2A4-1

深い言語理解と数式処理の接合による入試数学問題解答システム

Solving University Entrance Exam Math Problems
through the Integration of Deep Language Understanding and Computer Algebra

松崎 拓也 [*1] 岩根 秀直 [*2] 穴井 宏和 [*2 *3] 相澤 彰子 [*1] 新井 紀子 [*1]
Takuya Matsuzaki Hidenao Iwane Hirokazu Anai Akiko Aizawa Noriko Arai

[*1]国立情報学研究所 [*2]（株）富士通研究所 [*3]九州大学
National Institute of Informatics Fujitsu Laboratories Ltd. Kyushu University

We firstly elucidate what is “to solve math problems by a machine,” and then describe a symbolic system that instantiates it. In the system, a problem text is translated to a directive that commands to find an answer, i.e., a term in ZF set theory which is in a certain form and satisfies the condition given in the problem. The directive is then transformed to a formula in the language of a solver, which finds an answer through inference. A preliminary experiment using real entrance exam math problems showed positive results for the viability of the approach. Finally, we introduce an annotated math problem corpus that will be made available for the research community.

1. 「問題を解く」とは何か

そもそも数学の「問題を解く」とは何であろうか．どのような状態になれば問題は「解かれた」といえるのだろうか．例えば次のような問題について考えてみよう．

> xy-平面上の放物線 $y = x^2$ を C とする．C 上の点 (a, a^2) における C の接線の方程式を求めよ．

問題文に書かれた日本語を素直に解釈するならば，これは「C と名付けられた「$y = x^2$」というオブジェクトの「(a, a^2) における接線の方程式」というオブジェクト」を計算する問題ということになる．あらゆる数学のオブジェクトは Zermelo-Fraenkel の公理的集合論 (ZF) の（保存拡大の）項だと考えることができるので，ここでいう「計算」とは ZF の項の書き換えだと見做せよう．では，その計算をどこで止めるべきか．即ち，問題文の直訳である項を，それと同等であるような無数の項のうちから，どのようなものに書き換えれば問題が「解けた」ことになるのだろうか．それを考えるヒントは，解答群の中に見いだすことができる．大学入試を例にとると，証明問題以外では，解答に現れるのは，$y = 2ax - a^2$，$(x < 0 \to a = 3) \wedge (x \geq 0 \to a = 5)$，$a_1 + \cdots + a_n > 0$ のような限量記号をひとつも含まないような式である．しかも，その式は，実閉体の理論に三角関数や指数関数などの超越関数をシンボリックにしか利用しないような拡張を行った実閉体の体系（拡張 RCF）に弱いペアノ算術の体系を加えた体系（RCF^++PA) で表現されるようなものにほぼ限られる．また，模範解答に現れる式も，実は RCF^++PA で記述可能な式が圧倒的に多いことに気づく．となれば，問題文を同等の RCF^++PA の式に変換し，その式から限量記号を消去することが，大学入試の数学問題を「解く」ことだと考えてもよいだろう．

では，「(a, a^2) における $y = x^2$ の接線の方程式を求める」という問題は，どのような RCF^++PA の式に変換しうるのだろうか．直線の方程式を $y = bx + c$ と置き，「接する」を $\epsilon - \delta$ 記法を使って愚直に翻訳すると，次のようになる:

$$\left(\begin{array}{l} y = bx + c \wedge (a^2 = ba + c) \wedge \\ \forall \epsilon > 0 \exists \delta > 0 \forall h \left(\begin{array}{l} 0 < |h| < \delta \\ \to \left| \frac{(a+h)^2 - b(a+h) - a^2 + ba}{h} \right| < \epsilon \end{array} \right) \end{array} \right)$$

これは，RCF の言語に絶対値記号を付け加えた保存拡大 (RCF+a.v.) の式である．RCF に対しては限量記号消去（QE: quantifier elimination）定理 [Tarski 51] が成り立つが，RCF+a.v. についても同じく QE 定理が成り立ち，上の式と同等な限量記号を含まないような式を求めることができる．RCF の QE は Mathematica をはじめとして Maple 上で動作する SyNRAC[Iwane 13] などで実装されている．SyNRAC を用いて上の式に対して QE を実行した結果は以下のとおりである．

$$2a - b \leq 0 \wedge -2a + b \leq 0 \wedge -bx - x + y = 0 \wedge a^2 - ab - c = 0$$

