3O1-5

センター試験「数学」整数問題解答システムの開発

Automatically Solving Elementary Number Theory Problems in National Center Test

犬塚 慎也 *1　松崎 拓也 *2　佐藤 理史 *2
Shinya Inuzuka　Takuya Matsuzaki　Satoshi Sato

*1名古屋大学 工学部 電気電子・情報工学科
Electrical and Electronic Engineering and Information Engineering, School of engineering, Nagoya University

*2名古屋大学大学院 工学研究科 電子情報システム専攻
Graduate School of Engineering, Nagoya University

We have been challenging to develop a number problem solver for National Center Test for University Admissions (NCTUA). Quantifier elimination is a powerful method to solve math problems, but Peano arithmetic does not allow it. We introduced several higher-order predicates that represent typical sub-problems in NCTUA , and defined a heuristic procedure to output an equivalent quantifier free formula for each predicate. The solver also attempts to eliminate quantifiers from formulas of a restricted system in a special form, which frequently appear in the logical translations of linear integer problems where a coefficient of a bound variable is a free variable.

1. 序論

本研究では、「ロボットは東大に入れるか。」（略称：東ロボ）プロジェクト [新井 12] の一環として、センター試験「数学 I・A」整数問題の自動解答システムの演繹部の開発に取り組んだ。

センター試験の問題はマーク式であり、論理式に翻訳すると空欄部分のみを自由変数とする論理式になる。問題と論理式の例を図 1 に示す。この論理式から限量子を除去することで束縛変数が消去され、自由変数のみを変数とする論理式が得られる。これを整理することで解が得られる。したがって、空欄部分に入る数字を求めることは、論理式から限量子を除去することとほぼ等価であるため、数量を求める数学問題において限量子除去は強力なツールの 1 つである。

従来の東ロボ数学解答システム [岩根 13] では演繹に実閉体上の限量子除去を応用しており、変数の定義域が実数となっている数学問題を主な対象としてきた。一方で、実閉体上とは異なり整数領域上の汎用的な限量子除去手法は原理的に存在しない。加法および等号・不等号のみを含む自然数に関する一階述語論理体系はプレスバーガー算術と呼ばれ、プレスバーガー算術のような限られた体系の式であれば限量子を除去できる。しかし、センター試験の整数問題を論理式に翻訳すると、このような体系の式にならないことが多い。したがって、整数領域上で限量子除去を行う際の各課題は個別に解決するしかない。

本研究ではセンター試験の整数問題を観察した結果 2 つの課題を抽出した。第一の課題は、「条件を満たす変数の最小値」のようなセンター試験に頻出する表現に対応する論理式から限量子を除去することである。これに対しては、問題文に含まれていて人間が汲み取ることのできる意味が失われないような表記を中間表現で用いることで、ヒューリスティクスを用いた限量子除去を効果的に行えるようにした。第二の課題は、プレスバーガー算術において係数が自由変数となる場合を許容するように拡張した体系の式に対する限量子除去である。この形の式はセンター試験で頻出するが、Mathematica などの現行のソフトウェアでは限量子除去ができないことが多い。これに対しては、少数のルールによる論理式の書き換えを再帰的に適用し、限量子が除去できるような式変形順序を探索することにより解決を試みた。

k, l を自然数とし、$m = 6k,\ m + 56 = 44l$ と表すと、k, l について一次不定方程式 $\boxed{\text{ケ}}(k + \boxed{\text{コ}}) + 1 = \boxed{\text{サシ}}\, l$ が成り立つ。

↓（言語処理による論理式への変換）

$$\forall k \forall l \forall m \big((m = 6k \wedge m + 56 = 44l) \to \boxed{\text{ケ}}(k + \boxed{\text{コ}}) + 1 = \boxed{\text{サシ}}\, l\big)$$

↓（限量子除去）

$$22\boxed{\text{ケ}} - 3\boxed{\text{サシ}} = 0 \wedge -2\boxed{\text{ケ}} + \boxed{\text{ケ}}\,\boxed{\text{コ}} + 1 - \boxed{\text{サシ}} = 0$$

