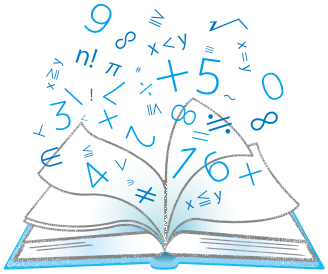

1 SAT技術の進化

番原睦則（神戸大学） 鍋島英知（山梨大学）

SATソルバー

The story of satisfiability is the tale of a triumph of software engineering, blended with rich doses of beautiful mathematics. Thanks to elegant new data structures and other techniques, modern SAT solvers are able to deal routinely with practical problems that involve many thousands of variables, although such problems were regarded as hopeless just a few years ago. —Donald E. Knuth, TAOCP, Vol.4, Fascicle 6

これはチューリング賞受賞者 Donald E. Knuth の有名な教科書“The Art of Computer Programming" の最新分冊の序文の一節である．2015 年末に出版されたこの分冊は，300 ページ以上を使って SAT ソルバーと呼ばれるプログラム（とその応用）について述べている．この SAT ソルバー，何やらただ者ではなさそうである．

実は，SAT ソルバーは近年その性能が飛躍的に向上し，プログラム検証，プランニング，スケジューリング，制約充足問題，制約最適化問題など，さまざまな分野への実用的応用が急速に拡大している．身近なところでは，インテル社の i7 プロセッサの設計，統合開発環境 Eclipse のコンポーネント間の依存解析，Fedora Linux のパッケージ管理システム☆1，ナンバーリンクパズル☆2，などに SAT ソルバーが使われていたりする．

このSAT ソルバーの正体を探るため，まず手始めに，四国のように 4 地区からなる白地図を，地区の境界が分かるように赤（R），緑（G），青（B）の 3 色で塗り分ける問題を，SAT ソルバーを使って解いてみよう．この地図は，各地区を頂点，隣接関係を辺として，**図 -1** のグラフで表せるものとする．いま，変数 p_{ic} で「地区 i は色 c で塗られる」を表すことにする．たとえば，p_{1R} は「地区 1 は赤で塗られる」を表す．各地区 i は R, G, B のいずれかの色で塗られるので，論理式（節）

$$p_{iR} \vee p_{iG} \vee p_{iB} \qquad (1)$$

が真となればよい．一方，隣接する地区 i と地区 j（$i < j$）は同じ色では塗れないので，3 つの節

$$\neg p_{iR} \vee \neg p_{jR} \quad \neg p_{iG} \vee \neg p_{jG} \quad \neg p_{iB} \vee \neg p_{jB} \qquad (2)$$

が真となればよい．全体で地区の数は 4 つ，隣接地区の組は 5 つなので，この塗り分け問題を解くために必要な節（1）（2）の総数は $4+5 \times 3=19$ となる．そしてこの 19 個の節がすべて真となればよい．

実はこれらの節☆3 をほぼそのまま SAT ソルバーに入力できる．ただし，実際の入力ファイルでは記号が使えないため，次のようになる．

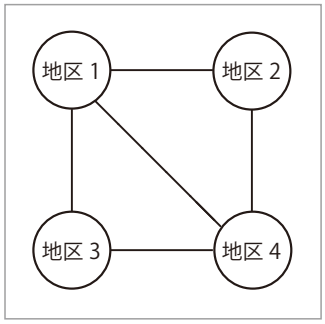

図 -1 塗り分け問題のグラフ表現

☆1 Fedora22 以降．
☆2 IC やボード上の配線経路を求める 1 層平面配線問題に相当．詳しくは本特集の「2. SAT とパズル」を参照．
☆3 多値符号化法と呼ばれる SAT への翻訳方法．各地区に塗られる色数を 1 色に制限しない，場合によっては多色を許す点が特徴である．

```
1  p cnf 12 19
2  1 2 3 0
3  4 5 6 0
4  7 8 9 0
5  10 11 12 0
6  -1 -4 0
7  -2 -5 0
8  -3 -6 0
9  -1 -7 0
10 -2 -8 0
11 -3 -9 0
12 -1 -10 0
13 -2 -11 0
14 -3 -12 0
15 -4 -10 0
16 -5 -11 0
17 -6 -12 0
18 -7 -10 0
19 -8 -11 0
20 -9 -12 0
```

