定理証明支援系 Coq を用いた アクチュアリー数学の形式化

伊藤洋介

ＳＯＭＰＯひまわり生命保険株式会社

2021 年 11 月 5 日

免責事項

本発表内容は全て発表者個人の見解であり, 所属する組織としての見解を示すものではありません.

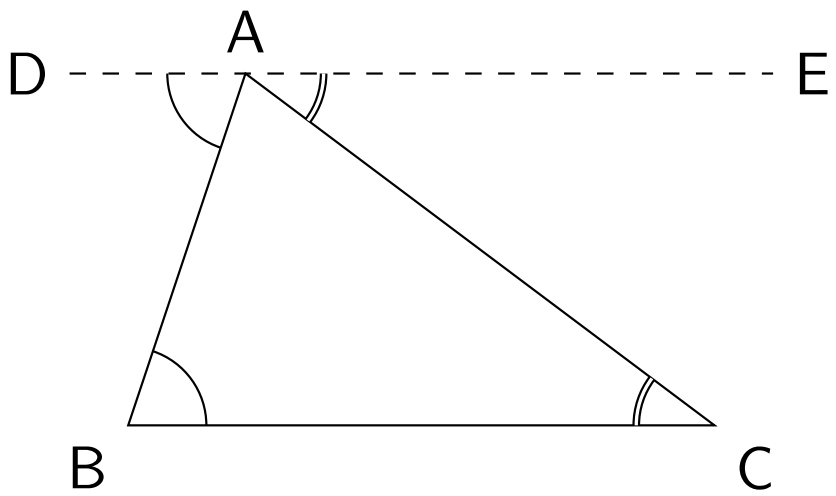
目次

目次

そもそも証明とは何か？

例

三角形の内角の和は $180°$ である.



証明. △ABC が与えられたとする.

点 A を通り, 直線 BC に平行な直線 DE を引く.

平行ならば錯角は等しいので, $\angle B = \angle BAD$, $\angle C = \angle CAE$.

すると, $\angle A + \angle B + \angle C = \angle BAC + \angle BAD + \angle CAE = 180°$. □

証明とは…

与えられた命題から, 論理的形式に頼って推論を重ね, 結論を導き出すこと.

証明の形式化

- 厳密な証明は, 直感に頼らない機械的な推論で記述できる.
- 主要な現代数学は, 記号論理を基礎として展開できる (形式化).
- 形式的証明は, “公理” と “推論規則” から成る.

  公理 理論の出発点となる仮定

  例 初等幾何学における平行線公理
  平面上で直線外の 1 点を通って, この直線と交わらない直線がただ 1 本存在する.

  推論規則 既に証明された命題 (あるいは公理) から別の命題を導く操作

  例 Modus Ponens

$$\frac{P \quad P \to Q}{Q}$$

- 形式的な証明はコンピュータ上に実装可能.
- 定理証明の検証・支援を行うソフトウェアを定理証明支援系と言う.

デモンストレーション (Coq)

定理 (Modus Ponens)

命題 A, B に対し, A かつ $A \to B$ ならば B が成り立つ.

```
Theorem mp : forall A B : Prop, A /\ (A -> B) -> B.
```

```
> Proof.
```

```
1 subgoal

  ============================
  forall A B : Prop, A /\ (A -> B) -> B
```

デモンストレーション (Coq)

定理 (Modus Ponens)

命題 A, B に対し, A かつ $A \to B$ ならば B が成り立つ.

```
Theorem mp : forall A B : Prop, A /\ (A -> B) -> B.
```

```
> intros A B.
```

```
1 subgoal

  A, B : Prop
  ============================
  A /\ (A -> B) -> B
```

デモンストレーション (Coq)

定理 (Modus Ponens)

命題 A, B に対し, A かつ $A \to B$ ならば B が成り立つ.

```
Theorem mp : forall A B : Prop, A /\ (A -> B) -> B.
```

```
> intro H.
```

```
1 subgoal

  A, B : Prop
  H : A /\ (A -> B)
  ============================
  B
```

デモンストレーション (Coq)

定理 (Modus Ponens)

命題 A, B に対し, A かつ $A \to B$ ならば B が成り立つ.

```
Theorem mp : forall A B : Prop, A /\ (A -> B) -> B.
```

```
> destruct H as [HA HAB].
```

```
1 subgoal

  A, B : Prop
  HA : A
  HAB : A -> B
  ============================
  B
```

