

# SAT ソルバーと利用技術

宋 剛秀 [▶ web](#)

神戸大学 DX・情報統括本部

2023年9月14日

電子情報通信学会 ソサイエティ大会

AT-1：組合せ論と情報理論－最新動向と展望－

本講演では、組合せ問題の求解技術として SAT ソルバーとその利用技術について説明する。

## AT-1 セッションの話題リスト

- 1 符号・マトロイドやグループテストの回顧と最新トピック
- 2 情報理論分野の人工知能への応用
- 3 組合せ問題の求解技術
- 4 情報理論に関連する組合せ構造の最近の研究進展

- ① (準備) SAT ソルバーと SAT 型システム
- ② 成功事例
- ③ 仕組みと求解性能
- ④ 充足可能性判定以外の機能
- ⑤ SAT 符号化
- ⑥ おわりに
- ⑦ 参考情報

- 1 (準備) SAT ソルバーと SAT 型システム
- 2 成功事例
- 3 仕組みと求解性能
- 4 充足可能性判定以外の機能
- 5 SAT 符号化
- 6 おわりに
- 7 参考情報

**SAT 問題** (Boolean satisfiability testing problem) は、与えられた命題論理式を真にする値割当てが存在するか否かを判定する問題である。

- NP-完全であることが最初に証明された問題である [Cook, 1971] .
- 計算理論にとって中心的である [Garey+, 1979] .

## SAT 問題 (続き)

具体的な SAT 問題例 (SAT Problem Instances) は, 連言標準形 (CNF) の命題論理式 (CNF 式) で与えられる.

### CNF 式

- **CNF 式** は, 複数の節の論理積 (連言) である.
- **節** (clause) は複数のリテラルの論理和 (選言) である.
- **リテラル** (literal) は, 命題変数かあるいはその否定である.

ソフトウェアの標準的フォーマットとしては DIMACS CNF [web](#) が用いられる.

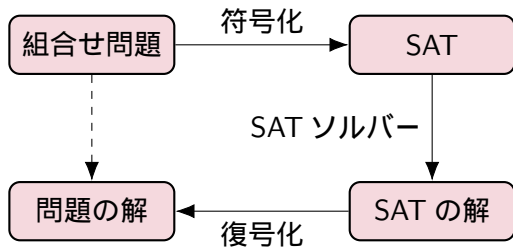
```
p cnf 3 4 ; 命題変数の数_節の数
1 2 3 0 ;  $p_1 \vee p_2 \vee p_3$ 
-1 -2 0 ;  $\neg p_1 \vee \neg p_2$ 
-1 -3 0 ;  $\neg p_1 \vee \neg p_3$ 
-2 -3 0 ;  $\neg p_2 \vee \neg p_3$ 
```

- **SAT ソルバー**は、与えられた SAT 問題が充足可能 (SAT) か充足不能 (UNSAT) かを判定するプログラムである。
- 通常、充足可能であればその値割当てを解として出力する。
- 確率的 SAT ソルバーは、SAT のみを判定する (UNSAT は判定できない)。
  - ▶ ローカルサーチアルゴリズムが用いられる。
- 系統的 SAT ソルバーは、SAT あるいは UNSAT を判定する。
  - ▶ ほとんどは **DPLL** を基にしたアルゴリズムを用いている。

# SAT 型システム

**SAT 型システム**とは与えられた問題を **SAT 符号化** により SAT へと変換し, SAT ソルバーを用いて解くシステムである<sup>†</sup>.

<sup>†</sup> 符号化と SAT ソルバーによる求解を繰り返すような場合も含む.





- ① (準備) SAT ソルバーと SAT 型システム
- ② 成功事例
- ③ 仕組みと求解性能
- ④ 充足可能性判定以外の機能
- ⑤ SAT 符号化
- ⑥ おわりに
- ⑦ 参考情報

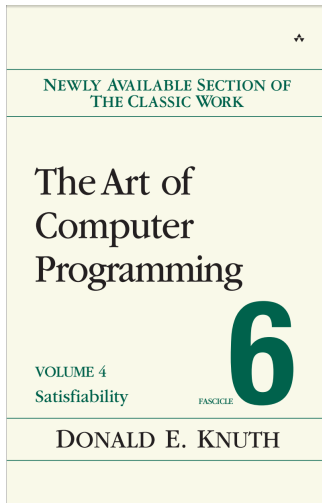
## 成功事例 (1): 推論基盤としての利用

- プランニング (SATPLAN, Blackbox) [Kautz+, 1992] [▶ web](#)
- 自動テストパターン生成 [Larrabee, 1992]
- ジョブショップスケジューリング [Crawford+, 1994]
- 有界モデル検査 [Biere, 2009]
- ソフトウェア検証 (Alloy) [Jackson, 2006] [▶ web](#)
- 書換えシステム (AProVE) [Jürgen+, 2004] [▶ web](#)
- インテル社の i7 プロセッサの検証 [Kaivola+, 2009]
- Eclipse のコンポーネント間の依存解析 [Le Berre+, 2009] [▶ web](#)
- 解集合プログラミング (clasp) [Gebser+, 2012] [▶ web](#)
- Linux のパッケージマネージャである DNF の依存性解決 [▶ web](#)
- 制約プログラミング (Sugar) [Tamura+, 2009] [▶ web](#)
  - ▶ オープンジョブショップスケジューリング問題の未解決問題の求解 [Tamura+, 2009]
  - ▶ パッキング配列問題の未解決問題の求解 [則武+, 2013]
- この他ペトリネットの検証, システム生物学, グラフ理論の問題などにも応用されている [Ogata+, 2004, Soh+, 2010, Soh+, 2014] .

## 成功事例 (2): 未解決問題への適用

- 2014年2月
  - ▶ Erdős Discrepancy Conjecture の  $C = 2$  の場合の解決 [Konev+, 2014]
- 2015年10月
  - ▶ Ramsey Number  $R(4, 3, 3) = 30$  が SAT ソルバーによって求められる .
- 2016年5月
  - ▶ Boolean Pythagorean Triple 問題の解決 [Heule+, 2016]
    - ★ 発表当時 Nature 誌へこの話題が掲載される [▶ web](#)
- 2017年2月
  - ▶ 同一の SHA-1 (Security Hash Algorithm 1) ハッシュ値をもつ 2 つのファイルを作成する過程で SAT ソルバーが使われる [▶ web](#)
- 2020年7月
  - ▶ ケラー予想の次元 7 の場合が成り立つことを SAT ソルバーで証明 [Brakensiek+, 2020] [▶ web](#)
- 2023年6月
  - ▶ Kochen–Specker vector system の下限が 24 に更新される [Li+, 2023].

## 成功事例 (3): TAOCP への掲載



→ Knuth 先生と食事した  
写真 (@Trento, SAT2012)

- スタンフォード大学 Knuth 教授著 “**The Art of Computer Programming**” の 4 巻分冊 6 [Knuth, 2015] では SAT が 300 ページ以上にわたって取り上げられている。
- 序文には「SAT 問題は、非常に多くの問題を解くためのキーであることから、明らかに **killer app** である」と述べられている。



# SAT ソルバーが利用される理由

- 高い求解性能
  - ▶ 2000 年頃に高速な **CDCL ソルバー** が提案されたこと
  - ▶ 研究の蓄積と **国際競技会** により継続的に性能が改善していること
- 高い信頼性
  - ▶ ソフトウェアの継続した更新により動作が堅牢になっていること
  - ▶ 入力 that 充足可能である場合は解を出力し, 入力 that 充足不能である場合は検証可能な **証明** を出力できること
- 汎用性・記述力
  - ▶ NP 完全問題である SAT 問題が入力であること
  - ▶ 複雑な制約でも連言標準形に符号化 (**SAT 符号化**) する方法が研究されていること

# 目次

- 1 (準備) SAT ソルバーと SAT 型システム
- 2 成功事例
- 3 仕組みと求解性能**
- 4 充足可能性判定以外の機能
- 5 SAT 符号化
- 6 おわりに
- 7 参考情報

# SAT ソルバーの仕組みの歴史

- 1960 年代
  - ▶ **DPLL** (Davis--Putnam--Logemann--Loveland) [Davis+, 1962]
- 1990 年代
  - ▶ **CDCL** (Conflict Driven Clause Learning)  
[Bayardo Jr.+, 1997, Marques-Silva+, 1999]
- 2000 年以降
  - ▶ 変数選択ヒューリスティック VSIDS [Moskewicz+, 2001]
  - ▶ 2 リテラルウォッチ [Moskewicz+, 2001]
  - ▶ リスタート [Luby+, 1993, Selman+, 1996, Eén+, 2003]
  - ▶ Phase Saving [Pipatsrisawat+, 2007]
  - ▶ 学習節の評価尺度 [Audemard+, 2009, 鍋島+, 2012]

# SAT ソルバーの仕組みの歴史

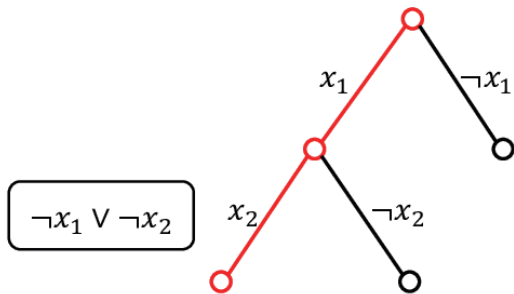
- 1960 年代
  - ▶ **DPLL** (Davis--Putnam--Logemann--Loveland) [Davis+, 1962]
- 1990 年代
  - ▶ **CDCL** (Conflict Driven Clause Learning)  
[Bayardo Jr.+, 1997, Marques-Silva+, 1999]
- 2000 年以降
  - ▶ 変数選択ヒューリスティック VSIDS [Moskewicz+, 2001]
  - ▶ 2 リテラルウォッチ [Moskewicz+, 2001]
  - ▶ リスタート [Luby+, 1993, Selman+, 1996, Eén+, 2003]
  - ▶ Phase Saving [Pipatsrisawat+, 2007]
  - ▶ 学習節の評価尺度 [Audemard+, 2009, 鍋島+, 2012]

SAT ソルバーの基本動作を簡単な例を用いて説明する。  
より詳しく知りたい方は以下を参照ください。

- 高速 SAT ソルバーの原理 (解説論文) [鍋島+, 2010] [▶ web](#)

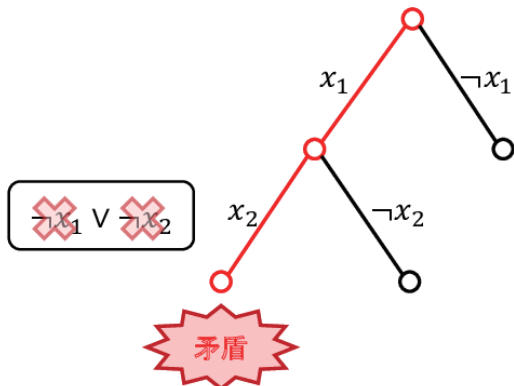


# SAT ソルバーの基本 二分木の深さ優先探索



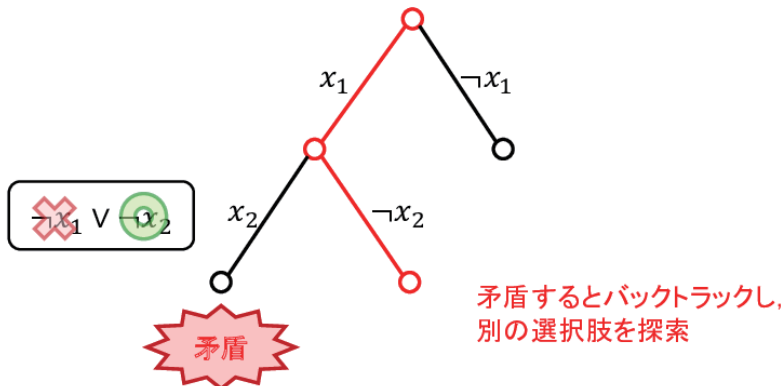
(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 二分木の深さ優先探索



