専門用語リスト

2018SE034　児玉 春司
指導教員　横山 哲郎

2021 年 05 月 08 日

南山大学 理工学部 ソフトウェア工学科

第 1 章

専門用語リスト

1.1　計算

チューリング機械：アラン・チューリングが提唱した抽象的な機械。
計算：チューリング機械は計算の一部を定義している。現在は TM で計算できるものを計算できる問題と呼ぶことにしている。

1.1.1　計算能力

現状最も高い計算能力をもつ計算モデルの一つはチューリング機械である。チューリング機械と同等の計算能力をもつものには、ラムダ計算、帰納的関数がある。計算能力が下がると元の計算モデルに比べて制限がかかる。
計算可能関数:チューリング機械で表現できるもの、帰納的関数、ラムダ計算など様々な定義がある。
計算可能性理論：計算可能関数に解法があるかどうかの理論。
計算万能性:チューリングマシンで計算可能であれば、時間とメモリを大きくすることで必ず計算できる性質。プログラム言語で計算可能関数を全て計算できるもの万能言語という。

1.1.2　計算量

計算量：計算において消費した資源の量のことである。
時間計算量：チューリング機械のステップ数、プログラムの実行・評価を何回行ったかの量を表す。
漸近的な時間計算量：大雑把にみる時間計算量、オーダー記法で表される。漸近的な時間計算量は無限大に飛ばすとそれぞれの項の重みがどれだけ違うかを無視できる。例えばハードごとに計算量が違うが大まかに比較したいときに適している。
$2^n + n^2 + n$
$n = 1$ のとき $1 : 1 : 1$
$n = 10000000$ のとき $10^o : 10^p : 10^8$、o, p は多量
一次の項は無視できる
空間計算量：チューリング機械のテープ長、メモリにどこまで書きこんだかの量を表す。
回路計算量：回路規模の大きさを表す。

1.1.3 計算複雑性

computational complexity theory（計算複雑性理論）：あるアルゴリズムに対する計算機の実際の計算限界を求める理論。理論を応用して問題を複雑性クラスに分類する。
複雑性クラス：P、NP などそれぞれのクラスは計算量 $t_{classname}$ を持つ。
クラスに属している場合、計算量 $t_{classname}$ 以下で計算できる。$t_{classname}$ が上界になる。より良い上界を求めると計算量が小さくなり、より小さなコストで計算できる。
クラスに属さない場合、計算量 $t_{classname}$ で計算できないことになる。つまり下界である。その計算量で解読できないことが証明できる。
決定的チューリング機械：P クラス：決定的チューリング機械で多項式時間で計算可能なクラス。
NP クラス：非決定的チューリング機械で多項式時間で計算可能なクラス。
EXPTIME クラス：決定性チューリング機械で指数時間で計算可能なクラス。

1.2 可逆計算

可逆計算：計算過程において、任意の計算ステップでもたかだか（0 個か 1 個）前の状態が一意に定まる計算モデル。非可逆な解を A としたら、可逆な解 A と G が出力されるとき G をごみという。可逆な計算で入力がｘ、関数をｆ（ｘ）とする出力Ａにごみ情報Ｇがついてる。非可逆な計算機で可逆計算を行う途中のごみデータは消される。戻すための結果のみが残る。
中間ごみ：最終的に戻すために必要なデータ。
可逆性制約：計算可能関数のなかで単射関数しか扱えないこと。単射関数はごみを付ければどんな計算可能関数を模倣可能である。
可逆チューリング機械が計算できるのは単射な計算可能関数である。
投機的実行：必要かわからない計算も事前に計算をしておくこと。
可逆プログラミング言語：プログラムを書いたら常に可逆性が保証される言語。通常のプログラミング言語でも可逆なプログラムは作成可能である。

1.3 変数

局所変数：任意の変数、その範囲内でのみ成り立つ。
束縛変数：変数名を変えても問題ない変数。
プログラミングにおいては、自由変数とは関数の中で参照される局所変数や引数以外の変数である。
自由変数を全く含まない項や式を閉項または閉論理式または閉式と呼ぶ。

1.4 関数型言語における直交の条件

パターンマッチングにおいて Data が直交するとは、パターンを具体化したときに重なる Data がないこと。制御フローで直交するとは、ある状態の値域の Data を具体化したときに重なる Data がないこと。

1.5　可逆な原子的ステップ

atomic: 原始的
local: 局所的
joinPoints: 合流点
関数としてみると中身は気にせず、入出力が可逆可逆な原始的ステップとは、それ以上分割すると可逆性を保てない、単射部分関数を表すコード片のことである。アサーションとは、どこから合流したかを表す。

1.6　プログラミングパラダイム

Java に Janus というフレームワークがある。
Agent：環境に応じて、自動的に手続きを始めるプログラムの単位。メッセージの送受信で手続きを始めるオブジェクトに近い概念だが、環境によることが異なる。
ModularProgramming：モジュールに分解してプログラムを書くこと
ProgrammingParadiam：プログラミング言語の分類、 ˜　志向など。
AgentOrientedProgramming：MultiAgentPlatform：
SARL：ModularMultiAgentOrientedLanguage

Modulality を増す：モジュール性を増すことである。
構造化：繰り返し、連接、条件分岐のみで表現することである。
オブジェクト指向：あらゆるものがオブジェクト、クラス変数と関数やメソッドでまとめてオブジェクトを作る。　オブジェクト間でメッセージを送受信して計算を実行する。

1.7　Landauer

情報が消失すると熱が散逸することを提唱。Bennet は各ステップで情報を消失しないといけないわけではない。埋め込み法とは，演算のトレースを作成するために，追加の情報を記録する方法である。

1.8　グラフ

頂点と辺の集合の組
$G = (V, E)$
$E \subseteq V * V$

1.9　Janus

変数のエイリアス：プログラムの実行中に、複数の変数が記憶域の同一の記憶場所を指すことがある、この変数をエイリアスという。エイリアスはコンパイラによる最適化の妨げになる。

Principlesof a Reversible Programming Language の Janus

- 大域変数が使えない
- プロシージャ呼び出しの実引数は互いに異なる変数もしくは配列変数を用いる必要がある
- プロシージャの仮引数には同一の変数を渡すことが制限されている

- プロシージャ呼び出し以外の構文でエイリアスが生み出されることはない
- 引数付きプロシージャがある

A Reversible ProgrammingLanguage and its Invertible Self-Interpreter の Janus

- 大域変数あり
- 引数付きプロシージャが存在しないためエイリアスなし

参考文献

[1] 新海 由侑, 田中 秀明, 横山 哲郎：“可逆プログラミング言語の引数渡し機構の拡張”, 情報処理学会論文誌プログラミング Vol.7 No.4 21–36 (Aug. 2014)

[2] German Vidal:“Reversible Computations in LogicProgramming”, Springer Nature Switzerland AG 2020I. Lanese and M. Rawski (Eds.): RC 2020, LNCS 12227, pp. 246–254, 2020.

[3] Yokoyama, T., Axelen, H.B. and Glück, R.: Fundamentals of reversible flowchart languages, Theoretical Computer Science, Vol.611, pp.87–115 (2016).

[4] Lars Kristiansen:“Reversible Programming LanguagesCapturing Complexity Classes”,Springer Nature Switzerland AG 2020I. Lanese and M. Rawski (Eds.): RC 2020, LNCS 12227, pp. 111–127, 2020