デモンストレーション (Coq)

定理 (Modus Ponens)

命題 A, B に対し, A かつ $A \to B$ ならば B が成り立つ.

```
Theorem mp : forall A B : Prop, A /\ (A -> B) -> B.
```

```
> apply HAB.
```

```
1 subgoal

  A, B : Prop
  HA : A
  HAB : A -> B
  ============================
  A
```

デモンストレーション (Coq)

定理 (Modus Ponens)

命題 A, B に対し, A かつ $A \to B$ ならば B が成り立つ.

```
Theorem mp : forall A B : Prop, A /\ (A -> B) -> B.
```

```
> assumption.
```

```
No more subgoals.
```

デモンストレーション (Coq)

定理 (Modus Ponens)

命題 A, B に対し, A かつ $A \to B$ ならば B が成り立つ.

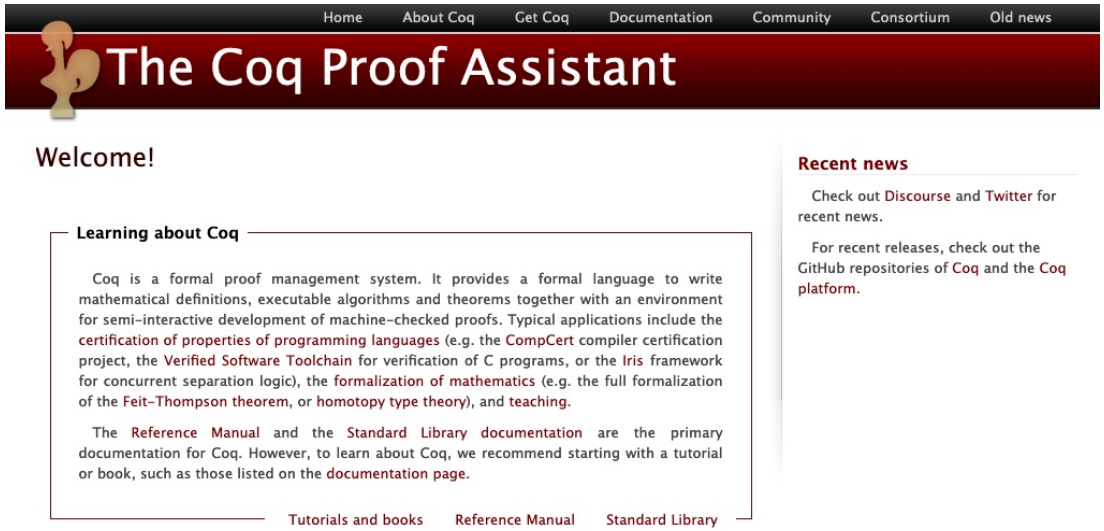
```
Theorem mp : forall A B : Prop, A /\ (A -> B) -> B.
```

```
> Qed.
```

定理証明支援系 Coq

Coq

- `https://coq.inria.fr/`



- 1980 年代〜. Thierry Coquand and Gérard Huet.
- ACM ソフトウェアシステム賞受賞 (2013 年).
- 日本語の入門書あり [5].

Coq の標準ライブラリ

- 標準ライブラリには形式的検証を経た定理が格納されている.
  - https://coq.inria.fr/library/

- 基礎的な数学は網羅されている.
  Arith 自然数に関する諸定理
  Reals 実数の定義・連続性・初等解析
  Sets 集合論
- その他, Coq ユーザにより多くの理論が形式化されている.
  - Coq Package Index (https://coq.inria.fr/packages.html)

Coq 標準ライブラリの中身

- Coq では数学の対象が基礎から組み上げられている.
  - 自然数の定義

```
Inductive nat : Set :=
  | O : nat
  | S : nat -> nat.
```

  - 加法の定義

```
Fixpoint add n m :=
  match n with
  | O => m
  | S p => S (p + m)
  end
where "n + m" := (add n m) : nat_scope.
```

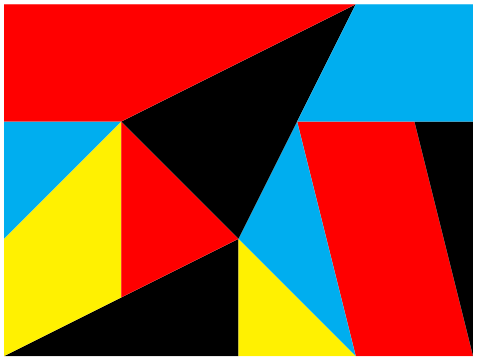
  - 加法の交換律

```
Theorem add_comm : forall n m, n + m == m + n.
```

定理証明支援系の応用例 (四色定理)

定理 (K. Appel and W. Haken, 1976 [1])

平面上の地図は *4* 色で塗り分けられる.