(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

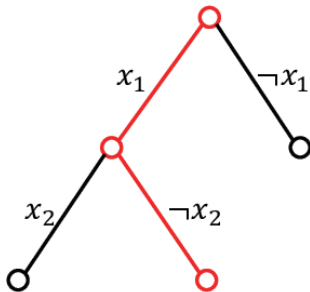
# SAT ソルバーの基本 二分木の深さ優先探索



(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 単位伝搬

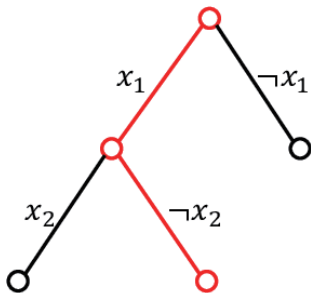
- 単位伝搬により必然的な真偽値割当を検出
- 探索木の深さを低減



(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 単位伝搬

- 単位伝搬により必然的な真偽値割当を検出
- 探索木の深さを低減

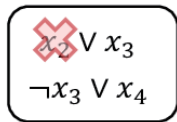
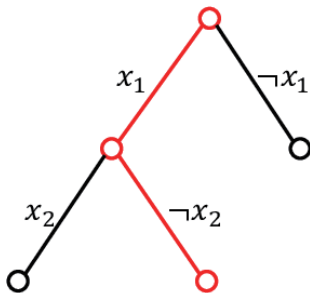


$$\begin{array}{l} x_2 \vee x_3 \\ \neg x_3 \vee x_4 \end{array}$$

(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 単位伝搬

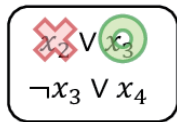
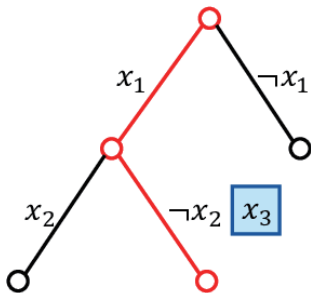
- 単位伝搬により必然的な真偽値割当を検出
- 探索木の深さを低減



(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 単位伝搬

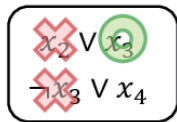
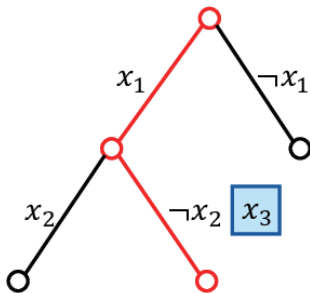
- 単位伝搬により必然的な真偽値割当を検出
- 探索木の深さを低減



(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 単位伝搬

- 単位伝搬により必然的な真偽値割当を検出
- 探索木の深さを低減

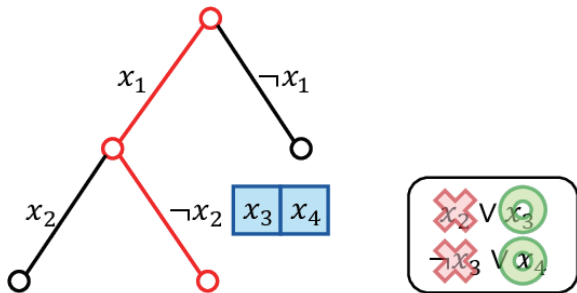


(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.



# SAT ソルバーの基本 単位伝搬

- 単位伝搬により必然的な真偽値割当を検出
- 探索木の深さを低減

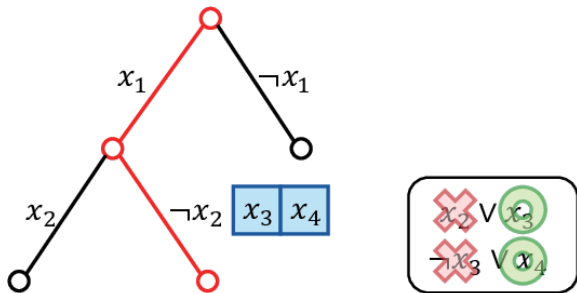


$x_1 \wedge \neg x_2$  の下では  $x_3 = x_4 = \text{真}$  が確定

(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 単位伝搬

- 単位伝搬により必然的な真偽値割当を検出
- 探索木の深さを低減

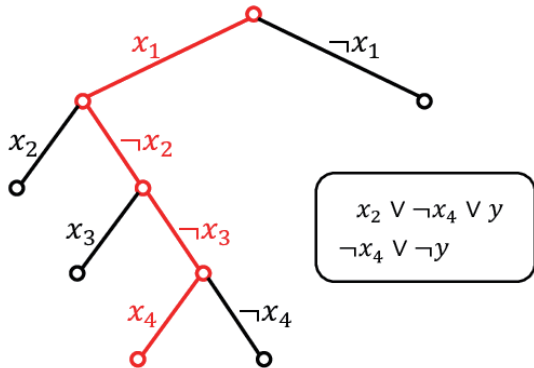


$x_1 \wedge \neg x_2$  の下では  $x_3 = x_4 = \text{真}$  が確定

二分木の深さ優先探索 + 単位伝搬  $\doteq$  **DPLL**

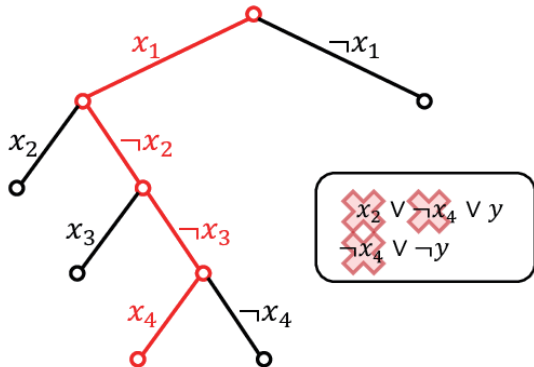
(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 節学習 (CDCL)



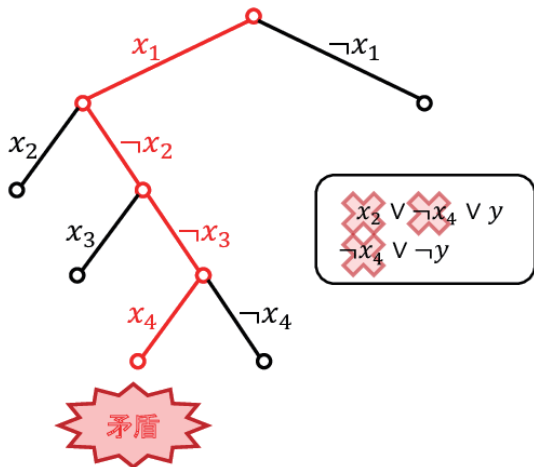
(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 節学習 (CDCL)



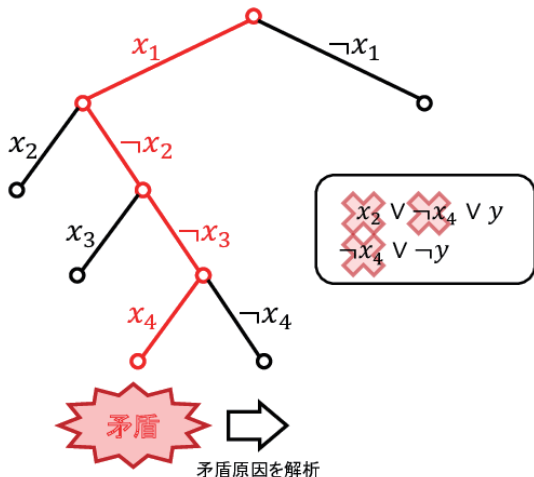
(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 節学習 (CDCL)



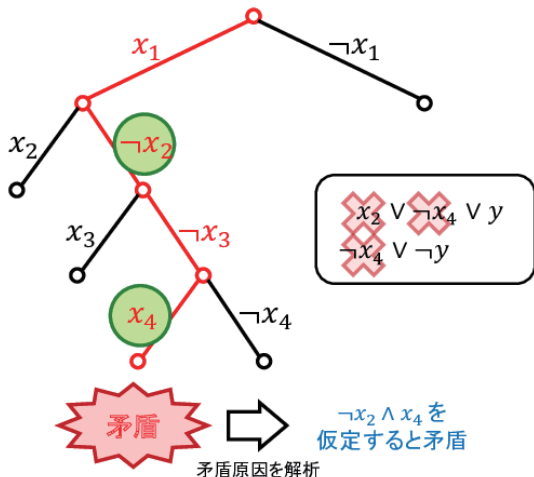
(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 節学習 (CDCL)



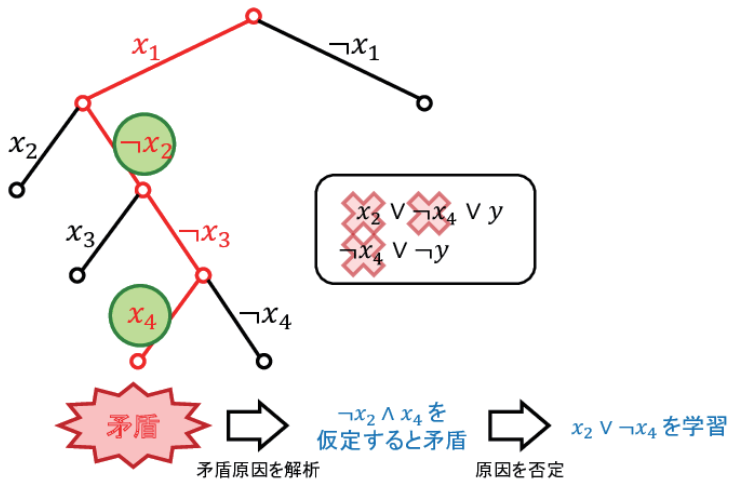
(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 節学習 (CDCL)



(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

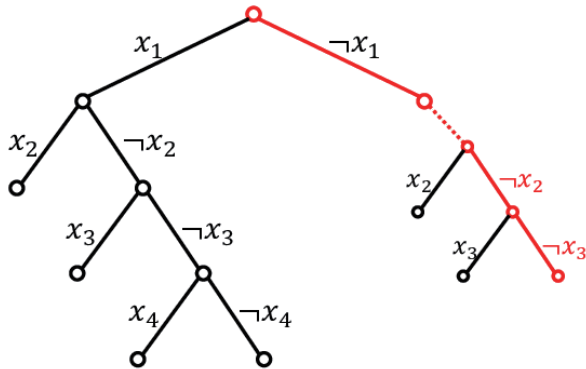
# SAT ソルバーの基本 節学習 (CDCL)



(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.



# SAT ソルバーの基本 節学習 (CDCL)



**矛盾**



矛盾原因を解析

$\neg x_2 \wedge x_4$  を  
仮定すると矛盾

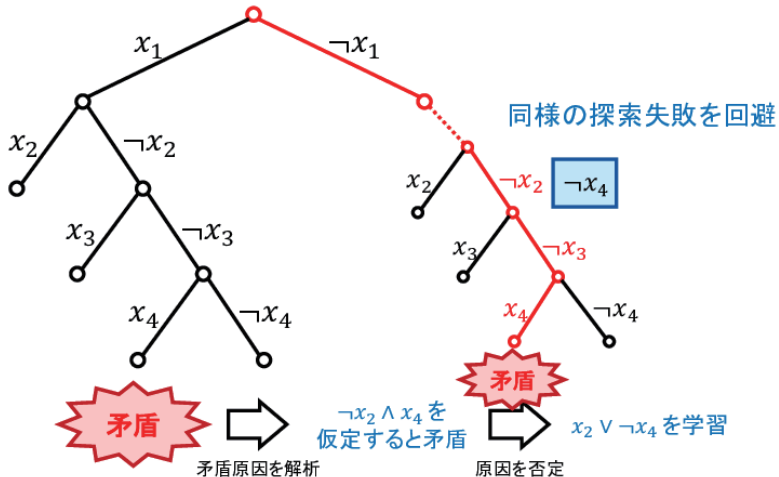


原因を否定

$x_2 \vee \neg x_4$  を学習

(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの基本 節学習 (CDCL)



(次より引用) 鍋島英知, SAT ソルバーの最近の技術動向, 2016 年度人工知能学会全国大会  
オーガナイズドセッション「OS-2 SAT 技術の理論, 実装, 応用」, 1D4-OS-02a-1, 2016.

