形式仕様記述シリーズ

シリーズリーダー

国立情報学研究所
石川　冬樹

講師

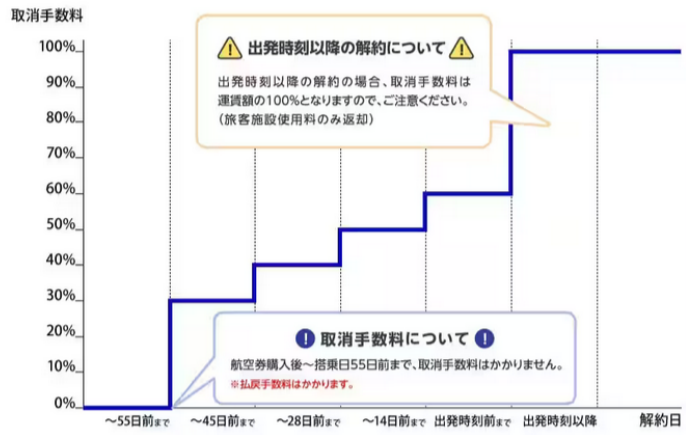
中央大学　松崎　和賢
宇宙航空研究開発機構　小林　努
株式会社 proof ninja　今井　宜洋

仕様(論理・ルール)って難しいですよね！

取消手数料について

ANA SUPER VALUE PREMIUM 28・ANA SUPER VALUE 75・ANA SUPER VALUE 55・ANA SUPER VALUE 45・ANA SUPER VALUE 28・ANA SUPER VALUE 21・ANA SUPER VALUE SALE・ANA SUPER VALUE TRANSIT 75・ANA SUPER VALUE TRANSIT 55・ANA SUPER VALUE TRANSIT 45・ANA SUPER VALUE TRANSIT 28・ANA SUPER VALUE TRANSIT 21

取消手数料イメージ



変更・払い戻しの特例

お客様が病気などの理由で旅行不可能な場合、次のいずれかの特別対応をいたします。

・予約便出発予定日から30日間以内の便への変更を承ります。

・払戻手数料・取消手数料を適用せず、全額払い戻しいたします。

＊ 予約の変更・払い戻しの特例は、ご予約便の出発前かつ払い戻し前にご連絡いただいた場合に限ります。
すでに払い戻し済みのご予約は、対象外となりますのでご注意ください。

＊ 病気などの理由で変更・払い戻しをする場合は、医師の診断書等の提出が必要です。
書面には、当初の旅程で搭乗ができない状況であることの明記が必要です。

お手続きは、**ANA国内線予約・案内センター**、ANA国内線空港カウンターにて承ります。
特典航空券は、**ANA国内線予約・案内センター**でのみ承ります。
なお、インフルエンザ（または新型コロナウィルス感染症）に感染した場合の払い戻しは、お問い合わせフォームから手続きが可能です。**【国内線】インフルエンザ（または新型コロナウィルス感染症）にかかってしまい搭乗できません。**
をご確認ください。

不整合はない？

あいまいさや状況の抜けはない？

上位にある意図・目的・制約を遵守している？

[https://www.ana.co.jp/ja/jp/guide/plan/fare/domestic/charge/]
（2024/11/16アクセス時）

扱う技術アプローチ

- 厳密な記述を用い強力な検証技術も活用
  - 仕様や設計のあいまいさに起因する問題を排除
  - 上流工程での問題検出による効率化・高品質化
  - プログラムに対する強力な検査も

記述イメージ

```
class イベント参加登録管理システム
  登録済みユーザ集合 : set of「ユーザ識別子」;
  定員 : nat1;
  inv card 登録済みユーザ集合 <= 定員

  抽選登録する : set of「ユーザ識別子」==>「ユーザ識別子」
    抽選登録する(引数ユーザ集合) == is not yet specified
    pre
      card 登録済みユーザ集合 < 定員
      and exists ユーザ in set 引数ユーザ集合 & ユーザ not in set 登録済みユーザ集合
    post
      登録済みユーザ集合 = 登録済みユーザ集合~ union {RESULT}
      and RESULT in set 引数ユーザ集合 and RESULT not in set 登録済みユーザ集合~;
```