1行目は変数の数（12個）と節の数（19個）を表している．各行は1つの節を表し，最後の0は節の終わりを表している．1〜12の整数値は変数p_{ic}に対応している☆4．記号“-”は否定（¬）を表す．また，2〜5行目は節（1）に，6〜20行目は節（2）にそれぞれ対応している．たとえば，2行目の節“1 2 3 0”は，$p_{1R} \vee p_{1G} \vee p_{1B}$を表しているといった具合である．

せっかくなので，SATソルバーを実行してみよう．一般的なSATソルバーは，節をすべて満たせた場合，各変数が真か偽かを出力してくれる．実際の出力はこんな感じになる．

```
s SATISFIABLE
v -1 2 -3 4 -5 -6 7 -8 -9 -10 -11 12 0
```

1行目のSATISFIABLEは「充足可能」，すなわち，節をすべて満たせたことを意味している．上では，変数p_{1G}を表す2，p_{2R}を表す4，p_{3R}を表す7，p_{4B}を表す12が真になり，結果として地区1は緑，地区2と3は赤，地区4は青に塗り分けられたことになる．この塗り分け問題，4地区であればたいしたことはない．が，たとえば世界地図のすべての国を4色で塗り分ける，となると人手では一仕事である．しかし，最新のSATソルバーならこれも即座に解くことができる．

この一見単純だが実は強力なSATソルバー☆5について，ページの許す限りじっくり見ていくことにしよう．正体探しの性格上，少々専門的な部分に足を踏み入れることもあるが，その点はご容赦願いたい．より専門的な日本語の解説としては文献1）がある．

☆4 1：p_{1R}，2：p_{1G}，3：p_{1B}，4：p_{2R}，5：p_{2G}，6：p_{2B}，7：p_{3R}，8：p_{3G}，9：p_{3B}，10：p_{4R}，11：p_{4G}，12：p_{4B}

☆5 SATソルバーは系統的ソルバーと確率的ソルバーの2つに大別されるが，後者については触れない．

SATソルバーの進化

SATとは命題論理式の充足可能性判定問題である．ある論理式ϕが与えられたとき，ϕを真にする変数の真偽値割当てが存在するならばϕは**充足可能**（SAT）であるといい，存在しないならば**充足不能**（UNSAT）である．例として次のような論理式を考えてみよう．

$$\phi = (x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4) \wedge (x_1 \vee x_4) \wedge (\neg x_2 \vee x_3 \vee \neg x_4)$$

ここでx_1やx_2が**命題変数**（真または偽の値をとる変数）であり，命題変数とその否定（x_1や$\neg x_2$）を**リテラル**，リテラルを論理和（∨）で結合したもの（$x_1 \vee x_2$など）を**節**という．SAT問題は一般に節を論理積（∧）で結合した式により与えられる☆6．節は論理積で結ばれるため，SATはすべての節を同時に満たす変数の真偽値割当ての存在を判定する問題ともいえる．ϕはx_1に真，x_2に偽を割り当てると，すべての節が充足できるため充足可能であり，一方$\phi \wedge \neg x_1$は，それを充足する真偽値割当てが存在しないため充足不能である．

SATの定義はすこぶる単純であるが，計算理論においては求解困難な問題の代表例である．たとえば与えられた論理式にn個の命題変数が含まれるとしよう．単純に2^n個のすべての真偽値割当てを列挙すれば充足可能性は判定できる．しかし応用問題を符号化して得られるSAT問題は，しばしば数百万個もの変数を含むことがある．もし1秒間に1京個の真偽値割当てを試したとしても，すべてを調べきるには変数が100個の場合でさえ単純計算で400万年かかってしまう☆7．

この組合せ爆発に対し，最新SATソルバーはどん

☆6 これを連言標準形（CNF）という．

☆7 $2^{100}/10^{16}$/60/60/24/365≈400万年

な武器（テクニック）を使って対抗しているのだろうか？　ここでは，SAT ソルバーの三種の神器と言える「深さ優先探索」,「単位伝播」,「矛盾からの節学習」の 3 つの基本テクニックを紹介する．

