付値のもとで $u + 1$ という値を取る。

よって、「 $I^1_{\rho[y \mapsto v][x \mapsto u]}[s(x) = y]$ がすべての $u \in D$ に対して真となる」ためには、 $v = 1 = 2 = 3 = \dots$ が成立しなければならないが、そんなことは不可能である。したがって、「 $I^1_{\rho[y \mapsto v][x \mapsto u]}[s(x) = y]$ がすべての $u \in D$ に対して真となる」ことができないので、 $I^1_{\rho}[\exists y \forall x s(x) = y]$ は、偽である。

3 演習問題

問題: 言語 L_2 に対する解釈 I^2 に対して、例 11,12,13 の論理式の解釈を行いなさい。

解答 (例 11 のみ): I^2 における付値を ρ とする。(実は、 ρ は何であっても同じである)。 $I^2_{\rho}[\forall x \forall y x * y = y * x]$ が真となるのは、すべての $u_1, u_2 \in M_{2,2}$ に対して、 $I^2_{\rho[x \mapsto u_1][y \mapsto u_2]}[x * y = y * x]$ が真となる時である。(ここで、 2×2 行列すべての集合を $M_{2,2}$ とあらわした。)

これは、すべての $u_1, u_2 \in M_{2,2}$ に対して、 $u_1 * u_2 = u_2 * u_1$ が真となるということである。しかし、

$$u_1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \quad u_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$$

と置くと、 $u_1 * u_2 \neq u_2 * u_1$ となるので、「すべての。。。に対して。。。が真」ということはない。よって、 $I^2_{\rho}[\forall x \forall y x * y = y * x]$ は偽である。

4 一階述語論理の解釈の方法 (任意の言語に対して)

L_1 とは限らない任意の言語に対する解釈について、簡単に言及する。

言語 L_1 は、定数、関数記号の解釈をそれぞれ決めたが、一般の言語では、これら以外に、述語記号がある。たとえば、 P という 2 引数の述語記号があったとき、どのように解釈すればよいか、といえば、

述語記号 P (アリティ 2 とする) に対する解釈:

- $D \times D \rightarrow \text{bool}$ となる関数 f を対応付ける。
- $I_{\rho}[P(t_1, t_2)]$ の値は、 $f(\rho(t_1), \rho(t_2))$ の値 (bool 型の値、つまり、真か偽か) とする。

例: $D = \mathbb{N}$ とし、 P に対して、 $f(u, v) = \text{if } u < v \text{ then true else false}$ となる関数 f を対応付ける。

すると、 $I_{\rho}[P(s(0), s(s(0)))]$ は真となり、 $I_{\rho}[P(s(s(0)), s(0))]$ は偽となる。

5 恒真性と充足可能性

意味論で特に重要なのは、以下の 2 つの概念である。

- 論理式 A が、すべての I, ρ, D のもとで真であれば、 A は恒真 (valid) であるという。($\models A$)
- 論理式 A が、ある I, ρ, D のもとで真であれば、 A は充足可能 (satisfiable) であるという。

恒真性と充足可能性は表裏一体の関係であり、「 A が恒真」と、「 $\neg A$ が充足可能でない」ことは同じである。つまり、恒真性判定プログラムを作れば、充足可能性判定プログラムもできるし、逆も同様である。

例: P を 1 引数の述語記号とすると、 $P(x) \supset P(x)$ は恒真かつ充足可能であり、 $P(x) \supset (\neg P(x))$ は恒真ではないが、充足可能である。($P(x)$ が成立しない解釈を取れば、真となるので。) また、 $P(x) \wedge (\neg P(x))$ は、恒真でも充足可能でもない。

6 健全性と完全性

論理式 F に対して、「形式的体系で証明できる」と「恒真である」とはまったく独立に定義された。では、それらの関係はどうだろうか？

定理 1 (一階述語論理の健全性と完全性) 一階述語論理の論理式 F に対して、以下の 2 つは同値である。

- F が (NK あるいは、それと同等な他の証明体系で) 証明可能ならば、 F は恒真である。 [健全性 *soundness*]
- F が恒真ならば、 F は証明可能である。 [完全性 *completeness*]

