計算機数理科学専攻 計算論講座

酒 井 研 究 室

酒井正彦 教授　　草刈圭一朗 准教授

INFORMATION SCIENCE NAGOYA UNIVERSITY
計算機数理科学専攻

研究概要

『虫のいないプログラムはない』と言われるほど，誤りのないプログラムを作るのは大変な作業です．プログラムの正しさをきちんと表すためには，プログラムの意味を数学的に定めてやる必要があります．

本研究室では，プログラムが持つ基本的な性質の研究を進めており，効率的な計算のための計算戦略やプログラムの持つ性質を自動証明するための枠組みやその基礎技術，与えられたプログラムと意味が同じであるがより効率的なプログラムに変換する方法などについて研究しています．これらを通して，信頼性が高くて効率の良いソフトウエアの作成法の開発を目指しています．

主な研究テーマ

- 関数プログラムの持つ性質を検証するための高階定理自動証明
- SAT ソルバや SMT ソルバの開発と問題解法への応用
- 木オートマトンを用いた決定問題の解法
- プログラム変換による計算の効率化
- 難読言語プログラムの作成法

共同研究者

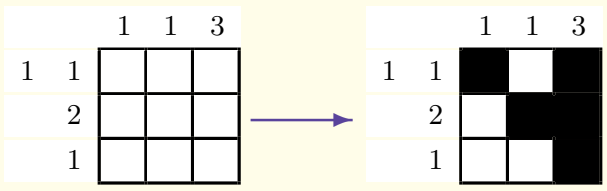
- 本研究科情報システム学専攻
  坂部俊樹 教授，西田直樹 准教授，濱口毅 助教，橋本健二 助教
- オーストリアとの二国間共同研究 (JSPS) ※酒井教授が日本側代表
  Aart Middeldorp 教授，Georg Moser 准教授 (インスブルック大学)
  Bernhard Gramlich 教授 (ウィーン工科大学)
  栗原正仁 教授，佐藤晴彦 助教 (北海道大学)
  小川瑞史 教授，廣川直 准教授 (JAIST)
  岩沼宏治 教授，鍋島英知 准教授 (山梨大学)
- Florent Jacquemard 研究員 (フランス INRIA/IRCAM)

充足可能性判定ソルバ (SAT ソルバ) の飛躍的発展

- 命題論理の充足可能性判定問題は，最初に NP 完全性が証明された計算が困難なはずの問題です．
- SAT ソルバの性能が近年飛躍的に向上し，数百万変数を含む巨大な命題論理式の充足可能性を多くの場合は実用的な時間で解いてしまうようになりました．
- 基本対称節を効率的に SAT ソルバに組み込む手法を研究することにより一意性制約を導入し，効率的に解を求める SAT ソルバを実現しました．
  [馬野, 酒井, 西田, 坂部, 草刈 (2009), 2010 年度電子情報通信学会論文賞受賞]

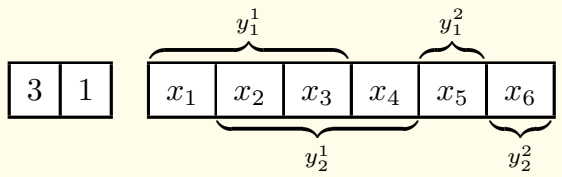
SAT ソルバを用いたお絵かきロジック作成

お絵かきロジックとは，縦と横の数字列をヒントに塗り潰す升目を割り出し，最終的に絵を完成させるパズルです．



お絵かきロジック作成支援ツールによる作成例

SAT 問題への変換



- 各升目に対し塗り潰す事を表す変数 $x_1, \ldots, x_6$ を用意
- ヒントで可能な組合せを列挙することにより制約論理式を生成

$$(x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4} \wedge x_5 \wedge \overline{x_6}) \vee (x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4} \wedge \overline{x_5} \wedge x_6) \vee (\overline{x_1} \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge \overline{x_6})$$

- このままでは式のサイズが爆発してしまいます．30×30 の問題で，最終的に得られる CNF 論理式の節数が 650 万，生成に 15 秒，SAT 問題を解くのに 40 秒かかってしまいました．
- 塗り潰す領域を表す変数 y_i^j を追加で導入．我々が導入した一意に存在することを意味する Unique を用いて，各 j で $\mathrm{Unique}\{y_1^j, y_2^j\}$ を追加．さらに，領域が塗り潰されてたならば領域内の升目は塗り潰され，直後の升目は塗り潰されない事を表現する次の制約を追加．

$$y_1^1 \Rightarrow x_1 \wedge x_2 \wedge x_3 \wedge \overline{x_4} \qquad y_2^1 \Rightarrow x_2 \wedge x_3 \wedge x_4 \wedge \overline{x_5} \qquad y_1^2 \Rightarrow x_5 \wedge \overline{x_6} \qquad y_2^2 \Rightarrow x_6$$

- 最終的に得られる CNF 論理式の節数が 3 万，生成に 0.1 秒，SAT 問題を解くのに 0.01 秒と劇的に効率が上昇します．

高階定理自動証明

- 関数プログラムでは高階関数が広く利用されているため，高階関数を許した系における定理自動証明の研究成果は実在の関数プログラムの検証に直接適用可能となります．
- 計算機は閃きに基づく発見や技巧的な式変形が苦手であるので，『証明』を『計算』に帰着させる証明法が重要となります．
  このような証明法として 1990 年に Reddy が書換え帰納法を提案し，我々が高階の系に拡張しました [草刈, 酒井, 坂部 (2005)]．
- 高階の系に拡張した書換え帰納法を実現する上で障壁なっていたのは停止性自動証明法でした．
  我々は，静的な再帰構造解析に基づく停止性証明法であり実装も容易な静的依存対法を提案しました [草刈, 酒井 (2007)]．この成果により実用的な高階定理自動証明の設計が可能になりました．

アッカーマン関数の通常の定義

$$\begin{cases} \mathtt{ack}_1\ 0\ y & \to & \mathtt{s}\ y \\ \mathtt{ack}_1\ (\mathtt{s}\ x)\ 0 & \to & \mathtt{ack}_1\ x\ (\mathtt{s}\ 0) \\ \mathtt{ack}_1\ (\mathtt{s}\ x)\ (\mathtt{s}\ y) & \to & \mathtt{ack}_1\ x\ (\mathtt{ack}_1\ (\mathtt{s}\ x)\ y) \end{cases}$$

高階原始再帰を用いたアッカーマン関数の定義

$$\begin{cases} \mathtt{iter}\ f\ 0 & \to & f\ (\mathtt{s}\ 0) \\ \mathtt{iter}\ f\ (\mathtt{s}\ x) & \to & f\ (\mathtt{iter}\ f\ x) \\ \mathtt{ack}_\mathtt{h}\ 0 & \to & \mathtt{s} \\ \mathtt{ack}_\mathtt{h}\ (\mathtt{s}\ x) & \to & \mathtt{iter}\ (\mathtt{ack}_\mathtt{h}\ x) \end{cases}$$

書換え帰納法による関数等価性『$\mathtt{ack}_1\ x\ y = \mathtt{ack}_\mathtt{h}\ x\ y$』の証明の流れ