限量記号を含まない RCF^++PA の式から所望の答え，

$$y = 2ax - a^2$$

を得ることは難しくはない．どの程度の RCF の拡張に対して QE 定理が成り立つかについては未解決な点が多い [Dries 88]．たとえば，RCF の言語に指数関数を加えた体系に対して限量記号消去が成り立つかは未解決である．

RCF は，数学的に意味をもつ体系であるにも関わらず QE が成り立つ稀有な例のひとつだが，それにかかる計算量は与えられた式に含まれる変数の $\forall$ と $\exists$ の交替の数の二重指数関数になる．実用に供されているほぼすべての RCF-QE が採用している CAD と呼ばれるアルゴリズムの計算量は式に含まれる変数の数の二重指数関数である [Collins 75]．よって，実際に問題を機械で解く上では，変数を抑制する必要があるため，「接する」は $\epsilon - \delta$ 記法は用いずに判別式を用いて限量記号を消去するほうが賢明である:

$$\left(y = bx + c \wedge b^2 + 4c = 0 \wedge a^2 = ba + c\right)$$

RCF-QE を用いて問題を「解く」には，QE を低コストで実行できるような RCF の拡張言語と変数を抑制するための工夫が必要になる．これについては別稿 [岩根 13] で詳しく触れる．

一方，ペアノ算術は，Robinson の Q のような極めて弱い部分体系でも強い意味で決定不能であることが知られている [Robinson 50]．当然ながら QE 定理は成り立たない．そこで，(何らかの）帰納法を要する問題では，問題文と同等の論理式に現れる自由変数を文脈にそって全称限量記号・存在限量記号で束縛し，自動証明機にかけた上で証明図から存在限量記号が指し示す「答え」を抽出するという方法が考えられる．たとえ

連絡先: 松崎拓也，国立情報学研究所，東京都千代田区一ツ橋 2-1-2，takuya-matsuzaki@nii.ac.jp

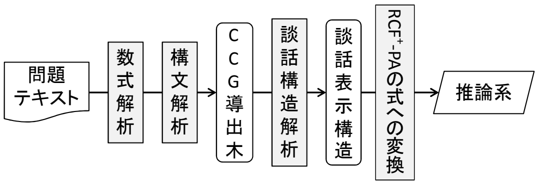

図 1: 解答システム処理フロー

ば,「奇数 m を平方数の差で表せ」という問題を,「奇数 m に対してある整数 n と p が存在し，$m = n^2 - p^2$ が成り立つ」を表わす PA の式に翻訳する．

$$\forall m(\exists k(m = 2k+1) \to \exists n \exists p(m = n^2 - p^2))$$

この証明図を自動的に生成すると，その証明図に,

$$m = 2k+1 \to m = (k+1)^2 - k^2$$

という式が現れざるを得ない．このとき，$(k+1)$ を n の，k を p の witness といい，所望の答えとなる．一般には証明図が与えられたからといって存在限量記号の witness が常に得られるとは限らない．また，問題文の答えのすべてが得られた証明図の witness によって網羅される保証もない．だが，その可能性は高い．

以下では，まず本節で述べた方針に基づく解答システムの概要を示し (§2)，言語理解すなわち問題テキストを同等な ZF の式へ翻訳するための形式的枠組みについて詳述する (§3)．ZF の式を RCF$^+$+PA の式へ変換し，推論器 (ソルバー) を適用して答えを得る部分については別稿 [岩根 13] で詳説する．次に，実際の入試問題を用いた実験結果について報告し (§4)，公開の準備を進めている数学問題コーパスを紹介する (§5)．

2. 解答システムの概要

図 1 に，解答システム全体の処理の流れを示す．以下では，次の問題を例として，各ステップの処理について概説する:

> $a > 0$，$b \leq 0$，$0 < p < 1$ とし，関数 $y = ax - bx^2$ のグラフは点 $P(p, p^2)$ を通るとする．このとき，b を a と p を用いて表せ．

システムへの入力は，数式部分に注釈づけを施した問題テキストである．煩雑なため実例は示さないが，注釈はおおむね下記の LaTeX 記法のように，記号の配置を指定するものである:

