論理と形式化 No.7
命題論理と充足可能性判定

2025年度

担当教員　亀山幸義 (logic@logic.cs.tsukuba.ac.jp)

後半４週の授業の計画

7. 充足可能性判定とSATソルバ
    1. 命題論理式から節形式への変形
    2. 節形式に対する充足可能性判定（SATソルバ）
    3. SATソルバを利用した問題解決
8. 導出原理と論理プログラミング
    1. ホーン節形式の述語論理式
    2. 導出原理
    3. Prolog言語の使い方
9. 論理プログラミングと解集合プログラミング
    1. Prologプログラミング
    2. 解集合プログラミング
10. コンピュータを使った定理証明
    1. SMTソルバによる全自動証明、Coqを使った対話的定理証明、モデル検査など。

今日の目標

命題論理に対する決定手続き（アルゴリズム）について学ぶ。

- 証明可能性判定問題：論理式Aが与えられた時、Aが証明可能かどうかをyes/noで答える問題
- 充足可能性判定問題：論理式Aが与えられた時、Aが充足可能かどうかをyes/noで答える問題

論理学の立場：

- これらは決定可能：判定問題を解くプログラム(Turing機械）が存在する。

情報科学の立場：

- 判定問題を高速に解くアルゴリズムはあるか？
- yes/noだけでなく、yesの場合はもっと知りたい。
  - 証明可能なら、その証明を出してほしい。
  - 充足可能なら、充足されるその割り当てを出してほしい。
- このような判定問題に帰着できる問題はどんなものがあるか？

準備：命題論理の用語

- 論理式　　$A, B ::= P \mid A \wedge B \mid A \vee B \mid \neg A$
  - Pは命題変数（原子命題、原子文ともいう）
  - ⊃（ならば）と ≡（同値）など、他の論理記号を追加することもある。
- 真理値の割り当て
  - 割り当て$[P \mapsto T, Q \mapsto F, R \mapsto F]$のもとで命題 $P \wedge (Q \vee \neg R)$ は T
- 証明可能、充足可能
  - 命題 $P \vee \neg P$はNKで証明可能
  - 命題 $\neg P \vee Q$は、割り当て $[P \mapsto T, Q \mapsto T]$ のもとで T なので、充足可能
- 節、論理積標準形
  - 節(clause)：$\neg P \vee Q \vee P \vee \neg R$
  - 論理積標準形(CNF)：$(\neg P \vee Q \vee P) \wedge (P \vee \neg Q \vee \neg R) \wedge (\neg Q)$

7-1. 標準形への変換

論理式の標準形

標準形の定義はいろいろだが、共通するのは以下の性質

- 任意の論理式を、それと同値な標準形に変形できる。

命題論理の論理式の標準形（代表的なもの）

- 論理積標準形（連言標準形、Conjunctive Normal Form, CNF）
- 論理和標準形（選言標準形、Disjunctive Normal Form, DNF）

練習問題１.

- 以下の命題論理式をそれと同値なCNFおよびDNFに変形せよ。
  - $\big((P \land \neg Q) \lor R\big) \supset \neg(S \land T)$

CNFにすると何が良いか？

練習問題２.

- 以下のCNFが証明可能かどうか判定せよ。
  - $(\neg P \vee Q \vee R) \wedge (P \vee Q \vee \neg P) \wedge (\neg P \vee \neg Q \vee R)$

- 以下のCNFが充足可能かどうか判定せよ。
  - $(\neg P \vee Q) \wedge (P \vee Q) \wedge (\neg Q \vee R) \wedge (\neg R \vee P)$
  - $(\neg P \vee Q \vee R) \wedge (P \vee Q \vee R) \wedge (\neg P \vee \neg Q \vee R)$

命題論理の証明可能性判定は簡単か？

練習問題3.

