一階述語論理とそれを拡張した体系のおはなし

〜数理論理学の初歩から分離論理まで〜
Director's cut

Yukihiro Masuoka

NII, SOKENDAI 1st year, TATSUTA Lab

April 6, 2019

自己紹介

- 名前: 益岡 幸弘
- 所属: 総合研究大学院大学複合科学研究科情報学専攻
  - 情報学なにもわからない. いろいろあって迷い込んでしまった.
- 学年: D1(ただし, M1 相当)
- Twitter: YukihiroMasuoka @Yukihiro0036

今日の講演の方針

- あまりガッツリした数学の議論はやらない.
- 数理論理学の雰囲気を感じることが一番の目的.
    - 高校生にもわかるように書いてある (つもり)。
- モチベーションなどの話が主になるので, 細かい話は参考文献であげるものなどで勉強してください.

Outline

Outline

数理論理学とは?

- 数理論理学とは?
- ざっくり言うと形式化 (記号化) された論理・数学を '数学' を使って調べる分野
  - 異論は認める.

数理論理学とは?

- 少なくとも次のような分野ではない.
  - 理性の限界を示した.
  - 自然言語を使って議論すると怒られる.
  - 数学じゃない. 哲学である.
  - 事実上死に絶えた分野

数理論理学とは?

- 数理論理学の原初の問題意識 (の一つ)
- 数学の証明の構造を研究したい!!
- 数学の証明を文章に書き表したものは'記号列'なのだから数学的に扱えるはず.

立ちはだかる自然言語の壁

- 自然言語で書かれた証明をそのまま数学的に扱うの難しい.
  - 自然言語の良くないところ 例 1(Berry Paradox)
    - “19 文字以内で記述できない最小の自然数”
    - この文字列自体が 19 文字なので何を指し示しているのかわからん.
  - 自然言語の良くないところ 例 2(多義語がある)
    - Distribution
    - 超関数
    - (確率) 分布

立ちはだかる自然言語の壁

- 自然言語を直接扱うの難しい.
- 昔のすごい人 (たしかフレーゲ)
  そうだ. 数学を記述するのに特化した人工的な言語を作ろう.

$\Rightarrow$ 人工的な言語 (形式言語 Formal Language) を定義しよう.

- 何かに対する予防線
  - 歴史的な数理論理学の始まりはもう少し複雑です. 詳しくは歴史に詳しい人に聞いてください.

記号列 (形式言語) と意味

- ここからの話の前提：紙などに書かれた文字とそれが指し示すものは別
  - 犬 を欲しがる子供に “ 犬 ” と書かれた紙 をあげても喜ばれることはない.

記号列 (形式言語) と意味

- これから我々がやること その 1
  - 形式言語の ‘ 文法 ’ を決める.
    - Syntax を定める.
    - Syntax は 構文論 , “ 統語論 ” や “ 統辞論 ” などと訳される.

  ...... 字面はすごそう.
  - 数理論理学の文脈で構文論を定めるとは

    ‘ 記号 ’ の正しい並べかたを決める.

    ことを指す.

記号列 (形式言語) と意味

- これから我々がやること その 2
  - 形式言語の‘意味’を決める.
    - Semantics を定める.
    - Semantics は 意味論 などと訳される.

  ...... 字面はすごそう.
  - 数理論理学の文脈で 意味論 を定めるとは

    形式言語の‘正しい記号列’から
    ‘何らかの数学的対象’への‘関数’
    を決める.

    ことを指す.

証明体系

- これから我々のやること その 3
  - 定義された形式言語の上での ‘ 証明’ を定義する.
    - Proof System を与える.
    - Proof System は 証明体系 と訳される.

  ...... 字面はすごそう.
  - 証明体系 を定めるとは
    - 議論の前提となる記号列 (公理) を定める.
    - 新しい記号列を導くための規則を定めること.

  を指す.

証明体系

Note

‘普通の数学の証明’と“Proof System”とを区別したいとき, “Proof System”を特に Formal Proof System(形式的証明体系) などと呼ぶことがある.