そもそもすべての真偽値割当てを試す必要はない．式 ϕ の場合，x_1=x_2= 偽とすると，最初の節（$x_1 \vee x_2$）を充足できない．このときほかの変数の真偽値の組合せを試す必要はない．よって，まずある変数 x_i を選択して真または偽を割り当て，そこで矛盾（充足できない節）が発生しないならば新たな変数を選択して同様の操作を繰り返す．もし矛盾が発生するならば，x_i への割当てを反転して再度試し，もし真と偽の両方を試しても矛盾するならば，さらに 1 つ前の選択肢にバックトラックする．すなわち**深さ優先探索**である．この深さ優先探索により，大規模な問題であってもメモリ不足に陥ることなく，真偽値の組合せを調べることが可能になる．

さらに節（$x_1 \vee x_2$）を考えると，x_1 または x_2 のいずれかの変数を真にする必要があるため，x_1= 偽とした時点で x_2= 真が確定する．すなわち各節を充足するために必然的な真偽値割当てが発生することがある．これを**単位伝播**と呼ぶ．単位伝播によって前述の深さ優先探索における探索木の高さを大きく低減させることが可能になる．これは探索空間の削減に大きく貢献する．

この深さ優先探索と単位伝播を組み合わせた手続きを DPLL と呼ぶ．1962 年に提案された古典的なアルゴリズムである．これに 1990 年代末に提案された**矛盾からの節学習**（Conflict Driven Clause Learning：CDCL）を組み合わせた手続きを CDCL といい，これが SAT ソルバーの飛躍的な性能向上をもたらした．

いま x_2, x_1, x_4 の順に真を割り当てたとしよう．すると節（$\neg x_2 \vee \neg x_3 \vee \neg x_4$）と（$\neg x_2 \vee x_3 \vee \neg x_4$）の両方を満たせないため矛盾が発生する（$x_3$ を真にしても偽にしても両者を充足できない）．この矛盾は x_1 の値にかかわらず発生している．すなわち矛盾の原因は x_2 および x_4 に真を割り当てたことである．これは $x_2 \wedge x_4$ と表すことができる．CDCL では，矛盾が発生したときに，その原因を解析し，原因を否定した $\neg x_2 \vee \neg x_4$ を節として学習する（論理式に追加する）．これを**学習節**という．この学習節は x_2 と x_4 が同時に真にできないことを表している．つまり一番最初の x_2 に真を割り当てた時点で x_4 を偽にできることを意味する．このように学習節は同じ探索失敗を繰り返すことを防ぎつつ，新たな単位伝播を生み出すことを可能にする．SAT ソルバーは矛盾が発生するたびに学習節を獲得するが，矛盾は無数に起こるため定期的に役立ちそうにない学習節を破棄することも行う．「失敗から学習」しつつも，「学習したことを忘れる」ことがある点は人間のようでもある．

計算環境や問題にも依存するが，最新 SAT ソルバーは 1 秒間に深さ優先探索における分岐を数万回繰り返し，単位伝播によって数百万変数に値を割り当て，数千回の矛盾を発生させながら動作する．つまり，SAT ソルバーの正体は，単位伝播や学習節により探索空間を削減しつつ，猛烈な勢いで真偽値の組合せを調べる判定器なのである．

SAT ソルバーは CDCL を契機にその性能を飛躍的に向上させ，その後も継続的にその性能を向上させている．その SAT ソルバーの進化を示すため，代表的なソルバーの性能推移を見てみよう．**図 -2** は，2012 ～ 2015 年に開催された SAT ソルバーの競技会で使用された産業応用問題計 1,130 問に対する 5 種類の SAT ソルバーの求解数と求解時間の分布を示したものである☆8．図中のソルバー Chaff と MiniSat は，それぞれ 2004 年および 2008 年の競技会で優勝したソルバーであり，Glucose と Lingeling は最新の 2 大ソルバーであり，GlueMiniSat は筆者らが継続的に開発している SAT ソルバーである．この図は縦軸から見るとよい．たとえば 1 問あたりの制限時間を 800 秒とすると MiniSat は約 600 問求解できることを表している．2004 年，2008 年時点の最速ソルバーと比べて，最新ソルバーは大きく性能を向上させていることが分かる．

☆8 実験環境は Core i5 2.3GHz 8GB RAM であり，1 問あたりの制限時間を 1,000 秒とした．

では，それぞれのソルバーを簡単に紹介しよう．

Chaff☆9は現在の高速SATソルバーの基盤を作った画期的なCDCLソルバーである．Chaffにおいて提案された変数選択ヒューリスティクスVSIDS（Variable State Independent Decaying Sum）と高速な単位伝播を実現するための監視リテラル技術は，最新のSATソルバーでも利用され続けている．

