等価変換モデルにおける情報付き変数を用いた制約充足問題としてのクロスワードパズルの効率的な解法

山田 敬三，馬淵 浩司

岩手県立大学 ソフトウェア情報学部，{k-yamada,mabu}@iwate-pu.ac.jp

1　はじめに

近年，宣言型言語を用いた制約充足問題 (CSP:Constraint Satisfaction Problems) の解法に関する研究が盛んに行われてきた [1]．特に，論理プログラミング言語による研究が多く，Prolog はその代表的な言語である．制約充足問題は NP 困難に属し，一般的には多項式時間で解くことができない難しい問題とされている．こういった問題は，サイズの大きな問題は現実的に解くことはできないが，できるだけ計算時間の増え方を緩やかにすることで，比較的小さなサイズの問題を現実的な時間で解くことが可能となる．しかしながら，Prolog には次の 3 つの計算効率に関する問題点があるため，制約充足問題に関して十分な計算効率を実現することが容易ではない．

1. 計算が推論であり，バックトラック処理を行う．
2. 問題の記述と解法のための記述がセットであるため，プログラムの表現力が低い．
3. 1 つの述語が複数の節からなる場合，上から順に処理を行うため，計算の柔軟性に欠ける．

そこで，我々は，これらの問題点を克服するために，等価変換計算モデル [2] を採用する．制約充足問題を解く場合，問題の制約条件を満たすルールが必要になる．制約条件は問題により異なり，各制約充足問題ごとのルールが必要である．現在までに，等価変換計算モデルを用いて，ナンバープレースなどが制約充足問題として解かれている [3]．この研究では，情報付き変数[1]を使用することを基本とした解法により，効率な情報の伝播を可能にした．このように，ナンバープレースなどの一部の制約充足問題を解く際の有効性は既に示されている．ナンバープレースのような制約充足問題は，1 度の動作では，1 つのマス目 (1 つの変数) の解が決定される．しかしながら，1 度の動作で複数のマス目の解が決定される制約充足問題も存在する．本研究では，このような制約充足問題の 1 つである，クロスワードパズルを対象とする．本研究では，等価変換計算モデルを用いて，[3] とは異なる制約ルールが必要になる制約充足問題を解き，等価変換計算モデルの有効性を示す．具体的には，member 制約を用いてクロスワードパズルを解く．また，member 制約に新たなルールを追加し，より効率的な member 制約を提案する．member 制約に追加するルールは，効率的な情報の伝播や，等価変換計算モデルの特徴である，計算の柔軟性を考慮したルール，および情報付き変数である．本研究により，等価変換計算モデルが，クロスワードパズルを解く際にも有効であることを示すことができる．

本論文では，第 2 章では，本研究で扱うクロスワードパズルを導入する．第 3 章では，等価変換計算モデルの概念について説明する．第 4 章では，等価変換モデルに基づく member 制約を用いた解法について説明する．第 5 章では，本研究で提案する解法を説明する．第 6 章では，従来の member 制約を用いた解法と提案手法の比較を行い，提案手法の有効性を示す．第 7 章は，まとめである．

2　制約充足問題

制約充足問題 [4, 5] とは，与えられた次の 3 つの集合に対して，制約条件 $C_1, C_2, \ldots, C_l$ をすべて満たす変数への割り当てが存在するか否かを問う問題である．

1. 変数の集合 $\{X_1, X_2, \ldots, X_n\}$,
2. ドメイン $D = \{D_1, D_2, \ldots, D_n\}$,
3. 制約条件 $C = \{C_1, C_2, \ldots, C_k\}$.