論理学の体系

- ◯◯ Logic や ◯◯ 論理と呼ばれるものはだいたい
  - Syntax 構文論
  - Semantics 意味論
  - Proof System 証明体系

  の 3 つを持っている (※一部例外あり．上 2 つだけのものがあれば十分と感じる瞬間もある).

論理学の体系

- かなり乱暴な言い方だが,
  - Syntax 構文論 に最も焦点を当てて研究しているのが 形式言語理論 という分野
  - Semantics 意味論 に最も焦点を当てて研究しているのが モデル理論 という分野
  - Proof System 証明体系 に最も焦点を当てて研究しているのが 証明論 という分野

  である (異論は認める).

今回扱う体系

- First Order Logic (一階述語論理)
  - 基本的 (Basic) な論理の一つ.
  - 多くのこの体系を手本としている.
  - 少し知っている人向けの注意: 直観主義論理などについては今日は話したくありません.
- Separation Logic (分離論理)
  - 一階述語論理の拡張の一つ.
  - プログラム安全性の検証に使うことができる.

Outline

一階述語言語の記号

Definition 1 (一階述語言語の記号 Alphabet)

次の 4 種類の記号を **一階述語言語の記号** という.

- 変数記号 $x, y, z, \ldots, x_1, x_2, \ldots$
- 関数記号 $f, g, h, c, s, \ldots, f_1, f_2, \ldots$
- 述語記号 $P, Q, R, =, \ldots, P_1, P_2, \ldots$
- 論理記号 $\neg, \wedge, \vee, \rightarrow, \exists, \forall$

Remark

大学生以上の人々へ：言語の記号の集合の濃度は (技術的な面倒をさけるため) 高々可算だと思っていることが多いが, 非可算でもよい.

一階述語言語の記号

以降, Σ と書いたら一階述語言語の記号の集合を表す.
また, $\mathrm{Var}(\Sigma)$, $\mathrm{Func}(\Sigma)$, $\mathrm{Pred}(\Sigma)$ はそれぞれ一階述語言語の変数記号, 関数記号, 述語記号の集合を表す.

一階述語言語の記号

Definition 2 (arity)

$\mathbb{N}$ を自然数の集合とする[1].
関数 $\# : \mathrm{Func}(\Sigma) \cup \mathrm{Pred}(\Sigma) \to \mathbb{N}$ であって, 任意の述語記号 $\mathbf{P}$ について $\#(\mathbf{P}) \neq 0, \#(=) = 2$ を満すものを arity 関数と言う.
このとき, 記号 $\tau \in \mathrm{Func}(\Sigma) \cup \mathrm{Pred}(\Sigma)$ について $\#(\tau)$ を τ の arity と言う.

以降 arity 関数 $\#$ は固定されているものとする.

[1] 0 は自然数. 異論は認めない.

メタ記号

Remark

arity の定義中において $\mathbf{P}$ や τ は 一階述語言語の記号の上を走る‘記号’である.
そのような記号のことを**メタ記号** と言い, Σ などの考えている対象の記号 (オブジェクト記号) と区別する.
同様に, 一階述語言語の記号列の上を走る記号が, このあと出てくるが, これも**メタ記号**と呼ぶ.

メタ記号

Remark

arity の定義中に表れている “$\#(=) = 2$” の 1 つめの $=$ と 2 つめの $=$ は別物である.
前者は一階述語論理の記号の一つであるが, 後者は日本語で「$\#(=)$ と 2 は等しい」ということを略記するための ‘記号’ である.
これらの区別をしたいとき,

- $= \in \Sigma$ の $=$ を
  (一階述語) 言語の中のイコール,
- “$\#(=) = 2$” の 2 つめの $=$ などの ‘普通の数学’ の $=$ を
  日本語のイコール や メタなイコール

と呼ぶことにする.

一階述語言語の項

Definition 3 (項 Term)

以下のようにして構成されるものを **一階述語言語の項** あるいは単に **項** と言う.

i) $\mathrm{Var}(\Sigma)$ の元 $\mathbf{x}$ は **項** である.

ii) $\mathbf{f} \in \mathrm{Func}(\Sigma), \#(\mathbf{f}) = n$ とする.
n 個の記号列 $\mathbf{t}_1, \ldots, \mathbf{t}_n$ が項であるとき,

$$\mathbf{f}\mathbf{t}_1 \ldots \mathbf{t}_n$$

も **項** .

iii) 以上のように構成されるものだけが, **項**である.