技術の利用イメージ（一例）

実装詳細を捨象

重要な制約を明記
（これからできるコードは
何を守るべきなのか）

論理に基づく
＝見落としがない
などの保証付き

厳密化

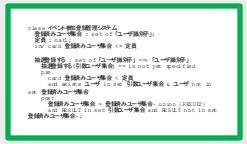
自然言語による
仕様書や設計書

フィードバック

```
class イベント参加登録管理システム
  登録済みユーザ集合 : set of「ユーザ識別子」;
  定員 : nat1;
  inv card 登録済みユーザ集合 <= 定員

  抽選登録する : set of「ユーザ識別子」==>「ユーザ識別子」
    抽選登録する(引数ユーザ集合) == is not yet specified
    pre
      card 登録済みユーザ集合 < 定員
      and exists ユーザ in set 引数ユーザ集合 & ユーザ not in set 登録済みユーザ集合
    post
      登録済みユーザ集合 = 登録済みユーザ集合~ union {RESULT}
      and RESULT in set 引数ユーザ集合 and RESULT not in set 登録済みユーザ集合~;
```

上流の記述との
整合性確認も



様々な検査
- テスト
- 定理証明
- 具体例生成による
  妥当性確認なども

※ 自然言語文書は
「概要」程度と位置づけ，
厳密な記述を中心として
進めている実例もある

ここの対象はプログラムでもよい

技術の利用イメージ：有名な産業界事例

- FeliCaチップ：外部仕様を厳密に記述，テスト
  - 多数の実装者や外部パートナーが活用する対象
  - 後工程で検出された不具合のうち，記述の問題に起因するものはゼロに
- パリ地下鉄や空港シャトル，世界各国に展開：正しさが保証されたコードを仕様から導出
  - 重要な部品について，数学的証明を通して仕様遵守が保証された形で段階的に詳細化
  - 高信頼プロセス：仕様の議論・定義に大半の工数，「正しさを維持し実装へ」，単体テストは不要

受講生のこれまでの試み

- トップエスイーでのこれまでの演習・制作
  - ビジネスプロセス，チップ処理ワークフロー，クラウドストレージ管理，アクセス制御，など多様な領域での試行
  - 海外外注先への仕様書記述（日本語問題回避）
  - 仕様書・設計書に関する定形化・機械処理（ルールベースのテスト仕様導出自動化など）
  - 鉄道運賃規定やTDLファストパス使用ルールの記述と分析

扱う技術

強力な推論や検査・保証

厳密な記述

論理の基礎

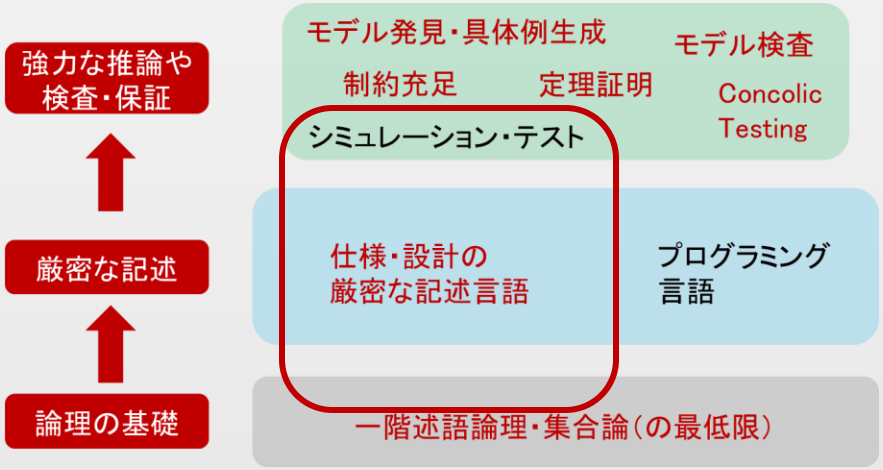
モデル発見・具体例生成　モデル検査
制約充足　定理証明　Concolic Testing
シミュレーション・テスト