- 当初の証明はコンピュータプログラムを用いて膨大な場合分けを行うもの.
- プログラム自体にバグがないことの検証が困難.
- 2005 年に G. Gonthier が四色定理を Coq で形式化したことにより完全に検証された [3].

目次

形式的証明とは

- 論理記号

論理記号	意味
$\neg P$	P でない
$P \wedge Q$	P かつ Q
$P \vee Q$	P または Q
$P \to Q$	P ならば Q
$\forall x\, L(x)$	任意の x に対し $L(x)$
$\exists x\, L(x)$	ある x に対し $L(x)$

- 証明図

$$\dfrac{\dfrac{\dfrac{[P]^1 \qquad \dfrac{[(P \to Q) \wedge (Q \to R)]^2}{P \to Q}}{Q} \qquad \dfrac{[(P \to Q) \wedge (Q \to R)]^2}{Q \to R}}{R}\,1}{P \to R}\,2$$
$$(P \to Q) \wedge (Q \to R) \to (P \to R)$$

形式化の意義

- 無謬性の保証
  - 形式化された証明には論理的誤りが無い.
  - 学術誌に掲載された論文を形式化する際, 論理の飛躍が見つかることもある [4].
- システムの形式的検証
  - ソフトウェアの品質確保に応用可能.
  - 情報システム等のセキュリティ評価に関する国際規格 “コモンクライテリア”[7]

    EAL7 (形式的検証済み設計及びテスト) リスクが非常に高いか、高い資産価値により、さらに高い開発コストが正当化される場合に適用される。EAL7 は、EAL6 の保証に加え、数学的検証を伴う形式的表現と対応、広範囲のテストを使用する包括的分析を要求する。
- 自動推論の強化
  - 自動定理証明機とは, 目的の定理を与えることで自動的に形式的証明を生成するソフトウェア.
  - 形式化された数学は, 自動推論の質・スピードを上げるための機械学習にも応用される [8].

数学の形式化の進捗

- 形式化された証明の例 [6, p. 1372; 抜粋; 拙訳]

年	定理	定理証明支援系	実装者
1986	第一不完全性定理	Boyer-Moore	Shankar
1990	平方剰余の相互法則	Boyer-Moore	Russinoff
1996	微分積分学の基本定理	HOL Light	Harrison
2000	代数学の基本定理	Coq	Geuvers et al.
2004	四色定理	Coq	Gonthier
2004	素数定理	Isabelle	Avigad et al.
2005	Jordan 曲線定理	HOL Light	Hales
2007	留数定理	HOL Light	Harrison
2008	素数定理	HOL Light	Harrison

- 過去にアクチュアリー数学を形式化した例は無い.

目次

Coq の文法

- 論理式の記法

論理記号	Coq の記法
$\neg$	~
$\wedge$	/\
$\vee$	\/
$\rightarrow$	->
$\forall$	forall
$\exists$	exists

例 (ϵ-δ 論法)

$$\lim_{x \to a} f(x) = b \overset{\text{def}}{\iff} \forall \epsilon > 0, \exists \delta > 0, |x - a| < \delta \Rightarrow |f(x) - b| < \epsilon$$

```
Definition is_lim_R (f : R -> R) (a b : R) :=
  forall eps:R, 0 < eps -> exists delta:R, 0 < delta /\
    (forall x:R, Rabs (x-a) < delta -> Rabs (f x - b) < eps).
```