# SAT ソルバーの仕組みの歴史

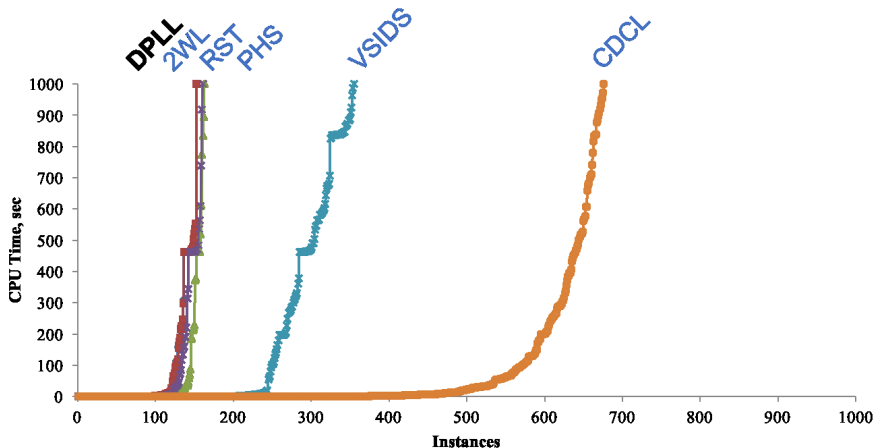
- 1960 年代
  - ▶ **DPLL** (Davis--Putnam--Logemann--Loveland) [Davis+, 1962]
- 1990 年代
  - ▶ **CDCL** (Conflict Driven Clause Learning) [Bayardo Jr.+, 1997, Marques-Silva+, 1999]
- 2000 年以降
  - ▶ 変数選択ヒューリスティック VSIDS [Moskewicz+, 2001]
  - ▶ 2 リテラルウォッチ [Moskewicz+, 2001]
  - ▶ リスタート [Luby+, 1993, Selman+, 1996, Eén+, 2003]
  - ▶ Phase Saving [Pipatsrisawat+, 2007]
  - ▶ 学習節の評価尺度 [Audemard+, 2009, 鍋島+, 2012]

どの技術がどのくらい効いているのか?

# 高速化技術の貢献度 (Katebi 他による) [Katebi+, 2011]

- 2WL: 2 リテラルウォッチ, RST: リスタート, PHS: Phase Saving

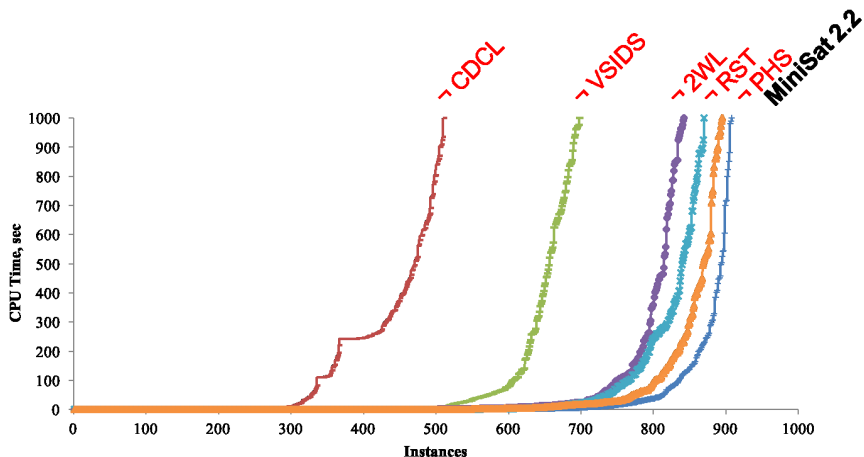
青字: DPLL にその技術を追加したソルバー



# 高速化技術の貢献度 (Katebi 他による) [Katebi+, 2011]

- 2WL: 2 リテラルウォッチ, RST: リスタート, PHS: Phase Saving

赤字: MiniSat 2.2 からその技術を削除したソルバー

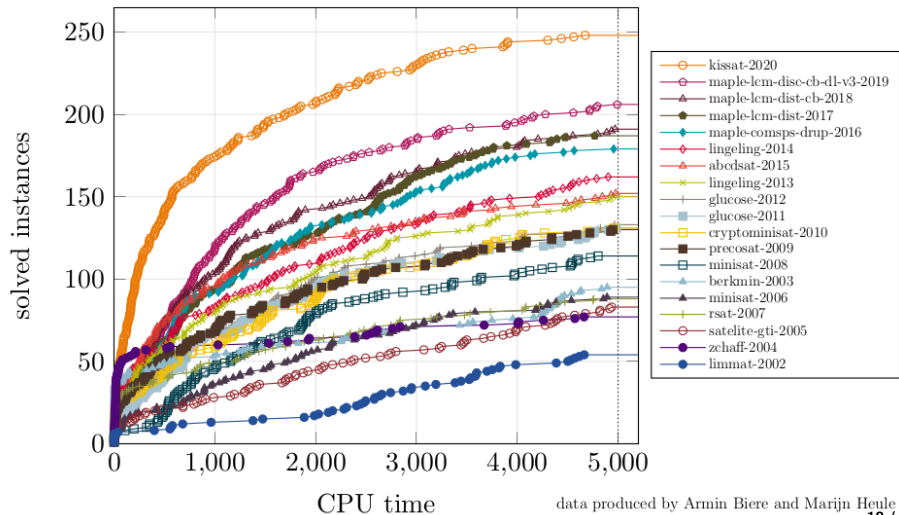


a

# 求解性能の進化

Armin Biere 先生の Twitter (@ArminBiere) より (2020年7月29日)

SAT Competition Winners on the SC2020 Benchmark Suite



# SAT ソルバーの国際競技会 (SAT Competition)

参考資料: SAT Competition 2023 [▶ web](#)

## 目的

- 挑戦的なベンチマークを見つけること
- SAT ソルバー開発の促進すること
- 現状のソルバーのスナップショット評価を行うこと

## 歴史

1992 年から **SAT ソルバーの国際競技会** (SAT Competition, SATRace, SAT Challenge) が開催されている .

- 3 competitions in the 90s (1992, 1993, 1996)
- 15 SAT Competitions (2002-)
- 5 SAT Races (2006, 2008, 2010, 2015, 2019)
- 1 SAT Challenge (2012)

# SAT ソルバーの国際競技会

## 開催方法

- 参加者は自分の SAT ソルバーを指定されたサーバにアップロードする。
- 参加者によるテスト期間後，最終版がオーガナイザ側で実行されて，SAT 国際会議の時に結果が公表される。

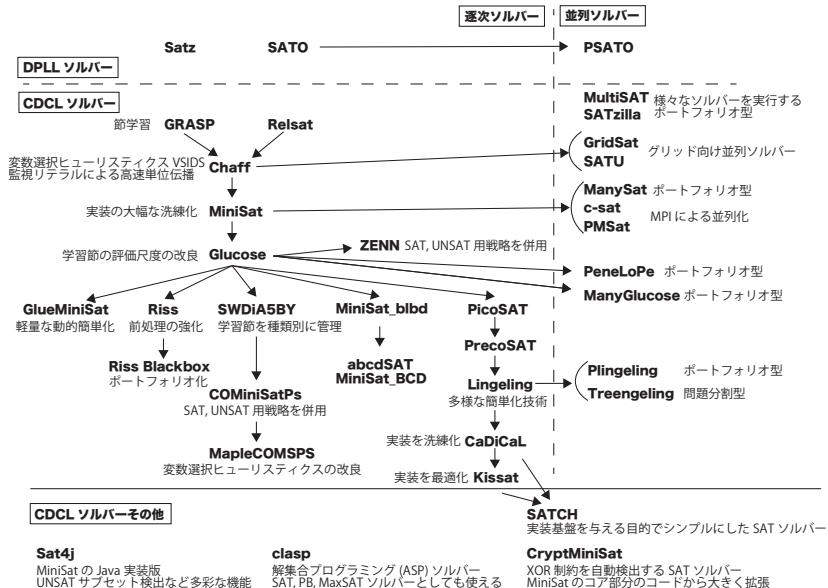
## 参加ソルバーのソースコードが原則公開

- 最新の技術・実装を誰でも入手でき，自分のアイデアを上乗せできる
- 新規参入の敷居が低くなる
- 競争が激しくなる
- ほぼ毎年優勝ソルバーが変わり，優勝ソルバーには何らかの新しいアイデアが入っている。

この活発な競技会が SAT ソルバーの性能向上とコミュニティの醸成に大きく貢献している！



# 主な SAT ソルバー: [番原+, 2016] の図 3 に追記



# SAT ソルバーの機能

SAT ソルバーの国際競技会は単純な求解性能の進化だけでなく、様々な競技カテゴリを設けることで **SAT ソルバーの機能の拡張と標準化**にも貢献している。

- **Certified UNSAT** トラック

- ▶ 問題が UNSAT の時に判定結果だけでなく、その証明を DRAT 形式 [Wetzler+, 2014] で出力・検証する必要があるトラック。

- **Minimal Unsatisfiable Subset** (MUS) トラック

- ▶ 問題が UNSAT の時に判定結果だけでなく、与えられた問題が UNSAT となる極小の部分節集合を 1 つ出力する必要があるトラック。

- **インクリメンタル SAT** トラック

- ▶ 少しずつ異なる複数の SAT 問題を連続的に解く性能を競うトラック。
- ▶ この機能のための標準 API である IPASIR [▶ web](#) の仕様が公開されている。

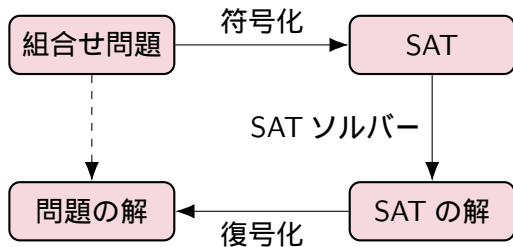
# SAT ソルバーの探索の様子

- SATracer [堀岡+, 2021] [▶ web](#)

# 目次

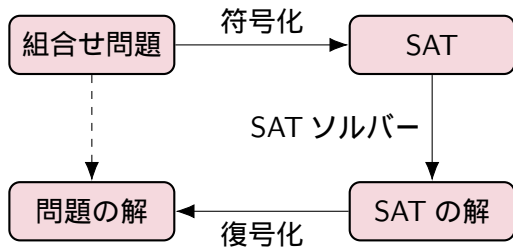
- ① (準備) SAT ソルバーと SAT 型システム
- ② 成功事例
- ③ 仕組みと求解性能
- ④ 充足可能性判定以外の機能
- ⑤ **SAT 符号化**
- ⑥ おわりに
- ⑦ 参考情報

# SAT ソルバーを応用するには?



- ここまで SAT ソルバーの求解性能や機能の進化を説明した。
- しかし、解きたい問題が CNF 式で記述されていることは非常に少ないため、SAT ソルバーを利用するには、与えられた問題を SAT ソルバーの入力に変換する **SAT 符号化** が必須になる。

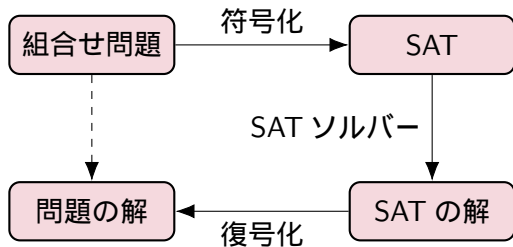
# SAT ソルバーを応用するには?



- ここまで SAT ソルバーの求解性能や機能の進化を説明した。
- しかし、解きたい問題が CNF 式で記述されていることは非常に少ないため、SAT ソルバーを利用するには、与えられた問題を SAT ソルバーの入力に変換する **SAT 符号化** が必須になる。

CNF 式にさえすれば良い? SAT ソルバーが勝手に高速に解く?

# SAT ソルバーを応用するには？



- ここまで SAT ソルバーの求解性能や機能の進化を説明した。
- しかし、解きたい問題が CNF 式で記述されていることは非常に少ないため、SAT ソルバーを利用するには、与えられた問題を SAT ソルバーの入力に変換する **SAT 符号化** が必須になる。

CNF 式にさえすれば良い？ SAT ソルバーが勝手に高速に解く？

No!