[1]等価変換モデルでは，効率的な情報伝搬を目的とした，情報付き変数が用意されている．これは，変数に数字などの情報を付加することが可能であり，変数からの情報の取得，追加，削除ができる．ナンバープレースの場合，情報付き変数には，マス目に入る候補，1 から 9 が付加される．

ここで，ドメインの要素 $D_i\,(1 \leq i \leq n)$ は，変数 X_i の取る値の範囲を表している．

本論文では，応用の目的から解の存在を問うだけでなく，すべての解を求めることを目指す．

2.1 クロスワードパズル

クロスワードパズルとは，与えられた盤面の，縦あるいは横の連続した升目に，与えられた単語を入れ，升目をすべて埋めるパズルである．

例 1 図 1 の盤面と，単語として，(アザ)，(アテ)，(タン)，(ラン)，(カタカナ)，(カラアゲ)，(ゲンザイ)，(ナイテイ) が与えられたとき，解は，図 2 のようになる．

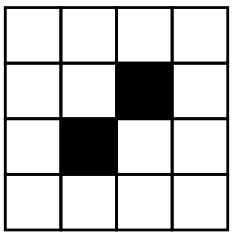

図 1: クロスワードパズル

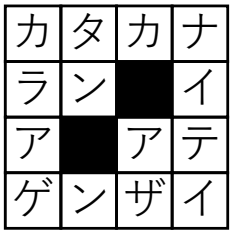

図 2: クロスワードパズル

クロスワードパズルを制約充足問題として定式化すると，与えられた盤面の各白マスに，それぞれ変数を割り当て，各変数のドメインは，単語を表すのに用いるアルファベットとし，制約条件は，以下の 2 つとなる．

1. 盤面の，縦あるいは横の連続したマス目に単語が入る．
2. 重複しているマス目は，2 つの単語に共通した文字が入る．

例 2 先ほどの例 1 を制約充足問題として定式化すると，以下のようになる．

1. 変数の集合 $\{X_1, X_2, X_3, X_4, X_5, X_6, X_8, X_9, X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}\}$.
2. ドメイン $D_i = \{$ ア, イ, カ, ゲ, ザ, テ, タ, ナ, ラ, ン $\}$ $(i \in \{1,2,3,4,5,6,8,9,11,12,13,14,15,16\})$.
3. 制約条件 $C = \{C_1, C_2, \ldots, C_8\}$,

   $C_1 : (X_1X_2X_3X_4) =$ (カタカナ)or(カラアゲ)or(ゲンザイ)or(ナイテイ),

   $C_2 : (X_5X_6) =$ (アザ)or(アテ)or(タン)or(ラン),

   $C_3 : (X_{11}X_{12}) =$ (アザ)or(アテ)or(タン)or(ラン),

   $C_4 : (X_{13}X_{14}X_{15}X_{16}) =$ (カタカナ)or(カラアゲ)or(ゲンザイ)or(ナイテイ),

   $C_5 : (X_1X_5X_9X_{13}) =$ (カタカナ)or(カラアゲ)or(ゲンザイ)or(ナイテイ),

   $C_6 : (X_2X_6) =$ (アザ)or(アテ)or(タン)or(ラン),

$C_7 : (X_{11}X_{15}) = (アザ)or(アテ)or(タン)or(ラン)$,

$C_8 : (X_4X_8X_{12}X_{16}) = (カタカナ)or(カラアゲ)or(ゲンザイ)or(ナイテイ)$.

上の変数の集合と制約条件で，図 3 の盤面を表している．

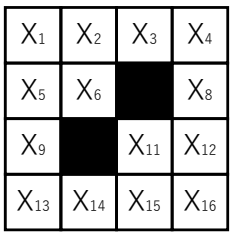

図 3: 盤面に割り当てられる変数

3 等価変換計算

等価変換 [2] とは，記述の意味を変えずに表現だけを変える変換である．等価変換計算モデルにおいては，プログラムは等価変換ルールの集合であり，計算は問題の意味を保持した等価な変換として行われる．

与えられた問題に対して，本質的な意味を変えない変換，すなわち等価変換を繰り返すことによって，最終的な解を得ることができる．等価変換計算モデルの特徴は以下の通りである：

1. 計算が等価な変換で行われるため，Prolog のようなバックトラック処理は行わない．
2. 問題の記述と解法のための記述が分離されているため，プログラムの記述能力が高い．
3. ルールの適用を制御できるため，問題の状況に応じた最適なルールを適用することができる．

