SAT ソルバーを用いた命題論理問題の説明と具体的問題に対する計算機実験

文理学部情報科学科
5419045 高林 秀

2021 年 10 月 31 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「命題論理の説明」及び「SAT ソルバーを使用した、その具体的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を SAT ソルバーを使用して解答する。なお、本演習には SAT ソルバーとして clasp を使用した。

1　目的

本稿は、今年度論理と計算 2 における課題学習として、SAT ソルバーを用いた命題論理による宣言的問題解決を通じ，命題論理に関する学修内容を振り返ることを目的とする。

本稿は大まかに次のように構成される。

1. 計算理論説明
   （a）命題論理における解釈とモデル、その他関連する事項について
   （b）SAT 問題とはなにか
   （c）DPLL アルゴリズムの解説
2. 計算機実験
   （a）N 人の女王
   （b）グラフ頂点の彩色
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 命題論理とは

まず命題論理とはなにか説明する。命題論理という言葉の意味はデジタル大辞泉に次のように書かれている。

> 記号論理学の基礎的部門。個々の命題を結合する「かつ」「または」「ならば」「でない」などの関係を、論理記号を用いて論理積（＞）・論理和（＜）・含意（→）・否定（～）などにより記号化して演算形式に表し、複合された命題を研究する学問。命題計算。

そもそも命題は、数学では「真偽の判断の対象となる文章または式」であり、論理学においては「判断を言葉で示したもので真または偽」という性質を持つもの、という意味である。したがって、命題論理とは、命題同士の関係性を論理記号を使用して記号化し、演算できるようにしたものということだ。

■結合子　今説明したように、命題論理では命題同士の性質、関係性を扱う。それを説明する上で「結合子（論理記号）」と呼ばれるものが定義されている。

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。直感的には「p が真であるとき、必ず q は真である」
$\Leftrightarrow, \equiv$	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
$\top$	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。

■命題文～原子文, 複合文　命題論理は次の要素から構成される。

- 文、命題文 (sentence) ※これは命題論理式とも言う。
  - 原子文 (atomic formula)：これ以上分解することができない命題。最も単純な文。いわゆる 1 つの命題であり、以下の例のようにそれぞれ固有の記号で示すことができる。
    （例）p：「動物はいつか死ぬ」,q：「人間は動物である」,z：「人間はいつか死ぬ」
  - 複合文 (complex sentence)：文同士を「結合子」で連結した文。命題同士の関係性を結合子を利用して連結し、新たな文を作ることができる。
    （例）$p \wedge q \Rightarrow z$：「動物はいつか死ぬ」かつ「人間は動物である」ならば「人間はいつか死ぬ」。

まとめると、命題文には原子文や複合文と呼ばれる区分けが存在し、原子文は「真 (true)、偽 (false)、それ以上分解できない命題（記号）」であり、複合文は「命題文同士を結合子で連結した新たな命題文」ということになる。

なお、true,false と呼ばれる命題の真偽を示すこれらの記号は論理定数と呼ばれる。原子文の例で示した命題文を各記号に置き換えたもの p, q, z は命題記号ないしは命題変数と呼ばれる。

■命題文の構文　これらの要素を組み合わせて作られる命題文は、使用する結合子によって以下に区分けされる。

- 否定文：結合子 $\neg$ で連結されている複合文。
- 連言文：結合子 $\wedge$ で連結されている複合文。
- 選言文：結合子 $\vee$ で連結されている複合文。
- 含意文：結合子 $\Rightarrow$ で連結されている複合文。
- 同値文：結合子 $\Leftrightarrow$ で連結されている複合文。

とくに含意文 $\alpha \Rightarrow \beta$ については α を前提、β を帰結と呼ぶ。また、原子文とその否定文 $\neg p, \neg q, \neg z$ をひとくくりにしてリテラルと呼ぶ。

■知識ベース　また、命題文の集合において、各文を結合子 $\wedge$ で連結したものを知識ベースと呼ぶことがある。（例：KB[KnowledgeBase] = {S1, S2, S3} $\Rightarrow$ S1 $\wedge$ S2 $\wedge$ S3)