## 例えば，線形比較の符号化の場合

$$\sum_{i=1}^n a_i x_i \leq k \quad (a_i > 0)$$

- 全ての  $a_i$  が 1,  $x_i$  が 0-1 変数の場合 (基数制約) の符号化



## 例えば，線形比較の符号化の場合

$$\sum_{i=1}^n a_i x_i \leq k \quad (a_i > 0)$$

- 全ての  $a_i$  が 1,  $x_i$  が 0-1 変数の場合 (基数制約) の符号化
  - ▶ この単純な場合でも，数多くの SAT 符号化が研究されている!  
[Warners, 1998, Bailleux+, 2003, Sinz, 2005, Chen, 2010, Roberto+, 2011, Abío+, 2013, Nguyen+, 2015]

## 例えば，線形比較の符号化の場合

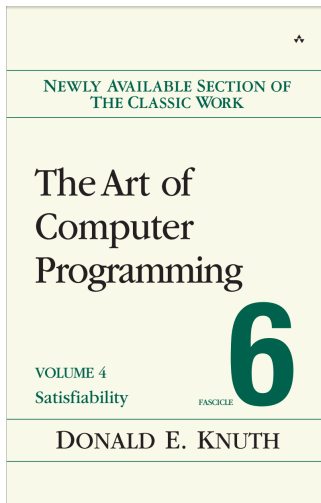
$$\sum_{i=1}^n a_i x_i \leq k \quad (a_i > 0)$$

- 全ての  $a_i$  が 1,  $x_i$  が 0-1 変数の場合 (基数制約) の符号化
  - ▶ この単純な場合でも，数多くの SAT 符号化が研究されている!  
[Warners, 1998, Bailleux+, 2003, Sinz, 2005, Chen, 2010, Roberto+, 2011, Abío+, 2013, Nguyen+, 2015]
- $x_i$  が 0-1 変数の場合 (擬似ブール制約) の符号化 [Eén+, 2006, Bailleux+, 2006, Bailleux+, 2009, Roussel+, 2009b, Abío+, 2012, Ogawa+, 2013, Tamura+, 2013, Philipp+, 2015, Sakai+, 2015]

## 例えば，線形比較の符号化の場合

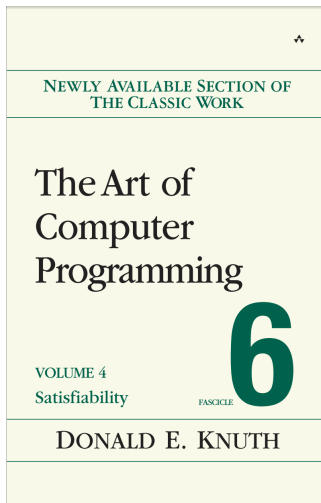
$$\sum_{i=1}^n a_i x_i \leq k \quad (a_i > 0)$$

- 全ての  $a_i$  が 1,  $x_i$  が 0-1 変数の場合 (基数制約) の符号化
  - ▶ この単純な場合でも，数多くの SAT 符号化が研究されている!  
[Warners, 1998, Bailleux+, 2003, Sinz, 2005, Chen, 2010, Roberto+, 2011, Abío+, 2013, Nguyen+, 2015]
- $x_i$  が 0-1 変数の場合 (擬似ブール制約) の符号化 [Eén+, 2006, Bailleux+, 2006, Bailleux+, 2009, Roussel+, 2009b, Abío+, 2012, Ogawa+, 2013, Tamura+, 2013, Philipp+, 2015, Sakai+, 2015]
- $x_i$  が有限整数領域上のドメインを持つ 場合の符号化 [de Kleer, 1989, Kasif, 1990, Iwama+, 1994, Walsh, 2000, Gent, 2002, Gavanelli, 2007, Prestwich, 2009, Tamura+, 2009, 田村+, 2010, 丹生+, 2013, Soh+, 2017]



The Art of Computer Programming 4 巻分冊 6 [Knuth, 2015] の 97 ページには以下のように記載されている .

“Thus the art of problem encoding turns out to be just as important as the art of devising algorithms for satisfiability.”



The Art of Computer Programming 4 巻分冊 6 [Knuth, 2015] の 97 ページには以下のように記載されている .

“Thus the art of problem encoding turns out to be just as important as the art of devising algorithms for satisfiability.”

符号化の研究は SAT 分野において重要なテーマの 1 つ!

## (例) 直接符号化と順序符号化

### 直接符号化 [de Kleer, 1989]

各整数変数  $x$  とそのドメインの各値  $i$  に対して,  $x = i$  を表す命題変数  $p_{x=i}$  を用いる. 各整数変数  $x$  に対して用いる命題変数

$$p_{x=i} \quad (lb(x) \leq i \leq ub(x))$$

命題変数  $p_{x=i}$  が  $x = i$  の時かつその時に限って真となるように, 以下の at-least-one 節と at-most-one 節を追加する.

$$p_{x=lb(x)} \vee \cdots \vee p_{x=ub(x)}$$

$$\neg p_{x=i} \vee \neg p_{x=j}$$

$$(lb(x) \leq i < j \leq ub(x))$$

### 順序符号化 [Tamura+, 2009]

各整数変数  $x$  とそのドメインの各値  $i$  に対して,  $x \leq i$  を表す命題変数  $p_{x \leq i}$  を用いる. 各整数変数  $x$  に対して用いる命題変数

$$p_{x \leq i} \quad (lb(x) \leq i < ub(x))$$

命題変数  $p_{x \leq i}$  が  $x \leq i$  の時かつその時に限って真となるように, 以下の節を追加する.

$$\neg p_{x \leq i-1} \vee p_{x \leq i} \\ (lb(x) < i < ub(x))$$

# 直接符号化 [de Kleer, 1989, Walsh, 2000]

制約  $x + y \leq 7$  ( $x, y \in \{2, 3, 4, 5, 6\}$ ) は, 違反点 (図中の  $\times$  点) を列挙することで以下の 15 節に符号化される .

$$\neg(p_{x=2} \wedge p_{y=6})$$

$$\neg(p_{x=3} \wedge p_{y=5})$$

$$\neg(p_{x=3} \wedge p_{y=6})$$

$$\neg(p_{x=4} \wedge p_{y=4})$$

$$\neg(p_{x=4} \wedge p_{y=5})$$

$$\neg(p_{x=4} \wedge p_{y=6})$$

$$\neg(p_{x=5} \wedge p_{y=3})$$

$$\neg(p_{x=5} \wedge p_{y=4})$$

$$\neg(p_{x=5} \wedge p_{y=5})$$

$$\neg(p_{x=5} \wedge p_{y=6})$$

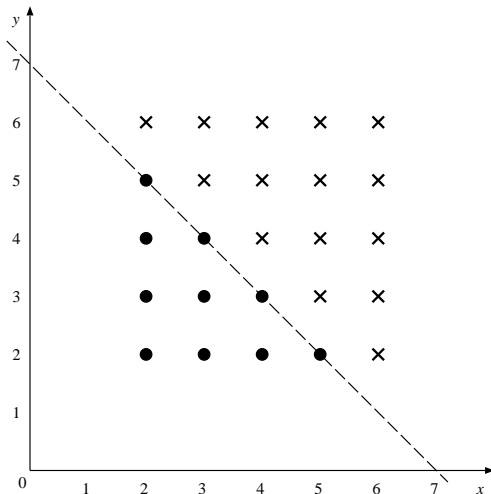
$$\neg(p_{x=6} \wedge p_{y=2})$$

$$\neg(p_{x=6} \wedge p_{y=3})$$

$$\neg(p_{x=6} \wedge p_{y=4})$$

$$\neg(p_{x=6} \wedge p_{y=5})$$

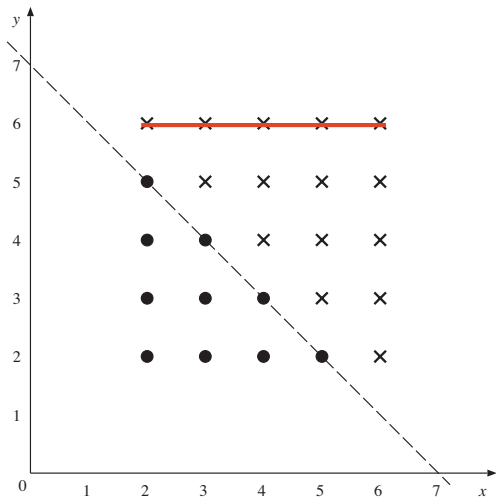
$$\neg(p_{x=6} \wedge p_{y=6})$$



# 順序符号化 [Tamura+, 2009]

制約  $x + y \leq 7$  ( $x, y \in \{2, 3, 4, 5, 6\}$ ) は, 違反する範囲を表すことで以下の5節に符号化される.

$$\begin{aligned} & \neg p_{y \geq 6} \\ & \neg(p_{x \geq 3} \wedge p_{y \geq 5}) \\ & \neg(p_{x \geq 4} \wedge p_{y \geq 4}) \\ & \neg(p_{x \geq 5} \wedge p_{y \geq 3}) \\ & \neg p_{x \geq 6} \end{aligned}$$

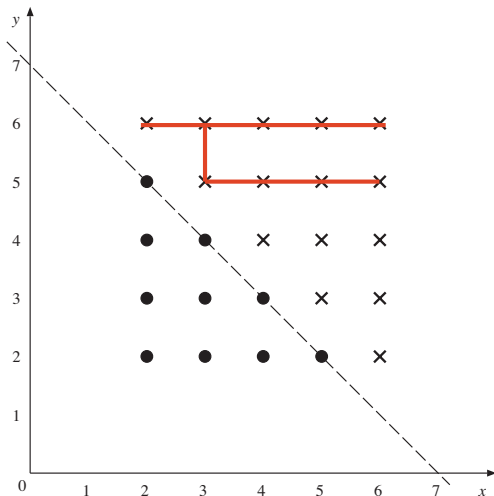




# 順序符号化 [Tamura+, 2009]

制約  $x + y \leq 7$  ( $x, y \in \{2, 3, 4, 5, 6\}$ ) は, 違反する範囲を表すことで以下の5節に符号化される.

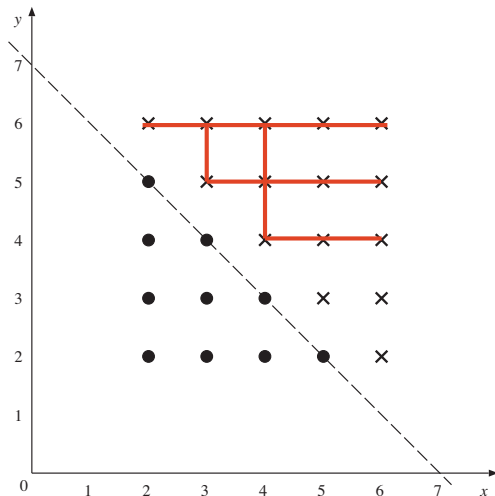
$$\begin{aligned} & \neg p_{y \geq 6} \\ & \neg(p_{x \geq 3} \wedge p_{y \geq 5}) \\ & \neg(p_{x \geq 4} \wedge p_{y \geq 4}) \\ & \neg(p_{x \geq 5} \wedge p_{y \geq 3}) \\ & \neg p_{x \geq 6} \end{aligned}$$



# 順序符号化 [Tamura+, 2009]

制約  $x + y \leq 7$  ( $x, y \in \{2, 3, 4, 5, 6\}$ ) は, 違反する範囲を表すことで以下の5節に符号化される.