一階述語言語の項

Remark

$c \in \mathrm{Func}(\Sigma), \#(c) = 0$ であるとき,

$$c$$

は項である.
上のような特殊性のため, arity が 0 の関数記号を**定数記号**と呼んで区別する流儀もある.

帰納的な定義

Remark

“ 項 ” の定義のように

- hoge は foo である (最低限入って欲しい要素を指定).
- foo をみたす hogehoge から hoo という方法で生成される fogefoge は foo である (構成方法を提示する).
- foo であるものは以上のようにして構成されるもののみである (構成される集合の最小性).

という形の定義を帰納的な定義 (Inductive Definition) と言う.

帰納的な定義

一階述語論理とそれを拡張した体系のおはなし

Yukihiro Masuoka

Remark

帰納的な定義 は数理論理学ではたくさん出てくる.
そのため, 誤解がおきないとき (最初に帰納的に定義するなどと宣言してあるときなど), 暗黙の了解で最後の 最小性 を表す文が省略されてしまっていることがある.
(追記 69 ページに帰納的定義の最小性についてもう少し詳しく書きました.)

項の例

Example (項の例)

次の記号列は項である (ただし, $\#(c) = 0$, $\#(f) = 1$, $\#(g) = 2$, $\#(h) = 3$).

$$fx, gxc, hgxcfyy$$

Example (項ではない例)

次の記号列は項ではない (ただし, $\#(c) = 0$, $\#(f) = 1$, $\#(g) = 2$, $\#(h) = 3$).

$$f, gx, hxfcgycz, Pfx, f\neg z$$

一階述語言語の論理式の定義

Definition 4 (原子論理式 atomic formula)

$\mathbf{P} \in \mathrm{Pred}\left(\Sigma\right), \#\left(\mathbf{P}\right) = n$ とする.
n 個の記号列 $\mathbf{t}_1, \ldots, \mathbf{t}_n$ が項であるとき,

$$\mathbf{P}\mathbf{t}_1 \ldots \mathbf{t}_n$$

を**一階述語言語の原子論理式** あるいは単に **原子論理式** と言う.

一階述語言語の論理式の定義

Definition 5 (論理式 formula)

以下のようにして帰納的に定義されるものを **一階述語言語の論理式** あるいは単に **論理式** と言う.

i) 原始論理式は **論理式** である.

ii) φ を論理式とする. このとき, $\neg\varphi$ は論理式である.

iii) φ, ψ を論理式とする. このとき, $\wedge\varphi\psi, \vee\varphi\psi, \to\varphi\psi$ はそれぞれ論理式である.

iv) φ を論理式とする. また, $\mathbf{x} \in \mathrm{Var}(\Sigma)$ とする. このとき, $\exists\mathbf{x}\varphi, \forall\mathbf{x}\varphi$ はそれぞれ論理式である.

一階述語言語の論理式の例

Example (論理式の例)

次の記号列は論理式である (ただし, $\#(c) = 0$, $\#(f) = 1$, $\#(g) = 2$, $\#(P) = 1$, $\#(Q) = 2$).

$$Px, = xfc, \neg = xx, \wedge = xxQfcgxc,$$

$$\exists x Px, \forall x \exists x Px, \forall y \to = yfxQyx$$

一階述語言語の論理式の例

Example (論理式ではない例)

次の記号列は項ではない (ただし, $\#(c) = 0$, $\#(c_1) = 0$, $\#(f) = 1$, $\#(s) = 1$, $\#(P) = 1$).

$$Pxy, Px \wedge Qxy, \forall P \to \wedge Pc \to Pc_1 Psc_1 \forall x Px$$

一階述語言語の論理式の略記

論理式の定義に前置記法 ($\to, \wedge, \vee, =$ などを一番前に書く記法) を採用した.
しかし, 人類には中置記法 ($\to, \wedge, \vee, =$ などを真ん中に書く記法) の方が見やすい.
中置記法は **略記** として定義する.