■真理値表 (trueth table)　命題文の真偽は先に上げた論理定数 true, false で示す。複合文の真偽を表にまとめて示したものを真理値表と呼ぶ。先に上げた各構文の真理値は、真理値表を用いて次のように定義されている。

定義 1. 複合文の真理値は以下のとおりである。

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \Rightarrow q$	$p \Leftrightarrow q$
false	false	true	false	false	true	true
false	true	true	false	true	true	false
true	false	false	false	true	false	false
true	true	false	true	true	true	true

上記真理値表より、含意文に関して重要なことを以下に列挙する。

- 前提 α が false の場合、帰結 β の真理値に関係なく、式として true になる。
- 前提 α と帰結 β の間に因果関係や関連性は要求されない。すなわち、前提文と帰結文がかけ離れた話題であっても、前提が真ならば含意文としては真となる。（例：p : のび太は人間である $\Rightarrow$ q : スネ夫は金持ちのボンボンである、の p, q に直接的な関連性はないが、前提の p が true であるので、文としては true（正しい）となる。）

ここまで、基本用語の説明を行った。以降は、命題論理における「解釈、モデル、伴意関係」とはなにか、加えて SAT 問題や DPLL アルゴリズムについて説明する。

2.2　解釈について

解釈という言葉の意味は日本国語大辞典に次のように書かれている (一部抜粋)。

> 語句や物事の意味、内容などを説明すること。解き明かすこと。また、その解説。
> 物事、特に表現されたものを、自分の経験や判断力などによって理解すること。

法令の意味を明確にして、その内容が動かないように定めること。

命題論理における「解釈」とは、これらの意味に関係なく次のように定義する。

定義 2. 命題文 α に現れる原子文（または命題記号）を $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n$ とする。このとき、各原子文 $\alpha_{i(1\leq i\leq n)}$ に対する真理値（*truem false*）の割当を「α の解釈」と呼ぶ。

命題記号 p, q, r が定義されているとする。このとき、命題文「$p \wedge q \vee r$」の解釈は次のものが考えられる。

$$\{p = true, q = true, r = true\}, \{p = true, q = true, r = false\},$$
$$\{p = false, q = true, r = true\}, \{p = false, q = true, r = false\},$$
$$\{p = true, q = false, r = true\}, \{p = true, q = false, r = false\},$$
$$\{p = false, q = false, r = true\}, \{p = false, q = false, r = false\}$$

そしてその組み合わせの数は $2^3 = 8$ より 8 個である。一般化すると、「n 個の原子文がある時、解釈の個数は 2^n個」である。

ただし、こうズラズラ書いていては非常に煩雑であるので、以下のように true が割り当てられている命題記号だけ抜き出して記述する略記法も定義されている。上記例の場合は以下のようになる。

$$\{p, q, r\}, \{p, q\}, \{q, r\}, \{q\}, \{p, r\}, \{p\}, \{r\}$$

すなわち、解釈とは命題記号に対する true,false の割当パターンのことである。この解釈のうち、命題文 α の真理値を true にする解釈が、次に示す「モデル」と呼ばれる。

2.3 モデルについて

モデルという言葉の意味には次のようなものが挙げられるだろう。

- ある事柄の手本や見本
- ある事象について、様々な要素とそれらの相互関係を定式化したもの
- 機械学習モデルなど、データ解析の手法のこと。

命題論理における「モデル」とは先に示したとおり、「命題文の真理値を true にする解釈」のことである。

定義 3. 命題文 α の真理値を真 *(true)* にする解釈 I を α のモデルと呼ぶ。

繰り返すが、命題文を true にする解釈をモデルと呼ぶ、ただそれだけである。

真理値表では、各行がそれぞれ一つの解釈となっており、全体の命題文を真にする行（解釈）がモデルということだ。

2.3.1 トートロジーと充足可能

任意の命題文の真理値表を作成すると、すべての解釈で true になる命題文がでてくることがある。この様な命題文ないしは命題論理式を「トートロジー (tautology)」と呼ぶことがある。トートロジーとは日本語で「恒真」、すなわち常に真である、ということを意味する。以下、`https://qiita.com/kimunny/items/195f45154b6cc6a2940b` より引用する。