MiniSat☆10はSAT Race 2008競技会にて優勝したソルバーであり，Chaffで導入された技術を簡潔かつ効率良く実装し直すことで，大きな性能向上を果たした．そのソースコードは拡張しやすい設計になっており，ほかの多くのSATソルバーがMiniSatをベースに開発されていることから，MiniSatはこの分野を代表的するSATソルバーとなっている．

Glucose☆11は2009年以降の競技会において何度も優勝しており，現在も継続的に開発されている最新ソルバーの1つである．MiniSatをベースに開発され，**UNSAT**問題に対する求解能力を飛躍的に向上させている．これには単位伝播の発生を促進する学習節の評価尺度と，良い学習節を求めて積極的にリスタートする戦略が貢献している．

Lingeling☆12もまた競技会において何度も優勝する成果を挙げている最新ソルバーの1つである．Glucoseと同様の戦略とともに，論理式から冗長な節を除去するなどの簡単化技術を多数実装しており，求解前だけでなく求解中にも簡単化を行うことが特徴である．

GlueMiniSat☆13は筆者らが開発しているSATソルバーであり，求解中に論理式の簡単化を頻繁に試みることが特徴である．GlueMiniSatは2011，2013年のSAT競技会のUNSAT部門で入賞しており，**UNSAT**問題を比較的得意としている．

