

お絵書きツールを用いた制約プログラミング体験ソフトの開発

高橋 祐樹[†] 西田誠幸[‡]
 拓殖大学工学部情報工学科[§]

1 はじめに

プログラミングの手法の1つに、制約プログラミングがある。制約プログラミングは、「制約条件を満たす答えを探して見つけ出す問題」を解くことを目的としたプログラミングの手法である。制約プログラミングは主にスケジューリングや要因配置、資源割当てといった計画系のアプリケーションを中心に実用化に至っており、この種の問題を解決することに対して優れた能力を持っている。

本発表では、我々が開発した制約プログラミングの機能を付加したドローツールについて述べる。このドローツールは制約プログラミング・ライブラリ iZ-C^[1]を用いている。本ソフトによって、ユーザはパズルの問題を解くための制約条件を考えながら制約プログラミングの手法を体験することができる。

2 制約プログラミング・ライブラリ iZ-C

iZ-C は、整数上の制約プログラムの作成を支援する C 言語のライブラリである。iZ-C は、int 型変数にその値の取り得る範囲の情報を付加した CSint 型変数 (ドメイン変数) を提供する。また、ドメイン変数同士を四則演算する関数や、複数のドメイン変数から 1 つのドメイン変数を作る関数、解を導出する関数などを提供する。

3 制約プログラミング体験ソフトの概要

本節では制約プログラミング体験ソフトの概要を述べる。本ソフトは制約の設定と解決の機能を持つドローツールである。このソフトを使用して数理パズルを作成して解を求めることができる。例えば、魔方陣の場合、ユーザは本ソフト上で図形を $3 \times 3 = 9$ 個配置し、制約として「すべての図形には 1 から 9 までの整数が割り当てられる」、「すべての図形には異なる整数が割り当てられる」、「行と列と対角のそれぞれの和は互いに等しい」などの制約を設定して、解決することによって、魔方陣の解が得られる。本ソフトの作成の方針として、1) 制約解決の処理は iZ-C を使用する、2) ドローツールの図の大きさ、形状、位置は制約に使用しない。位置情報を利用しないで、例えば魔方陣の行の和についての制約を作成するには、和を取る図形のグループをユーザに指定させることにする。

3.1 ソフトの実行手順

本ソフトは以下の 3 つの実行手順で動作する。ユーザはすべての操作をドローツールを通して行う。

1) 問題の作成

ユーザはドローツール上で図形を配置する。本ソフトで配置できる図形として制約に関わる図形と、制約と無関係な図形がある。2 種類の図形にはともに 3 段階の大きさを用意する。

制約に関わる図形として四角と丸の 2 種類があり、本ソフトは、これらの図形を虫食い算や数独の枠として扱う。また、図形 1 個にはそれぞれ 1 つのドメイン変数を割り当て、制約の作成に使用する。

制約と無関係の図形として直線と、+ や - といった演算子の形をした図形がある。

2) 制約条件の設定

ユーザは問題の中のドメイン変数の関係を表す制約条件を設定する。制約条件を設定する図形やドメイン変数を選択し、対応した制約式を設定することで、ソフト内部に制約条件を登録する。

3) 解の導出

すべての制約条件を設定し終えた後にユーザが解導出の命令を下すことで、本ソフトは解を導出し、その結果を表示する。また、複数の解が存在する場合には表示する解を切り替える機能を持つ。