命題論理において、パラメータの命題の値にかかわらず、常に真になる論理式をトートロジー、あるいは常真式と呼びます。

以下にトートロジーになる命題論理式の例を示す。これらは後述する「命題論理式の標準形」に変形する際に必要となる考え方である。また、真理値表を書くとトートロジーであることが確認できる。

α	β	γ	α	$\vee$	$(\beta \wedge \gamma)$	$\Leftrightarrow$	$(\alpha \vee \beta)$	$\wedge$	$(\alpha \vee \gamma)$
false	false	false	false	false	false	true	false	false	false
false	false	true	false	false	false	true	false	false	true
false	true	false	false	false	false	true	true	false	false
false	true	true	false	true	true	true	true	true	true
true	false	false	true	true	false	true	true	true	true
true	false	true	true	true	false	true	true	true	true
true	true	false	true	true	false	true	true	true	true
true	true	true	true	true	true	true	true	true	true

α	β	$\neg$	$(\alpha \wedge \beta)$	$\Leftrightarrow$	$\neg\alpha$	$\vee$	$\neg\beta$
false	false	true	false	true	true	true	true
false	true	true	false	true	true	true	false
true	false	true	false	true	false	true	true
true	true	false	true	true	false	false	false

これとは反対に、どの解釈でも常に偽になる命題論理式を「矛盾 (contradictory well-formed formula)」または「恒偽」と呼ぶ。

また、真にも偽にもなりうる命題論理式を「整合式 (consistent well-formed formula)」または「充足可能」と呼ぶ。

これらの言葉を使うと、先に述べた論理定数である true と false はそれぞれ、「トートロジー, 恒偽」であり、任意の命題変数 p は true,false どちらも取り得るため充足可能である、と言える。整理すると以下。

- 論理定数 true:トートロジーである。
- 論理定数 false:恒偽である。
- 任意の命題変数 p:充足可能である。

すなわち、ある命題文、命題論理式の真理値表を作成するということは「その命題文がトートロジーであるか、それとも恒偽であるか、それとも充足可能であるか」を調査、判定していることになる。

2.4 伴意関係 (Entailment)

命題論理における伴意関係（別名：論理的帰結）とは以下のような意味をもつ。以下 Wikipedia の `https://ja.wikipedia.org/wiki/%E8%AB%96%E7%90%86%E7%9A%84%E5%B8%B0%E7%B5%90` より引用する。

論理的帰結（ろんりてききけつ、伴意、英: logical consequence, entailment）は、論理学における最

も基本的な概念であり、複数の文（または命題）の集合と 1 つの文（命題）の間が「〜だから、当然〜」という繋がり方をする関係を指す。

すなわち、「命題文の集合」と「命題文」の関係のことを示している。この伴意関係を式で示したものを「伴意式」と呼ぶ。このとき、命題文の集合のことを「理論」と呼ぶ。加えて、この「理論」は先に示した知識ベース「KB」のことであり、各命題文を連言記号で連結したもの「連言文」と同じである。

■伴意式　一般的に伴意式は次のような表記をする。

$G \models \alpha$
意味：「理論 G を文（命題文）αが伴意する」＝理論 G からαが論理的に導出できる。

これは、理論 G の解釈が true であるとき、α も true となることを意味している。すなわち、G が true のとき、α は常に true になるということだ。よって、含意文で示すと $G \Rightarrow \alpha$がトートロジー となる。つまり伴意関係とは、G が true（G の全要素が true）であるとき、α が false になることはありえない、トートロジーであると言っているのである。したがって、命題文が伴意関係にあるか否かはトートロジーであることを確認すればよいことになる。