- 命題論理式の証明可能性を判定する以下のアルゴリズムの計算量を見積ろう。
  1. 命題論理式をCNFに同値変形する。
  2. CNFに対する証明可能性を判定する。
- $A \coloneqq (\mathrm{P}_1^0 \wedge \mathrm{P}_1^1) \vee (\mathrm{P}_2^0 \wedge \mathrm{P}_2^1) \vee \cdots \vee (\mathrm{P}_k^0 \wedge \mathrm{P}_k^1)$ のとき、
  - $|A| = 2k - 1$　(論理式のサイズは、論理記号の数とする。)
  - $\mathrm{B} \coloneqq \wedge \left(\mathrm{P}_1^{f_1} \vee \mathrm{P}_2^{f_2} \vee \cdots \vee \mathrm{P}_k^{f_k}\right)$ はAと同値なCNF
    - ただし、$f_1, f_2, \cdots, f_k \in \{0,1\}$
    - $|B| = 2^k(k-1) + 2^k - 1 = k \cdot 2^k - 1$
  - ステップ１の出力となる論理式が指数関数的なサイズなので、アルゴリズム全体の計算時間も指数関数以上

証明可能性と充足可能性

命題論理では、充足可能性を判定するアルゴリズムがあれば、証明可能性も判定できる。逆もそう。

- Aが証明可能⇔$\neg A$が充足可能でない
- Aが充足可能⇔$\neg A$が証明可能でない

演習問題４.

- 上のことはなぜ成立するか、命題論理の完全性や健全性などの言葉を使って説明せよ。

最古のNP完全問題

定理 [Cook 1971]

- CNFに対する充足可能性判定(SAT)問題は、NP完全である。
  - CNF-SATは、歴史上初めてNP完全であることが証明された問題
  - CNFに限定してもNP完全なので、任意の命題論理式に対する充足可能性は、NP完全かそれ以上の計算量。証明可能性も同様。

課題.

以下のものに対する正確な定義を自分で調査しよう。

- NP問題
- NP完全
- P≠NP予想

※Turing機械：専門授業「オートマトンと形式言語」の対象

情報科学の立場

数学者が考えること

- 解けるかどうかが問題、解くのにかかる時間は考慮外。
- 問題が解けないなら、解けないことを証明しよう。

情報科学者が考えること：

- 現実的な時間・メモリで解きたい。
- 問題が解けないなら、解けるように問題を変形しよう。
- 全部解くことが不可能でも、一部でも解けるとうれしい。

命題論理式の充足可能性判定

- 命題論理式Aが与えられた時、
  1. CNFへの変形は、論理的な同値変形でなくてよい。
  2. CNFに対して、多くの場合に高速に動作するアルゴリズムであればよい。