↓（簡約）

$$\boxed{\text{ケ}} = 3 \wedge \boxed{\text{コ}} = 9 \wedge \boxed{\text{サシ}} = 22$$

図 1: 整数問題とその計算処理の例

人手で作成した中間表現を入力とした評価実験の結果、本研究で開発した解答システムは Mathematica に比べて約 3 倍の点を獲得することができた。ヒューリスティクスを用いた限量子除去や書き換え探索による限量子除去の効果が確認できた一方、次数の高い式を含む論理式については課題が残った。

2. 問題の分析

センター試験「数学」整数問題は、倍数・約数・剰余、不定方程式、n 進法などが題材となっている。整数問題解答システムを構成するにあたり、どのような数学的課題が存在するのかを確認するために問題の分析を行い、解決すべき目下の課題として以下の 2 つを抽出した。

1 つ目は、「条件を満たす変数の最小値」のようなセンター試験に頻出の表現に対応する論理式から限量子を除去できないという問題である。例えば、次のような問題を考えると、一階述語論理式による表現は可能だが、このように書き下してしまうと Mathematica では限量子を除去できない。

連絡先: 犬塚慎也，名古屋大学工学部 電気電子・情報工学科，名古屋市千種区不老町 C3-1(631)，052-789-4435，s_inuzuk@nuee.nagoya-u.ac.jp

表 1: 論理式の種類による問題の分類と出題数

論理式の種類	出題数 (問)
① 直接計算	23
② プレスバーガー算術	19
③ 高階述語	30
④ プレスバーガー算術の拡張	14
⑤ 高次方程式・不等式	38

$\sqrt{756m}$ が自然数となる最小の m は $\boxed{\text{カキ}}$ である。

$$\updownarrow$$

$$\exists k(756\,\boxed{\text{カキ}} = k^2) \land \forall m'\bigl(\exists k'(756m' = k'^2) \to \boxed{\text{カキ}} \le m'\bigr)$$

このような問題は、一階述語論理式ではなく高階論理による記述を用いることで、問題文に含まれていて人間が汲み取ることのできる意味が失われないようにすることが適切であると考えた。そのために、整数問題に頻出の表現に対応する専用の述語・関数をあらかじめ定義しておき、これらの述語・関数を簡約するヒューリスティックな処理をあらかじめ定義しておくことで限量子を含まない等価な式を作り出すことを試みる。

2 つ目の課題は、プレスバーガー算術において係数が自由変数となる場合を許容するように拡張した体系の式に対する限量子除去が Mathematica などの現行のソフトウェアによって行えないという問題である。センター試験で頻出する、次のような一次式を題材とした問題には係数が空欄となったものが多く登場するため、論理式にすると自由変数と束縛変数の積、すなわち 2 次式を含む論理式となってしまう。

方程式 $4x - 5y = 1$ の整数解は、整数 k を用いて

$x = \boxed{\text{ア}}\,k + 4,\ y = \boxed{\text{イ}}\,k + 3$ と表される。

$$\updownarrow$$

$$\forall x \forall y\bigl(4x - 5y = 1 \to \exists k(x = \boxed{\text{ア}}\,k + 4 \land y = \boxed{\text{イ}}\,k + 3)\bigr)$$

このような論理式に対してはプレスバーガー算術の定理証明機は使えず、Mathematica による限量子除去も行えないことが多い。しかし、このような数式を人手で処理する際の手続きは一定であるため、これは数ステップ程度の論理式の書き換えで代替することができると期待される。そこで、あらかじめ用意した少数のルールによる論理式の書き換えを再帰的に適用し、限量子が除去できるような式変形順序を探索することにより限量子除去を試みる仕組みを実装した。

