

計算モデル特論

国立情報学研究所 ソフトウェア研究系 /
科学技術振興事業団 さきがけ研究21

佐藤一郎

E-mail: ichiro@nii.ac.jp

Ichiro Satoh

▶ 講義内容

- 抽象機械モデル
 - チューリングマシン
- 関数型計算モデル
 - 型なしラムダ計算
 - ラムダ計算の性質
 - 型付きラムダ計算
- 並行計算モデル(プロセス代数)
 - 抽象並列機械モデル
 - CCS (Calculus for Communicating Systems)
 - π -calculus

Ichiro Satoh

▶ 計算モデル

計算モデルの必要性

- 世の中の情報処理システムは非常に複雑
- この情報処理システムがどのような原理に基づくかを知るには数学的基盤を持つ**抽象的な枠組み**を導入する必要がある。

情報処理システムの抽象的な枠組み = 計算モデル

計算モデルを通じて「計算」という概念を明確化する

Ichiro Satoh

▶ 履修者への質問

講義内容とレベルは履修者の背景知識と関心により決める

履修者の背景知識に関する質問

- 集合または代数論の講義の履修(有/無)
- オートマトンの講義の履修(有/無)
- 一階述語論理の講義の履修(有/無)
- 計算モデルに関する講義の履修(有/無)

卒論または修論テーマは？

Ichiro Satoh

▶ 教科書 & 参考書

ラムダ計算

- 大畑淳著: “プログラミング言語の基礎理論”, 共立出版
- 井田哲雄: “計算モデルの基礎理論”, 岩波書店

プロセス代数

- R.Milner: “Communication and Concurrency”, Prentice Hall
- R.Milner: “ π -calculus”

Ichiro Satoh

▶ 講義形態

評価: レポート(数回)

休講日: 10/22, 11/12, 12/3,

Ichiro Satoh

▶ 講義の目標

- 各種の計算モデルについて学ぶこと
- 計算の数学的な取り扱いに慣れること
- 計算モデルを通じて計算という概念を理解すること
- ものの原理を見抜くセンスを養うこと

Ichiro Satoh

▶ 「計算」とは

「計算」とは、

ある「アルゴリズム」に基づいて、ある基本的な手順を実行し目的とする結果を得る作業

しかし、実際の情報処理システムは複雑であり、本来の基本原則である「計算」が不明確

- 数学的な裏付けをもつ抽象的な枠組み (= 計算モデル) を導入し、
- 記号の書き換えとして計算を再現する

Ichiro Satoh

▶ 計算モデルの必要性

- 計算機のハードウェア
複雑すぎて扱えない
- CやPascalなどで書かれたプログラム
数学的な厳密さを持つとは限らない
- プログラムを入力から出力を得るものとする
例: 数学的な関数
ただし、実際に入力を与えても、その答え(出力)が導けるとは限らない

Ichiro Satoh

▶ 代表的な計算モデル

(逐次)計算モデル:

- 抽象機械計算モデル
- 関数型計算モデル
- 論理型計算モデル
- 項書き換え型計算モデル

(並列分散)計算モデル

- 抽象機械計算モデル
- プロセス代数(プロセスカルキュラス)
- イベント機械モデル(ペトリネット)

Ichiro Satoh

▶ チューリング機械

A. M. チューリングが提案した仮想機械

すべてのプログラム内蔵計算機を模倣する抽象機械

- 基本的な動作(特定の作業)
 - テープの読み書き
 - テープの送り、戻し
 - 内部状態の変更
- 記憶領域
 - 内部記憶
 - テープ(外部記憶)

Ichiro Satoh

▶ チューリング機械

定義: 6項組($Q, \Sigma, \Gamma, \delta, m, q_0$)

ここで、 Q : 状態集合、 Σ : テープ上の記号の集合、 q_0 : 初期状態、 f : 状態遷移関数 $f: Q \times \Sigma \rightarrow Q$
 d : 次動作関数 $d: Q \times \Sigma \rightarrow \{\text{右, 止, 左}\}$
 m : 出力関数 $m: Q \times \Sigma \rightarrow \Sigma$

両方向に無限に長い1本のテープ
1マスに1つの上の記号が書かれる

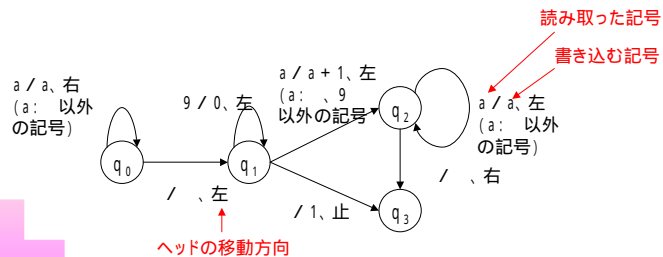
テープを操作しながら規則に従って計算を行う本体
内部状態((1)の規則と(2)の情報)が記憶される