証明は省略する。(参考書としてあげた小野先生の本や林先生の本などを参考にされたい。)

7 証明可能性と恒真性

一階述語論理に対しては、健全性と完全性が両方とも成立するので、証明可能性と恒真性は同値である。しかし、それらの「使い勝手」は異なる。

- 形式的体系: 証明できることは示しやすいが、証明できないことを示すのは困難である。
- 意味論: 恒真でない (ある解釈で偽である) ことは示しやすいが、恒真であることを示すのは大変である。

よって、論理式が成立する場合は形式的体系を使って証明し、論理式が成立しない場合は意味論を使って偽になる解釈の存在を示す、というのが、賢いやりかたである。

例: $(\forall x.\exists y.A(x,y)) \supset (\exists y.\forall x.A(x,y))$ は、領域 $D = \mathbb{N}$ とし、 $A(x,y)$ を、「 $x < y$ 」という論理式に対応付ける解釈において、偽となる。すなわち、恒真ではない。よって、証明可能でない。

8 意味論のまとめ

意味論～ 論理式があらわしているものの世界

- 領域
- 付値
- 項の意味
- 原子論理式の意味
- 論理式の意味

形式的体系と意味論の関係

- 意味論の概念: 充足可能, 充足不能, 恒真, 恒偽
- 形式体系の概念: 証明可能, 証明不能
- 健全性, 完全性

9 Prolog の続き

Prolog の動作:

- ゴールに含まれる原子論理式を、プログラム (ホーン節の集合) により、次々と書換えることにより、計算が進む。
- プログラム中のどのホーン節を使うか、「ホーン節のヘッド」と「ゴールの中の原子論理式」が「単一化可能かどうか」により定まる。(単一化可能であれば、書換え可能)。
- このように次々と書き換えて、ゴールが空になれば、成功して終了し、書き換え可能なホーン節がなくなったら (すべて試し終わったら) 失敗して、バックトラックする。

書き換え:

- ホーン節のヘッドとマッチした、(ゴールに含まれる) 原子論理式を、ホーン節の本体部分に置きかえること。

単一化:

- 変数を含む表現が2個以上与えられたとき、これらの表現に「変数への代入」を適用して、完全に一致させられるかどうかを調べる手続き。
- 単一化可能なとき、「最も一般的な代入」を返す。

バックトラック:

- プログラム中のホーン節のうち、複数のもので書換え可能なとき、それらを順番に試すのだが、これらは「どれか1つが成功すればよい」というものである。
- よって、1つ目が失敗しても、全体が失敗とは限らず、2つ目以降を試しに行く。

解:

- ゴールを解くことに成功したとき、最初にゴールに含まれていた変数がどのような値を取るか知りたい。
- Prolog は、途中ででてきた (単一化の答えとしての) 代入を集めており、その代入を全部適用することにより、成功したときの変数の値がわかる。
- バックトラックにより複数の (場合によっては無限個の) 解をもつことがある。

本日の演習

basic.pl のファイルを参考に、以下のプログラムを書きなさい。(ただし、 $0, s(0), s(s(0)), \dots$ であらわされる自然数の範囲とする。)

- 「X は Y の 2 倍であるか、3 倍であるか、どちらかである」ことを意味する関係 `foo(X,Y)`
- 「X を `s(Y)` で割った余りが Z である」ことを意味する関係 `rem(X,Y,Z)`

ヒント: いろいろなやりかたがある。basic.pl の `div` を参考にして直接実装してもよいし、`div` と `times` と `subtract` をつかって、`x - (x/y)*y` といった計算をしてもよい。

- (発展課題) list.pl の Quick Sort を参考に Merge Sort を実装せよ。