Actuary パッケージ

研究成果

Coq を用いて生命保険数学の基本的な公式を形式化し，一連の結果を Actuary パッケージとして公開した．

GitHub Yosuke-Ito-345
`https://github.com/Yosuke-Ito-345/Actuary`

- Actuary パッケージ (Version 2.0) の概要

ファイル名	SLOC	内容
`Basics.v`	1000	基本的な数学の補題群
`Interest.v`	794	金利の理論 (確定年金現価等)
`LifeTable.v`	827	生命関数の定義や諸性質
`Premium.v`	1863	保険料の計算式や公式群
`Reserve.v`	727	責任準備金の計算式や公式群
`all_Actuary.v`	5	上記のライブラリ全体
`Examples.v`	187	Actuary パッケージの応用例

アクチュアリー記号の実装

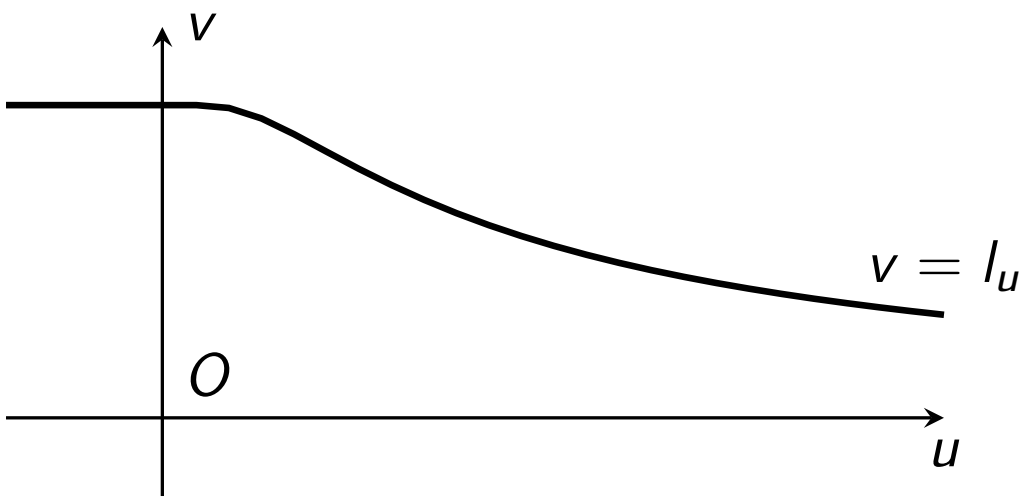
意味	アクチュアリー記号	Coq の記法
生存数	l_x	`\l_x`
死亡数	d_x	`\d_x`
生存確率	${}_tp_x$	`\p_{t & x}`
死亡確率	${}_{f\|t}q_x$	`\q_{f|t & x}`
死力	μ_x	`\μ_x`
定期保険の現価	$A^1_{x:\overline{n}\|}$	`\A_{x`1:n}`
生命年金現価	$\ddot{a}_{x:\overline{n}\|}$	`\a''_{x:n}`
定期保険の平準純保険料	$P^1_{x:\overline{n}\|}$	`\P_{x`1:n}`
定期保険の責任準備金	${}_tV^1_{x:\overline{n}\|}$	`\V_{t & x`1:n}`

生命関数の定義

定義 (生命関数)

実数上の実数値関数 l が生命関数であるとは, 次の 4 条件を全て満たすことを言う:

1. $l_0 > 0$,
2. 任意の負の実数 u に対し $l_u = l_0$ が成り立つ,
3. $\lim_{u\to\infty} l_u = 0$,
4. l は広義単調減少である.



生命関数の有限性・連続微分可能性

定義 (有限性)

1. 生命関数 l が有限であるとは, $l_x = 0$ となる自然数 x が存在することを言う.
2. 有限な生命関数 l に対し, $l_x = 0$ となる最小の自然数 x を l の最終年齢と呼び, ω と記す.

定義 (連続微分可能性)

生命関数 l が連続微分可能であるとは, 関数 l が $\mathbb{R}$ 上で連続微分可能であることを言う.