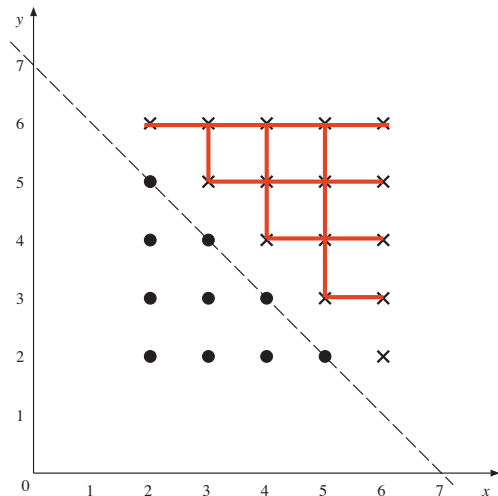
$$\begin{aligned} & \neg p_{y \geq 6} \\ & \neg(p_{x \geq 3} \wedge p_{y \geq 5}) \\ & \neg(p_{x \geq 4} \wedge p_{y \geq 4}) \\ & \neg(p_{x \geq 5} \wedge p_{y \geq 3}) \\ & \neg p_{x \geq 6} \end{aligned}$$



# 順序符号化 [Tamura+, 2009]

制約  $x + y \leq 7$  ( $x, y \in \{2, 3, 4, 5, 6\}$ ) は, 違反する範囲を表すことで以下の5節に符号化される.

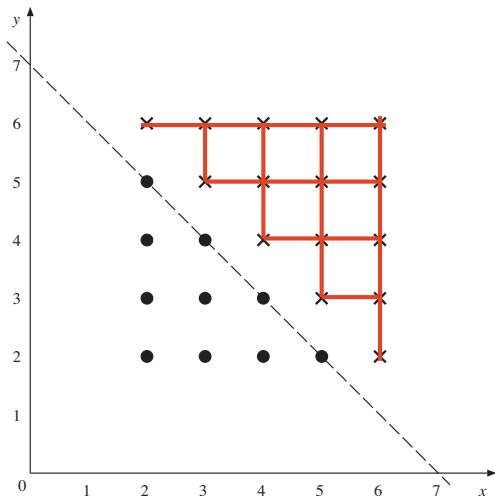
$$\begin{aligned} & \neg p_{y \geq 6} \\ & \neg(p_{x \geq 3} \wedge p_{y \geq 5}) \\ & \neg(p_{x \geq 4} \wedge p_{y \geq 4}) \\ & \neg(p_{x \geq 5} \wedge p_{y \geq 3}) \\ & \neg p_{x \geq 6} \end{aligned}$$



# 順序符号化 [Tamura+, 2009]

制約  $x + y \leq 7$  ( $x, y \in \{2, 3, 4, 5, 6\}$ ) は, 違反する範囲を表すことで以下の5節に符号化される.

$$\begin{aligned} & \neg p_{y \geq 6} \\ \neg & (p_{x \geq 3} \wedge p_{y \geq 5}) \\ \neg & (p_{x \geq 4} \wedge p_{y \geq 4}) \\ \neg & (p_{x \geq 5} \wedge p_{y \geq 3}) \\ & \neg p_{x \geq 6} \end{aligned}$$



# 順序符号化 [Tamura+, 2009]

制約  $x + y \leq 7$  ( $x, y \in \{2, 3, 4, 5, 6\}$ ) は, 違反する範囲を表すことで以下の5節に符号化される.

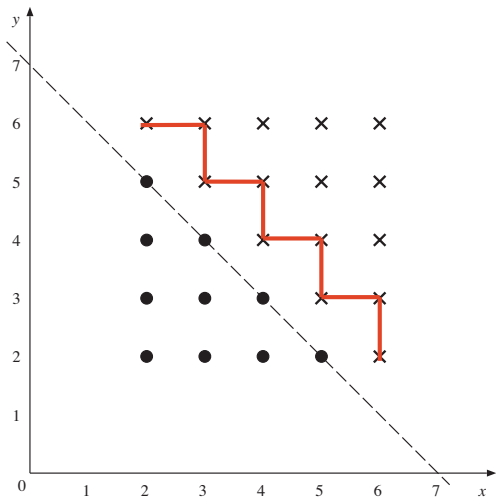
$$\neg p_{y \geq 6}$$

$$\neg(p_{x \geq 3} \wedge p_{y \geq 5})$$

$$\neg(p_{x \geq 4} \wedge p_{y \geq 4})$$

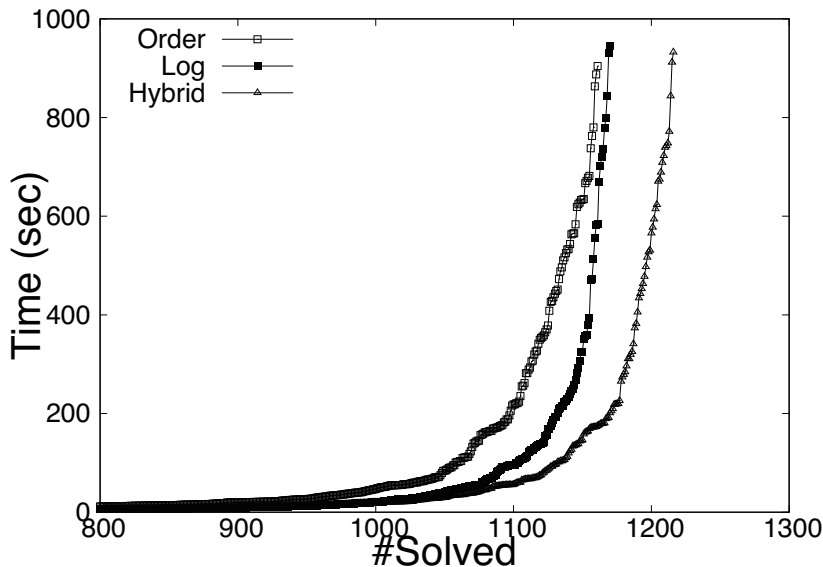
$$\neg(p_{x \geq 5} \wedge p_{y \geq 3})$$

$$\neg p_{x \geq 6}$$



- 順序符号化 [Tamura+, 2009] (実装: Sugar [▶ web](#))
  - ▶ 整数の順序関係を用いており, 線形比較  $\sum a_i x_i \geq b$  の符号化に適している.
  - ▶ 順序符号化を実装した Sugar は 2008 年, 2009 年に CSP ソルバー競技会のグローバル部門で優勝した.
  - ▶ 理論的にも他の符号化にはない良い性質を持っていることが証明されている [Petke+, 2011].
  - ▶ しかし, 問題の規模が大きくなると性能が悪くなる.
- 対数符号化 [Iwama+, 1994]
  - ▶ 整数の 2 進数表現を用いておりコンパクトな SAT 符号化が可能だが, 線形比較について一般に順序符号化より性能が悪い.
  - ▶ 問題の規模が大きくても解くことができる.
- ハイブリッド符号化 [Soh+, 2017] (実装: Diet-Sugar [▶ web](#))
  - ▶ **順序符号化と対数符号化を融合した SAT 符号化法**.
  - ▶ 各整数変数は順序符号化もしくは対数符号化のどちらか一つで符号化され, 各制約は両方の符号化変数を含むことができる.

# 順序符号化, 対数符号化, ハイブリッド符号化の比較 (CSC2009 ベンチマーク 1458 問で評価) [Soh+, 2017]

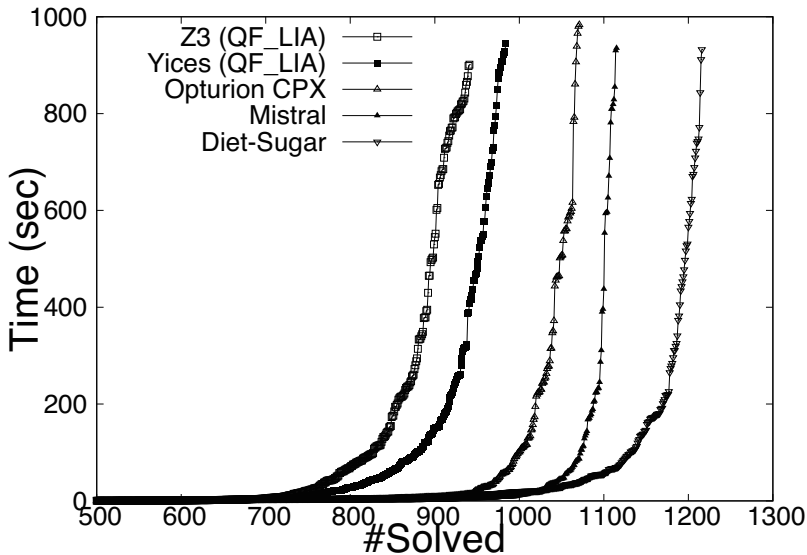


## 比較したソルバー

- **Mistral** (version 1.550)
  - ▶ ピュアな (SAT 型でない) CSP ソルバー . 2009 年の CSP ソルバー競技会の優勝ソルバー .
- **Opturion CPX** (version 1.0.2)
  - ▶ CSP ソルバーと SAT ソルバーのハイブリッドソルバー . 2015 年の Minizinc チャレンジ “Fixed/Free” カテゴリの優勝ソルバー .
- **Yices** (version 2.4.2) — QF\_LIA
  - ▶ 2015 年の SMT ソルバー競技会 “Quantified Free Linear Integer Arithmetic (QF\_LIA)” 部門の優勝ソルバー .
- **z3** (version 4.3.2) — QF\_LIA
  - ▶ 2015 年の SMT ソルバー競技会 QF\_LIA 部門の優勝ソルバー .



# Diet-Sugar と CSP/SMT ソルバーとの比較 (CSC2009 ベンチマーク 1458 問で評価) [Soh+, 2017]



# おわりに

- SAT ソルバーと利用技術について以下の説明を行った。
  - ▶ 成功事例と話題
  - ▶ SAT ソルバーの仕組みと求解性能と機能の進化
  - ▶ SAT ソルバー利用に必須である符号化技術

## SAT ソルバーが利用される理由

- 高い求解性能
  - ▶ 2000 年頃に高速な **CDCL ソルバー** が提案されたこと
  - ▶ 研究の蓄積と **国際競技会** により継続的に性能が改善していること
- 高い信頼性
  - ▶ ソフトウェアの継続した更新により動作が堅牢になっていること
  - ▶ 入力が充足可能である場合は解を出力し，入力が充足不能である場合は検証可能な **証明を出力** できること
- 汎用性・記述力
  - ▶ NP 完全問題である SAT 問題を入力にとれること
  - ▶ 複雑な制約でも連言標準形に符号化 (SAT 符号化) する方法が研究されていること

- SAT ソルバーは，教科書通りの充足可能性を判定するプログラムではなく，インクリメンタル SAT，Certified UNSAT，UNSAT コアなど様々な機能を提供するようになっている．
- SAT 型システムもそれらを問題解法に利用する研究が行われている．
- SAT ソルバーを使った問題解法は決して「ソルバーまかせ」ではなく，問題のモデリング，符号化，SAT ソルバーの機能の活用などを深く検討しないと良い解法にはならない．
- しかし，それらがうまくはまった時には既存の専用解法を大きく上回ることも珍しくない．

## 参考情報

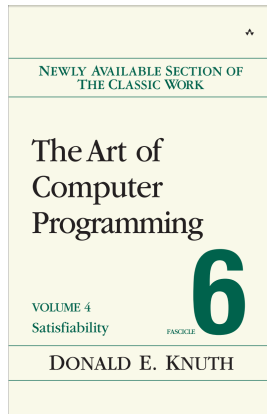
# SAT Competition 2023 の内容

- Main 트랙
  - ▶ 最も重要視される逐次ソルバーの部門
  - ▶ 1つのCPUコア上での性能を競う
  - ▶ UNSAT の証明も出力する必要がある
- CaDiCaL 1.5.3 Hacks
  - ▶ CaDiCaL 1.5.3 との編集距離が 1000 文字以下のソルバーの性能を競う部門
- Parallel 트랙
  - ▶ 並列部門-共有メモリ
  - ▶ AWS の m4.16xlarge インスタンス 1 台上での性能を競う部門 (64 virtual CPU cores, 256GB RAM)
- Cloud 트랙
  - ▶ 並列部門-分散メモリ
  - ▶ AWS の m4.4xlarge インスタンス 100 台上での性能を競う部門 (16 virtual CPU cores, 64GB RAM)
  - ▶ 各 Worker は MPI もしくは SSH で通信する .

