2007
言語モデル論－1（序論）

米澤明憲
コンピュータ科学専攻
情報理工学系研究科

2007. 10. 1

スケジュール等

- 講義
  - 「コンパイラ講義・演習」と対
  - 合計12、3回の予定
- 試験　2月はじめ　closed book　１．５～2時間
- レポート:
  - 試験の出来が悪いときなどボーナス
  - 講義中に課題等の形あるいは自分で考えた課題

記号とモデル

- 記号と言語
- 言語と記述
- 記述とモデル
- 言語と意味

言語とは？

- 「はじめに言葉(logos)ありき」
  - 名前のないものに名前を付けると概念ができて、その存在が始まる、e.g., 新しい分野など
  - 「ギリシャ語のlogos」とは制度・法律・約束ごと„
- 言語とは何かと問われると、普通は
  - コミュニケーションの手段
  - 表現の手段
  - 考え・思考を進める手段
  - 概念体系・文化（の記述）
  - „

言語の分類

1. 自然言語 (自然に存在する、発生した)
   - 日本語、英語、中国語、ヒエログリフィクス、…
2. 人工言語
   - エスペラント
   - 手話
   - 形式言語(正規言語、文脈自由言語、…)
   - プログラミング言語
   - 述語論理
   - ・・・・
   - (数学)

「古典的言語学」における３つアプローチ

ここでは以下、対象は記号・文字の並びを想定。

1. 統語論（構文論、Syntax)

- 文中の語（記号）と語（記号）の関係を論じる。
- 文面上の正しい語の並びが中心課題（「意味」でなく「文法」）
- 非文例：“には東大生あるもつ創造性を者も”

ー “syntacticには” とは、文字列・記号列としてのみ考えると。。。

2. 意味論(Semantics)

- 文中の語（記号）とその語が指し示す(denote、意味するもの）との関係を論じる。
- 非文例：“無色の緑が憤激して眠っている。”Colorless green sleeps furiously.”– N. Chomsky: “Syntactic Structures” Mouton (1956)
- （F. ソシュール）「文面そのものが意味・意義」という考えもある。

３．　語用論(pragmatics, semiotics)

- 語（記号）と、語（記号）が指示するものと、語（記号）を使うもの、との関係を論じる。
  - 敬語の使用様式
  - 間接的表現
  - ．．．
- 言語行為(Speech Acts)
  - **denote**(指示）するという関係とは捕らえ難いものがある
  - 例えば、仕事をさせるロボットに話しかける言語を考えると。。。
  - 命令、要請、忠告、約束
- 非文例：　花子が太郎が一郎が地球は小さいと思っていないと思っていると思っていない

課題：

命令・要請・約束をどう数学的に表現するか考えよ。

プログラム言語論の3つのアプローチ

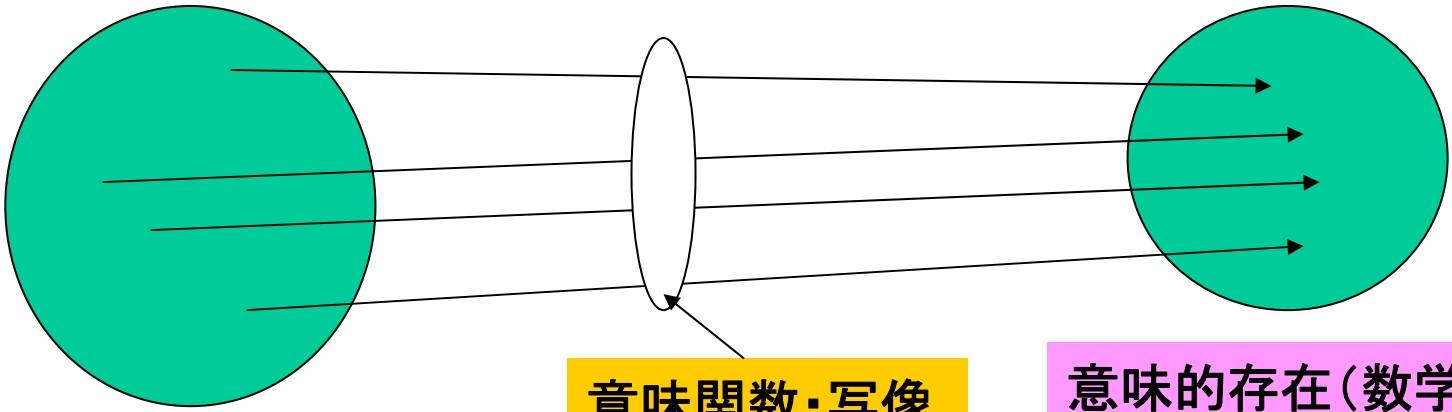
1. 統語論／構文論

重要な問題は構文の指定法：

- BNF，構文チャート、正規表現、CSG、…
- 受理型オートマトン(acceptor automaton),
  構文解析器(parsers)、…

2.（表示的）意味論とは

- プログラム、変数、名前、値、等の文面(text)の世界から、それらが指示(denote)する意味世界の存在要素への「写像」を「意味」と看做す。
- 意味を与えるとはこの写像を定義すること。



プログラムの意味の与え方

1. 操作的意味論

プログラムの評価・実行の方法を厳密に記述する。

- Definitional Interpreterや仮想機械を指定する
- 指定の方法に色々ある。例えば、
  - Schemeプログラムで与える
  - 推論規則・評価規則群で与える

$$\frac{E \mid\!- e1: \mathbf{n1},\ E \mid\!- e2: \mathbf{n2}}{E \mid\!- e1+e2: \mathbf{n1+n2}}$$