Definition 6 (論理式の略記)

φ, ψ, ϑ を論理式, $\star, \heartsuit$ を $\to, \wedge, \vee$ のいずれかとする.

i) $\varphi \star \psi$ は $\star\varphi\psi$ の略記である.
ii) $(\varphi \star \psi)\heartsuit\vartheta$ は $\heartsuit \star \varphi\psi\vartheta$ の略記である.
iii) $\varphi \star (\psi\heartsuit\vartheta)$ は $\star\varphi\heartsuit\psi\vartheta$ の略記である.

" $=$ " などに対する中置記法も同様に定義する.

一階述語言語の論理式の略記

Remark

前置記法を採用したのは“(”,“)”のことを考えたくないから.

Remark

前置記法 (ポーランド記法): $= a\ b$
中置記法: $a = b$
後置記法 (逆ポーランド記法): $a\ b =$

一階述語論理の構造 Structure

Definition 7 (構造 Structure)

一階述語論理の構造 (以下単に **構造**) とは次の条件を満たす二つ組 $(|\mathcal{M}|, \mathcal{M})$ のことである.

- $|\mathcal{M}|$ は空でない集合 (**対象領域**と呼ぶ).
- $\mathcal{M}$ は定義域が $\mathrm{Func}(\Sigma) \cup \mathrm{Pred}(\Sigma)$ の関数であり, 次の条件を満たす (**意味関数**と呼ぶことにする).
  - $\mathbf{f} \in \mathrm{Func}(\Sigma)$, $\#(\mathbf{f}) = n$ とする.
    このとき, $\mathcal{M}(\mathbf{f})$ は $|\mathcal{M}|^n \to |\mathcal{M}|$ という関数である.
    ただし, $n = 0$ のとき, $\mathcal{M}(\mathbf{f})$ は $|\mathcal{M}|$ の元である.
  - $\mathbf{P} \in \mathrm{Pred}(\Sigma) \setminus \{=\}$, $\#(\mathbf{P}) = n$ とする.
    このとき, $\mathcal{M}(\mathbf{P})$ は $|\mathcal{M}|^n$ の部分集合である.
  - $\mathcal{M}(=)$ は $\{(c, c) \,|\, c \in |\mathcal{M}|\}$ である.

一階述語論理の論理式の解釈

以下, “ 構造 $\mathcal{M}$ ” と書いたら “ $(|\mathcal{M}|, \mathcal{M})$ ” という構造を表すことにする.

Definition 8 (変数に対する値の割り当て valuation)

構造 $\mathcal{M}$ について関数 $\rho : \mathrm{Var}(\Sigma) \to |\mathcal{M}|$ を構造 $\mathcal{M}$ における変数に対する値の割り当てという.

一階述語論理の論理式の解釈

Definition 9 (項の解釈)

構造 $\mathcal{M}$, 構造 $\mathcal{M}$ における変数に対する値の割り当て ρ に対して項 $\mathbf{t}$ の解釈 $[\![\mathbf{t}]\!]_\rho$ を次のように帰納的に定義する.

I) $\mathbf{t}$ が変数記号 $\mathbf{x}$ であるとき,

$$[\![\mathbf{x}]\!]_\rho = \rho\left(\mathbf{x}\right).$$

II) $\mathbf{t}$ が $\mathbf{f}\mathbf{t}_1 \ldots \mathbf{t}_n$ という形 ($\mathbf{f}$ は arity n の関数記号, $\mathbf{t}_1, \ldots, \mathbf{t}_n$ は n 個の項) であるとき,

$$[\![\mathbf{f}\mathbf{t}_1 \ldots \mathbf{t}_n]\!]_\rho = \mathcal{M}\left(\mathbf{f}\right)\left([\![\mathbf{t}_1]\!], \ldots, [\![\mathbf{t}_n]\!]\right).$$

一階述語論理の論理式の解釈

構造 $\mathcal{M}$, 構造 $\mathcal{M}$ における変数に対する値の割り当て ρ, 論理式 φ について,

$$\mathcal{M}, \rho \models \varphi$$

(φ が $\mathcal{M}, \rho$ で成り立つ) という関係を次の I) – VIII) によって帰納的に定める.