# SAT Competition 2023 の結果 (抜粋)

	Gold	Silver	Bronze
Main	SBVA-CaDiCaL (284)	Kissat-MAB-prop* (272)	MCaDiCaL-PPD-500 (260)
Parallel	PRS-parallel (320)	Mallob64 (301)	p-Kissat (297)
Cloud	Mallob1600 (328)	PRS-distributed (305)	Malloblin (297)

- 上記3つのトラックでは同じベンチマーク 400 問で評価される
  - ▶ Main/Parallel は各問 5000 秒の制限時間
  - ▶ Cloud は各問 1000 秒の制限時間
- カッコ内は制限時間内に解けた問題数 (競技会では PAR2 スコアで評価される).
- 台数効果は出ないものの Parallel と Cloud で性能向上が確認できる .
- **CaDiCaL/Kissat/**は Armin Biere 先生がスクラッチから開発した SAT ソルバー .



The Art of Computer Programming  
4 卷分冊 6 [Knuth, 2015]



Handbook of Satisfiability  
[Biere+, 2009] (2nd Edition 2020)

# SAT 技術の進化と応用

～ パズルからプログラム検証まで～,  
情報処理 57 巻 8 号 (2016 年 8 月号)

- SAT 技術の進化 [番原+, 2016] [▶ web](#)
- SAT とパズル ～問題をいかに SAT ソルバーで解くか～ [田村+, 2016] [▶ web](#)
- SAT とラムゼー数 ～数学の未解決問題への挑戦～ [藤田+, 2016] [▶ web](#)
- SAT と AI [井上, 2016] [▶ web](#)
- SAT ソルバーの最近の進展 [鍋島+, 2016] [▶ web](#)
- MaxSAT : SAT の最適化問題への拡張 ～MaxSAT ソルバーの活用法～ [越村+, 2016] [▶ web](#)
- SMT ソルバーによるプログラム検証 [石井+, 2016] [▶ web](#)



- **SAT ソルバ・SMT ソルバの技術と応用** [梅村, 2010] [▶ web](#)
- **SAT 問題と他の制約問題との相互発展** [酒井+, 2015] [▶ web](#)
- **SAT 型制約プログラミングシステムと周辺技術** [宋+, 2017] [▶ web](#)
- **SAT ソルバーの最新動向と利用技術** [宋+, 2018] [▶ web](#)

# 特集「最近の SAT 技術の発展」, 人工知能学会誌 25 巻 1 号 (2010 年 1 月号)

- SAT ソルバーの基礎 [井上+, 2010] [▶ web](#)
- 高速 SAT ソルバーの原理 [鍋島+, 2010] [▶ web](#)
- 制約最適化問題と SAT 符号化 [田村+, 2010] [▶ web](#)
- SMT: 個別理論を取り扱う SAT 技術 [岩沼+, 2010] [▶ web](#)
- モデル列挙とモデル計数 [長谷川+, 2010] [▶ web](#)
- \*-SAT: SAT の拡張 [平山+, 2010] [▶ web](#)
- SAT によるプランニングとスケジューリング [鍋島, 2010] [▶ web](#)
- SAT によるシステム検証 [番原+, 2010] [▶ web](#)

# その他の情報

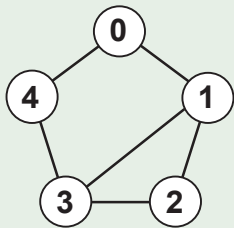
- **Constraint Solvers Catalog** [▶ web](#)
  - ▶ 制約ソルバーを実装言語，モデリング言語，利用可能な制約等の様々な観点でまとめたカタログサイト．
- **Satisfiability modulo theories** [▶ web](#)
  - ▶ Wikipedia の SMT のページ．SMT ソルバーを背景理論や実装言語でまとめた表が記載されている．
- **私のブックマーク (SAT)** [▶ web](#)
  - ▶ SAT に関するプロジェクト，ソルバー，競技会，ハンドブック・解説・スライドなどの情報が URL と共に紹介されたブックマーク集．
- **CSPSAT プロジェクト** [▶ web](#)
  - ▶ 2009 年にスタートした日本国内のプロジェクト．2017 年 3 月時点では CSPSAT3 が進行中．
- **発表者らのソフトウェア**
  - ▶ SAT 型制約ソルバー sCOP [▶ web](#)
  - ▶ SAT 型制約ソルバー Sugar [▶ web](#)
  - ▶ SAT 型制約ソルバー Diet-Sugar [▶ web](#)
  - ▶ SAT 型制約プログラミングシステム Scarab [▶ web](#)
    - ★ Scarab チュートリアル [▶ web](#)

**CSP , 言語 , 競技会**

- **制約充足問題 (Constraint Satisfaction Problem; CSP)** は、与えられた制約を満たす解を探索する問題である。
  - ▶ CSP には産学問わず様々な応用があり、かつ、求解が困難な問題として人工知能分野などにおける重要な研究課題となっている。
- **制約言語** は、CSP を記述するための言語であり、汎用プログラミング言語を利用したものや、専用の言語が研究開発されている。
- **CSP ソルバー** は、CSP を解くプログラムであり、欧米の研究機関を中心に活発に研究開発されている。
- **CSP ソルバーの国際競技会** は、CSP ソルバーの性能を競う競技会であり、ほぼ毎年開催されている。

- ① 制約言語，CSP ソルバー，競技会の動向について
- ② SAT 型 CSP ソルバー sCOP
- ③ 2018 年 XCSP3 競技会における成績

# 例: グラフ彩色問題



## 整数変数とドメイン

$$n_0, n_1, n_2, n_3, n_4 \in \{0, 1, 2\}$$

## 制約

$$n_0 \neq n_1 \quad n_0 \neq n_4 \quad n_1 \neq n_2$$

$$n_1 \neq n_3 \quad n_2 \neq n_3 \quad n_3 \neq n_4$$

```
<instance format="XCSP3" type="CSP">
  <variables>
    <array id="n" size="[5]"> 0..2 </array>
  </variables>
  <constraints>
    <intension>
      and(ne(n[0],n[1]),
          ne(n[0],n[4]),
          ne(n[1],n[2]),
          ne(n[1],n[3]),
          ne(n[2],n[3]),
          ne(n[3],n[4]))
    </intension>
  </constraints>
</instance>
```



# 代表的な制約言語

制約言語はこれまで数多く提案されているが、中でも **XCSP3** [▶ web](#) と **MiniZinc** [▶ web](#) が、事実上の標準となりつつある。

## XCSP3 [Roussel+, 2009a]

- 制約の構造を保持したまま XML ベースで記述する言語
- ブラックボックスソルバーを想定している (探索はソルバーにまかせる).
- 23000 問以上のインスタンスをベンチマークとして公開
- 公式 parser (Java, C++), verifier (Java) が公開

## MiniZinc [Nethercote+, 2007]

- モデリング言語. ソルバーの入力には変換後の FlatZinc 形式で渡される (構造は壊れる).
- 探索方法や決定変数などの情報を記述可能 (人が探索方法を指定できる)
- 17000 問以上のインスタンスをベンチマークとして公開
- Coursera のコースがある.
- 専用の IDE が提供されている.

# 制約の種類 (XCSP3, MiniZinc)

## ● 内包的制約

- ▶ 論理演算: 論理積 ( $\wedge$ ), 論理和 ( $\vee$ ), 同値 ( $\leftrightarrow$ ), 否定 ( $\neg$ ) 等
- ▶ 比較演算:  $=, \neq, \geq, >, \leq, <$  等
- ▶ 算術演算: 加算 ( $+$ ), 減算 ( $-$ ), 乗算 ( $\times$ ), 除算 ( $\div$ ), 剰余 ( $mod$ ) 等

## ● 外延的制約

- ▶ 以下の表のように変数の取り得る値の組合せを陽に記述したもの. この例は, 変数  $x, y, z$  上の外延的制約を表している.

$x$	$y$	$z$
1	0	1
0	2	0
1	1	0

## ● グローバル制約

- ▶ 与えられたすべての変数が互いに異なることを表す `alldifferent` 制約をはじめとして, 数十種類以上のグローバル制約を記述可能.
- ▶ 経路, 閉路, 木などグラフに関するグローバル制約もある.

## ● その他

- ▶ 単一, 多目的の目的関数を用いて制約最適化問題を記述できる.
- ▶ ソフト制約, 実数変数など

## 使われている技術

- アーク整合性維持法 (MAC) [Rossi+, 2006]
- SAT ソルバー [Roussel+, 2009b]
- Lazy Clause Generation (LCG) [Ohrimenko+, 2009]

## 代表的な CSP ソルバー

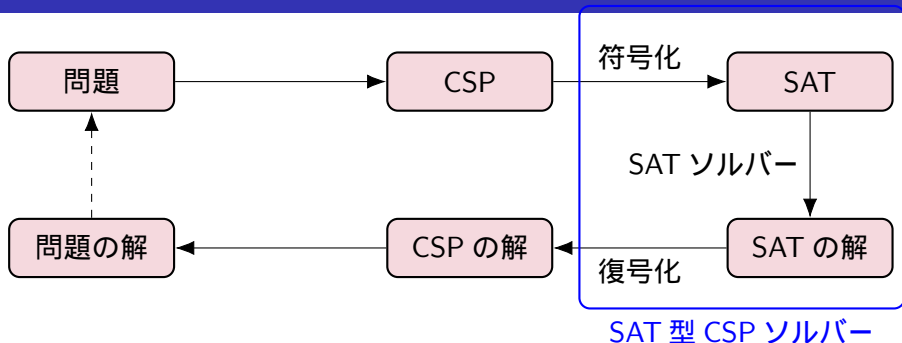
	サポート言語	技術	実装言語等	開発機関
AbsCon	XCSP3	MAC	Java	CRIL-CNRS
Choco	XCSP3, MiniZinc	MAC	Java	LS2N-CNRS
Mistral	XCSP3, MiniZinc	MAC	C++	LAAS-CNRS
or-tools	MiniZinc	LCG	C++	Google
chuffed	MiniZinc	LCG	C++	Monash Univ.
PicatSAT	XCSP3, MiniZinc	SAT	C, C++	City Univ. of NY
Sugar	XCSP2	SAT	Java, C++	神戸大学
sCOP	XCSP3	SAT	Scala, C++	神戸大学など

# CSP ソルバーの国際競技会

- **XCSP3** と **MiniZinc** それぞれについて競技会が開催されている。
- 両競技会ともに、以下のような手順で行われる。
  - ① 参加者はソルバーを事前にオーガナイザに提出 (5-7 月)
  - ② オーガナイザは研究機関のクラスタマシンや Amazon AWS を利用してソルバーを評価
  - ③ 結果を CP 国際会議で発表 (8-9 月)

競技会	XCSP3 Competition	MiniZinc Challenge
言語	XCSP3	MiniZinc
歴史	2005 年に始まって 5 回	2008 年に始まって 11 回
評価方式	解けた問題の数	ボルダ得点
制限時間	1 問 40 分	1 問 20 分
問題	CSP, COP	COP
問題数	数百問 ~ 数千問	100 問
ソルバー	逐次, 並列, ミニ	逐次, 並列, 探索指定, オープン
提出方法	実行形式 (or ソース)	Docker イメージ

# SAT 型 CSP ソルバー sCOP の概要



- sCOP は, XCSP3 に対応した Scala 版 Sugar [Tamura+, 2008] を目指して開発した SAT 型 CSP ソルバー .
  - ▶ 順序符号化 [Tamura+, 2009]
  - ▶ 対数符号化 [Iwama+, 1994]
  - ▶ MDD と順序符号化を用いた外延的制約の SAT 符号化
  - ▶ ハイブリッド符号化 [Soh+, 2017] (2019 年に追加)
- SAT ソルバー: MapleCOMSPS (逐次), glucose-syrup (並列)

# sCOP の 2018 年 XCSP3 競技会における成績

順位	ソルバー	技術	#Solved	SAT/UNSAT	%VBS
0	VBS		163	103/60	100
1	<b>scop-order+maple</b>	SAT	146	92/54	90
2	<b>scop-both+maple</b>	SAT	140	87/53	86
3	PicatSAT	SAT	138	85/53	85
4	Mistral-2.0	MAC	116	80/36	71
5	Choco-solver-4.0.7b seq	MAC	115	77/38	71
6	Concrete-3.9.2	MAC	92	64/28	56
7	OscAR - Conf. Ord. Res.	MAC	90	62/28	55
8	Concrete-3.9.2-SuperNG	MAC	84	55/29	52
9	Sat4j-CSP	SAT	83	40/43	51
10	OscAR - Conf. Order.	MAC	81	51/30	50
11	cosoco-1.12	MAC	79	53/26	48
12	BTD	MAC	76	31/45	47
13	BTD_12	MAC	76	32/44	47
14	macht	MAC	66	33/33	40

- SAT ソルバーを使った CSP ソルバーが多くの問題を解いた .
- 2018 年は比較的ドメインサイズの小さな問題がベンチマークとして採用されていることも原因と考えられる .