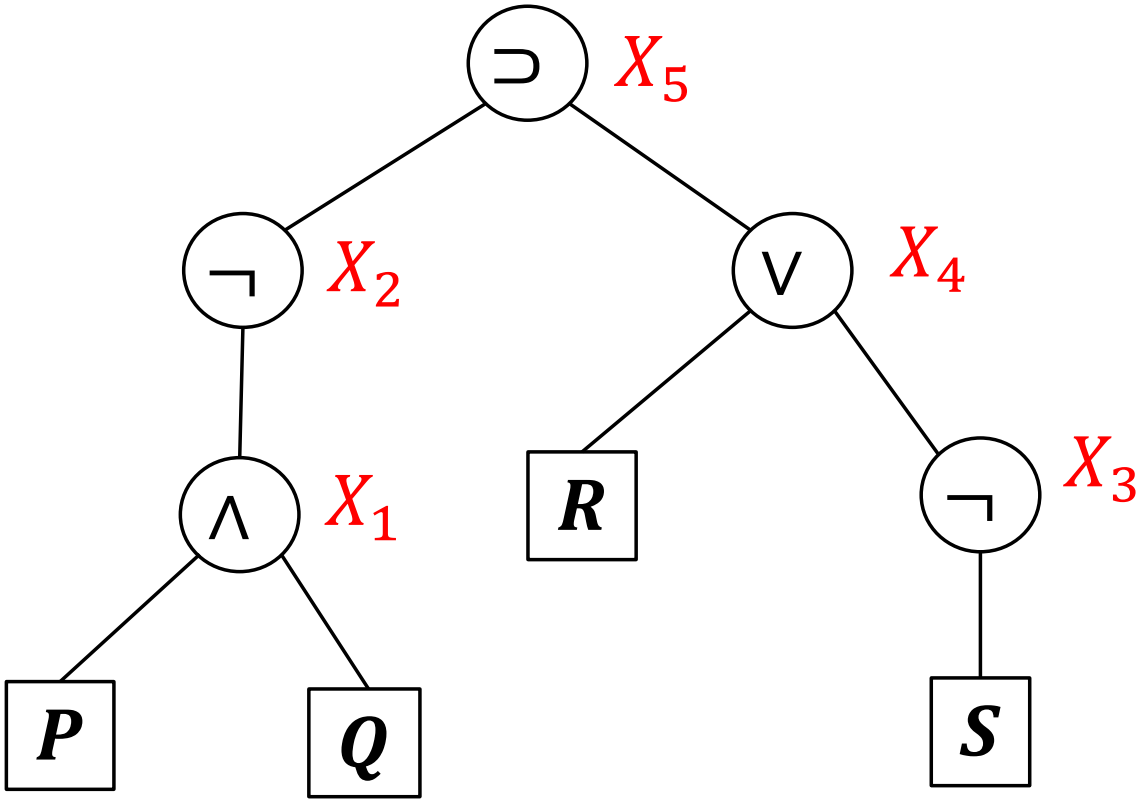
命題論理式からCNFへの変形

命題論理式Aから$\mathbf{CNF}$ $\phi(A)$への変形アルゴリズムの満たすべき条件：

- $\phi(A)$は$\mathbf{CNF}$である。
- 「Aが充足可能」と「$\phi(A)$が充足可能」は同値。
- ϕ は計算量的に良い。つまり、$|A| = n$とするとき、
  - $|\phi(A)|$は$O(n)$
  - ϕの時間計算量・空間計算量は$\mathrm{O}(n)$
- さらに、$\phi(A)$を充足する割り当てから、Aを充足する割り当てを簡単に計算できると、うれしい。

命題論理式からCNFへの変形 [Tseitin 1968]

$A \coloneqq (\neg(P \wedge Q)) \supset (R \vee \neg S)$



1. 論理記号ごとに命題変数を導入

- $X_1 \equiv P \wedge Q$
- $X_2 \equiv \neg X_1$
- $X_3 \equiv \neg S$
- $X_4 \equiv R \vee X_3$
- $X_5 \equiv X_2 \supset X_4$

命題論理式からCNFへの変形(1)

$A \coloneqq (\neg(P \wedge Q)) \supset (R \vee \neg S)$

1. 論理記号ごとに命題変数を導入

- $X_1 \equiv P \wedge Q$
- $X_2 \equiv \neg X_1$
- $X_3 \equiv \neg S$
- $X_4 \equiv R \vee X_3$
- $X_5 \equiv X_2 \supset X_4$

2. それぞれの式をCNFに変形

- $(\neg X_1 \vee P) \wedge (\neg X_1 \vee Q) \wedge (X_1 \vee \neg P \vee \neg Q)$
- $(\neg X_2 \vee \neg X_1) \wedge (X_2 \vee X_1)$
- $(\neg X_3 \vee \neg S) \wedge (X_3 \vee S)$
- $(\neg X_4 \vee R \vee X_3) \wedge (X_4 \vee \neg R) \wedge (X_4 \vee \neg X_3)$
- $(\neg X_5 \vee \neg X_2 \vee X_4) \wedge (X_5 \vee X_2) \wedge (X_5 \vee \neg X_4)$

命題論理式からCNFへの変形(2)

$(\neg(P \wedge Q)) \supset (R \vee \neg S)$

3. 得られたCNFすべてと、全体をあらわす命題変数の論理積をとる

$$\phi(A) \coloneqq (\neg X_1 \vee P) \wedge (\neg X_1 \vee Q) \wedge (X_1 \vee \neg P \vee \neg Q)$$
$$\wedge (\neg X_2 \vee \neg X_1) \wedge (X_2 \vee X_1)$$
$$\wedge (\neg X_3 \vee \neg S) \wedge (X_3 \vee S)$$
$$\wedge (\neg X_4 \vee R \vee X_3) \wedge (X_4 \vee \neg R) \wedge (X_4 \vee \neg X_3)$$
$$\wedge (\neg X_5 \vee \neg X_2 \vee X_4) \wedge (X_5 \vee X_2) \wedge (X_5 \vee \neg X_4)$$
$$\wedge X_5$$