> `$a>0$`，`$b≤0$`，`$0<p<1$`とし，関数`$y=ax-bx^2$`のグラフは点`$P(p，p^2)$`を通るとする．このとき，`$b$`を`$a$`と`$p$`を用いて表せ．

最初のステップである数式解析は，数式の問題文中での意味を予測する処理である．例えば，上の例の`$y=ax-bx^2$`という数式には,「a, b, x, y の間の関係を主張する命題」という解釈と,「x から $ax - bx^2$ への関数」という解釈の 2 通りが少なくとも考えられる．数式解析では可能な解釈を少数に絞り込み，後続の構文解析処理で用いる辞書項目として生成する:

`$y=ax-bx^2$` := $\mathsf{S} : y = ax - bx^2$

`$y=ax-bx^2$` := $\mathsf{NP} : \Lambda x.ax - bx^2$

辞書項目は表層形 w に対し，その統語カテゴリ c と意味表現 s を $w := c : s$ という形で指定する．上の 2 つ目の辞書項目の中の Λ は通常のラムダ抽象と同じ意味だが，意味表現を合成するためのラムダ抽象 (λ で表す) と区別する．これらの辞書項目から，問題文における統語・意味的な文脈と整合的なものを選択する処理は，構文解析以降の処理の中で行なわれる．

構文解析処理では組合せ範疇文法 (CCG)[Steedman 01, 戸次 10] を用いて問題中の各文を解析する．CCG による解析では，文中の各単語に辞書項目を割り当て，句どうしを階層的に組み合わせて文全体の意味を導出する．図 2 に,「C_1 と C_2 の中心が一致するとき」という句に対する解析結果 (導出木) の一部を示す．句と句を組み合わせる規則 (図中の $>$，$<$ など) にはいくつかの種類があり，各規則は，親の句の統語カテゴリと意味表現を子の句から合成する方法を定める．

上の例題については，問題中の 2 文に対し，それぞれ以下の D_1，D_2 という意味表現が得られる:

$$\begin{aligned} D_1 &= a > 0; b \leq 0; 0 < p < 1; P = (p, p^2); \mathrm{on}(P, \Lambda x.ax - bx^2) \\ D_2 &= \mathrm{Find}(b') \left[cc; \exists^{-1} a; \exists^{-1} p; b = b'\right] \end{aligned}$$

この意味表現は，次節で説明するように，談話表示構造 (DRS)[Kamp 93] を拡張した iDRS という形式で記述されている．談話構造解析は，上記のような各文に対する iDRS を組み合わせ，問題全体を表す iDRS を構成する処理である．この処理では，問題文に含まれる「その点」などの照応表現や空の照応詞 (この例題はいずれも含まない) の先行詞を同定し (照応解析)，さらに，文間の論理的関係を同定することで問題全体の論理構造を定める．この例題の場合は，D_1 と D_2 の関係は単なる累加であり，累加演算 (;) によって D_1 と D_2 を接合した構造 $D_1; D_2$ が問題全体の意味表現となる．

システムの言語処理部分の最後のステップは，文全体を表す iDRS に対して次節で述べる変換を適用し，問題文と同等な ZF の項 X を「求めよ」という要求を表す表現 (以下，指令) を得る処理である．例題の場合は，下記の指令が得られる：

$$\mathrm{Find}(b') \left[\begin{array}{l} a > 0 \wedge b \leq 0 \wedge 0 < p \wedge p < 1 \wedge \\ \exists P. \left(\begin{array}{l} P = (p, p^2) \wedge \\ \mathrm{on}(P, \Lambda x.ax - bx^2) \wedge \ b = b' \end{array} \right) \end{array} \right]$$

ここで，$\mathrm{Find}(v)[\phi(v)]$ という指令は，変数 v を自由変数として含む論理式 $\phi(v)$ が定める v の値の範囲 (すなわち $X = \{v|\phi(v)\}$) を，限量記号を含まない RCF$^+$-PA の式として求める操作を意味する．また，その特別な場合として，与えられた命題 ϕ を証明する操作を $\mathrm{Show}[\phi]$ で表す．