4 本ソフトの構成

本ソフトの構成を図 1 に示す。

本ソフトはドローツールと、制約管理部から構成される。ここで生成されるソースコードは、iZ-C の使用を前提とした制約プログラムである。

特に、解導出の際の処理について述べる。解導出の命令により、制約管理部に保存されたドメイン変数や制約条件を元にソースコードを生成し、生成された

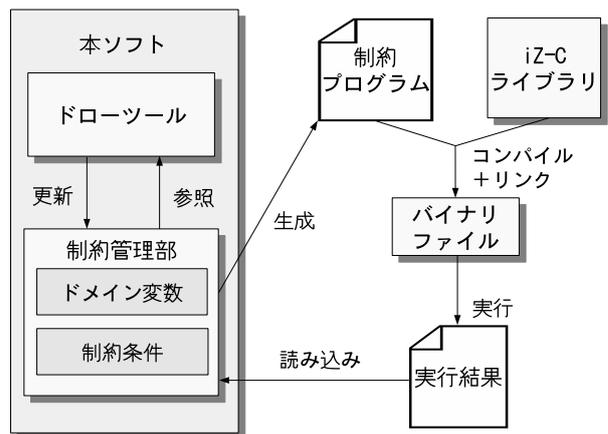


図 1 本ソフトの構成

Development of Drawtool to Experience Constraint Programming

[†]Yuuki TAKAHASHI

[‡]Seikoh NISHITA

[§]Dept. of Computer Science, Faculty of Engineering, Takushoku University

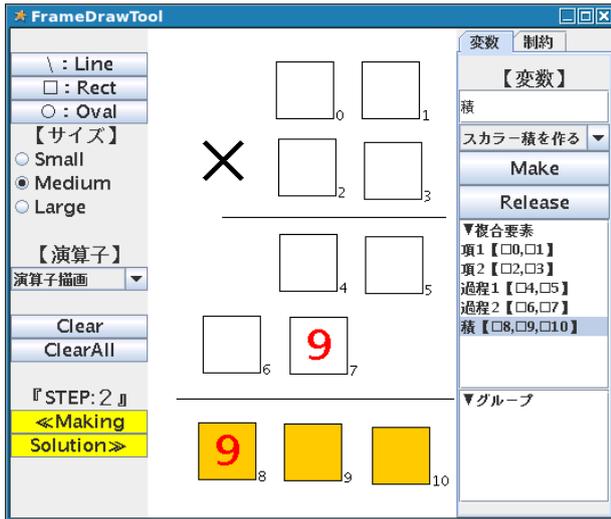


図 2 本ソフトの実行画面

ソースコードを iZ-C とともにコンパイルして実行形式のファイルを実行し解を列挙した実行結果のファイルを生成する。生成されたファイルを再び制約管理部が読み込み、ドローツールを通して解を表示する。

5 本ソフトの実装

本ソフトはプログラム言語として Java5.0 を用いて作成した。図 2 に本ソフトの実行画面を示す。

5.1 制約のデータ構造

本ソフトでは、iZ-C が提供する関数の一部を使用する。iZ-C が提供する関数は、大きく分けて制約プログラムでの「制約」を扱う関数とそれに含まれる「データ」を扱う関数からなる。このうち「制約」を扱う関数に対応するデータ構造として、制約条件のクラスを用意した。この制約条件の種類は単一のドメイン変数と、後述するドメイン変数のグループ配列とで異なる。単一のドメイン変数に対する制約条件には、ドメイン変数同士の四則演算や等式、不等式、比較の関係式、ドメイン変数どうしのたたみ込みがある。また、グループ配列に対する制約条件には、グループ配列に属するドメイン変数すべての同値や非同値の関係などがある。

また、「データ」を扱う関数に対応するデータ構造として制約プログラムでいう「変数名」と「1つのドメイン変数」、「四則演算」などのクラスを用意した。

5.2 ドメイン変数のたたみ込み

iZ-C が提供する関数には、複数のドメイン変数を同一の演算子により計算し、1つのドメイン変数を作成するたたみ込みを行う関数がいくつか存在する。 $a + b + c$ のようなたたみ込みを行った後の値に対して複数回制約条件を設定する場合、何度も同じ図形を選択することがユーザにとって手間となる。そこで、本ソフトではたたみ込みを行った結果と等価なドメイン変数を新たに用意し、ユーザがこのドメイン変数を使用できるようにした。

本ソフトではユーザが図形を複数選択し、たたみ込みの演算を指定することでドメイン変数のたたみ込みを行う。また、たたみ込みドメイン変数にはユーザが任意の名前を付けられる。

5.3 ドメイン変数のグループ化

数独などのパズルでは、「縦一列の要素は全て異なる値を持つ」などのルールが存在する。このルールを制約条件として生成するにはまず「縦一列の要素」のデータ構造を用意しなければならないが、iZ-C にはこのようなデータ構造は存在しない。そこで、複数のドメイン変数の集合を 1つの配列として作成する C 言語の関数を実装し、生成するソースコード内に埋め込むようにした。本ソフトでは、このグループ化によって作成されるドメイン変数の配列をグループ配列と呼ぶことにする。ユーザはたたみ込みと同様の操作でグループ化を行うことができる。

5.4 削除機能

本ソフトではドメイン変数を割り当てられた図形や、ユーザが作成したたたみ込みドメイン変数やグループ配列、それらに対して設定した制約条件などのデータがユーザの操作によって削除された場合、削除されたデータを含むたたみ込みドメイン変数やグループ配列、制約条件をすべて削除する。

5.5 実行手順の後戻り

ユーザが制約条件を設定する段階で、作成した問題に誤りがあることを気づいた場合や、解を導出した後に制約条件を変更して再び解を導出したい場合などに対応するため、本ソフトには実行手順の後戻り機能がある。

後戻り後に、ドメイン変数を割り当てた図形の削除を指定すると、5.4 節同様に削除対象となった図形を含むたたみ込みドメイン変数やグループ配列、制約条件がすべて削除される。

5.6 ソースコードの表示

解を導出し終了後、生成された制約プログラムのソースコードを表示することができる。ソースコードには、プログラムを読みやすくするためのコメント文を添えている。

6 おわりに

本発表では、お絵書きツールを用いた制約プログラミング体験ソフトについて述べた。本ソフトは制約プログラミング・ライブラリ iZ-C を用いてユーザがパズルを解くための制約条件を考えながら制約プログラミングの手法を体験することを目的としたドローツールである。

参考文献

- [1] NTT データセキュリティシステムズ: “iz : Constraint programming”, <http://www.isac.co.jp/iz> (2002).