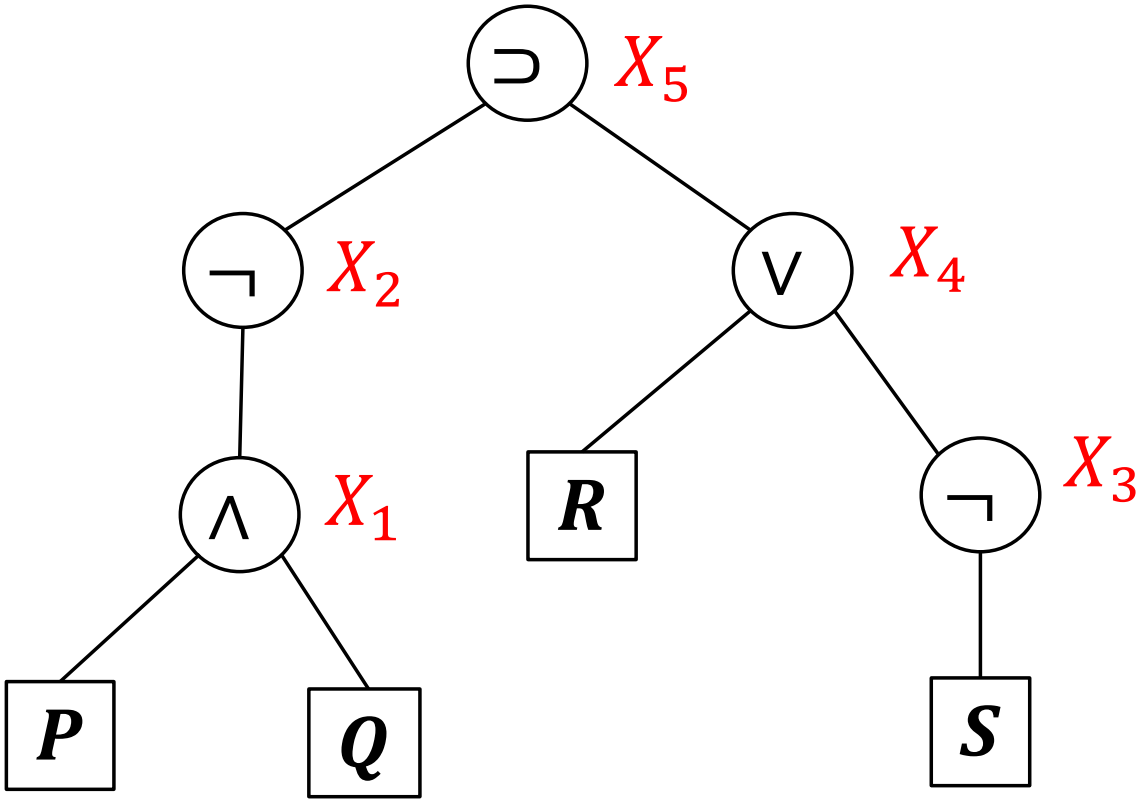
これで完成

考察

- Aと$\phi(A)$は同値ではない。
- しかし、「 Aが充足可能 」と「 ϕ(A)が充足可能 」は同値。

Tseitin変換は、充足可能性を保つ(1)