上の例のように，言語処理の結果として直接得られる指令には，ZF を（非本質的に）拡大し，自然言語での表現に近づけるための要素，例えば “on” などの述語や関数記号，さらに，$\Lambda x.ax - bx^2$ のような高階の表現が含まれる．このような，(保存拡大された)ZF の式を，RCF$^+$+PA の等価な式へと変換する処理が，ここまでの言語理解処理と，QE アルゴリズムや定理証明器などのソルバーをつなぐ鍵となる．

そのような変換は，当然，どのような問題に対しても可能な訳ではない．しかし，指令が含む ZF の式 ϕ と等価で，かつ「逐語訳」の関係にある元の問題を非専門家が読んだときに，例えば，それが実数，あるいは，2 次元/3 次元/... のユークリッド空間の問題だと認識しうるのであれば，ϕ を機械的に RCF の命題 ϕ' に書き換える手続きが存在することが期待できる．この書換えの手段には，問題テキストから (ϕ を経由せず) 直接 ϕ' を予測するような方法も含め，さまざまな手段が考えられる．別稿 [岩根 13] では，ZF の式からの等価な書換えにより RCF$^+$+PA の式へ変換する現在の実装，および，実

$$
\dfrac{\dfrac{\dfrac{C_1 \text{ と } C_2 \text{ の中心が}}{\mathsf{S/(S\backslash NP_{ga})} : \lambda P.(\exists x; \exists x_1; x_2; x = [x_1, x_2]; x_1 = \mathrm{center\_of}(C_1); x_2 = \mathrm{center\_of}(C_2); Px)} \quad \dfrac{\text{一致する}}{\mathsf{S\backslash NP_{ga}} : \lambda x.\mathrm{coincide}(x)}}{\mathsf{S} : (\exists x; \exists x_1; \exists x_2; x = [x_1, x_2]; x_1 = \mathrm{center\_of}(C_1); x_2 = \mathrm{center\_of}(C_2); \mathrm{coincide}(x))}> \quad \dfrac{\text{とき}}{\mathsf{S/S\backslash S} : \lambda P.\lambda Q.P \to Q}}{\mathsf{S/S} : \lambda Q.((\exists x; \exists x_1; \exists x_2; x = [x_1, x_2]; x_1 = \mathrm{center\_of}(C_1); x_2 = \mathrm{center\_of}(C_2); \mathrm{coincide}(x)) \to Q)}<
$$

図 2: CCG 導出木

際に数式処理アルゴリズムを適用し答えを得る部分について詳述する.

3. 問題文から意味表示へ

ここまで述べたように，機械によって「数学の問題を解く」ためには，問題テキストと同等な ZF の式と RCF$^+$+PA の式という 2 つの中間表現を得るための手法が鍵となる．本節では，その第一段階の基礎となる，問題テキストと同等な ZF の式 (意味表示) を得るための形式的な枠組みについて述べる．

§2 で概説したように，問題全体の意味表示は，構文解析によって文ごとの意味表示を得た後，それらを文間の論理的関係に従って合成することで得られる．談話表示理論 (DRT)[Kamp 93] は，そのような複数文からなる談話の意味を DRS という表示言語で捉える枠組みである．数学テキストの解析では一文を超える論理的関係の正確な記述・認識が必須であり，DRT に基づく数学テキストの分析にはいくつかの先行例がある (例えば [Cramer 10])．しかし，本研究で目指すような広範囲の数学問題の自動的な意味解析を実現した例は無い．

我々は実際の試験問題を分析する過程で，数学問題においては，上記のような既往研究では考慮されていない，いくつかの言語的要素が頻出することに気付いた:

変数・数式による特殊な参照　「$a+b=3$ とする．ab の最大値を求めよ.」という談話において，2 文目に現れる「ab」は，「ab」の値を指すのではない．この例の「ab」は，先行文脈によって定まる，ab が取り得る値全体の集合を指すというのはひとつの自然な解釈だろう．このような，文脈に直接現れない対象を指示する変数・式の用法をどう分析するか?

限量しないことの指定　様々な限量の表現に加え，数学問題には「限量しない」（自由変数とみなす）ことを指定する表現がしばしば現れる．例えば,「$a+b=3$ とする．b を a で表せ.」に対応する指令は $\mathrm{Find}(b)[a+b=3]$ であって，論理式部分 $(a+b=3)$ では a は自由変数であるべきである．対照的に,「$a>0$ かつ $a+b=3$ とする．b の範囲を求めよ.」は指令としては $\mathrm{Find}(b)[\exists a(a>0 \wedge a+b=3)]$ と書け，ここでは a は束縛されている．この違いをどう捉えるか?