ここで、上記 2 つの問題の重要性を確認するため、論理式で表現した場合に、どのような種類の式になるのかという観点から問題を分類した。調査対象として、センター試験過去問 4 問（本試験 2 問・追試験 2 問）・予備校マーク式問題集から 34 問（河合塾 7 問、駿台 16 問、代ゼミ 5 問、Z 会 6 問）、実教出版重要問題集から 21 問の計 59 問を用いた。

問題観察の結果、論理式の形を見たときに、①素因数分解のように直接計算可能なものや、②プレスバーガー算術の式のように計算の手続きが明らかなもの以外に、③高階の述語・関数による記述が適当なもの、④プレスバーガー算術において係数が自由変数となる場合を許容するように拡張した体系の式、があることがわかった。これ以外のものを⑤高次方程式・不等式を含む論理式、として、図 2 に記載した方法で問題を分類したところ、問題数は表 1 の通りであった。なお、1 つの大問に複数の種類の論理式が含まれている場合は各種類に算入しているため、表 1 の出題数の合計は大問数の合計とは一致しない。

問題の数としては⑤高次方程式・不等式 が最も多いが、この形の式に対する限量子除去は一般には不可能である。一方で、③高階述語、④プレスバーガー算術の拡張 はともに多く出題されている。③④に含まれる問題は現状のソフトウェアでは解くことができないため、前述の 2 つの課題の解決は重要であるといえる。

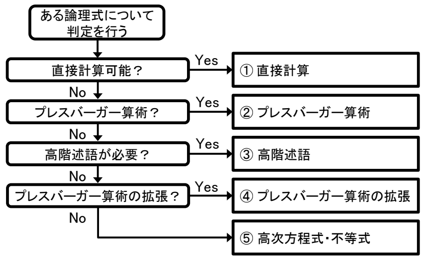

図 2: 論理式の種類による問題分類の流れ

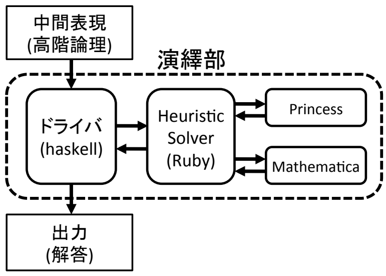

図 3: 整数問題解答システムの構成

3. システム概要

整数問題解答システムの構成を図 3 に示す。解答システム演繹部は高階論理によって記述された数学問題を入力とし、解答を出力とする。演繹部においては、入力された問題はまずドライバによって複数の小問題に分割され、それぞれが整数問題用ソルバー（Heuristic Solver）に入力される。整数問題用ソルバーでは、ヒューリスティクスを適宜用いて入力された式に対して限量子の除去および述語・関数に対応する計算処理を試みる。その過程の一部で、限量子の除去に Princess[Philipp 08] および Mathematica を用いる。処理が終了した式はドライバに返され、解答欄の桁数の条件を加えて解が求められる。なお、言語処理部との接続を念頭に、本稿では演繹部の入力を「中間表現」と呼ぶ。

3.1 中間表現

演繹部の入力となる中間表現は一階述語論理式による記述を基本とし、それに特殊な述語を加えた体系の式として記述する。特殊な述語による記述の例を次に示す。

$92x + 197y = 10$ をみたす整数 x, y の組の中で、

x の絶対値が最小のものは $x = \boxed{\text{ア}}$, $y = \boxed{\text{イ}}$ である。

$$\updownarrow$$

$$\text{nth-smallest-elem}(\{\boxed{\text{ア}}, \boxed{\text{イ}}\}, \{x, y\}, 92x + 197y = 10, |x|)$$

この例では、“nth-smallest-elem” という述語を用いている。この述語は 5 つの引数をとり、第 2 引数で指定した変数 (x, y) が第 3 引数の条件 $(92x + 197y = 10)$ を満たすときに、第 4 引数 $(|x|)$ の値を最小にするような x, y の組が第 1 引数 $(\boxed{\text{ア}}, \boxed{\text{イ}})$ になる、という意味である。