$A \coloneqq (\neg(P \wedge Q)) \supset (R \vee \neg S)$

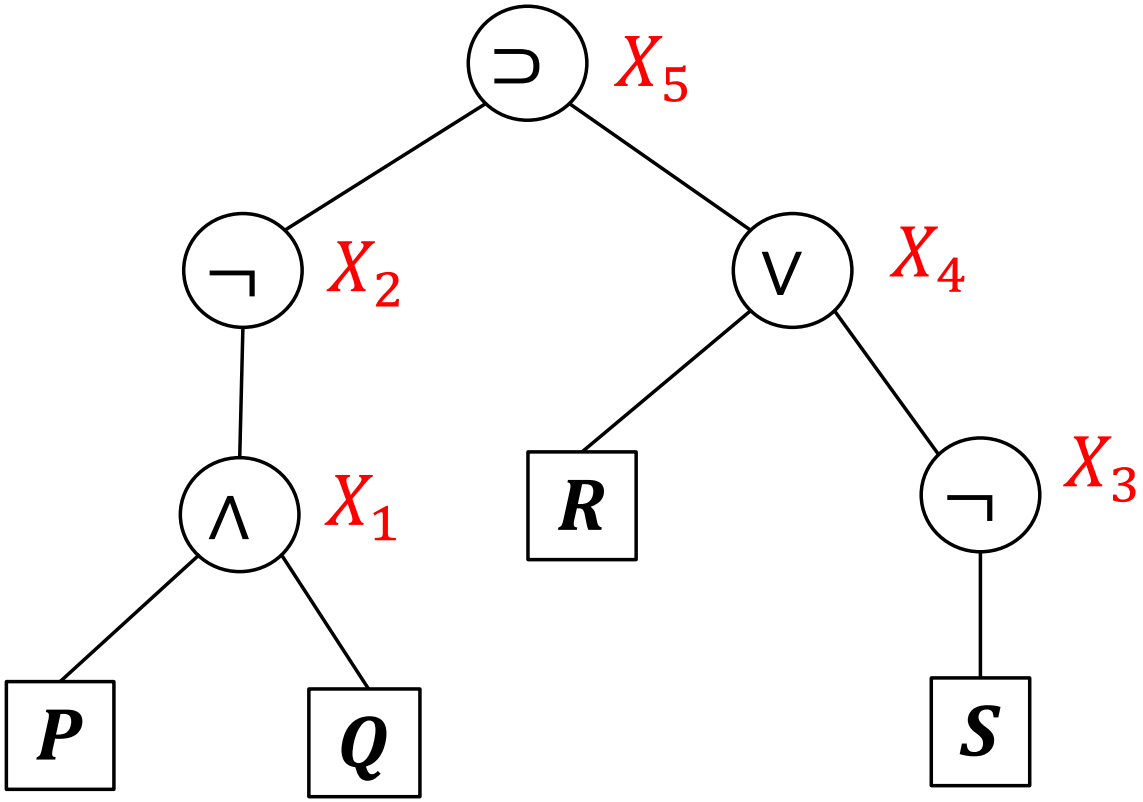


Aが充足可能
⇒Aを真にするP, Q, R, Sへの割り当てが存在

⇒$\phi(A)$は充足可能

Tseitin変換は、充足可能性を保つ(2)

$A \coloneqq (\neg(P \wedge Q)) \supset (R \vee \neg S)$



$\phi(A)$が充足可能
⇒ $\phi(A)$を真にする
$P, Q, R, S, X_1, X_2, X_3, X_4, X_5$への割り当てが存在

⇒Aは充足可能

Tseitin変換のまとめ

ϕ：命題論理式→CNF

- 「Aが充足可能」と「$\phi(A)$が充足可能」は同値。
- $\phi(A)$がsatのとき、それを充足する割り当てから、Aを充足する割り当てが簡単に得られる。
- ϕ は計算量的に優れている。$|A| = n$とすると、
  - $|\phi(A)|$は$O(n)$
  - ϕの時間計算量は$O(n)$

課題．任意の命題論理式Aに対して、 $|\phi(A)| < C_1|A| + C_2$ となる定数C_1, C_2が存在することを示しなさい。

練習問題５．$\neg(((P \supset Q) \supset P) \supset P)$をTseitin変換せよ。

7-2. 充足可能性判定

CNFの充足可能性判定

CNF に対する充足可能性(satisfiability)判定

- 与えられたCNFに対して、satまたはunsatを返す。
  - CNFを真とする割り当てが1つでもあればsat（と割り当て）を返し、
  - なければunsatを返す。
- 決定可能だがNP完全

割り当て

- 命題変数$P, Q, R, \cdots$に対して真理値を対応付け
- 例：命題 $P \wedge (Q \vee \neg R)$ は、割り当て$[P \mapsto T, Q \mapsto F, R \mapsto F]$で真

SATソルバの基本アルゴリズム

DPLL (Davis-Putnam-Logemann-Loveland)アルゴリズム

入力：CNF

内部変数：作りかけの割り当てθ　(初期値は空の割り当て)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