命令文　「求めよ」,「示せ」など，数学問題は通常なんらかの命令文 (ないし疑問文) で終わる．これらの命令文の内容と，先行する文脈の内容をどう合成し，ひとつの指令を導出するか?

これらの要素に対応するために，Eijck と Kamp による DRS の合成の枠組み [Eijck 11] を以下のように拡張した．まず，意味表示の単位として，図 4 に示す iDRS という構造を導入する．図 3，図 4 を比較すると分かるように，iDRS は通常の DRS における条件にいくつかの要素を加えたものになっている．それらは，文脈への変数の導入を表す $\exists v$（単一の変数を持つ DRS $(\{v\}, \{\})$ に相当），累加演算 $I_1; I_2$,「限量しない」という指定を表す $\exists^{-1}v$，2 種類の指令オペレータ Find および Show，そして先行文脈全体を iDRS の形で保持する特殊な「変数」cc である．

これらの要素を用いると，例えば「最大値」の辞書項目は，

項　$t ::= v \mid f(t_1, \ldots, t_k) \mid \Lambda v.t \mid \Lambda v.D$

条件　$C ::= P(t_1, \ldots, t_k) \mid \neg D \mid D_1 \to D_2$

DRS　$D ::= (\{v_1, \ldots, v_k\}, \{C_1, \ldots, C_m\})$

図 3: DRS の定義

項　$t ::= v \mid f(t_1, \ldots, t_k) \mid \Lambda v.t \mid \Lambda v.I$

iDRS　$I ::= P(t_1, \ldots, t_k) \mid \neg I \mid I_1 \to I_2 \mid$
$\exists v \mid I; I \mid \exists^{-1} v \mid \mathrm{Find}(v)[I] \mid \mathrm{Show}[I] \mid cc$

図 4: iDRS の定義

$$
\begin{aligned}
\{\!\!\{ I_1; I_2 \}\!\!\}_c &:= \{\!\!\{ I_1 \}\!\!\}_c; \{\!\!\{ I_2 \}\!\!\}_{c;[\![I_1]\!]_c} \\
\{\!\!\{ \mathrm{Find}(v)[I] \}\!\!\}_c &:= \mathrm{Find}(v)\ [[\![I]\!]_c] \\
\{\!\!\{ \mathrm{Show}[I] \}\!\!\}_c &:= \mathrm{Show}\ [[\![I]\!]_c] \\
\{\!\!\{ I_1 \to I_2 \}\!\!\}_c &:= \{\!\!\{ I_2 \}\!\!\}_{c;[\![I_1]\!]_c} \\
\{\!\!\{ I \}\!\!\}_c &:= \epsilon
\end{aligned}
\qquad
\begin{aligned}
[\![cc; I]\!]_c &:= c; [\![I]\!]_c \\
[\![cc]\!]_c &:= c \\
[\![I_1; I_2]\!]_c &:= [\![I_1]\!]_c; [\![I_2]\!]_{c;[\![I_1]\!]_c} \\
[\![I_1 \to I_2]\!]_c &:= [\![I_1]\!]_c \to [\![I_2]\!]_{c;[\![I_1]\!]_c} \\
[\![\mathrm{Find}(v)[I]]\!]_c &:= \exists v; [\![I]\!]_c \\
[\![\mathrm{Show}[I]]\!]_c &:= [\![I]\!]_c
\end{aligned}
$$

図 5: iDRS から指令列への変換 (一部)

左にノ格の名詞句 ($\mathsf{NP_{no}}$) を取って，普通名詞 (N) のように振る舞う統語カテゴリ（関係名詞）をもつ

「最大値」$:= \mathsf{N\backslash NP_{no}} : \lambda x.\lambda m.\max(\Lambda y.(cc; y = x), m),$

という形に書ける．意味表示に現れる $\max(S, m)$ という述語は，特性関数によって表された集合 S の最大の要素が m であることを表すとする．この max の定義の下で，上記の「最大値」の意味表示は，左のノ格名詞句が表す項 (x) を受け取り，先行文脈全体 (cc) の下で項 x が取り得る値 (y) の全体を表す集合 $\Lambda y.(cc; y = x)$ の最大値が m であること，を表す．