一階述語論理の論理式の解釈

I) φ が原子論理式 $\mathbf{P}\mathbf{t}_1 \ldots \mathbf{t}_n$(ただし, $\mathbf{P}$ は $=$ ではない arity n の述語記号, $\mathbf{t_1}, \ldots, \mathbf{t_n}$ は n 個の項) という形であるとき,

$$\mathcal{M}, \rho \models \mathbf{P}\mathbf{t}_1 \ldots \mathbf{t}_n :\Longleftrightarrow (\llbracket \mathbf{t}_1 \rrbracket, \ldots, \llbracket \mathbf{t}_n \rrbracket) \in \mathcal{M}(P).$$

II) φ が原子論理式 $= \mathbf{t}_1\mathbf{t}_2$(ただし, $\mathbf{t}_1, \mathbf{t}_2$ は項) であるとき,

$$\mathcal{M}, \rho \models = \mathbf{t}_1\mathbf{t}_2 :\Longleftrightarrow \llbracket \mathbf{t}_1 \rrbracket = \llbracket \mathbf{t}_2 \rrbracket.$$

一階述語論理の論理式の解釈

III) φ が $\neg\psi$ という形のとき,

$$\mathcal{M}, \rho \models \neg\psi :\Longleftrightarrow \mathcal{M}, \rho \not\models \psi.$$

IV) φ が $\wedge\psi\vartheta$ という形のとき,

$$\mathcal{M}, \rho \models \wedge\psi\vartheta :\Longleftrightarrow \mathcal{M}, \rho \models \psi \text{ かつ } \mathcal{M}, \rho \models \vartheta.$$

V) φ が $\vee\psi\vartheta$ という形のとき,

$$\mathcal{M}, \rho \models \vee\psi\vartheta :\Longleftrightarrow \mathcal{M}, \rho \models \psi \text{ または } \mathcal{M}, \rho \models \vartheta.$$

VI) φ が $\rightarrow\psi\vartheta$ という形のとき,

$$\mathcal{M}, \rho \models \rightarrow\psi\vartheta :\Longleftrightarrow \mathcal{M}, \rho \not\models \psi \text{ または } \mathcal{M}, \rho \models \vartheta.$$

一階述語論理の論理式の解釈

VII) φ が $\exists x \psi$ という形のとき,

$$\mathcal{M}, \rho \models \exists \mathbf{x} \psi$$

$$:\Longleftrightarrow \text{ある } v \in |\mathcal{M}| \text{ に対して } \mathcal{M}, \rho\left[\mathbf{x} \mapsto v\right] \models \psi.$$

ただし,

$$\rho\left[x \mapsto v\right](\mathbf{y}) := \begin{cases} v & \text{if } \mathbf{y} = \mathbf{x}, \\ \rho(y) & \text{otherwise.} \end{cases}$$

一階述語論理の論理式の解釈

VIII) φ が $\forall x \psi$ という形のとき,

$$\mathcal{M}, \rho \models \forall \mathbf{x} \psi$$
$$:\Longleftrightarrow \text{すべての } v \in |\mathcal{M}| \text{ に対して}$$
$$\mathcal{M}, \rho[\mathbf{x} \mapsto v] \models \psi.$$

一階述語論理の論理式の解釈の具体例

Example

$|\mathcal{M}| = \{0, 1\}$ として, 関数 $+ : |\mathcal{M}|^2 \to |\mathcal{M}|$ を

$$+(x, y) = \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{otherwise.} \end{cases}$$

と定義する.
$\mathcal{M}(g) = +$ とすると, 任意の変数に対する値の割り当て ρ に対して

$$\mathcal{M}, \rho \models \forall y \exists x = gxy0$$

が成立する.

一階述語論理の証明体系を作ろう

さて, ここから証明体系を作っていくわけだが……
証明体系には大きく分けて 4 通りある.

- 仮定がとても多く, 新しい論理式を導く規則が一つしかない. (ヒルベルト流の体系 HK)
  - 数学的には準備が一番少なくて済む体系.
  - 人間の直感に反する証明を書かなければならないことがほとんど.
- 仮定がとても少なく (場合によっては 0) だが, 新しい論理式を導く規則がいっぱい. (自然演繹体系 NK)
  - 人間の直感に一番近い (ように見える).
  - 一番メジャーな体系 (だと思う).
  - 数学的にはちょっと定義が面倒.