生命関数の形式的定義

```
Record life : Type := Life {
  l_fun :> R -> R;
  l_0_pos : 0 < l_fun 0;
  l_neg_nil : forall u:R, u <= 0 -> l_fun u = l_fun 0;
  l_infty_0 : is_lim l_fun p_infty 0;
  l_decr : decreasing l_fun
}.
Notation "\l[ l ]_ u" := (l_fun l u) (at level 9).
```

```
Definition ages_dead (l:life) : Ensemble nat := fun x:nat => \l[l]_x = 0.
Definition l_finite (l:life) := exists x:nat, (ages_dead l x).
Definition ult_age (l:life) (l_fin : l_finite l) := sval (leastN l_fin).
Notation "\ω [ l , l_fin ]" := (ult_age l l_fin) (at level 9).
```

諸公式の形式化

$$A^{(m)}_{x:\overline{n}|} = 1 - d\ddot{a}^{(m)}_{x:\overline{n}|}$$

```
Lemma ins_endow_ann_due : forall (m x n : nat),
  0 < m -> x < \ω ->
    \A^{m}_{x:n} = 1 - \d^{m} * \a''^{m}_{x:n}.
```

Fackler の再帰式 ${}_{t-1}V^1_{x:\overline{n}|} + P^1_{x:\overline{n}|} - vq_{x+t-1} = vp_{x+t-1} \cdot {}_tV^1_{x:\overline{n}|}$

```
Lemma res_term_rec : forall t x n : nat,
  0 < t <= n -> x+t < \ω ->
    \V_{(t-1) & x`1:n} + \P_{x`1:n} - \v * \q_(x+t-1) =
      \v * \p_(x+t-1) * \V_{t & x`1:n}.
```

$$\lim_{m\to\infty} A^{1\,(m)}_{x:\overline{n}|} = \bar{A}^1_{x:\overline{n}|}$$

```
Lemma lim_ins_term_cont : forall x n : nat,
  is_lim_seq (fun m:nat => \A^{m}_{x`1:n}) \Abar_{x`1:n}.
```

Actuary パッケージの応用例 I

生命年金現価が予定利率に関し広義単調減少であること

$i \leq i'$ ならば $\ddot{a}_{x:\overline{n}|}(i) \geq \ddot{a}_{x:\overline{n}|}(i')$

```
Lemma ann_due_decr_i : forall (i i' : R) (x n : nat),
  0 < i -> 0 < i' -> x < \ω -> i <= i' -> \a''[i']_{x:n} <= \a''[i]_{x:n}.
```

```
Proof.
  move => i i' x n Hipos Hi'pos Hx Hleii'.
  have Hvpos : 0 < \v[i] by apply /v_pos /Hipos.
  have Hv'pos : 0 < \v[i'] by apply /v_pos /Hi'pos.
  rewrite !ann_due_annual.
  apply Rsum_le_compat => k /andP; case => /leP Hmk /ltP Hkn.
  apply Rmult_le_compat_r; [by apply (p_nonneg _ l_fin) |].
  case: (zerop k) => [Hk0 | Hkpos].
  - rewrite Hk0 !Rpower_0 //; lra.
  - case: (Rle_lt_or_eq_dec i i') => // [Hlt | Heq].
    + rewrite /Rpower.
      apply /Rlt_le /exp_increasing.
      apply Rmult_lt_compat_l; [rewrite (_ : 0 = INR 0%N) //; apply lt_INR => // |].
      apply ln_increasing => //.
      rewrite /v_pres.
      apply Rinv_1_lt_contravar; lra.
    + rewrite Heq; lra.
Qed.
```

Actuary パッケージの応用例 II

将来法の責任準備金と過去法の責任準備金が一致すること

$$ {}_tV_{x:\overline{n}|} = P_{x:\overline{n}|}\ddot{s}_{x:\overline{t}|} - A^{1}_{x:\overline{t}|}/A_{x:\overline{t}|}^{1} $$

```
Theorem res_eq_pros_retro : forall (t x n : nat),
  x+t < \ω -> t <= n -> 0 < n ->
    \V_{t & x:n} = \P_{x:n} * \s''_{x:t} - \A_{x`1:t} / \A_{x:t`1}.
```

Thiele の微分方程式 $\frac{d}{dt}{}_t\bar{V}^{(\infty)}_{u:\overline{n}|} = \bar{P}^{(\infty)}_{u:\overline{n}|} - \mu_{u+t} + (\delta + \mu_{u+t}) \cdot {}_t\bar{V}^{(\infty)}_{u:\overline{n}|}$

```
Theorem Thiele_ODE : forall t u n : R, u+t < \ψ ->
  is_derive (fun t:R => \Vbar^{p_infty}_{t & u:n}) t
    (\Pbar^{p_infty}_{u:n} -
      \μ_(u+t) + (\δ + \μ_(u+t)) * \Vbar^{p_infty}_{t & u:n}).
```

アクチュアリー業務への応用可能性

- アクチュアリー試験出題ミスの防止
  - 問題文から解答を形式的に証明して検証する.
- 算出方法書の検証
  - 営業保険料や保険料積立金が商品仕様と整合していることを形式的に証明して検証する.
- 保険数理に関するプログラムの検証
  - 保険数理計算プログラムのソースコードにバグがないことを形式的に検証する.