DPLLの例(1)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

$$(P \vee Q \vee \neg R)$$
$$(P \vee \neg Q \vee R)$$
$$(Q)$$
$$(\neg P \vee R \vee S)$$
$$(R \vee \neg S)$$
$$(\neg P \vee \neg R)$$

割り当て$\theta = [\,]$

単位節は、原子命題だけ節、および、
原子命題の前に否定記号を付けただけの節のこと

DPLLの例(2)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

~~$(P \vee Q \vee \neg R)$~~

$(P$ ~~$\vee \neg Q$~~ $\vee R)$

~~(Q)~~

$(\neg P \vee R \vee S)$

$(R \vee \neg S)$

$(\neg P \vee \neg R)$

$\theta = [\, Q \mapsto T \,]$

単位節は、原子命題だけ節、および、
原子命題の前に否定記号を付けただけの節のこと

DPLLの例(3)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

$$(P \vee R)$$
$$(\neg P \vee R \vee S)$$
$$(R \vee \neg S)$$
$$(\neg P \vee \neg R)$$
$$\theta = [\, Q \mapsto T, P \mapsto T \,]$$

DPLLの例(4)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

$$\cancel{(P \vee R)}$$
$$(\cancel{\neg P \vee} R \vee S)$$
$$(R \vee \neg S)$$
$$(\cancel{\neg P \vee \neg} R)$$
$$\theta = [\, Q \mapsto T, P \mapsto T \,]$$

DPLLの例(5)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

$$(\cancel{R \vee S})$$

$$(\cancel{R \vee \neg S})$$

$$\cancel{(\neg R)}$$

$$\theta = [\, Q \mapsto T, P \mapsto T, R \mapsto F \,]$$

DPLLの例(5)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

~~(S)~~

$(\neg S)$ **矛盾節**

$$\theta = [\, Q \mapsto T, P \mapsto T, R \mapsto F, S \mapsto T \,]$$

DPLLの例(6)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

$$(P \vee R)$$
$$(\neg P \vee R \vee S)$$
$$(R \vee \neg S)$$
$$(\neg P \vee \neg R)$$

$$\theta = [Q \mapsto T, P \mapsto T, \cancel{R \mapsto F, S \mapsto T}]$$

$$P \mapsto F$$

DPLLの例(7)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

$$(\cancel{P \vee R})$$

$$\cancel{(\neg P \vee R \vee S)}$$

$$(R \vee \neg S)$$

$$\cancel{(\neg P \vee \neg R)}$$

$$\theta = [Q \mapsto T, P \mapsto F]$$

DPLLの例(8)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

~~(R)~~

~~$(R \vee \neg S)$~~

$\theta = [\, Q \mapsto T, P \mapsto F, R \mapsto T]$

DPLLの例(8)

1. 単位節がある時、単位伝搬を行い、CNFを単純化
2. 単位節がない時、まだ真理値が割り当てられていない命題変数を１つ取り、その真理値をT（あるいはF）にした割り当てをθに追加して、CNFを単純化
3. 上記を繰り返して、CNFの節がなくなったら（すべての節が真になって削除されたら） satを返して終了
4. 上記を繰り返して、CNFに矛盾節が生じたら、前回の２の選択までバックトラックして、T/Fを反転させた割り当てにして続行。
5. バックトラックするポイントがない（２の選択の全ての場合を尽くした）なら、unsatを返して終了。

CNF（以下の節を∧で結合したもの）

$\theta = [Q \mapsto T, P \mapsto F, R \mapsto T]$

satを返して終了
割り当ては$\theta + [S \mapsto T]$
（この場合、Sの真理値は何でもよい）

練習問題６.

以下の節を∧でつないだCNFに対して、「このテキストで紹介したアルゴリズム」を走らせなさい。

$$(\neg P \vee Q \vee R)$$
$$(P \vee R \vee S)$$
$$(P \vee R \vee \neg S)$$
$$(P \vee \neg R \vee S)$$
$$(P \vee \neg R \vee \neg S)$$
$$(\neg Q \vee \neg R \vee S)$$
$$(\neg P \vee Q \vee \neg R)$$
$$(\neg P \vee \neg Q \vee R)$$