例を挙げる。$G = \{p, q, (p \wedge q) \Rightarrow r\}$ としたとき、$G \models r$ または、G を展開して、$\{p, q, (p \wedge q) \Rightarrow r\} \models r$ といったように表記する。このとき、$\{p, q, (p \wedge q) \Rightarrow r\}$ は、$(p \wedge q \wedge (p \wedge q) \Rightarrow r)$ と同じであるので、真理値表では以下のように示すことができる。

p	q	r	$p \wedge q$	$p \wedge q \Leftrightarrow r$	$p \wedge q \wedge (p \wedge q \Rightarrow r)$
false	false	false	false	true	false
false	false	true	false	true	false
false	true	false	false	true	false
false	true	true	false	true	false
true	false	false	false	true	false
true	false	true	false	true	false
true	true	false	true	false	false
true	ture	true	true	true	true

以上より、G が true であるとき、r も true となっていることが確認できる。したがって、伴意式「$G \models r$」は成立する（伴意関係にある）。

2.5　SAT 問題とは

前章では、命題論理における解釈とモデルや伴意関係とはなにか、加えてトートロジーや恒偽、充足可能とはなにかについて説明してきた。この章では、先に示した「充足可能」についてより詳細に扱う。

初めに繰り返しになるが、「充足可能」とは「真にも偽にもなりうる命題論理式」のことであることは示した。ある命題論理式が充足可能であるか否かを判定するのが、SAT 問題[*1]（充足可能性問題）である。より厳

[*1] ・SAT : satisfiability problem の頭 3 文字
・SAT 問題 : Boolean Satisfiability Testing

密には以下のように言われる※引用元 https://ja.wikipedia.org/wiki/%E5%85%85%E8%B6%B3%E5%8F%AF%E8%83%BD%E6%80%A7%E5%95%8F%E9%A1%8C。

> 充足可能性問題（じゅうそくかのうせいもんだい、satisfiability problem, SAT）は、一つの命題論理式が与えられたとき、それに含まれる変数の値を偽 (False) あるいは真 (True) にうまく定めることによって全体の値を' 真' にできるか、という問題をいう。

すなわち、与えられた命題論理式にモデルが存在するかを判定する問題ということになる。SAT 問題は、命題論理式が以下のどちらに属するか決定する問題と言える。

- モデルが存在する→充足可能
- モデルが存在しない→恒偽

これを利用して、背理法を使用して命題論理式にモデルが存在しないことを証明することもできる。

余談だが、この SAT 問題は「NP 完全[*2]」であることが最初に証明された問題でもあることが知られている。

では、実際どのように命題論理式にモデルが存在するか否かを求めるのかについて述べる。SAT 問題は、与えられた命題論理式が充足可能かどうか言えれば良い。したがってまず思いつく手法としては先に示した「真理値表を作成する」という方法が思いつくだろう。しかし、真理値表はすべての命題変数に真理値を割り当てた後、解釈を見ることができる。したがって、論理式中の命題変数の個数が多ければ多いほど、充足可能かどうか判定するのに非常に時間がかかる。そこで、後述する SAT ソルバーと呼ばれるものが存在する。この SAT ソルバーは、単位伝搬や DPLL アルゴリズムなど様々な仕組みによって、効率よく命題論理式が充足可能かどうか判定することができる。単位伝搬や DPLL アルゴリズムについての説明は後述する。まずは、この SAT ソルバーを用いた問題解決システムである「SAT 型システム」について紹介する。

■SAT 型システム　与えられた問題を SAT 符号化し、SAT ソルバーを使用して解くシステムを SAT 型システムと言う。SAT 型システムが生まれた背景としては、問題ごとに特化（適した）アルゴリズムを作成するのではなく、SAT 符号化して SAT ソルバーに解いてもらおうとする考え方があった。問題ごとに特化（適した）アルゴリズムを作成するのは手間と時間がかかることがある。しかし、問題を SAT 符号化すれば、SAT ソルバーに入力するだけで良いので、手間と時間がかからなくなる。他の問題に対しても、SAT ソルバーに対する入力を変更しさえすればよいので、前者よりもはるかに効率が良い。

近年、SAT ソルバーの解を求める性能が飛躍的に向上したこともあり、SAT 型システムは以下のような分野において成功を収めている。

- 自動テストパターン生成
- ソフトウェア検証
- 解集合プログラミング
- Linux パッケージマネージャー（DNF）の依存性解決

この他にも多種多様な分野で SAT 型システムは成功を収めている。他の事例については https://www.