## 制約種類毎の解けた問題数 (上位のソルバー)

制約の種類	問題数	chco	mist	pics	scopB	scopO
外延的制約 (sup.)	89	50	45	62	58	63
外延的制約 (sup. and conf.)	37	14	11	15	18	18
内包的制約のみ	110	51	60	61	64	65
	236	115	116	138	140	146

- どの制約種類毎で見ても sCOP が最も多くの問題を解いている (同じ SAT 型 CSP ソルバーである PicatSAT 同様)
- 但し, どの問題においても優れているわけではなく, 問題シリーズ毎に見ると SAT 型 CSP ソルバーが苦手な問題があることがわかる.

# 問題シリーズ毎の解けた問題数

シリーズ	問題数	chco	mist	pics	scopB	scopO	制約
Bibd	12	4	8	8	8	8	pure int
CarSequencing	17	11	6	17	12	17	sup.
ColouredQueens	12	3	4	4	3	3	pure int
Crossword	13	5	3	2	3	3	sup.
Dubois	12	7	7	12	12	12	sup.
Eternity	15	7	7	6	6	6	sup.
Frb	16	3	3	3	5	5	conf./sup.
GracefulGraph	11	6	6	6	6	6	pure int
Haystacks	10	4	2	10	10	10	conf./sup.
Langford	11	7	9	9	9	9	pure int
MagicHexagon	11	4	5	3	3	3	pure int
MisteryShopper	10	10	10	10	10	10	sup.
PseudoBoolean-dec	13	4	6	4	7	8	pure int
Quasigroups	16	6	6	7	8	8	pure int
Rlfap-dec-scens11	12	11	10	12	12	12	pure int
SocialGolfers	12	6	6	8	8	8	pure int
SportsScheduling	10	3	4	4	4	4	sup.
StripPacking	12	7	8	11	11	11	sup.
Subisomorphism	11	7	6	2	3	3	conf./sup.
	236	115	116	138	140	146	



- 参考まで (教科書などと企業における CP, CP と SAT のサーベイ論文)
  - ▶ CP 分野の教科書: Handbook of Constraint Programming (教科書)
    - ▶ [web](#)
  - ▶ Operations Research and Constraint Programming at Google
    - ★ [https://link.springer.com/chapter/10.1007%2F978-3-642-23786-7\\_2](https://link.springer.com/chapter/10.1007%2F978-3-642-23786-7_2)
  - ▶ IBM ILOG CP optimizer for scheduling
    - ★ <https://link.springer.com/article/10.1007%2Fs10601-018-9281-x>
  - ▶ Propositional Satisfiability and Constraint Programming: A comparative survey [▶ web](#)
- SAT 型 CSP ソルバー
  - ▶ Sugar [▶ web](#)
  - ▶ sCOP [▶ web](#) (XCSP3, ハイブリッド符号化に対応)
- SAT 型制約プログラミングシステム
  - ▶ Copris [▶ web](#)
  - ▶ Scarab [▶ web](#) (lightwait. コード少ない)

## 参考文献

[Abío+, 2013] Abío, Ignasi, Nieuwenhuis, Robert, Oliveras, Albert , and Rodríguez-Carbonell, Enric (2013).

A parametric approach for smaller and better encodings of cardinality constraints.

*In Principles and Practice of Constraint Programming - 19th International Conference, CP 2013, Uppsala, Sweden, September 16-20, 2013. Proceedings*, pages 80–96.

[Abío+, 2012] Abío, Ignasi, Nieuwenhuis, Robert, Oliveras, Albert, Rodríguez-Carbonell, Enric , and Mayer-Eichberger, Valentin (2012).

A new look at BDDs for pseudo-Boolean constraints.

*Journal of Artificial Intelligence Research*, 45:443–480.

## Reference II

- [Roberto+, 2011] Roberto As<sup>á</sup>in, Robert Nieuwenhuis, Albert Oliveras ,  
Enric Rodr<sup>í</sup>guez-Carbonell (2011).  
Cardinality networks: a theoretical and empirical study.  
*Constraints*, 16(2):195–221.
- [Audemard+, 2009] Audemard, Gilles and Simon, Laurent (2009).  
Predicting learnt clauses quality in modern SAT solvers.  
In *Proceedings of the 21st International Joint Conference on Artificial  
Intelligence (IJCAI 2009)*, pages 399–404.
- [Bailleux+, 2003] Bailleux, Olivier and Boufkhad, Yacine (2003).  
Efficient CNF encoding of Boolean cardinality constraints.  
In *Proceedings of the 9th International Joint Conference on Principles  
and Practice of Constraint Programming (CP 2003)*, LNCS 2833, pages  
108–122.

## Reference III

[Bailleux+, 2006] Bailleux, Olivier, Boufkhad, Yacine , and Roussel, Olivier (2006).

A translation of pseudo Boolean constraints to SAT.

*Journal on Satisfiability, Boolean Modeling and Computation*,  
2(1-4):191–200.

[Bailleux+, 2009] Bailleux, Olivier, Boufkhad, Yacine , and Roussel, Olivier (2009).

New encodings of pseudo-Boolean constraints into CNF.

In *Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing (SAT 2009)*, LNCS 5584, pages 181–194.

[番原+, 2016] 番原 睦則 , 鍋島 英知 (2016).

SAT 技術の進化.

情報処理, 57(8):704–709.

## Reference IV

[番原+, 2010] 番原 睦則, 田村 直之 (2010).

SAT によるシステム検証.

人工知能学会誌, 25(1):122–129.

[Bayardo Jr., 1997] Bayardo Jr., Roberto J. and Schrag, Robert (1997).

Using CSP look-back techniques to solve real-world SAT instances.

In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI 1997)*, pages 203–208.

[Biere, 2009] Biere, Armin (2009).

Bounded model checking.

In *Handbook of Satisfiability*, pages 457–481. IOS Press.

[Biere+, 2009] Biere, Armin, Heule, Marijn, van Maaren, Hans , and Walsh, Toby, editors (2009).

*Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications (FAIA)*. IOS Press.

[Brakensiek+, 2020] Brakensiek, Joshua, Heule, Marijn, Mackey, John , and Narváez, David (2020).

The resolution of keller's conjecture.

In Peltier, Nicolas and Sofronie-Stokkermans, Viorica, editors, *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I*, volume 12166 of *Lecture Notes in Computer Science*, pages 48–65. Springer.

[Chen, 2010] Chen, Jingchao (2010).

A New SAT Encoding of the At-Most-One Constraint.

In *The 9th International Workshop on Constraint Modelling and Reformulation (ModRef 2010) at the 16th International Conference on the Principles and Practice of Constraint Programming (CP 2010)*, pages 1–8.

## Reference VI

[Cook, 1971] Cook, Stephen A. (1971).

The complexity of theorem-proving procedures.

In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC 1971)*, pages 151–158.

[Crawford+, 1994] Crawford, James M. and Baker, Andrew B. (1994).

Experimental results on the application of satisfiability algorithms to scheduling problems.

In *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI 1994)*, pages 1092–1097.

[Davis+, 1962] Davis, Martin, Logemann, George , and Loveland, Donald W. (1962).

A machine program for theorem-proving.

*Communications of the ACM*, 5(7):394–397.



## Reference VII

[de Kleer, 1989] de Kleer, Johan (1989).

A comparison of ATMS and CSP techniques.

In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI 1989)*, pages 290–296.

[Eén+, 2003] Eén, Niklas and Sörensson, Niklas (2003).

An extensible SAT-solver.

In *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*, LNCS 2919, pages 502–518.

[Eén+, 2006] Eén, Niklas and Sörensson, Niklas (2006).

Translating pseudo-Boolean constraints into SAT.

*Journal on Satisfiability, Boolean Modeling and Computation*, 2(1-4):1–26.

## Reference VIII

[藤田+, 2016] 藤田 博, 越村 三幸 (2016).

SAT とラムゼー数 ~ 数学の未解決問題への挑戦 ~ .  
情報処理, 57(8):716–719.

[Garey+, 1979] Garey, Michael R. and Johnson, David S. (1979).

*Computers and Intractability: A Guide to the Theory of  
NP-Completeness.*

W. H. Freeman and Company, New York.

[Gavanelli, 2007] Gavanelli, Marco (2007).

The log-support encoding of CSP into SAT.

In *Proceedings of the 13th International Joint Conference on Principles  
and Practice of Constraint Programming (CP 2007)*, LNCS 4741, pages  
815–822.

## Reference IX

- [Gebser+, 2012] Gebser, Martin, Kaufmann, Benjamin , and Schaub, Torsten (2012).  
Conflict-driven answer set solving: From theory to practice.  
*Artif. Intell.*, 187:52–89.
- [Gent, 2002] Gent, Ian P. (2002).  
Arc consistency in SAT.  
In *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 2002)*, pages 121–125.
- [Jürgen+, 2004] Jürgen Giesl, René Thiemann, Peter Schneider-Kamp , and Stephan Falke (2004).  
Automated termination proofs with *aprove*.  
In *Rewriting Techniques and Applications, 15th International Conference, RTA 2004, Aachen, Germany, June 3-5, 2004, Proceedings*, pages 210–220.

# Reference X

[長谷川+, 2010] 長谷川 隆三, 藤田 博, 越村 三幸 (2010).

モデル列挙とモデル計数.

人工知能学会誌, 25(1):96–104.

[Heule+, 2016] Heule, Marijn J. H., Kullmann, Oliver, and Marek, Victor W. (2016).

Solving and verifying the boolean pythagorean triples problem via cube-and-conquer.

In *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*, pages 228–245.

[鍋島+, 2012] 鍋島 英知, 岩沼 宏治, 井上 克巳 (2012).

Glueminisat 2.2.5: 単位伝搬を促す学習節の積極的獲得戦略に基づく高速 SAT ソルバー.

コンピュータソフトウェア, 29(4):211–230.

# Reference XI

[平山+, 2010] 平山 勝敏, 横尾 真 (2010).

\*-SAT: SAT の拡張.

人工知能学会誌, 25(1):105–113.

[堀岡+, 2021] 堀岡 真未, 宋 剛秀, 田村 直之 (2021).

Cdcl 型 sat ソルバーの内部動作可視化ツール.

B3.

[井上, 2016] 井上 克巳 (2016).

SAT と AI.

情報処理, 57(8):720–723.

[井上+, 2010] 井上 克巳, 田村 直之 (2010).

SAT ソルバーの基礎.

人工知能学会誌, 25(1):57–67.

## Reference XII

- [石井+, 2016] 石井 大輔, 上田 和紀 (2016).  
SMT ソルバーによるプログラム検証.  
情報処理, 57(8):734–737.
- [Iwama+, 1994] Iwama, Kazuo and Miyazaki, Shuichi (1994).  
SAT-variable complexity of hard combinatorial problems.  
In *Proceedings of the IFIP 13th World Computer Congress*, pages  
253–258.
- [岩沼+, 2010] 岩沼 宏治, 鍋島 英知 (2010).  
SMT: 個別理論を取り扱う SAT 技術.  
人工知能学会誌, 25(1):86–95.
- [Jackson, 2006] Jackson, Daniel (2006).  
*Software Abstractions - Logic, Language, and Analysis*.  
MIT Press.