2. 公理的意味論

- Hoare論理系 -- P{Q}R 実行前・実行後の状態

3. 表示的意味論(denotational semantics)

意味関数と意味空間

- 各命令の「意味」を状態から状態への関数と仮定する
- プログラムは命令の組み合わせと考える
- 意味関数／空間が整合的に構成できるためには：
  1. 命令の「逐次実行」は関数合成に対応させられる。
  2. 命令の「無限の繰返し実行」は関数に対応させられる。
  3. 命令を適用する対象（即ちデータ）も命令である場合で、その命令も関数に対応させられる。
- この要件を満たす関数空間をD.Scottが初めて構成した。
  （1978年にTuring賞受賞）

算法・プログラムの表現方式

1. 作用型／関数型(applicative/functional)
2. 命令型／手続き型／ノイマン型
   (imperative/von-Neumann style)
3. 論理型(logic)
4. オブジェクト指向(object-oriented)

命令型表現

- ノイマン型計算機

  その2大構成要素（I/O部を除く）

  1. 記憶部＝番地付けされた語の並び
  2. 演算部＝ “命令読出ー＞解釈ー＞実行”の繰り返す

  – 命令の実行＝ 1. 語の内容の読み・書き（更新）
  2. 算術・論理演算
  3. 分岐・条件付分岐

- 計算機の実行
  – 命令の実行による「状態」の遷移の繰り返し
  – 「状態」とは記憶部のその時点おける記憶内容
  – 一回の遷移で巨大な記憶内容の一部のみが逐次的更新

命令型言語・手続き型言語

- 言語における「変数」が
  ノイマン型計算機の「語」に直接的に対応する。
- 文(statement)と式(expression)の２大要素の世界
  「文」 ＝ 1. 記憶（語の中味）の更新
  2. 制御の流れを表現
  「式」 ＝ 1. 記憶（語の中味）を読みだす操作
  or その結果
  2. 評価操作（計算方法）orその結果
- 文の世界 ＝ 記憶／状態を変化させる(stateful)世界
- 式の世界 ＝ 値をあらわす、
  記憶／状態のない(stateless)世界

記憶／状態のないシステム

- 同じ入力に対して常に同じ出力が対応する
  or そのように常に外部から観測(observe)される!
- 数学的関数は同じ引数に対して常の同じ値
  => 数学的関数は記憶のないシステムの例

今後の講義内容予定

1. 形式的計算モデル
   - チューリング機械（省略）、項書き換えシステム（省略）
   - 計算可能な関数： 原始帰納法的関数、帰納的関数、Kleeneの定理、smn定理（部分評価の基礎）、決定可能集合、枚挙可能集合、帰納的集合、計算不可能、 対角線論法、Churchの提唱
   - λ計算 ー 直感説明、プログラミング言語の基礎概念、定義、Normal forms、コーディング、Yーコンビネータ、Boehm木、評価順序、合流性定理

２．作用型プログラミング

- 一般的性質 ー referential transpaency,,,
- 5つの並列性表現
  1. parallel evaluation/fork-join
  2. futures
  3. lazy evaluation and infinite data-stream
  4. network of prcesses
  5. data flow and Petri-nets

３．プログラミング言語の意味論

- 操作的意味論 i. Definitional Interpreter
  ii. Natural semantics Notation
- 公理的意味論 Hoare Logic
- 表示的意味論 continuation概念、連続関数に合成。。。

４．　型理論の初歩

- 型とは、型の歴史
- 抽象データ型　－　CLUの例
- 型多相
- 型システム
  - simply-typed λ計算
  - 単相型システム
  - μMLの型（型多相システム）
  - 型推論アルゴリズム

以上

参考書・文献

- The Formal Semantics of Programming Languages, G. Winskel, MIT Press, 1997.
- Concepts in Programming Languages, J. Mitchell, Cambridge Univ. Press, 2003
- Types and Programming Languages, B.C. Pierce MIT Press, 2002.
- 「計算論--計算可能性とラムダ計算」 高橋正子著　近代科学社 （1991）
- 「モデルと表現」岩波講座ソフトウエア科学第17巻　米澤・柴山（1992）

計算モデル、ソフトウエアのモデル

情報の世界
（対物理世界）

0. 「ビット（記号・文字）」と「アトム」（N.ネグロポンテ)
1. 情報の世界はデジタル／離散的である。
   – 永久に動き続けるプログラムのモデル化には連続（アナログ）な数学的な概念が必要。(D∞モデル）
2. **バーチャルである。**
   – 「実際には存在しない」という意味
     or「実質的に存在するも同然」ということ。
   – 自然科学が、たった一つの客観的現実世界を相手にしている、一方コンピュータサイエンスでは、プログラムとデータとして表現・実行・モデル化できる世界を、物理的真の世界とは無関係に表現できる。

3. 記号による記述の世界である。

４　正確な記述は記述対象のモデルを前提にしている

5.　記述において自己参照が可能である。

ある意味で論理的に矛盾しても手続き・アルゴリズム・プログラムの実行においては、可能

モデルの階層

- モデルは階層をなす。
  - 自然科学と同じように、普遍性の高いミクロモデルから問題領域に依存したマクロなモデルへの階層がある。
    1. プログラミングパラダイムのモデル
    2. 個別のソフトウエアのモデル
    3. 計算のモデル