これらの点から NP 困難問題である制約充足問題においても効率的な計算が可能になる．また，等価変換ルールは，単独で宣言的な意味を変えない変換を行うことを前提として作成されるので，仕様に対して常に正当性が保証された結果を導くことが可能であり，合流性を考慮する必要がない．さらに，等価変換モデルで用いられる各ルールは独立性が高いため，プログラムの保守性を高く保つことが比較的容易である．

等価変換計算モデルには，決定的な動作である D 規則と共に，非決定的な動作が可能な N 規則が用意されている．等価変換計算モデルでは，非決定的計算を用いることで，Prolog に見られるバックトラック処理を防ぐことが可能であるため，効率的な計算が可能となる．

3.1 D 規則

等価変換計算モデルで用いられる規則は，アトムと呼ばれる，それ以上分解できない最小の単位の文によって構成される．

D 規則は次の形をしている：

$$\begin{aligned}(Head),\{(Cond_1),(Cond_2),\ldots,(Cond_n)\}\\ \longrightarrow (Body_1),(Body_2),\ldots,(Body_m).\end{aligned} \tag{1}$$

式 (1) の先頭にある 1 個のアトム $(Head)$ はヘッドと呼ばれ，変換対象となるアトムのパターンを表す．次にある中括弧で括られたアトム列 $(Cond_1),(Cond_2),\ldots,(Cond_n)$ は，条件列と呼ばれ，規則を適用するための条件が記述される．条件列は 0 個以上のアトム列からなる．なお，条件列のアトムが 0 個の場合は直前にあるコンマと中括弧を省略することができる．矢印の次にあるアトム列 $(Body_1),(Body_2),\ldots,(Body_m)$ はボディ列と呼ばれ，ヘッドと置き換えるアトム列が記述される．ボディ列は 0 個以上のアトムの列からなる．他にも，実際のプログラムでは，B 規則が用意されている．B 規則とは，Bulit-in 規則の略称で，基本的な代入や演算を実装する規則であり，D 規則と同様な決定的な計算である．

3.2 N 規則

N 規則では，先頭に規則名を記述する．規則名は規則の優先度を決定するためにのみ使用され，優先度を指定しない場合は，規則名を省略することができる．N 規則は，次の形をしている：

$$
\begin{aligned}
&RuleName\\
&\quad (Head), \{(Cond_1), (Cond_2), \ldots, (Cond_n)\}\\
&\qquad \Longrightarrow \{(Atom_1), (Atom_2), \ldots, (Atom_k)\},\\
&\qquad\quad (Body_1), (Body_2), \ldots, (Body_m);\\
&\qquad\quad \vdots\\
&\qquad \Longrightarrow \{(Atom_1), (Atom_2), \ldots, (Atom_{k'})\},\\
&\qquad\quad (Body_1), (Body_2), \ldots, (Body_{m'}).
\end{aligned}
\tag{2}
$$

N 規則のヘッド，条件列，ボディ列の動作は，基本的に D 規則と同様である．式 (2) のように，矢印の次にある中括弧で括られたアトム列は実行列と呼ばれ，規則適用に関して必要な代入操作を行う．その際に与えられるアトムは D 規則か B 規則で与えられる．また，実行列は 0 個以上のアトム列からなり，アトムが 0 個の場合，条件列と同様に中括弧とその直後のコンマを省略することができる．N 規則は D 規則と異なり，複数のボディに分岐することができる．その際，ボディアトムを列挙する場合，セミコロンで区切ることで列挙できる．この「動作を分岐することができる」という特徴は，他の宣言的言語には見られない特徴であり，等価変換計算モデルでは，これによって非決定的計算を実現している．

4 member 制約を用いた解法

4.1 概要

この解法は，述語 member を用いたもので，この述語を用いて問題と解法を記述する．述語 member は，次のように用いる：

$$(member \quad 盤面の要素 \quad 単語リスト \quad 残りの単語リスト) \tag{3}$$

この述語は，単語が入るタテまたはヨコの盤面の要素を第 1 引数とし，その盤面の要素に入るべき単語を含む単語リストを第 2 引数，第 2 引数から盤面の要素に入れた単語を除いた差分を第 3 引数とする．