Ichiro Satoh

チューリング機械の例

具体的なチューリング機械

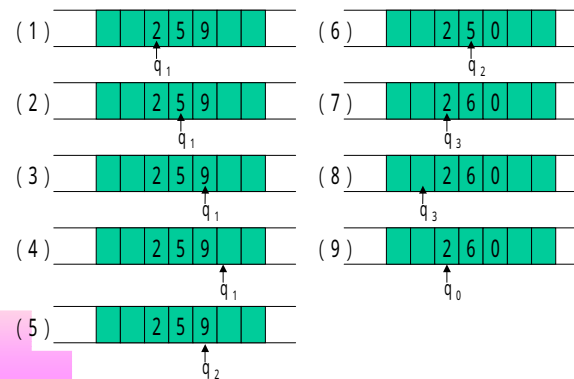
例: 自然数 k の 10 進表現を与えられたとき、 $k + 1$ の 10 進表現を得る機械 M
 M のテープ記号の集合 $A = \{ \text{ } , 0, 1, \dots, 9 \}$
 M の内部状態の集合 $Q = \{q_0, q_1, q_2, q_3\}$ (初期状態は q_0)
 M の規則を表す関数 T (以下の遷移図)



Ichiro Satoh

チューリング機械モデルの計算

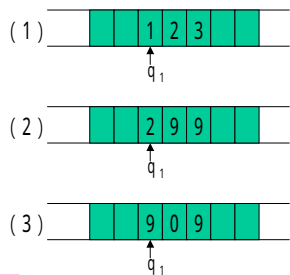
具体的なチューリング機械での計算ステップ



Ichiro Satoh

練習

次の入力をチューリング機械で実行せよ



Ichiro Satoh

チューリング機械と計算

抽象的な計算機 (計算という概念を定式化)

- チューリング機械で記述できる計算過程は現実の計算機でも実現できる
- 現実の計算機の計算過程はチューリング機械でも記述できる

チャーチの提唱

- チューリング機械で計算可能 一般にも計算可能
- チューリング機械で計算不能 一般にも計算不能

チューリング機械の停止性

- 一般のチューリング機械が停止することを判定するアルゴリズムは存在しない。

Ichiro Satoh

▶ 万能チューリング機械

任意のチューリング機械 T_1 を記号化してテープ上に表現し、それを入力とするチューリング機械 T_2 (T_2 は T_1 を模倣する)

▶ フォンノイマン型計算機

メモリ中にプログラムコードとデータが内蔵する計算機

コード化されたチューリング機械

万能チューリング機械

例:

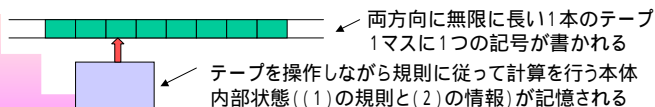
- プログラム
- 仮想機械

▶ チューリング機械モデル

計算する機械の本質を定義した

- (1) はじめに計算の手順を一種の規則として記憶する
- (2) 与えられたデータに上の規則を適用し、1ステップづつ計算を行う
 - 1ステップの動作は、データを読み込んで簡単な計算を行い、必要に応じて結果を計算用紙に書きとめたあと、いくらかの必要な情報を記憶して次のステップへ進む。
 - ここで記憶する情報は、前のステップで得られた結果のほか、今どの段階の計算をしているかの情報なども含む。

具体的な機械のイメージで実現



▶ チューリング機械の例(2)

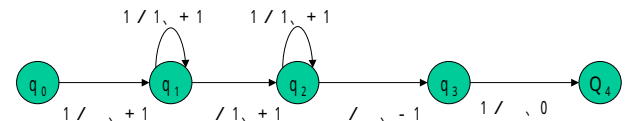
足し算をするチューリング機械M

自然数 k を $k+1$ 個の「1」の並びで表現する。0は1個の「1」で表現する。入力の2つの数値はブランク()で区切られて与えられる。結果はテープ上の1の数。

Mのテープ記号の集合 $A = \{ \quad, 1 \}$

Mの内部状態の集合 $Q = \{ q_0, q_1, q_2, q_3, q_4 \}$

Mの規則を表す関数 T (以下の遷移図)



引き算をする機械はどうやって定義するか?

▶ チューリング機械による自然数関数の計算

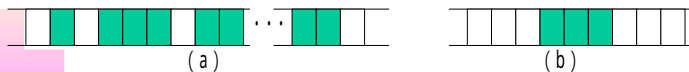
テープ記号に $0, 1, \dots, 9$ を含むチューリング機械 M と、関数 $f: \mathbb{N}^n \rightarrow \mathbb{N}$ について考える。

自然数 m_1, m_2, \dots, m_n に対し、それらの10進表現を下図(a)のように 0 をはさんで並べ、 0 でない最も左のマスに M のヘッドをセットする。

初期状態 q_0 から規則に従って M が計算を行った結果、下図(b)のように $(m_1, m_2, \dots, m_n) = m$ の10進表現をテープに記録してその左端にヘッドを置いて停止するとする。

このことがすべての自然数について成り立つとき、関数 f はチューリング機械 M で計算されるという。

\mathbb{N}^n から \mathbb{N} への部分関数 f がチューリング機械で計算されるならば、 f は帰納的関数である。また、その逆も成り立つ。



Ichiro Satoh



講義のURL

<http://research.nii.ac.jp/~ichiro/model>

メール

ichiro@nii.ac.jp

Ichiro Satoh