仕様・設計の厳密な記述言語　プログラミング言語

一階述語論理・集合論（の最低限）

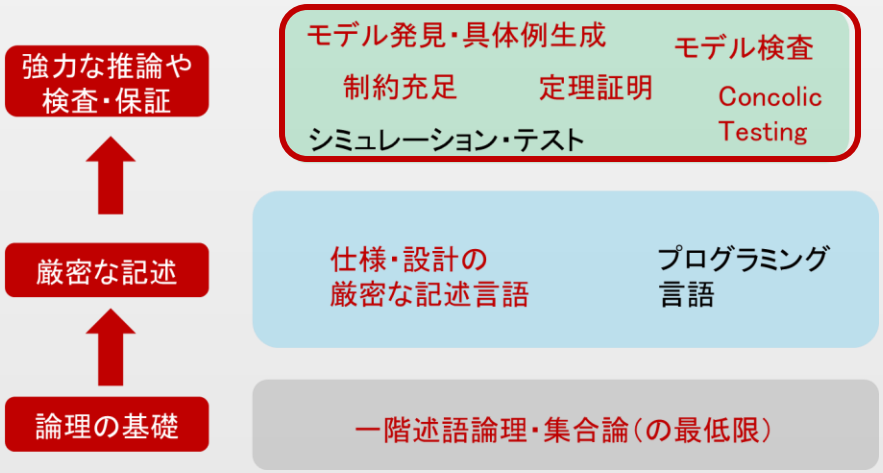
講義1：形式仕様記述入門

- 厳密な記述言語の活用に焦点
  - あいまいさの排除や，記述漏れや不整合の検出
  - 簡単な検証としてシミュレーションテストを扱う（仕様や設計を「動かしてみる」）
  - それに限らず，様々な手法・ツールを俯瞰する



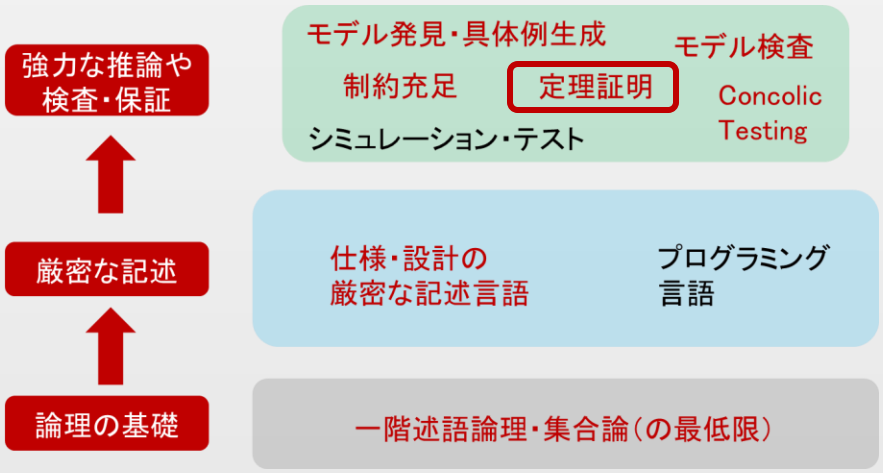
講義2：論理に基づくモデル・プログラム解析

- 推論・検証の技術に焦点
  - 制約ソルバーの活用するなど，論理に基づくことで，一定の保証を与えるような技術を扱う
  - 様々な種類のツールを俯瞰し，利点と限界を知る
  - 講義1を受けると理解が深まるが，必須ではない



講義3：高信頼ソフトウェアのための仕様記述と定理証明の活用

- 定理証明による高信頼性のための技術に焦点
  - 複雑なシステムを段階的に扱うモデリング
  - 定理証明を対話的に行ったり，「正しいとわかっている」プログラムを生成したりするツールの活用
  - 講義1の発展



形式仕様記述シリーズ

- （主に機能）仕様に関する下記アプローチを習得
  - 抽象モデル化と “What”の厳密な記述
  - 正しさの検証・保証
- 「形式仕様記述」という技術活用への第一歩
- 様々な課題に対して幅広く活きる原則・基盤
  - モデルの抽象度制御（WhatとHowの分離）
  - あいまいさへの対処
  - 正しさの定義や種類，様々な検証アプローチ
- LLMを補いうる「保証付き」の技術も知る