例 3 例 1 のクロスワードパズルは，member 述語を用いて図 4 のように表すことができる．図中 2〜5 行目で図 3 の盤面を表す．6〜8 行目は単語リストである．10〜13 行目は，盤面のヨコの要素を表しており，15〜18 行目は，タテの要素を表している．

member 述語の処理の本体は，図 5 のように表され，次の単一化規則と分岐規則からなる．

4.2 単一化規則

単一化規則は，第 1 引数である盤面の要素に単語を当てはめる規則であり，図 5 中の 2, 5, 9 行目で表される．2 行目は，単語リストに単語が 1 つしかない場合，その単語を盤面の要素に入れて，残りの単語リストを空とすることを表している．もし，盤面の要素に単語が当てはまらないときは，この分岐は失敗する．9 行目は，単語リストに単語が 2 つ以上ある場合，先頭の単語を盤面に入れ，先頭の単語以外を残りの単語リストとすることを表している．5 行目は，特に，盤面の要素の先頭に既に文字が入っている場合である．

```
 1: (cwet *all)
 2:   ==> {(= *all (*x1  *x2  *x3  *x4
 3:                 *x5  *x6       *x8
 4:                 *x9       *x11 *x12
 5:                 *x13 *x14 *x15 *x16)),
 6:        (= *A ((ア ザ) (ア テ) (タ ン) (ラ ン)
 7:               (カ タ カ ナ) (カ ラ ア ゲ)
 8:               (ゲ ン ザ イ) (ナ イ テ イ)))},
 9:
10:       (member (*x1 *x2 *x3 *x4) *A *A1),
11:       (member (*x5 *x6) *A1 *A2),
12:       (member (*x11 *x12) *A2 *A3),
13:       (member (*x13 *x14 *x15 *x16) *A3 *A4),
14:
15:       (member (*x1 *x5 *x9 *x13) *A4 *A5),
16:       (member (*x2 *x6) *A5 *A6),
17:       (member (*x11 *x15) *A6 *A7),
18:       (member (*x4 *x8 *x12 *x16) *A7 *A8).
```

図 4: 例 1 のパズルの member 述語による表現

```
 1: (member *x (*y) *z)
 2:   ==> {(= *x *y),(= *z ())}.
 3:
 4: (member (*a|*X) (*y *z|*W) *AA), {(symbol *a)}
 5:   ==> {(= (*a|*X) *y), (= (*z|*W) *AA)};
 6:   ==> {(= *AA (*y|*AAA))}, (member (*a|*X) (*z|*W) *AAA).
 7:
 8: (member *x (*y *z|*W) *AA)
 9:   ==> {(= *x *y), (= (*z|*W) *AA)};
10:   ==> {(= *AA (*y|*AAA))},(member *x (*z|*W) *AAA).
```

図 5: member 述語本体

4.3 分岐規則

分岐規則は，第 2 引数である単語リストの先頭以外の単語を盤面の要素にいれる候補とするための規則であり，図 5 中の 6, 10 行目で表される．10 行目は，単語リストの先頭の単語をスキップして 2 番目の単語が盤面の要素に当てはまるか否かを調べることを表しており，再帰的に呼び出されることで単語リスト中のすべての単語とのマッチングを試みる．6 行目は，特に，盤面の要素の先頭に既に文字が入っている場合である．

4.4 終了条件

クロスワードパズルにおける終了条件は，制約をすべて満たし，全ての盤面の要素に単語が入っている状態である．実際のプログラム (図 4) では，単語リストの単語と全ての盤面の要素が単一化された後，全ての規則が真であることを返すことがプログラムの終了条件となる．また，いずれかのタテ，ヨコの問題について偽が返され，全ての制約条件を満たす代入が存在しない場合，解が存在しないということになる．