[*2] NP-complete problem：クラス NP に属する決定問題かつクラス NP に属する任意の問題から多項式時間還元可能な問題のこと。詳細な説明は本稿では扱わないが下記に参考 URL を示す。
- うさぎでもわかる P vs NP 問題：https://www.momoyama-usagi.com/entry/info-p-np

jstage.jst.go.jp/article/jssst/35/4/35_72/_pdf を参照いただきたい。

2.5.1 SAT の基本アルゴリズム

■木構造化 さて、先に示したとおり命題論理式が true になる解釈を調べる（モデルが存在するか否かを調べる）のが SAT 問題である。問題を解くには各命題変数に true,false の 2 通りの真理値を割り当てていくことになる。よって、命題変数をノード、真理値の割り当てを分岐として捉えることで「二分木」を作成することが可能になる。

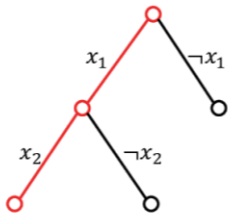

図 1 SAT の木構造化

※引用元：https://jssst-ppl.org/workshop/2017/slides/ppl2017_c4_soh.pdf

このように捉えることによって、木構造の特徴である「枝刈り」や「探索打ち切り」を行うことができる。SAT ソルバーはこのように、真理値割当中に命題論理式が true にならないと確定した時点で「探索打ち切り」や「分岐の削減」を行うことで効率を高めている。その方法が後述する DPLL アルゴリズムの章で触れる「早期停止」や「節学習」、「単位伝搬」と呼ばれる方法であり、それらを組み合わせた SAT アルゴリズムが「DPLL」と呼ばれるアルゴリズムである。

SAT 問題においては、モデルの有無のみが問われるので、探索中に 1 つのモデルが存在すればその時点で探索を終了すれば良いことになる。したがって、命題変数への真理値割当の際に命題論理式が true にならないことが確定した時点で探索を打ち切り、次の真理値割当へと進めば良い。

2.6 DPLL アルゴリズム

3 計算機実験

3.1 実験準備

3.1.1 実験環境

今回の実験は仮想マシン上で clasp のバイナリをダウンロードして行った。下記に実験時の環境を示す。

- ホスト OS：Window10 Home 20H2
- 仮想 OS：Ubuntu 20.04.2 LTS
- CPU：Intel(R)Core(TM)i7-9700K @ 3.6GHz
- GPU：Nvidia Geforce RTX2070 OC @ 8GB
- ホスト RAM：16GB
- 仮想 RAM：4GB

3.1.2 問題 1:N 人の女王

配布資料中に、Processing のプログラムが「nQueen.pde」として以下の関数が用意されている。

- バックトラック法を用いて nQueen を解く関数
- clasp への入力ファイルを作成する関数

1. この問題に対する SAT 符号化を詳細に説明せよ。
2. N の大きさを様々に変えながら，バックトラック法で解いた場合と SAT ソルバーで解いた場合とでの実行時間を比較・考察しなさい.

3.1.3 問題 2:グラフ頂点の彩色問題

配布資料中の「GraphColoring」フォルダに、「都道府県の隣接関係」を表すグラフの頂点彩色問題の CNF ファイルが用意されている。

1. この問題に対する SAT 符号化を詳細に説明せよ。
2. 関東地方を対象に、いくつの塗分け方法があるか調べなさい。
3. 47 都道府県を対象とした色塗りの例を一つ示しなさい。
   (a)（例）長野県：青色, 神奈川県：赤色、のように、どの都道府県をどの色で塗るのかを具体的に示すこと。

3.2 各問に対する解答・考察

3.2.1 問題 1:N 人の女王

3.2.2 問題 2:グラフ頂点の彩色問題

4 まとめ

5 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/1kOW_1KPUw_kBznaMWjge7HaBI7FoRAoq?usp=sharing`
- GitHub:`https://github.com/tsyu12345/logical_and_calculating_LectureCode/tree/master/No5`