このような単語ごとの意味表示は一文の意味を表す iDRS へと合成され，次いで問題全体を表す iDRS へと合成される．問題全体を表す iDRS は，さらに以下の手続きを経て，(保存拡大された)ZF の式 ϕ を含む指令 ($\mathrm{Find}(v)[\phi]$ および $\mathrm{Show}[\phi]$) の列へと変換される:

1. 自由変数の限量: iDRS 中の全ての自由変数を最もスコープが狭くなるように限量する．
2. 命令内容の取り出し: 図 5 にその定義 (一部のみ) を示す関数 $\{\!\!\{\cdot\}\!\!\}_c$ および $[\![\cdot]\!]_c$ によって，問題全体を表す iDRS からその命令内容を取り出す．$\{\!\!\{I\}\!\!\}_c$ と $[\![I]\!]_c$ は，それぞれ，文脈 c において iDRS I が表す意味内容のうち，命令内容に対応する指令列と平叙文の内容に対応する iDRS を取り出す．
3. 限量の打消し: 取り出された指令列に含まれる各指令 $\mathrm{Find}(v)[I]$ ないし $\mathrm{Show}[I]$ の iDRS I に $\exists^{-1}v$ が含まれるときは，I 中でそれより前方のすべての限量記号 $\exists v$ を消去し，$\exists^{-1}v$ 自体も消去する．
4. DRS への変換: ここまでの定義から，指令が含む iDRS は，累加演算を除き，DRS 条件に対応する要素か，DRS

$(\{v\}, \{\})$ に対応する $\exists v$ のみを含む．よって，条件に対応する要素 I を DRS $(\{\}, \{I\})$ へ，累加演算を併合演算 $(V_1, C_1); (V_2, C_2) := (V_1 \cup V_2, C_1 \cup C_2)$ へ読み替えることで，iDRS から DRS へと変換できる．

5. 最後に，通常の DRS と述語論理との対応 [Eijck 11] に従い，指令が含む DRS を高階の述語論理式へと変換する．

4. 入試問題を用いた実験結果

本節では実際の入試問題を用いた実験について報告する．この実験の目的は，§1 で示した方針，すなわち自然言語による問題記述を ZF の式として写し取った後，機械的な推論が現実的に可能な表現 (ここでは RCF の式) へと変換することで答えを得る，という流れの現実的な有効性を確かめることである．

テストデータとして，5 つの国立大学，各 6 年分の入試数学問題 (計 249 題) から，まず自然言語による記述をほぼ含まないような問題を除き，さらに，筆者のうち数式処理を専門とする 1 名が入力となる RCF の式を与えた場合に SyNRAC を用いて解けた問題を全て選び出し，結果として 32 題を得た．

実験設定として，上記の 32 題を，1 文ごとに，かつ逐語訳的に人手で iDRS へ翻訳し，問題全体に対する iDRS へと合成したもの（入力 A），および，入力 A の形式で SyNRAC によって完全に解けた問題のうちの 14 題に対し，人手で CCG 導出木を与え，単語ごとの意味表示から導出した iDRS（入力 B） の 2 種類の入力を用いた．これらの iDRS から ZF(の保存拡大) の式を含む指令列を導き (§4)，さらに，RCF の式へと書き換え，SyNRAC で解くまでの処理 [岩根 13] は自動的に行った．よって，ここでの結果は自然言語処理の過程での誤りを除いた場合のシステム性能の上限とみることができる．

結果として，入力 A の 32 題のうち，19 題に対し完全な正答を得た．詳細な分析のために 32 題を部分問題 78 問へ分割して評価した場合，このうち 56 問 (72%) に対して正答を得た．正答が得られなかった部分問題 22 問に対する原因の内訳は，(i) RCF-QE の time-out (最大 30 秒) によるものが 9 問，(ii) ZF の式に含まれる根号のために RCF の式へと変換できなかったものが 7 問，(iii) 自由変数と束縛変数の区別をする表層的な手掛かりが問題中にないために前節で述べた手続きでは正しい ZF の式が得られなかったものが 6 問であった．