## Reference XIII

[Kaivola+, 2009] Kaivola, Roope, Ghughal, Rajnish, Narasimhan, Naren, Telfer, Amber, Whittemore, Jesse, Pandav, Sudhindra, Slobodová, Anna, Taylor, Christopher, Frolov, Vladimir A., Reeber, Erik , and Naik, Armaghan (2009).

Replacing testing with formal verification in intel coretm i7 processor execution engine validation.

*In Proceedings of the 21st International Conference on Computer Aided Verification (CAV 2009), LNCS 5643, pages 414–429.*

[Kasif, 1990] Kasif, Simon (1990).

On the parallel complexity of discrete relaxation in constraint satisfaction networks.

*Artificial Intelligence*, 45(3):275–286.

## Reference XIV

- [Katebi+, 2011] Katebi, Hadi, Sakallah, Karem A. , and Silva, João P. Marques (2011).  
Empirical study of the anatomy of modern sat solvers.  
In *Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing (SAT 2011)*, LNCS 6695, pages 343–356.
- [Kautz+, 1992] Kautz, Henry A. and Selman, Bart (1992).  
Planning as satisfiability.  
In *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI 1992)*, pages 359–363.
- [Knuth, 2015] Knuth, Donald E. (2015).  
*The Art of Computer Programming, Volume 4, Fascicle 6: Satisfiability*.  
Addison-Wesley Professional.



[Konev+, 2014] Konev, Boris and Lisitsa, Alexei (2014).

A SAT attack on the erdős discrepancy conjecture.

In *Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, pages 219–226.

[越村+, 2016] 越村 三幸, 藤田 博 (2016).

MaxSAT: SAT の最適化問題への拡張 ~ MaxSAT ソルバーの活用法 ~ .  
情報処理, 57(8):730–733.

[Larrabee, 1992] Larrabee, Tracy (1992).

Test pattern generation using boolean satisfiability.

*IEEE Trans. on CAD of Integrated Circuits and Systems*, 11(1):4–15.

## Reference XVI

- [Le Berre+, 2009] Le Berre, Daniel and Rapicault, Pascal (2009).  
Dependency management for the eclipse ecosystem: Eclipse p2,  
metadata and resolution.  
*In Proceedings of the 1st International Workshop on Open Component  
Ecosystems, IWOCE '09*, pages 21–30.
- [Li+, 2023] Li, Zhengyu, Bright, Curtis , and Ganesh, Vijay (2023).  
A SAT solver and computer algebra attack on the minimum  
kochen-specker problem.  
*CoRR*, abs/2306.13319.
- [Luby+, 1993] Luby, Michael, Sinclair, Alistair , and Zuckerman, David  
(1993).  
Optimal speedup of Las Vegas algorithms.  
*Information Processing Letters*, 47(4):173–180.

## Reference XVII

- [Marques-Silva+, 1999] Marques-Silva, João P. and Sakallah, Karem A. (1999).  
GRASP: A search algorithm for propositional satisfiability.  
*IEEE Transactions on Computers*, 48(5):506–521.
- [Moskewicz+, 2001] Moskewicz, Matthew W., Madigan, Conor F., Zhao, Ying, Zhang, Lintao, and Malik, Sharad (2001).  
Chaff: Engineering an efficient SAT solver.  
In *Proceedings of the 38th Design Automation Conference (DAC 2001)*, pages 530–535.
- [鍋島, 2010] 鍋島 英知 (2010).  
SAT によるプランニングとスケジューリング.  
*人工知能学会誌*, 25(1):114–121.

## Reference XVIII

[鍋島+, 2016] 鍋島 英知, 岩沼 宏治, 井上 克巳 (2016).

SAT ソルバーの最近の進展.  
情報処理, 57(8):724–729.

[鍋島+, 2010] 鍋島 英知, 宋 剛秀 (2010).

高速 SAT ソルバーの原理.  
人工知能学会誌, 25(1):68–76.

[Nethercote+, 2007] Nethercote, Nicholas, Stuckey, Peter J., Becket, Ralph, Brand, Sebastian, Duck, Gregory J. , and Tack, Guido (2007).

MiniZinc: Towards a standard CP modelling language.

In *Proceedings of the 13th International Joint Conference on Principles and Practice of Constraint Programming (CP 2007)*, LNCS 4741, pages 529–543.

- [Nguyen+, 2015] Nguyen, Van-Hau and Mai, Son Thai (2015).  
A new method to encode the at-most-one constraint into SAT.  
In *Proceedings of the Sixth International Symposium on Information and Communication Technology, Hue City, Vietnam, December 3-4, 2015*, pages 46–53.
- [則武+, 2013] 則武 治樹, 番原 睦則, 宋 剛秀, 田村 直之, 井上 克巳 (2013).  
パッキング配列問題の制約モデリングと SAT 符号化.  
コンピュータソフトウェア, 31(1):116–130.

[Ogata+, 2004] Ogata, Shougo, Tsuchiya, Tatsuhiro , and Kikuno, Tohru (2004).

Sat-based verification of safe petri nets.

*In Automated Technology for Verification and Analysis: Second International Conference, ATVA 2004, Taipei, Taiwan, ROC, October 31-November 3, 2004. Proceedings, pages 79–92.*

[Ogawa+, 2013] Ogawa, Toru, Liu, Yangyang, Hasegawa, Ryuzo, Koshimura, Miyuki , and Fujita, Hiroshi (2013).

Modulo based CNF encoding of cardinality constraints and its application to maxsat solvers.

*In 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, Herndon, VA, USA, November 4-6, 2013, pages 9–17.*

## Reference XXI

- [Ohrimenko+, 2009] Ohrimenko, Olga, Stuckey, Peter J. , and Codish, Michael (2009).  
Propagation via lazy clause generation.  
*Constraints*, 14(3):357–391.
- [Petke+, 2011] Petke, Justyna and Jeavons, Peter (2011).  
The order encoding: From tractable csp to tractable sat.  
In *Proceedings of the 14th International Conference on Theory and Applications of Satisfiability Testing (SAT 2011)*, LNCS 6695, pages 371–372.
- [Philipp+, 2015] Philipp, Tobias and Steinke, Peter (2015).  
Pblib – a library for encoding pseudo-boolean constraints into cnf.  
In Heule, Marijn and Weaver, Sean, editors, *Theory and Applications of Satisfiability Testing – SAT 2015*, volume 9340 of *Lecture Notes in Computer Science*, pages 9–16. Springer International Publishing.

- [Pipatsrisawat+, 2007] Pipatsrisawat, Knot and Darwiche, Adnan (2007).  
A lightweight component caching scheme for satisfiability solvers.  
In *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT 2007)*, LNCS 4501, pages 294–299.
- [Prestwich, 2009] Prestwich, Steven David (2009).  
CNF encodings.  
In *Handbook of Satisfiability*, pages 75–97. IOS Press.
- [Rossi+, 2006] Rossi, Francesca, van Beek, Peter , and Walsh, Toby (2006).  
*Handbook of Constraint Programming*.  
Elsevier.



- [Roussel+, 2009a] Roussel, Olivier and Lecoutre, Christophe (2009a).  
XML representation of constraint networks: Format XCSP 2.1.  
*CoRR*, abs/0902.2362.
- [Roussel+, 2009b] Roussel, Olivier and Manquinho, Vasco M. (2009b).  
Pseudo-Boolean and cardinality constraints.  
In Biere, Armin, Heule, Marijn, van Maaren, Hans , and Walsh, Toby,  
editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial  
Intelligence and Applications*, pages 695–733. IOS Press.
- [酒井+, 2015] 酒井 政裕 , 今井 健男 (2015).  
SAT 問題と他の制約問題との相互発展.  
*コンピュータソフトウェア*, 32(1):103–119.

- [Sakai+, 2015] Sakai, Masahiko and Nabeshima, Hidetomo (2015).  
Construction of an ROBDD for a pb-constraint in band form and  
related techniques for pb-solvers.  
*IEICE Transactions*, 98-D(6):1121–1127.
- [Selman+, 1996] Selman, Bart, Kautz, Henry , and Cohen, Bram (1996).  
Local search strategies for satisfiability testing.  
In Johnson, David J. and Trick, Michael A., editors, *Cliques, Coloring,  
and Satisfiability: the Second DIMACS Implementation Challenge*,  
volume 26 of *DIMACS Series in Discrete Mathematics and Theoretical  
Computer Science*, pages 521–532. American Mathematical Society.

[Sinz, 2005] Sinz, Carsten (2005).

Towards an optimal CNF encoding of Boolean cardinality constraints.  
In *Proceedings of the 11th International Joint Conference on Principles and Practice of Constraint Programming (CP 2005)*, LNCS 3709, pages 827–831. Springer.

[Soh+, 2017] Soh, Takehide, Banbara, Mutsunori , and Tamura, Naoyuki (2017).

Proposal and evaluation of hybrid encoding of CSP to SAT integrating order and log encodings.

*International Journal on Artificial Intelligence Tools*, 26(1):1–29.

[宋+, 2017] 宋 剛秀, 番原 睦則 , 田村 直之 (2017).

SAT 型制約プログラミングシステムと周辺技術.  
*コンピュータソフトウェア*, 34(1):67–80.

[宋+, 2018] 宋 剛秀, 番原 睦則, 田村 直之 (2018).

SAT ソルバーの最新動向と利用技術.

コンピュータソフトウェア, 35(4):72–92.

[Soh+, 2010] Soh, Takehide and Inoue, Katsumi (2010).

Identifying necessary reactions in metabolic pathways by minimal model generation.

In *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, pages 277–282.

[Soh+, 2014] Soh, Takehide, Le Berre, Daniel, Roussel, Stéphanie,

Banbara, Mutsunori, and Tamura, Naoyuki (2014).

Incremental SAT-based method with native boolean cardinality handling for the hamiltonian cycle problem.

In *Proceedings of the 14th European Conference on Logics in Artificial Intelligence (JELIA 2014)*, LNAI 8761, pages 684–693. Springer.

## Reference XXVII

- [Tamura+, 2008] Tamura, Naoyuki and Banbara, Mutsunori (2008).  
Sugar: a CSP to SAT translator based on order encoding.  
In *Proceedings of the 2nd International CSP Solver Competition*, pages 65–69.
- [Tamura+, 2013] Tamura, Naoyuki, Banbara, Mutsunori , and Soh, Takehide (2013).  
PBSugar: Compiling pseudo-boolean constraints to SAT with order encoding.  
In *Proceedings of the 25th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2013)*, pages 1020–1027.
- [田村+, 2016] 田村 直之, 宋 剛秀 , 番原 睦則 (2016).  
SAT とパズル ~問題をいかに SAT ソルバーで解くか~.  
情報処理, 57(8):710–715.

- [Tamura+, 2009] Tamura, Naoyuki, Taga, Akiko, Kitagawa, Satoshi , and Banbara, Mutsunori (2009).  
Compiling finite linear CSP into SAT.  
*Constraints*, 14(2):254–272.
- [田村+, 2010] 田村 直之, 丹生 智也, 番原 睦則 (2010).  
制約最適化問題と SAT 符号化.  
人工知能学会誌, 25(1):77–85.
- [丹生+, 2013] 丹生 智也, 田村 直之, 番原 睦則 (2013).  
位取り記数法に基づく整数有限領域上の制約充足問題のコンパクトかつ効率的な sat 符号化.  
コンピュータソフトウェア, 30(1):211–230.
- [梅村, 2010] 梅村 晃広 (2010).  
SAT ソルバ・SMT ソルバの技術と応用.  
コンピュータソフトウェア, 27(3):24–35.

## Reference XXIX

[Walsh, 2000] Walsh, Toby (2000).

SAT v CSP.

In *Proceedings of the 6th International Conference on Principles and Practice of Constraint Programming (CP 2000)*, pages 441–456.

[Warners, 1998] Warners, Joost P. (1998).

A linear-time transformation of linear inequalities into conjunctive normal form.

*Inf. Process. Lett.*, 68(2):63–69.

[Wetzler+, 2014] Wetzler, Nathan, Heule, Marijn , and Jr., Warren A. Hunt (2014).

Drat-trim: Efficient checking and trimming using expressive clausal proofs.

In *Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic*,

*VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings, pages 422–429.*