5 提案

クロスワードパズルは，盤面の要素のタテとヨコの交わり方によって，変数のドメインを絞り込むことができるが，member 述語を用いた解法では，この情報を全く用いていない．そこで，我々は，ドメインの絞り込みを情報付き変数を用いて表現し，実行の分岐を抑えることで，高速化を図る．

互いに交差する盤面の要素のうち，要素の長さが異なるか，重なっている場所が異なるタテとヨコの盤面の要素の組を有効な組という．例えば，例 2 において，$(X_1X_2X_3X_4)$ と $(X_4X_8X_{12}X_{16})$ は，4 文字目と 1 文字目で交叉しているので，有効な組である．実際，このことから，変数 X_4 のドメインは { カ, ゲ, ナ } ∩ { ナ, ゲ, イ } = { ゲ, ナ } に絞ることができる．一方，$(X_1X_2X_3X_4)$ と $(X_1X_5X_9X_{13})$ は，長さが同じであり，かつ互いに 1 文字目で交わっているので，有効な組でない．本提案では，このように，有効な組に含まれる共通の変数のドメインを絞り込み，変数に情報を付与することで高速化を図っている．

また，制約充足問題を解く際に，分岐が増えれば，それだけ実行に時間がかかる．そこで，本提案では，なるべく分岐が増えないように，分岐数が最小となるよう実行を制御している．

6 評価

本節では，member 述語を用いた従来の方法と，提案手法とで，いくつかのクロスワードパズルを解いて，実行時間を比較する．実験で使用する問題 [6, 7, 8] の盤面は，図 6 の通りである．各問題の単語リストを以下に示す：

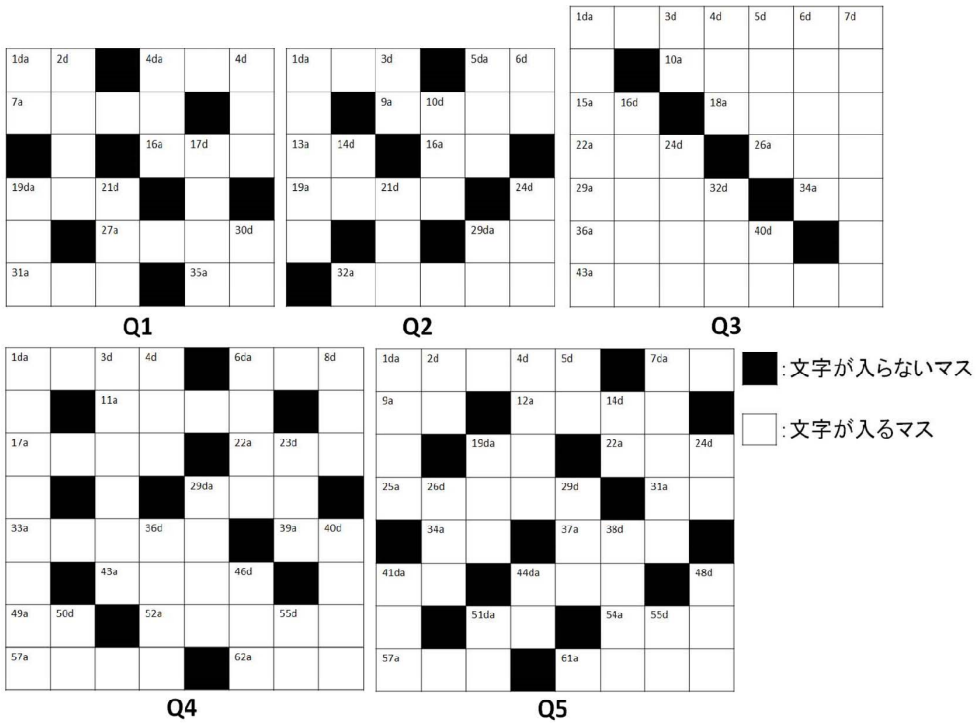

図 6: 使用するパズル

Q1: (カシ) (カグ) (メシ) (キシ) (カイガ) (トウフ) (カマド) (ハカマ) (カレハ) (ドラマ) (カブト) (ガクフ) (グローブ) (ラクガキ) (シロクマ) (ウミガメ)