これらの原因のうち，(i) は QE 実装の洗練，あるいは QE の計算コストがより小さい RCF 式へと変換する工夫によって解決すべき課題である．(ii) は ZF の式を RCF の式へと変換するステップでの失敗であるが，根号については機械的にそれを除去する方法があるため，現在の実装における変換手続きの延長としてカバーできる．(iii) は言語分析の問題であり，§3 で述べた枠組みを改善することで解決すべき課題である．

入力 B の CCG 導出木から得た iDRS に対しては，14 題のうち 13 題について完全な正答を得た．14 題に含まれる部分問題 33 問に対しては，うち 32 問について正答を得た．部分問題 1 問の失敗は RCF-QE の time-out によるものである．CCG 導出木に従って単語の意味表示から組み立てた ZF の式は人間の目にはかなり繁雑な式となることが多いが，この結果は，そのような繁雑さによる影響は RCF の式への変換，ソルバーでの求解のいずれに対しても小さいことを示唆している．

5. 研究資源

研究用途に対する公開に向け，大学入試センター試験 数学 IA および数学 IIB の 14 年分の本試験・追試験問題を注釈付きコーパスとして整備している．注釈としては，大問・小問といった文書構造に加え，数式部分に対する MathML markup，さらに，形態素区切りと文節間の係り受け情報を付与している．また、12 年分の本試験問題については，英訳をともに公開する予定である．様々なアプローチによる数学問題解答システムに関する研究，また，数式解析や構文解析といった部分問題の研究の資源として活用されることを期待している．

6. 結論と展望

本稿では,「機械によって数学問題を解く」という行為を (1) 問題文の ZF の式への翻訳，(2) ZF の式から現実的に機械による推論が可能な体系 (RCF^{+}+PA) の式への変換，(3) 「解けている」と認められる項・式を得るための推論の実行，という流れとして明確化し，これに基づく解答システムについて (1) の自然言語処理のステップを中心に概説した．言語処理部について理想化した実験設定ではあるが，RCF-QE ソルバーを用いてテストデータの約 7 割に現段階で正答が得られたことは，上記の方針について明るい見通しを与えるものである．今後の課題としては，まず (1) の言語処理部の自動化・高精度化がある．(2) については,「どのような問題であれば，どのような理由で」RCF^{+}+PA の式へと変換できるのか，ということの明確化，さらに，推論のコストを下げるための変換方法の洗練が課題として挙げられる．最後に，(3) についての展望として，RCF-QE ソルバー実装の洗練，および (弱い)PA を必要とする問題に対するソルバーの開発がある．

参考文献

[Collins 75] Collins, G. E.: Quantifier elimination for real closed fields by cylindrical algebraic decomposition, in *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern*, pp. 134–183 (1975)

[Cramer 10] Cramer, M., Fisseni, B., Koepke, P., Kühlwein, D., Schröder, B., and Veldman, J.: The naproche project controlled natural language proof checking of mathematical texts, in *Proc. CNL'09*, pp. 170–186, Springer-Verlag (2010)

[Dries 88] Dries, van den L.: Alfred Tarski's Elimination Theory for Real Closed Fields, *J. Symb. Log.*, Vol. 53, No. 1, pp. 7–19 (1988)

[Eijck 11] Eijck, van J. and Kamp, H.: Discourse Representation in Context, in Benthem, van J. and Meulen, ter A. eds., *Handbook of Logic and Language, Second Edition*, pp. 181–252, Elsevier (2011)

[Iwane 13] Iwane, H., Yanami, H., Anai, H., and Yokoyama, K.: An effective implementation of symbolic-numeric cylindrical algebraic decomposition for quantifier elimination, *Theoretical Computer Science* (2013), (in press)

[Kamp 93] Kamp, H. and Reyle, U.: *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, Kluwer Academic (1993)

[Robinson 50] Robinson, R. M.: An Essentially Undecidable Axiom System, in *Proc. of the International Congress of Mathematics*, pp. 729–730 (1950)

[Steedman 01] Steedman, M.: *The Syntactic Process*, Bradford Books, Mit Press (2001)

[Tarski 51] Tarski, A.: *A Decision Method for Elementary Algebra and Geometry*, University of California Press (1951)

[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 (第 27 回) 予稿集 (2013), (to appear)

[戸次 10] 戸次大介 : 日本語文法の形式理論: 活用体系・統語構造・意味合成, 日本語研究叢書, くろしお出版 (2010)