また**図-3**は，これらのソルバーと最近の競技会で入賞したソルバーを大まかにまとめた系統樹であ

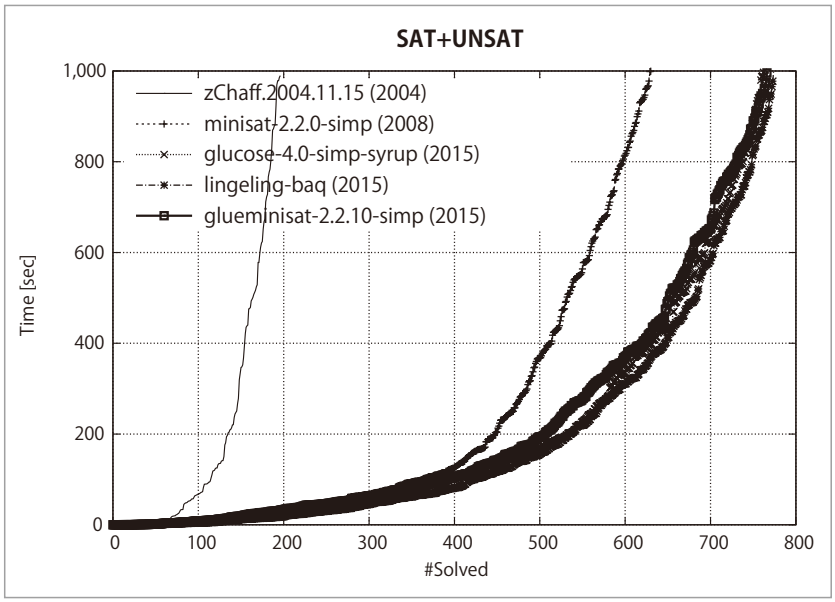

図-2　SAT競技会産業応用部門（2012～2015年の重複を除く1,130問）における求解数と求解時間の分布

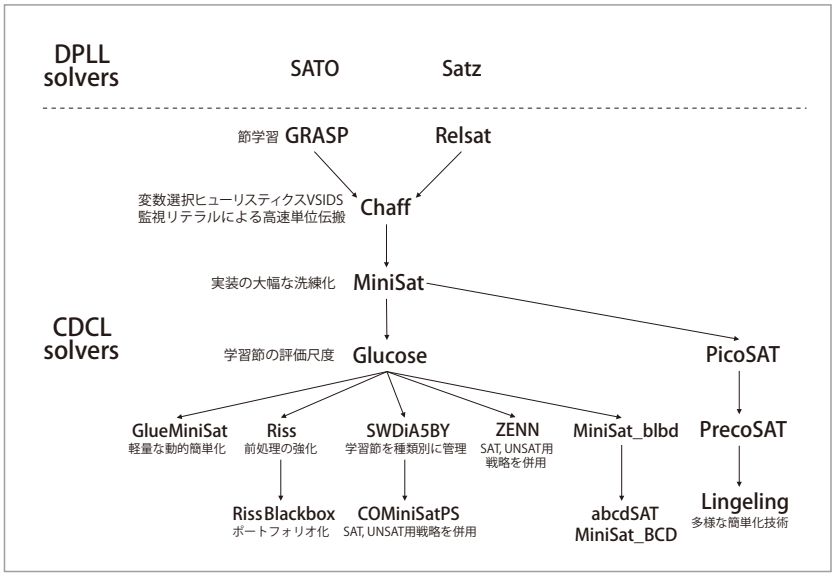

図-3　歴史的SATソルバーと2013～2015年の産業応用部門上位SATソルバーの系統樹

☆9 http://www.princeton.edu/~chaff/　Chaffは2007年以降後継バージョンは発表されておらず，開発は休止しているようである．
☆10 http://minisat.se/　残念ながらMiniSatの開発も2008年以降休止しているようである．
☆11 http://www.labri.fr/perso/lsimon/glucose/
☆12 http://fmv.jku.at/lingeling/
☆13 http://glueminisat.nabelab.org/

競技会	1位	2位	3位	出場ソルバー数
SAT Race 2015 ☆14	abcdSAT	MiniSat_BCD	Riss 5 BlackBox	28
SAT Competition 2014 ☆15	Lingeling ayv	SWDiA5BY A26	Riss 4.27 BlackBox	34
SAT Competition 2013 ☆16	Lingeling aqw	Lingeling 587f	ZENN 0.1.0	29

表-1　産業応用部門における上位SATソルバー

る．多くのソルバーがMiniSatをベースとしており，その拡張性や可読性の高さがうかがえる．

SATソルバー競技会

SATソルバーの実用面での性能を評価する場として，2002年以降，毎年SATソルバーの競技会が開催されている．数多くのソルバーが参加し性能を競い合っており，ソルバーの性能向上を促すよい場となっている．代表的な競技会がSAT Competitionであり，これが開催されない年はSAT Raceなどが開催されている．**表-1**はここ3年間の結果である．ここでは産業応用部門の結果のみを示しているが，競技会によっていくつかの部門がある．産業応用部門は比較的規模の大きな問題を対象とし，この部門ではMiniSat, Glucoseを改良した子孫ソルバーとLingelingが多数入賞していることが分かる．また小規模だが困難な組合せ問題を対象とした部門☆17ではclasp☆18またはその派生ソルバーがよい結果を示している．詳細は競技会のWebページを参照してほしい．

なお，SAT Competitionでは参加ソルバーのソースコードを原則公開することで新規参入時の障壁を下げている．また学生向けにMiniSatハック部門（2016年のSAT競技会からはGlucoseハック部門に継承される）を開催している．これは学生の参加を促すための部門であり，MiniSatにごく小規模な修正を加えて，その性能を競う．図-3におけるSWDiA5BYやZENN, abcdSATなどはこのハック部門をきっかけとして開発されたソルバーである．学生の皆さんにはぜひ参加を検討してもらいたい部門である．

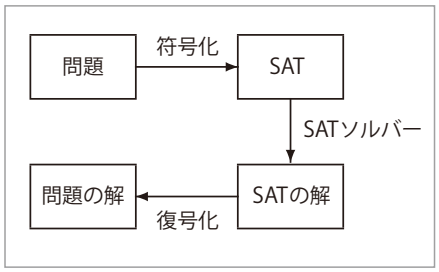

図-4
SATソルバーを利用した問題解法

SATを拡張・発展させた問題

SAT問題は構造が明確で単純であるが，解きたい問題を（CNF式として）直接記述するのは面倒だ．実際，SAT競技会で使われている問題のほとんどは，より複雑な応用問題からSAT問題に翻訳されて作られたものだったりする．**図-4**に問題をSATソルバーを利用して解く様子を描いてみた．まず，より高級な言語で記述された問題（左上）をSAT（右上）に翻訳する．続いてSATソルバーを実行する．最後に，得られたSATの解（右下）から元の問題の解（左下）を得るという流れである．見ての通り，SATソルバーはバックエンドとして優秀な黒子に徹している．また，SATソルバーを利用する際には，問題をいかにしてSAT問題に翻訳（SAT符号化）するかも大事となる．しかし，このSAT符号化の部分も別の優秀な黒子に任せたいところである．そんなこんなで，解きたい元の問題はより高級な言語・形式で記述されることが多い．以下では，この高級な言語・形式として，SATを拡張・発展させた問題を中心に紹介する．