7-3. SATソルバを用いた問題解決

SATソルバの急激な性能向上

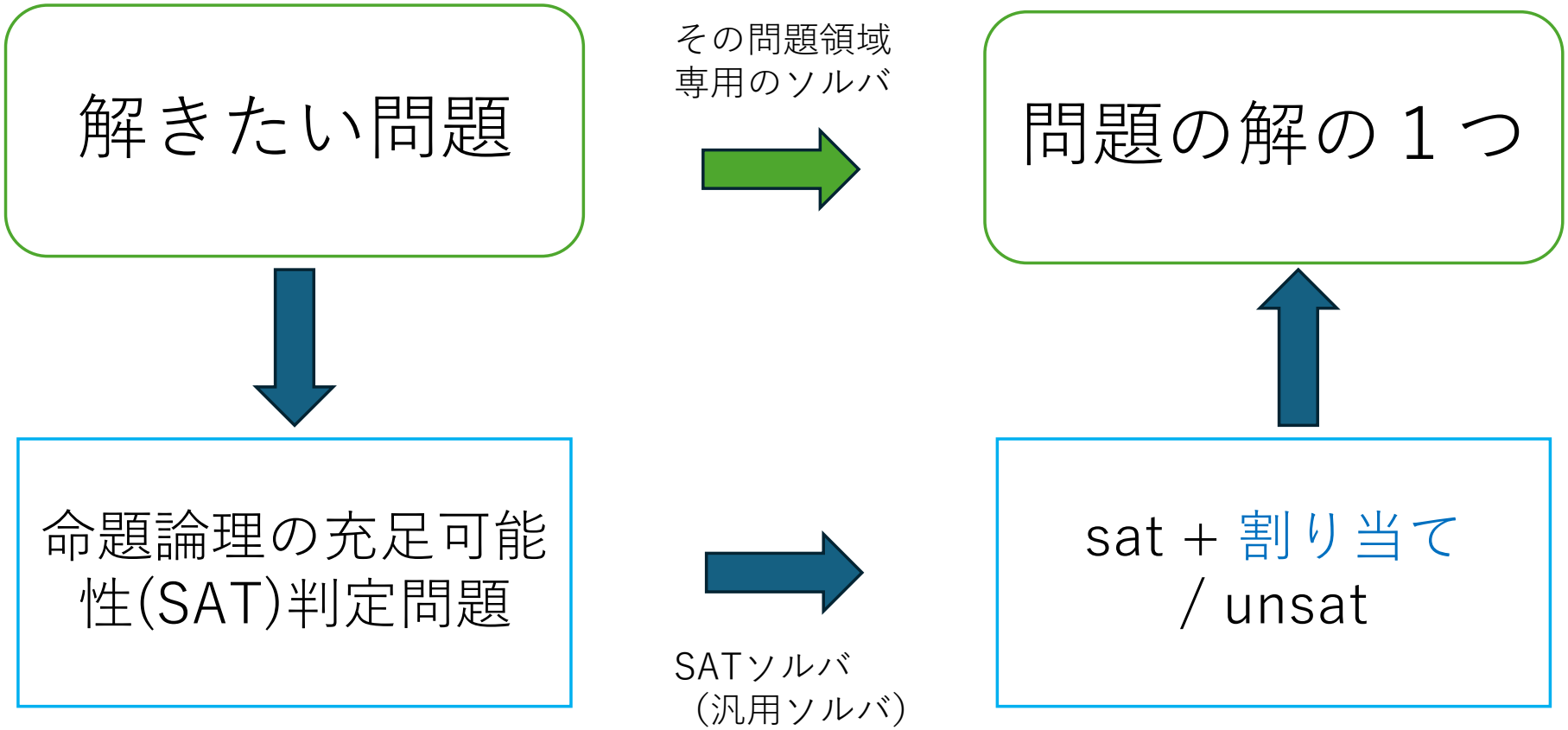
- NP完全問題であるにもかかわらず、多くの問題（人間が解きたい問題を符号化したものはたいてい含まれる）に対して、線形時間に近い性能（は、ちょっと言い過ぎかもしれないが。。。）でsat/unsat を判定できるSATソルバが出てきている。

- 原動力１．SAT competition **https://satcompetition.github.io/**
- 原動力２．予想外に多くの問題がSATに帰着できる。