中間表現では、加法・減法・乗法・べき乗、商・剰余、等号・

表 2: 中間表現で用いる述語・関数

述語	述語の意味
count-integer($\{x, y, ..\}, \phi$)	条件 ϕ を満たす解の組 $(x, y, \ldots)$ の数
nth-largest-elem(a, x, ϕ, f, n)	ϕ を満たす x のうち f を n 番目に大きくする組が a に等しい
nth-smallest-elem(a, x, ϕ, f, n)	ϕ を満たす x のうち f を n 番目に小さくする組が a に等しい
nth-largest-fun-value(a, x, ϕ, f, n)	x が ϕ を満たすとき f の n 番目に大きい値が a になる
nth-smallest-fun-value(a, x, ϕ, f, n)	x が ϕ を満たすとき f の n 番目に小さい値が a になる

Algorithm 1 整数問題用ソルバーが行う処理の全体像

```
procedure SOLVE(φ)                              ▷ φ から限量子を除去する
    for all φ_i (φ の部分式) do                  ▷ φ_i の詳細は本文を参照
        φ_i ← SOLVE(φ_i)                        ▷ 部分式から処理を行う
    end for
    if φ に限量子が含まれている then
        φ ← REDUCE(φ)                           ▷ 限量子の除去を試みる
    else if φ に専用の述語が含まれている then
        φ ← 専用述語処理手続 (φ)                  ▷ 専用の述語の処理を試みる
    end if
    return φ
end procedure
```

不等号、通常の論理結合子・限量子のほかに、表 2 に示す専用の述語を用いて論理式を記述する。

3.2 演繹部

処理の流れは Algorithm 1 の通りである。Algorithm 1 では、まず入力の根になっている演算子に関してその引数となっている部分式を再帰的に処理した後に限量子除去あるいは述語・関数に対応した計算を行う。

(1) 独自の述語・関数の処理

述語・関数部分の処理内容は述語・関数ごとにあらかじめ定義する。処理では適宜ヒューリスティクスを用いる。

たとえば、nth-smallest-elem であれば Algorithm 2 に従って計算処理を行う。論理式を組み立てたときにプレスバーガー算術の式となる場合は、プレスバーガー算術の定理証明機である Princess により限量子除去を試みる。その他の場合は論理式の組み立ては行わず、ヒューリスティクスにより解を求める。たとえば、$cond \equiv \exists k(1 \leq k \leq 9 \land x = 12k^2 + 5)$ のように束縛変数が有界であれば展開して限量子を除去し、$cond \equiv (x = 12{\cdot}1^2{+}5) \lor (x = 12{\cdot}2^2{+}5) \lor ... \lor (x = 12{\cdot}9^2{+}5)$ のようにして解の列挙に変形できるため、この中で評価関数を最小にするものを探す。また、$f = x, cond \equiv \exists k(x = 12k^2+5)$, $n = 1$ が入力されたときは、$12k^2 + 5$ の最小値が 5 であることより、$opt = 5$ を返す。

(2) グラフの探索による限量子除去

限量子除去は Algorithm 4 に従って行われる。入力がプレスバーガー算術の式でない場合はまずグラフ探索による限量子除去を行い、続いて Mathematica により限量子除去を試みる。

グラフの探索による限量子除去では、入力された論理式の形に従い、適用可能な式変形規則を幅優先探索で再帰的に適用することにより限量子の除去を試みる。式変形規則は表 3 のものを用いる。限量子の除去が達成された、あるいは試行回数が閾値を超過した時点で探索を終了し、変形結果のうち束縛変数の個数が一番少ないものを最良の結果として返す。