- 制約充足問題（Constraint Satisfaction Problem：CSP）は，整数・実数等の各種ドメインに対して，

☆14 http://baldur.iti.kit.edu/sat-race-2015/
☆15 http://www.satcompetition.org/2014/
☆16 http://www.satcompetition.org/2013/
☆17 Hard Combinatorial または Crafted 部門
☆18 clasp は SAT 技術をベースに開発された ASP ソルバーであるが，SAT や PB 問題を解くこともできる．また解を列挙することも可能である．http://potassco.sourceforge.net/

算術上の制約式を簡潔に記述することができる．また，組合せ問題を解く上で頻繁に現れる複数の制約式を1つにまとめた *alldifferent* $(x_1, ..., x_n)$ ☆19 などの組込み制約を利用できる点も特長である．実際，人工知能分野の多くの問題は，CSPとして定式化できることが知られている．CSPのSAT符号化を利用した高速なCSPソルバーも開発されている☆20．

- 有界モデル検査（Bounded Model Checking：BMC）は，SATソルバーの応用の中で最も成功したものの1つであろう．BMCはシステムの信頼性を高めるための形式的検証手法である．近年，プログラム検証の分野にも応用され注目を集めている．プログラム検証では，配列やリストなどのデータ構造や算術式を記述できるように拡張されたSMT（Satisfiability Modulo Theories）が使われることが多い☆21．
- 問題をSATに翻訳する際に，知識の確からしさや変更可能性の尺度に応じて，節に重みをつけることがある．こうしてできる問題は重み付きMaxSATと呼ばれ，ソルバーも数多く開発されている．このうち，必ず満たしてほしいハード制約とできれば満たしてほしいソフト制約の2種類の重みだけを持つ問題がMaxSATであり，すべてがハード制約である問題がSATに相当する☆22．

上記以外にも，擬似ブール制約（PB），解集合プログラミング（ASP），モデル列挙（#SAT），限量ブール式（QBF）などがあり，SAT技術の進化にはいとまがない．

☆19 変数 x_1 ～ x_n がお互いに異なる値をとることを強制．
☆20 本特集の「2. SATとパズル」を参照．
☆21 本特集の「7. SMTソルバーによるプログラム検証」を参照．
☆22 本特集の「6. MaxSAT：SATの最適化問題への拡張」を参照．

どのソルバーを使えばいい？

与えられた問題を論理式で表現しSATソルバーを用いて解くことから，SAT技術は現代の推論システムにおいて基盤的役割を果たしているといえる．本稿では，こうしたSAT技術の中でも中心的な役割を果たしているSATソルバーに焦点をあて，その正体を探るべく進化の過程を概観した．

冒頭でも紹介したように，SATソルバーはさまざまな分野で実用的に使われ始めている．SAT技術の進化も現在進行形である．一方で，自分が解きたい問題に対して「どのソルバーを使えばいいか？」という質問をよく耳にする．最新のSATソルバー競技会の優勝ソルバーを使えばいいのか．必ずしもそうではない．筆者も明確な答えを持っているわけではない．

とは言え，まず解きたい問題を直接SATに翻訳できる方は，Glucose, claspの順に試していただきたい．翻訳込みでもっと楽にSATソルバーの黒子パワーを享受したい方は，問題を制約充足問題のレベルで記述できるSugar☆23を使ってみてはいかがだろうか．

参考文献
1) 井上克巳，田村直之：SATソルバーの基礎，人工知能学会誌，Vol.25, No.1, pp.57-67 (2010).

(2016年4月29日受付)

☆23 http://bach.istc.kobe-u.ac.jp/sugar/

番原睦則（正会員） banbara@kobe-u.ac.jp

神戸大学情報基盤センター准教授．博士（工学）．制約/論理/解集合プログラミング，SAT等に興味を持つ．

鍋島英知 nabesima@yamanashi.ac.jp

山梨大学大学院医学工学総合研究部准教授．博士（工学）．SAT，プランニング，自動推論等に興味を持つ．