解集合プログラミングを使用した宣言的問題解決に関する計算機実験

文理学部情報科学科
5419045 高林 秀

2021 年 12 月 4 日

概要

本稿は、今年度論理と計算 2 における課題学習として「解集合プログラミング」及び「具体的な問題の解決を行う計算機実験」を行うものである。本稿の冒頭〜中盤では関係理論の説明を行い、終盤ではその理論を利用して、実際に具体的な問題に対する解答を提示する。なお、本演習にはソルバーとして clingo を使用した。

1 目的

本稿は、今年度論理と計算 2 の課題研究として、解集合プログラミングを使用した宣言的問題の解決と、その関係理論の説明を通して講義内容を振り返るものである。

以降、本稿の概要は次のとおりである。

1. 計算理論説明
   (a) 述語論理について
      i. 構文
      ii. 限量子
      iii. 解釈とモデル
      iv. 標準形
   (b) 論理プログラムについて
      i. エルブラン領域・基底
      ii. 論理プログラムのクラス区分
      iii. 確定論理プログラム
   (c) 標準論理プログラムについて
   (d) 安定モデルについて
      i. 導出アルゴリズム
   (e) 解集合プログラミングについて
2. 計算機実験
   (a) clingo の説明
   (b) ハミルトン経路

(c) 数独
3. 各問に関する考察
4. まとめ
5. 巻末資料

2 計算理論説明

この章では、今回の計算機実験に使用した各計算理論の解説を行う。

2.1 述語論理について

ここでは論理プログラムに入る前に前提知識となる、述語論理に関する説明を行う。前提となる命題論理に関する説明は下記 URL から参照いただきたい。以下のレポートでは、原子文、複合文等基本用語についてまとめたものである。

- 命題論理に関するレポート：https://drive.google.com/drive/folders/1kOW_1KPUw_kBznaMWjge7HaBI7FoRAoq?usp=sharing

述語論理とは、デジタル大辞泉によると以下のように書かれている。

> 記号論理学の一部門。命題内部の論理構造である主語と述語の関係「すべての主語は…である」「ある主語は…である」などを、論理記号（全称∀・存在∃など）によって記号化して研究するもの

これまで扱ってきた命題論理は、命題のみ扱うことができた。したがって、多数のオブジェクト[*1]間の関係性[*2]を記述することは難しく、それぞれの関係性ごとに逐一命題変数などを用意して記述する必要があった。より具体的には、命題論理はその命題の内容にかかわらず真偽のみに着目する。各命題文同士の関係性を説明するとき、命題記号（命題変数）に変形し推論を行うので、その妥当性を評価するのがむずかしくなる。以下参考となるページのリンクを挙げる。

- 論理学補足文書：http://student.sguc.ac.jp/i/st/learning/logic/%E8%BF%B0%E8%AA%9E%E8%AB%96%E7%90%86.pdf

上記ではこのことを「命題論理の限界」と説明しており、述語論理はそのような弱点を克服した上位の論理言語であると捉えることができる。

述語論理はその関係性に焦点をおいた論理言語で、オブジェクト間の関係性を簡単に示すことができる。また、命題論理では扱わなかった、推論の妥当性に関して扱うことができる。

述語論理は、以下のように区分けされている。

- 一階述語論理：オブジェクトの変数化ができる。
- 二階述語論理：オブジェクトの変数化に加え、述語、関数記号の変数化ができる。
  - 高階述語論理：引数として、1 つ以上別の述語ないしは関数記号をとることができる。一般化する

[*1] オブジェクト：主に名詞、またはそのかたまり（名詞句、名詞節）
[*2] オブジェクト間の関係：そのオブジェクトの動詞にあたるもの。

と、 n 階述語の引数は 1 つ以上の $(n - 1)$ 階の述語である。

以下、その表記の仕方と、登場する記号に意味について説明する。

2.1.1　構文

述語論理において、主語や目的語に相当するものを「対象」と呼ぶ。この「対象」は変数、定数のいずれでもよい。加えて、動詞や形容詞に相当するものを「述語」と呼ぶ。

これらの語句を用いると、述語論理の表記は次のように表すことができる。

- 述語（対象, 対象,...）

また、「対象」は「項 (term)」とも言われ、項には定数、変数、関数記号が存在する。すなわち、述語、関数記号の引数に該当する。

■述語論理における関数　ここで「関数記号」というものが登場したがこれは、構造を持つような複雑なオブジェクトを形式的に示すものである。一般的な数学やプログラムの場合、関数記号は引数になにか入力が与えられ、何かしら結果を出力するものであるが、述語論理における「関数」はそれとは無関係である。つまり、単に 1 つの記号として扱われるということである。

■基礎項　変数を 1 つも含まない項を「基礎項」と呼ぶ。これは、具体的な項、すなわちオブジェクトを示す。

■基礎原子文 (ground atom)　引数すべてが、基礎項であるような原子文 (atom)、すなわち変数がない原子文を「基礎原子文」と呼ぶ。これは、真理値を割り当てる対象となる文である。