一階述語論理の証明体系を作ろう

- 仮定も新しい論理式を導く規則もどちらも同じくらいある. (Shoenfield の体系など)
  - 最初の二つの間の子みたいな体系.
  - 最近はほとんど見かけない.
- “Sequent” というものを使う体系. (Sequent 計算 LK)
  - 証明論のための体系 (と言われることがある).
  - 性質がとても良い.
  - 玄人向けの体系なので理解するのに時間がかかる傾向がある.

今回は (さまざまな事情から) ヒルベルト流の体系 HK を扱おう.

おわび

すいません．準備が間に合わずこのあと，駆け足，かつお気持だらけになります．
(後日，内容を改良したスライドを HP 上で公開するかもしれません．しないかもしれません)

論理的公理

下記の形をした論理式すべてを論理的公理とよぶ.

- $\varphi \to (\psi \to \varphi)$
- $(\varphi \to (\psi \to X)) \to ((\varphi \to \psi) \to (\varphi \to X))$
- $(\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$
- $\varphi \to (\varphi \vee \psi)$
- $\psi \to (\varphi \vee \psi)$
- $(\varphi \to X) \to (\psi \to X) \to (\varphi \vee \psi \to X)$
- $(\varphi \wedge \psi) \to \varphi$
- $(\varphi \wedge \psi) \to \psi$
- $\varphi \to \psi \to (\varphi \wedge \psi)$
- $\psi \to \varphi \to (\varphi \wedge \psi)$

論理的公理

下記の形をした論理式すべてを論理的公理とよぶ.

- $\forall \mathbf{x}\,(\psi \to \varphi[\mathbf{y} := \mathbf{x}]) \to (\psi \to \forall \mathbf{x}\varphi)$
  - $\mathbf{y}$ は ψ と $\forall \mathbf{x}\varphi$ の自由変数ではない.
- $\forall \mathbf{x}\varphi \to \varphi\,[\mathbf{x} := \mathbf{t}]$
- $\varphi\,[\mathbf{x} := \mathbf{t}] \to \exists \mathbf{x}\varphi$
- $\forall \mathbf{y}\,(\varphi\,[\mathbf{y} := \mathbf{x}] \to \psi) \to (\exists \mathbf{y}\varphi \to \psi)$
  - $\mathbf{x}$ は ψ と $\exists \mathbf{y}\varphi$ の自由変数ではない.

証明

T を論理式の集合とする.
論理式の有限列

$$\varphi_0, \varphi_1, \ldots, \varphi_n$$

が以下の条件 i), ii), iii) のいずれかを満たしているとき, その列を T での論理式 φ_n の**証明**という.
各 $k \leq n$ に対して

i) φ_k は論理的公理であるか理論 T の元である.

ii) $i, j < k$ が存在して, φ_j は $\rightarrow \varphi_i \varphi_k$ という形である（M.P.）.

iii) φ_k が $\forall \mathbf{x} \psi$ という形で, i が存在して, φ_i は $\psi[\mathbf{x} := \mathbf{y}]$ という形である. ただし, $\mathbf{y}$ は $j \leq k, j \neq i$ なる φ_j に自由変数として含まれない.（Generalization).

Soundness

T を論理式の集合とする.
T から証明できるすべての論理式 ψ について, 構造 $\mathcal{M}$ において, すべての $\varphi \in T$ が

$$\mathcal{M} \models \varphi$$

ならば

$$\mathcal{M} \models \psi$$

である.

Completeness

T を論理式の集合とする.
任意の論理式 ψ についてすべての $\varphi \in T$ に対して

$$\mathcal{M} \models \varphi$$

となるような任意の $\mathcal{M}$ について

$$\mathcal{M} \models \psi$$

であるならば, ψ は T から証明できる.

Outline

分離論理の言語の記号

Definition 10 (分離論理の言語の記号 Alphabet)

次の 4 種類の記号を **分離論理の言語の記号** という.