今後の課題

- 形式化する範囲の拡大
  - 多重脱退
  - 連合生命
  - 計算基数
  - 営業保険料
  - ...
- 適用範囲の一般化
  - 生命関数が有限でない場合は扱えない.
  - $\mathbb{R}$ 上で連続微分可能という仮定は強過ぎる.
  - 広義積分が扱えない.
- 推論の自動化
  - 生命保険数学でよく使われる式変形は自動化したい.

目次

形式化された数学は何をもたらすか

- 定理証明支援系や自動定理証明機は発展の渦中にある.
- 照井一成 “コンピュータは数学者になれるのか？:
  数学基礎論から証明とプログラムの理論へ” [10, p. 343]

  > コンピュータは数学者になれるのか？　この問いは、コンピュータと人間の対決として捉えられるべきではない。むしろ**コンピュータと人間の協調**、その延長線上にある目標として捉えた方がよい。

アクチュアリーが貢献できる余地は大きい

- Coq の入門書・解説書
  1. Y. Bertot and P. Castéran.
     “Interactive Theorem Proving and Program Development:
     Coq’Art: The Calculus of Inductive Constructions” [2]
  2. B. C. Pierce et al. “Logical Foundations” (Software Foundations) [9]
  3. 萩原, アフェルト. “Coq/SSReflect/MathComp による定理証明:
     フリーソフトではじめる数学の形式化” [5]
- Coq コミュニティ
  - coq-club – The Coq mailing list (`coq-club@inria.fr`)
  - Zulip chat (`https://coq.zulipchat.com`)

謝辞

アクチュアリー数学の形式化にあたり, Coq を始めとする定理証明支援系の研究状況につき, 産業技術総合研究所の Reynald Affeldt 氏より貴重な助言をいただきました. 心より感謝申し上げます.

参考文献 I

[1] Kenneth Appel and Wolfgang Haken.
Every planar map is four colorable.
Bulletin of the American Mathematical Society, Vol. 82, No. 5, pp. 711–712, 1976.

[2] Yves Bertot and Pierre Castéran.
Interactive Theorem Proving and Program Development: Coq'Art: The Calculus of Inductive Constructions.
Springer-Verlag Berlin Heidelberg New York, 2004.

[3] Georges Gonthier.
A computer-checked proof of the four colour theorem.
http://www2.tcs.ifi.lmu.de/~abel/lehre/WS07-08/CAFR/4colproof.pdf, 2005.

[4] Sébastien Gouëzel and Vladimir Shchur.
A corrected quantitative version of the Morse lemma.
Journal of Functional Analysis, Vol. 277, No. 4, pp. 1258–1268, 2019.

[5] 萩原学, アフェルトレナルド.
Coq/SSReflect/MathComp による定理証明: フリーソフトではじめる数学の形式化.
森北出版, 2018.

[6] Thomas C. Hales.
Formal proof.
Notices of the American Mathematical Society, Vol. 55, No. 11, pp. 1370–1380, 2008.

参考文献 II

[7] 情報処理推進機構.
セキュリティ機能と保証レベル.
https://www.ipa.go.jp/security/jisec/forusers/abouteal.html, 2020.

[8] Cezary Kaliszyk and Josef Urban.
Learning-assisted automated reasoning with Flyspeck.
Journal of Automated Reasoning, Vol. 53, pp. 173–213, 2014.

[9] Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hriţcu, Vilhelm Sjöberg, and Brent Yorgey.
Logical foundations.
https://softwarefoundations.cis.upenn.edu/, 2021.
Version 6.1.

[10] 照井一成.
コンピュータは数学者になれるのか？: 数学基礎論から証明とプログラムの理論へ.
青土社, 2015.