Q2: (フシ) (ジク) (ダウ) (イン) (チイ) (ダイ) (ジフ) (ドク) (パイ) (ヨウチ) (イドウ) (パーク) (オウジ) (クイーン) (ヨシヨウ) (キチヨウ) (オクジヨウ)

Q3: (L D) (W E) (E C) (A U) (A T) (M E) (E E) (B G) (V O W) (C A N) (I S M) (E T C) (E N D S) (T A L E) (O N T O) (A B L E) (H E A V I E R) (Y O U N G E R) (H O L I D A Y) (R E S C U E R)

Q4: (メシ) (シダ) (コメ) (サメ) (ノツク) (ニダイ) (シゲン) (ズメン) (マナー) (ノミズ) (ダンシ) (クハイ) (ツキモノ) (ジキヒツ) (シキモノ) (ノリニゲ) (イカサマ) (イナオリ) (ポーカー) (ダイキン) (ウシミツ) (ツマハジキ) (ヒノミサキ) (サイカイハツ) (イツポウツウコウ)

Q5: (アオ) (ドン) (メイ) (マイ) (トゲ) (ボブ) (ムチ) (マオ) (ガカ) (アセ) (クサ) (ダイ) (バチ) (ナウ) (カラー) (アンイ) (ニボシ) (ビカン) (ガクヤ) (イナカ) (アコウ) (ルイセン) (ゲーム) (ヤサイ) (サシエ) (ライニチ) (エダマメ) (イントロ) (クチブエ) (クローバー) (エメラルド) (メイコンビ)

これらの問題を解くためにかかった計算時間を表 1 にまとめる．表中 M の列は member 制約を用いた従来手法での計算時間を，P の列は提案手法での計算時間を，M:P の列は，計算時間の比を，それぞれ表す．なお，Q4, Q5 については，従来手法では解いていない．

表 1: 計算時間の比較

問題番号	M(秒)	P(秒)	M:P
Q1	778.29	0.11	7,075
Q2	506.86	0.08	6,336
Q3	19744.29	0.17	116,143
Q4	∞	0.19	—
Q5	∞	0.30	—

7　まとめ

本研究では，member 制約を用いた解法で，先行研究とは異なる制約ルールが必要な制約充足問題を解いた．本研究で扱った制約充足問題の 1 つであるクロスワードパズルは，1 度の操作で複数の変数の値が決定する問題であった．そして，member 制約を用いた解法に，変数のドメインを絞るための情報付き変数を用いた手法と，新たな規則を導入することで，計算効率の向上を実現した．その結果，等価変換計算モデルは，クロスワードパズル解く際にも有効であることを示すことができた．

参考文献

[1] 西原清一：制約充足問題の基礎と展望，人工知能学会誌，vol.12(3)，pp.351–358，1997.

[2] Akama, K. and Nantajeewarawat, E.: “Formalization of the Equivalent Transformation Computation Model,” Journal of Advanced Computational Intelligence and Intelligent Informatics, vol.10, no.3, pp.245–259, 2006.

[3] Mabuchi, H. and Fukuchi, K.: “Efficient Solution of Constraint Satisfaction Problems by Taking into Account the Relationship of Constraints,” International Journal of Innovative Computing, Information and Control, Vol.11, No.1, pp.137–151, 2015.

[4] Tsang, E.: “Foundations of Constraint Satisfaction,” Computation in Cognitive Science, Academic Press, 1993.

[5] van Hentenryck, P., Simonis, H. and Dincbas, M.: “Constraint satisfaction using constraint logic programming,” Artificial Intelligence, vol.58, pp.113–159, 1992.

[6] 山岡憲史：英語のクロスワード 101：基本単語徹底活用，大修館書店，2005.

[7] デラックスクロスワード，2014 年 11 月号，マガジン・マガジン.

[8] スーパークロスワード，2014 年 12 月号，マガジン・マガジン.