- 変数記号 $x, y, z, \ldots, x_1, x_2, \ldots$
- 関数記号 nil
- 述語記号 $=, \mapsto, \mathsf{ls}, \mathsf{emp}$
- 論理記号 $\neg, \wedge, \vee, \rightarrow, \exists, \forall, *, -\!\!*$

分離論理の言語の記号

以降, Σ' と書いたら分離論理の言語の記号の集合を表す. また, $\mathrm{Var}(\Sigma')$, $\mathrm{Func}(\Sigma')$, $\mathrm{Pred}(\Sigma')$ はそれぞれ分離論理の言語の変数記号, 関数記号, 述語記号の集合を表す.

分離論理の言語の記号

Definition 11 (arity)

$\mathbb{N}$ を自然数の集合とする[2].
関数 $\# : \mathrm{Func}(\Sigma') \cup \mathrm{Pred}(\Sigma') \to \mathbb{N}$ であって,
$\#(=) = 2$ $\#(\mathsf{nil}) = 0$, $\#(\mapsto) = 2$, $\#(\mathsf{ls}) = 2$,
$\#(\mathsf{emp}) = 0$ を満すものを arity 関数と言う.
このとき, 記号 $\tau \in \mathrm{Func}(\Sigma') \cup \mathrm{Pred}(\Sigma')$ について
$\#(\tau)$ を τ の arity と言う.

以降 arity 関数 $\#$ は固定されているものとする.

[2] 0 は自然数. 異論は認めない.

分離論理の項

Definition 12 (項 Term)

分離論理の項 (あるいは単に **項**) は $\mathrm{Var}\,(\Sigma')$ の元と nil だけである.

分離論理の論理式の定義

Definition 13 (原子論理式 atomic formula)

$\mathbf{P} \in \mathrm{Pred}\left(\Sigma'\right), \#\left(\mathsf{f}\right) = n$ とする.
n 個の記号列 $\mathbf{t}_1, \ldots, \mathbf{t}_n$ が項であるとき,

$$\mathbf{P}\mathbf{t}_1 \ldots \mathbf{t}_n$$

を**分離論理の原子論理式** あるいは単に **原子論理式** と言う.

分離論理の論理式の定義

Definition 14 (論理式 formula)

以下のようにして帰納的に定義されるものを **分離論理の論理式** あるいは単に **論理式** と言う.

i) 原始論理式は **論理式** である.
ii) φ を論理式とする. このとき, $\neg\varphi$ は論理式である.
iii) φ, ψ を論理式とする. このとき, $\wedge\varphi\psi$, $\vee\varphi\psi$, $\rightarrow\varphi\psi$, $*\varphi\psi$, $-\!*\ \varphi\psi$ はそれぞれ論理式である.
iv) φ を論理式とする. また, $\mathbf{x} \in \mathrm{Var}\,(\Sigma')$ とする. このとき, $\exists\mathbf{x}\varphi, \forall\mathrm{x}\varphi$ はそれぞれ論理式である.

分離論理の構造 Structure

Definition 15 (構造 Structure)

分離論理の構造 (以下単に **構造**) とは次の条件を満たす 4 つ組 (V, L, s, h) のことである.

- V, L は空でない集合 (それぞれ Values, Location と呼ぶ).
- $L \subset V$.
- $\mathrm{nil} \in V \setminus L$.
- $s : \mathrm{Var}\,(\Sigma') \to V$.
- $h : L \overset{\mathrm{fin}}{\rightharpoonup} V$ (h は有限部分関数).

分離論理の論理式の解釈

構造 (V, L, s, h), 論理式 φ について,

$$s, h \models \varphi$$

という関係を次のように帰納的に定める.
以下 $[\![x]\!]_s = s\,(x)\,, [\![\textsf{nil}]\!]_s = \text{nil}$ とする.

$$
\begin{aligned}
s, h \models E_1 = E_2 &:\Longleftrightarrow [\![E_1]\!]_s = [\![E_2]\!]_s \\
s, h \models \neg P &:\Longleftrightarrow s, h \not\models P \\
s, h \models \wedge\psi\vartheta &:\Longleftrightarrow s, h \models \psi \text{ かつ } s, h \models \vartheta \\
s, h \models \vee\psi\vartheta &:\Longleftrightarrow s, h \models \psi \text{ または } s, h \models \vartheta. \\
s, h \models \rightarrow \psi\vartheta &:\Longleftrightarrow s, h \not\models \psi \text{ または } s, h \models \vartheta.
\end{aligned}
$$