性能向上のための様々なアルゴリズム上の改善

miniSAT solver: 短いソースコード（公開）で高い性能

SATへの帰着


解きたい問題
その問題領域
専用のソルバ
問題の解の１つ
命題論理の充足可能性(SAT)判定問題
SATソルバ
（汎用ソルバ）
sat + 割り当て
/ unsat

SATへの帰着例(1)

数独パズル

1	**4**		**2**
3		**4**	
	1	**2**	
			4

解の条件

- 初期配置を満たす。
- 各セルに１－４の数字が１個はいる
- 「行」方向は、１，２，３，４が１回ずつ
- 「列」方向は、１，２，３，４が１回ずつ
- 小行列は、１，２，３，４が１回ずつ

これを命題論理で表したい。

命題 $P(a,b,c)$：
第a行、第b列のセルに、数字cがはいる
かどうかを表す命題

SATへの帰着例(2)

数独パズル

1	**4**		**2**
3		**4**	
	1	**2**	
			4

命題 $P(a,b,c)$を命題変数と見なす。

- 初期配置を満たす。
  - $P(1,1,1) \wedge P(1,2,4) \wedge P(2,1,3) \wedge \cdots$
- 各セルに１－４の数字が１個はいる
  - $\forall x \forall y \exists z\, P(x,y,z)$　[述語論理]
  - $P(1,1,1) \vee P(1,1,2) \vee P(1,1,3) \vee P(1,1,4)$
  - $\forall x \forall y \forall z \forall w\, (P(x,y,z) \wedge P(x,y,w) \supset z = w)$ [述語論理]
  - $\big(\neg P(1,1,1) \vee \neg P(1,1,2)\big) \wedge \big(\neg P(1,1,1) \wedge \neg P(1,1,3)\big) \wedge \cdots$

SATへの帰着例(3)

数独パズル

1	4		2
3		4	
	1	2	
			4

命題 $P(a,b,c)$を命題変数と見なす。

- 「行」方向は、１，２，３，４が１回ずつ
- 「列」方向は、１，２，３，４が１回ずつ
- 小行列は、１，２，３，４が１回ずつ

練習問題７．「第１行には1,2,3,4が１回ずつ現れる」という条件を
命題変数 $P(a,b,c)$たちを使って表しなさい。
ただし、他の条件は、すべて定式化できている（それらとの論理積を取る）ことを前提としてよい。

SATソルバの利用

良いSATソルバを手軽に使えるサイトの例
https://www.msoos.org/cryptominisat/

```
p cnf 3 2
1 2 0
-3 -1 -2 0
```

DIMACS format

3	命題変数の数
2	節の数
1 2 0	節 $(P1 \vee P2)$
-3 - 1 -2 0	節 $(\neg P3 \vee \neg P1 \vee \neg P2)$

練習問題８.

- 練習問題６のCNFを DIMACS formatにして、SATソルバで解きなさい。

- 余力がある人は、４ｘ４数独の条件をすべて命題論理式として定式化して、前のページにある数独をSATソルバで解きなさい。

まとめ

- 命題論理に対する充足可能性判定問題(CNF_SAT問題)
  - 理論：NP完全
  - 現実：多くの場合に高速なソルバが開発されている。
  - SATソルバは、satを返すときに「割り当て」も返す。
    - 「条件をすべて満たす解を1つ求める」形の問題を、SATに帰着できる。
  - 個々の問題領域専用のアルゴリズムを開発するのではなく、問題領域をSATに帰着してSATソルバで解くので十分、という場合が多数ある。

※SATソルバは通常、unsatのときには「なぜunsatか」の証明は返してくれない。命題論理の証明を返すためには、自動証明器を使う方がよい。[Wang's algorithm]

※SMTソルバ：　SATソルバの基本アルゴリズムに「理論(theory)」を上載せして構築されたソルバ。Z3など非常に強力なものが出てきて、多方面で使われている。

本日の課題

以下のいずれかのパターンで、レポートを作成しなさい。

- 練習問題５と練習問題８　（両方やること）
- 練習問題６　（１つだけでよい）

- 練習問題５，６，８すべてを解答してもよい。
- ここで指定していない練習問題等については、期末の確認レポートの題材にする予定である。