なお、表 3 に示した 2 番目の規則は、プレスバーガー算術を拡張した体系の式に対して適用することを想定している。変形規則の適用例は図 4 通りである。

Algorithm 2 $cond$ を満たす $\{x, y, \cdots\}$ のうち f を n 番目に小さくする組 opt を求める

```
procedure NTH-SMALLEST-ELEM(opt, {x, y, ..}, cond, f, n)
    if cond および f がプレスバーガー算術の式 then
        return NTH-SMALLEST-ELEM-PA(opt, {x, y, ..}, cond, f, n, 1)
    end if
    cond ← REDUCE(cond)
    if cond に解が列挙されている then
        return opt = (f を n 番目に小さくする組)
            ▷ 順番に代入して試した結果、f を n 番目に小さくしたものを返す
    else if パラメータとなる束縛変数 k の変域が上限・下限を持つ then
        return opt = (f を n 番目に小さくする組)
                                      ▷ 列挙できる場合と同様、順番に試す
    else if f ∈ {x, y, ..} ∧ パラメータ表示が 2 次式で下限を持つ then
        return opt = (f を n 番目に小さくする組)
            ▷ f の値が n 番目に小さくなるパラメータに関する組を返す
                                                          ▷ (本文参照)
    end if
    return error
end procedure
```

Algorithm 3 プレスバーガー算術の範囲で論理式を記述できるときの nth-smallest-elem の動作

```
procedure NTH-SMALLEST-ELEM-PA(a, x, φ(x), f(x), n, i)
    a_i ← PRINCESS(φ(a) ∧ ∀x(φ(x) → f(a) ≤ f(x)))
    if i = n then
        return a_i
    else
        φ(x) ← φ(x) ∧ x ≠ a_i
        i ← i + 1
        return NTH-SMALLEST-ELEM-PA(a, x, φ(x), f(x), n, i)
    end if
end procedure
```

Algorithm 4 限量子除去手続き

```
procedure REDUCE(φ)                          ▷ φ から限量子を除去する
    if φ がプレスバーガー算術の式 then
        return PRINCESS(φ)        ▷ Princess により限量子除去を試みる
    end if
    φ ← GRAPHSEARCH(φ)          ▷ グラフ探索により限量子除去を試みる
    if φ が限量子を含む then
        φ ← MATH_REDUCE(φ)
                              ▷ Mathematica により限量子除去を試みる
    end if
    return φ
end procedure
```

$$\forall x \forall y\big(4x - 5y = 1 \to \exists k(x = \boxed{\text{ア}}\,k + 4 \land y = \boxed{\text{イ}}\,k + 3)\big)$$
$$\updownarrow \text{(規則 1)}$$
$$\forall x \forall y\big(\exists l(x = 5l + 4 \land y = 4l + 3) \to \exists k(x = \boxed{\text{ア}}\,k + 4 \land y = \boxed{\text{イ}}\,k + 3)\big)$$
$$\updownarrow \text{(規則 7)}$$
$$\forall x \forall y \forall l\big((x = 5l + 4 \land y = 4l + 3) \to \exists k(x = \boxed{\text{ア}}\,k + 4 \land y = \boxed{\text{イ}}\,k + 3)\big)$$
$$\updownarrow \text{(規則 8)}$$
$$\forall l \forall x \forall y\big((x = 5l + 4 \land y = 4l + 3) \to \exists k(x = \boxed{\text{ア}}\,k + 4 \land y = \boxed{\text{イ}}\,k + 3)\big)$$
$$\updownarrow \text{(規則 5)}$$
$$\forall l \exists k(5l + 4 = \boxed{\text{ア}}\,k + 4 \land 4l + 3 = \boxed{\text{イ}}\,k + 3)$$
$$\updownarrow \text{(規則 2)}$$
$$\Big((\boxed{\text{ア}} = 5 \land \boxed{\text{イ}} = 4) \land \exists \alpha(4 = 5\alpha + 4 \land 3 = 4\alpha + 3)\Big)$$
$$\lor\Big((\boxed{\text{ア}} = -5 \land \boxed{\text{イ}} = -4) \land \exists \alpha(4 = -5\alpha + 4 \land 3 = -4\alpha + 3)\Big)$$
$$\updownarrow \text{(Mathematica により簡約)}$$
$$(\boxed{\text{ア}} = 5 \land \boxed{\text{イ}} = 4) \lor (\boxed{\text{ア}} = -5 \land \boxed{\text{イ}} = -4)$$

