

2020年度 数理論理学

講義資料(0)

青戸 等人 (知能情報システムプログラム)

その他の(英語の)参考書

定評のある参考書は前出の小野先生の本の最後を参照.

- 講義の内容に近い参考書: **Dirk van Dalen, Logic and Structure (3rd ed.), Springer-Verlag, 1991.**
- トピックも広く書き方も非常に優れた参考書(ただし, 数学書の読み方に慣れていないと難しい): **Herbert B. Enderton, A Mathematical Introduction to Logic (3rd ed.), Academic Press, 2001.**
- プログラムを使って説明している本(プログラム好きの人には良いかも): **John Harrison, Handbook of Practical Logic and Automated Reasoning, Cambridge University Press, 2009.**

4/9

目次

- 講義の情報
- 講義の概要

数理論理学 講義情報 (2)

講義ホームページ

<http://www.nue.ie.niigata-u.ac.jp/~aoto/lecture/Logic/>
(講義資料や連絡など. 随時更新予定)

オフィスアワー

Slack上で行います. 登録先をお知らせしますので, 希望者は参加してください.

成績評価の方法

- **期末試験(50%), レポート課題(50%)**で評価.
- 期末試験はオンラインで実施の予定
ノート・印刷物・参考書・PC等持ち込み可.
他人との相談は不可. 同じ部屋での受験も不可.

1/9

5/9

数理論理学 講義情報 (1)

講義の進め方

- スライド教材あり.
学務情報システムからの連絡に添付.
講義ホームページからもダウンロード可能.
- 講義時間に講義ビデオを **Youtube**で視聴してください.
学務情報システムで **URL**を前日~当日に連絡.
- 講義のときに用意するもの.
スライド教材, ノート, 筆記具
講義の途中に演習問題を解く必要あり.
- 演習問題および解答あり.
講義と間をおかずに復習すること.
いくつかの演習問題をレポートに課す.

2/9

講義の復習とレポート

- **復習**: 毎回の講義はそれ以前の講義内容の理解を前提とします. 講義について行くには, **毎回, 講義内容を余さず消化することが重要です**. 不明な点は復習し, 演習問題を解いて理解を確実にしておくこと.
- 講義のホームページに, スライド, 演習問題, 演習問題の解答を掲載します. **解答のチェックは, 自分でやってください**. 講師の方では採点しません.
- **レポート**: いくつか問題を選んで, レポート問題にします. 学務情報システムでレポート課題を出しますので, 期限までに, 学務情報システムから, 丸付け済みの解答の写真をアップロードしてください.

6/9

教科書・参考書

- 教科書: なし
- 参考書:
前原昭二著「記号論理入門」(日本評論社)
小野寛晰著「情報科学における論理」(日本評論社)

目次

- 講義の情報
- 講義の概要

数理論理学 講義概要

講義のねらい

数理論理学は、情報分野全般における数学的・理論的な思考技術の基本として、更には、ソフトウェア科学/計算論/人工知能分野等における基礎として重要である。このような観点から、数理論理学の基礎を修得することを目的とする。

講義の概要

数理論理学の基本である命題論理および述語論理について、構文論と意味論を紹介する。具体的には,... などについて講義する。

6/9

数理論理学 講義計画

- 命題論理の意味論
- 命題論理式の同値変形
- 帰納的定義
- 命題論理における自然演繹法
- 命題論理における健全性と完全性
- 述語論理式の同値変形
- 述語論理における自然演繹法
- 形式的証明と数学的論証の対応
- 第一階述語論理の意味論
- 述語論理における健全性と完全性

最も応用範囲が広く、最も基本的な数理論理学の基礎を身につける。

7/9

(補足) 論理学は先端的な研究で応用されていないの？

いくつかの主要な応用を挙げる：

(1) 対話的証明器

数学の証明を形式的に記述するソフトウェア。今まで人間の手で書かれていた証明では大幅に省略されるが、これを計算機上で厳密に(プログラムのように)記述。バグのない証明が可能になるばかりでなく、推論システムの出力検証やユーザー本位の定理証明への応用など。

(2) SAT/SMTソルバー

計算困難な論理式の充足可能性を高速に解くソフトウェア。高速に解けるとは限らないが、実は、人間が解きたい問題はかなり高速に解けることが多い。さまざまな問題を

論理式等でエンコードして、SAT/SMTソルバーを用いることで、解を高速に発見することができる。

(3) モデルチェッカー

ソフトウェアのバグを見つけるのにさまざまな入力を試すのではなく、どのように動作して欲しいか(仕様)を論理を用いて記述し、プログラムの入力や実行を抽象化することでモデルを構築し、モデルが仕様を満たすかを検査することで、あらゆる入力や動作で(その仕様について)問題がないかどうかを判定する。

情報 + 論理 を背景としたアプローチは、**Logic in Computer Science** とよばれており、情報科学/情報工学の基本的な分野の重要な柱の1つ。

9/9