分離論理の論理式の解釈

$$s, h \models \exists \mathsf{x} \psi$$
$$:\Longleftrightarrow \text{ある } v \in V \text{ に対して}$$
$$s\left[\mathsf{x} \mapsto v\right], h \models \psi.$$

$$s, h \models \forall \mathsf{x} \psi$$
$$:\Longleftrightarrow \text{すべての } v \in V \text{ に対して}$$
$$s\left[\mathsf{x} \mapsto v\right], h \models \psi.$$

分離論理の論理式の解釈

$$
\begin{aligned}
s, h \models \mathsf{emp} &:\Longleftrightarrow h = \emptyset \\
s, h \models \psi * \vartheta &:\Longleftrightarrow \text{there exists } h_1 \perp h_2 \text{ where} \\
&\qquad\qquad h = h_1 * h_2 \\
&\qquad\qquad \text{and } s, h_1 \models \psi \\
&\qquad\qquad \text{and } s, h_2 \models \vartheta \\
s, h \models \psi \mathrel{-\!\!*} \vartheta &:\Longleftrightarrow \text{ For all } h' \perp h \text{ where} \\
&\qquad\qquad s, h' \models \psi \text{ implies } s, h * h' \models \vartheta
\end{aligned}
$$

分離論理の論理式の解釈

$$s, h \models \mapsto E_1 E_2 E_3 :\Longleftrightarrow h = [\emptyset \,|\, [\![E_1]\!]_s \mapsto ([\![E_2]\!]_s, [\![E_3]\!]_s)]$$

$$s, h \models \mathsf{tree}\,(E_1) :\Longleftrightarrow s, h \models \mathsf{emp} \text{ または}$$

ある E_2, E_3 が存在して

$$s, h \models$$

$$\mapsto E_1 E_2 E_3 * \mathsf{tree}\,(E_2) * \mathsf{tree}\,(E_2)$$

Outline

BIBLIOGRAPHY1

- James Brotherston, “ An introduction to separation logic ”, Logic Summer School, ANU, 7 December 2015
- Kenneth Kunen, “The Foundations of Mathematics”, IfColog
- John C. Reynolds, “Separation Logic: A Logic for Shared Mutable Data Structure”, Proceedings of the 17th Anual IEEE Symposium on Logic in Computer Science
  - Separation Logic に興味を持ったかたはこちらをお読み下さい.
- Joseph R. Shoenfield, “Mathematical Logic”, CRC Press

BIBLIOGRAPHY2

- 鹿島亮『数理論理学』朝倉書店
- 田中一之他『ゲーデルと 20 世紀の論理学 (2) 完全性定理とモデル理論』東京大学出版会
- 戸次大介『数理論理学』東京大学出版会
- 細井勉『情報科学のための論理数学』日本評論社

Outline

5 Q and A

帰納的定義の最小性

講演中にも聞かれましたが, その後, 同じことを

- Q. 帰納的な定義の最小性を表す部分はいらないのではないですか......?
- A. 定義したい集合を一意にするために本質的に必要です.

詳しい例を次のスライドにあげます.

帰納的定義の最小性

たとえば, ‘ 項 ’ の定義の最小性を外してしまうと, たとえば,

i) $\mathrm{Var}(\Sigma) \cup \mathrm{pred}(\Sigma)$ の元 u は foo である.
ii) $\mathsf{f} \in \mathrm{Func}(\Sigma), \#(\mathsf{f}) = n$ とする.
n 個の記号列 $\mathsf{t}_1, \ldots, \mathsf{t}_n$ が foo であるとき,

$$\mathsf{f}\mathsf{t}_1 \ldots \mathsf{t}_n$$

も foo .

と構成した foo も ‘ 項 ’ の定義を満たしてしまいますので gxP のようなものも ‘ 項 ’ として認めても良くなってしまいます. ♠