

デジタル・アシスタントへの自然言語による入力の解釈結果をユーザがすばやく正確に確認するための情報提示技術に関する研究

代表研究者 松崎 拓也 名古屋大学 大学院工学研究科 准教授

1 はじめに

スマートフォンなど情報通信端末の小型化にともなって、音声インタフェースを備えたデジタル・アシスタントの重要性が高まっている。Apple の Siri や Microsoft の Cortana などを代表とするデジタル・アシスタントは実用化されて久しい。また、近年はスマートウォッチや AI スピーカーなど、音声による自然言語入力を基本インタフェースとした情報機器が普及しつつある。しかし、その主な利用方法は、「予定を確認する」「音楽を再生する」「指定した相手にメッセージを送信する」「Google で検索する」など比較的シンプルな機能の実行にとどまっている。

現在、言語理解を含む人工知能技術は急激に進展しており、IoT など電気通信を通じた実世界との接合を通じて、更なる発展が見込まれている。今後のデジタル・アシスタントは「出張先と日程および予算を指定して、航空券・鉄道およびホテルの予約をまとめて行う」「会議のメンバーと日程候補を指定してスケジュール調整のメールを一斉送信し、返信をまとめ、全員の都合が合う時間に会議室を予約する」など、実世界と結びついた、より高度な情報処理のための汎用インタフェースとして一層活用されることが期待される。

人工知能アシスタントを通じて、金銭の支払い・施設の予約など、誤りが許されないタスクを含む高度な応用を実行するためには、処理の実行に先だって、システムがユーザの意図を正しく理解したことをユーザ自身が確認する必要がある。これまでのシンプルなデジタル・アシスタント機能のためには、音声認識の結果を文字列として提示するだけで十分だった。しかし、上記のような複雑な指示に対しては、単に音声認識結果をテキストとして提示するだけでは、システムがそれをどう解釈したかが不明であり、望みの処理が確実に実行されるかどうか、ユーザには分からない。

本研究では、複雑な処理を指示する自然言語入力に対し、システムの解釈結果をグラフィックスやテキストを適切に組み合わせて提示することで、ユーザが自らの指示が正しく解釈されたことを素早く、正確に確認するための情報提示技術の研究開発を行う。AI スピーカーなど、ビジュアルインタフェースを持たない機器の場合はさらに音声のみを通じて解釈結果を伝えるという課題が加わるが、この場合も、音声化に先立ちシステムによるユーザ入力の解釈結果を適切に構造化する必要がある点は共通の課題である。

代表研究者は近年、自然言語解析技術と自動推論技術を接合した数学問題自動解答システムの開発に注力し、一定の成功を収めてきた。この自動解答システム開発の過程で明らかになった課題の一つは、人工知能システムがテキストの言語理解に失敗し、誤った解釈結果を出力した場合、どこで、どのように誤りが生じたかを発見することが容易ではない、という問題であった。システムによる言語理解の結果は、計算機上での処理が可能な形式的表現（論理式）として出力される。機械的な言語理解の結果は巨大な論理式となるのが普通であり、それを解読して、解釈のどの部分が誤りの原因になったのかを見つけることは人間にとって容易でない。

この問題は、数学問題に対する自動解答システムだけでなく、人工知能アシスタントなど、自然言語による入力を受け付け、高度な知的処理を行うシステム一般に生ずる問題である。人と人工知能システムの共同作業によって高度な知的処理を確実に、かつ効率よく行うためには、システムによる言語解釈の結果を人が理解しやすいかたちで表示し、実際に処理を行う前に、指令が正しく解釈されたことを簡単に確認できる必要がある。

本研究では、ユーザからの複雑な指示を模擬する入力として、数学問題テキストを利用する。数学問題テキストは AND、OR や場合分け、否定、量化など、多様な条件の表現を含み、かつ、既存の大量データが利用できる点で好適な研究課題である。システムによる問題の解釈をユーザが容易に確認するための基礎技術として、①システムによる解釈結果を表す形式的表現（論理式）の単純化、②図表を含んだ答案テキストの自動生成、③演繹の過程が人にとって理解しやすい発見的な規則に基づく演繹システムの開発、の3つのテーマに関して研究を行った。