図 4: 式変形規則の適用例

4. 評価

システムの評価に際し、予備校マーク式問題集に掲載されている整数問題 25 問（河合塾 7 問、駿台 8 問、代ゼミ 5 問、

表 3: グラフ探索で用いる式変形規則

番号	変形前 *1	変形後 *1
1*2	$\sum_{j=1}^{n} i_j x_j = i$	$\exists k_1, \ldots, k_{n-1} \left(\bigwedge_j x_j = \sum_{l=1}^{n-1} a_l k_l + b_j \right)$ (不定方程式をパラメータ表示する)
2*3	$\forall x \exists y \left(\bigwedge_k i_{k1} x + i_{k2} = a_{k1} y + a_{k2} \right)$	$\bigvee_d \left(\bigwedge_k a_{k1} = \frac{i_{k1}}{d} \wedge \exists \alpha \left(\bigwedge_k a_{k2} = \frac{i_{k1}}{d} \alpha + i_{k2} \right) \right)$ (d は全ての i_{k1} の公約数)
3	$\forall x \left(\sum_{i=0}^{n} \phi_i x^i = 0 \right)$	$\bigwedge_{i=0}^{n} \phi_i = 0$
4*4	$a \mid i$	$\bigvee_j a = j$ (j は i の因数)
5*5	$\forall \boldsymbol{x} (\boldsymbol{x} = \boldsymbol{a} \to \phi(\boldsymbol{x}))$	$\phi(\boldsymbol{a})$
6	$N \bmod \alpha = r \to \phi(N \mathrm{div}\, \alpha)$	$\forall q (N = \alpha q + r \to \phi(q))$
7	$\exists x \phi(x) \to \varphi$	$\forall x (\phi(x) \to \varphi)$
8	$\forall x \forall y \phi(x, y)$	$\forall y \forall x \phi(x, y)$
9	$\exists x \exists y \phi(x, y)$	$\exists y \exists x \phi(x, y)$
10	$(\phi_1 \wedge \phi_2) \to \phi_3$	$\phi_1 \to (\phi_2 \to \phi_3)$
11	$(\phi_1 \vee \phi_2) \to \phi_3$	$(\phi_1 \to \phi_3) \wedge (\phi_2 \to \phi_3)$
12	$\phi_1 \wedge \phi_2$	$\phi_2 \wedge \phi_1$
13	$\phi_1 \vee \phi_2$	$\phi_2 \vee \phi_1$

*1 i, i_j, i_{k1}, i_{k2}, j, d は定数。

*2 a_l, b は変形前の方程式により決定される定数である。変形前は一次方程式なので Mathematica によりパラメータ表示を行い、a_l, b を決定する。使用例は本文中の例における（規則 1）の部分を参照。

*3 a_{k1}, a_{k2} は自由変数でも定数でもよいが、主に自由変数である状況を想定している。使用例は本文中の例における（規則 2）の部分を参照。詳細は本文において解説している。

*4 a は自由変数。

*5 $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ である。

表 4: 開発用データに対する性能評価

-	得点	得点率 [%]	得点比
Mathematica	34/300	11.3	-
解答システム	106/300	35.3	3.12

表 5: 評価用データに対する性能評価

-	得点	得点率 [%]	得点比
Mathematica	26/200	13.3	-
解答システム	72/200	36.0	2.77

Z 会 5 問）を用いて、対応する中間表現を手作業により作成した。このうち 10 問を評価用データとし、残りの 15 問を開発用データとした。この中間表現に対し、Mathematica のみを用いた場合、および解答システムを用いた場合のそれぞれについて、得点と得点率を調べた。また、解答システムの性能を評価する指標として、Mathematica の得点に対する解答システムの得点の比率（得点比）を調べた。