1 つ具体例を挙げる。次のような普通の文を考えてみる。

- 「地球と太陽は惑星である」

これを述語論理の形式で示すと以下のようになる。

- $orbits(earth, sun)$

この場合は、$orbits$（惑星である）という述語の目的語、すなわち対象として「$earth$(地球)」と「sun(太陽)」が割り当てられている。述語論理ではこの様な形式（述語文）を最小単位として、命題論理と同様の結合子を用いて、複合文を形成することもできる。

記号	訳	意味
$\wedge$	連言	プログラミングではよく and、&&として扱われる。p かつ q
$\vee$	選言	プログラミングでは or, ||。p または q
$\neg$	否定	プログラミングでは not, !。p ではない
$\Rightarrow, \supset$	含意	～ならばの意味で使われる。
$\Leftrightarrow, \equiv$	同値	「p は q である」が true のとき、
		もしくはその時点に限り true であるとき。p と q は同値。
$\top$	トートロジー (恒真)	後述するトートロジーを示す記号
$\bot$	恒偽 (矛盾)	後述する恒偽を示す記号
(補足)$\veebar, \oplus$	排他的論理和	NAND と呼ばれるもの。

表 1　主要な結合子

命題論理のときと同様に、例えば「$have\_a\_fever(X)$（X は熱を持っている）ならば $take\_a\_drag(X)$（X は薬を飲む）」は、「$have\_a\_fever(X) \Rightarrow take\_a\_drag(X)$」というようにして 2 つの述語文をつなげることもできる。

2.1.2　限量子

ここでは、命題論理には存在しない「限量子」という記号について扱う。限量子とは一言で言えば「変数の範囲を規定するもの」である。例えば、先程の例で $have\_a\_fever(X)$「X は熱を持っている」としたが、この対象 X の範囲を規定する役割を果たす。

限量子には以下 2 種の記号が存在する。

- 全称限量子 : $\forall$:「すべての～、任意の～」というように全てが対象である事を示す。
- 存在限量子 : $\exists$:「少なくとも 1 つの～、ある～に対して」というように、1 つ以上の対象が存在することを示す。

より厳密に言うと、Wikipedia には以下のように書かれている。

- 全称限量子 : 引用元 `https://ja.wikipedia.org/wiki/%E5%85%A8%E7%A7%B0%E8%A8%98%E5%8F%B7`

> 全称記号（ぜんしょうきごう、universal quantifier）とは、数理論理学において「全ての」（全称量化）を表す記号である。通常「∀」と表記され、全称量化子（ぜんしょうりょうかし）、全称限量子（ぜんしょうげんりょうし）、全称限定子（ぜんしょうげんていし）、普遍量化子（ふへんりょうかし）、普通限定子（ふつうげんていし）などとも呼ばれる。
>
> 中略
>
> 「Px」という開論理式 (open formula) が与えられたとき、これが意味するところは「……は P である」ということだけで、これだけでは真偽が確定しない。
>
> 中略
>
> このうち全称記号「∀」によって束縛した場合には「$\forall x Px$」という閉論理式が得られ、これは「全

ての（任意の）x について、x は P である」（より簡単には「全ての x は P である」）という意味になる。

- 存在限量子：引用元 `https://ja.wikipedia.org/wiki/%E5%AD%98%E5%9C%A8%E8%A8%98%E5%8F%B7`

  存在記号（そんざいきごう、existential quantifier）とは、数理論理学（特に述語論理）において、少なくとも1つのメンバーが述語の特性や関係を満たすことを表す記号である。通常「∃」と表記され、存在量化子（そんざいりょうかし）、存在限量子（そんざいげんりょうし）、存在限定子（そんざいげんていし）などとも呼ばれる。

2.1.3　解釈とモデル

2.1.4　標準形

2.2　論理プログラムについて

2.2.1　エルブラン領域・基底

2.2.2　論理プログラムのクラス区分

2.2.3　確定論理プログラム

2.3　標準論理プログラムについて

2.4　安定モデルについて

2.4.1　導出アルゴリズム

2.5　解集合プログラミングについて

3　計算機実験

3.1　clingo の説明

3.2　問題１：ハミルトン経路

3.3　問題２：数独問題

4　各問の結果・考察

4.1　問題１：ハミルトン経路

4.2　問題２：数独問題

5　まとめ

6　巻末資料

本稿で使用した画像、プログラムコード等はすべて以下のリンク先に掲載している。必要に応じてご覧頂きたい。

- GoogleDrive:https://drive.google.com/drive/folders/1n5JPwW-wtBKLASNwndoPRlT7vyZHQvT2?usp=sharing
- GitHub:https://github.com/tsyu12345/logical_and_calculating_LectureCode/tree/master/No10