開発用データに対する評価、および評価用データに対する評価をそれぞれ表 4、表 5 に示す。開発用データ・評価用データともに約 35 %の得点率であり、解答システムでは Mathematica に比べて平均して約 3 倍の得点を得ることができた。

また、中間表現における論理式の種類ごとに得点と得点率を調べたところ表 6 の通りであった。直接計算可能なもの、プレスバーガー算術およびそれを拡張した体系の式、独自の述語・関数を用いた部分では正答率 50%以上を達成した。プレスバーガー算術を拡張した体系の式については正答率 72%となっており、これはグラフ探索による限量子除去の効果である。また、独自の述語を用いた部分は正答率 56%であり、こちらも効果が確認できた。しかし、それ以外の高次方程式・高次不等式を含む論理式は正答率 3%に留まった。なお、プレスバーガー算術にもかかわらず得点できていない部分は前の問題の結果を利用する形の問題だったためである。

さらに、点を落とした原因について調べたところ、主なもの

表 6: 論理式の種類に対する正答数と正答率

論理式の種類 1	得点 2	正答率 [%]
① 直接計算	15/15	100
② プレスバーガー算術	15/25	60
③ 高階述語	22/39	56
④ プレスバーガー算術の拡張	18/25	72
⑤ 高次方程式・不等式	2/79	3

1 論理式の種類の分類は表 1 と同じ。

2 中間表現として記述できなかった部分があるため、合計得点は 200 点とはならない。

表 7: 点を落とした原因と落とした点数

原因	落とした点数
次数の大きい式	69
述語・関数	17
前の問題の結果を用いる問題	14
代数的でない式	11

は表 7 の通りであった。原因として最も多いのは次数の大きい式から限量子を除去できないことであった。現在は次数が大きい式を含む論理式が現れたとき、Mathematica が解くことのできる既知のディオファントス方程式でない限りは限量子を除去できない。これら全てについて限量子を除去できる汎用的アルゴリズムを構成するのは原理的に不可能である。これらの式の一部について限量子除去を可能にするためには、限量子を除去できそうな部分的体系を切り出し、その体系における限量子除去手続きを検討する必要があり、これは今後の課題である。

また、述語・関数のうち処理ができなかったものの多くは引数の簡約に失敗しているが、それらのうち一部は引数の束縛変数が有界であるため、実際には限量子を除去可能である。式の形に従って論理式を展開するなどのヒューリスティクスの充実も課題である。オイラー関数などの代数的でない関数を含む式に対する限量子除去を行う際にも、そのようなヒューリスティクスが必要である。

5. 結論

本研究では、センター試験「数学」整数問題解答システムの演繹部を開発した。センター試験の整数問題に対応する論理式に見られる数学上の問題を解決するため、中間表現で高階の述語を用いて記述することで演繹時にヒューリスティクスを用いた処理を行えるようにし、また事前に用意した式変形規則を繰り返し適用することで限量子の除去を試みる仕組みを実装した。これらには一定の効果があったが、特に高次式を含む一階述語論理式からの限量子除去にはまだ課題が多い。

今後は、現段階で限量子を除去できない形の論理式のうち、限量子を除去できそうな部分的な体系を見つけていき、その体系の式に対する処理の方法を詳細に検討する必要がある。

参考文献

[新井 12] 新井 紀子, 松崎 拓也: ロボットは東大に入れるか?—国立情報学研究所「人工頭脳」プロジェクト—, 人工知能学会誌 Vol. 27, No. 5, pp. 463–469 (2012)

[岩根 13] 岩根 秀直, 松崎 拓也, 穴井 宏和, 新井 紀子: 数式処理による入試数学問題の解法と言語処理との接合における課題, 人工知能学会全国大会論文集 (CD-ROM) Vol. 27, ROMBUNNO.2A4-2 (2013)

[Philipp 08] Philipp Rümmer: A constraint sequent calculus for first-order logic with linear integer arithmetic, LNCS, vol. 5330, pp. 274–289. Springer (2008)