2 ユーザ入力の解釈を表す論理式の単純化

2-1 背景と目的

一般的に言語の意味解釈において最も頻繁にあらわれる論理結合子は AND であり、述語を AND のみで結合した意味内容を表現するためには単に述語の集合を用いればよい。しかし、場合分けや OR、否定や量化を含む複雑な情報処理要求を正確に表現するためには、少なくとも 1 階述語論理と同等の表現力をもつ形式体系が必要となる。さらに、プログラムコードやデータベース問い合わせ (SQL) など、人が書き下した形式表現と異なり、自然言語入力を機械的に処理することで得た論理式は、非常に長大かつ冗長なものとなるのが普通である (図 1)。このため、言語処理結果である論理式をそのままユーザに提示しても、それをもとにシステムの解釈が意図した通りであるか否かを判断することは困難である。このため本研究ではまず機械処理の結果得られた論理式を変換し、意味内容を変えずにより簡単な表示を得る方法についての研究を行った。



```
(Find (__gen775)
(exists (L_1 L_2 P Q R)
(&& (all (PLam __gen959)
(forall (__gen958)
(-> (2d.on __gen958 __gen959)
(2d.on __gen958 (2d.xy-plane))))))
(cons L_1 (cons L_2 (nil))))
(exists (__gen956)
(&& (exists (__gen95bz __gen95by __gen95bx)
(&& (= __gen956 (2d.line2d __gen95bx __gen95by __gen95bz))
(II (= __gen95by 1)
(&& (= __gen95by 0)
(= __gen95bx 1))))))
(exists (__gen957)
(&& (exists (__gen957z __gen957y __gen957x)
(&& (= __gen957 (2d.line2d __gen957x __gen957y __gen957z))
(II (= __gen957y 1)
(&& (= __gen957y 0)
(= __gen957x 1))))))
(= (cons L_1 (cons L_2 (nil))) (cons __gen956 (cons __gen957 (nil))))
(= (2d.graph-of (poly-fun (cons 0 (nil)))) (2d.x-axis))
(= L_1 (2d.graph-of (poly-fun (cons 0 (nil))))))
(= L_2 (2d.graph-of (poly-fun (cons 0 (cons (sqrt 3) (nil))))))
(= (cons L_1 (cons L_2 (nil)))
(cons (2d.graph-of (poly-fun (cons 0 (nil))))
(cons (2d.graph-of (poly-fun (cons 0 (cons (sqrt 3) (nil))))
(nil))))))
```

図 1: 言語処理によって得られる長大かつ冗長な論理式の例 (一部)

2-2 方法

一口に「意味表示の単純化」と言っても、背景知識を用いた演繹まで含めれば、意味を変えないまま論理式に対して行いうる変形のバリエーションは非常に幅広い。そのため、論理式の複雑さを表す指標 (例えば長さなど) に関する最適化問題として単純化処理を定式化したとしても、適当な公理系の下で入力と等価で、かつ複雑さの指標について最適な論理式を求めるような探索は、現実的には難しい。また、知識を用いた演繹によって意味表示が大幅に簡単になるケースは存在するものの、そのような、知識に基づく演繹を経た書き換えの結果が、ユーザにとってより分かりやすいかどうかを自動的に判別することは、論理式の複雑さの指標の観点からのみでは難しい。

本研究では、詳細な背景知識とほぼ無関係に (一般的な性質のみに基づき) 適用可能な変形操作をいくつかのグループに分け、グループ毎に、可能な変形をできる限り行うという手順で単純化アルゴリズムを構成した。変形操作のグループ分けや変形操作を行う順番を制御することで、結果として得られる論理式の複雑さを調整することができる。具体的な変形操作としては以下のようなものがある:

- ① 代入操作によって束縛変数の数を減らす変換:
 $\forall x(x=\eta \rightarrow \phi(x)) \Leftrightarrow \phi(\eta)$, $\exists x(x=\eta \wedge \phi(x)) \Leftrightarrow \phi(\eta)$
- ② ラムダ式に対する β 簡約: $(\lambda x. \phi(x))M \Leftrightarrow \phi(M)$
- ③ 幾何的な不変性に基づき、束縛変数を定数に置き換える操作
- ④ 定数のみからなる代数式に対する単純化
- ⑤ 代数演算における単位元・零元などの性質に基づく単純化:

- 例) $x + 0 \Rightarrow x$, $0 \cdot x \Rightarrow 0$, $1 \cdot x \Rightarrow x$, $x^1 \Rightarrow x$ など
- ⑥ 等号・不等号の基本的性質に基づく簡単化：
 例) $x = x \Leftrightarrow \text{True}$, $x \leq x \Leftrightarrow \text{True}$, $x < x \Leftrightarrow \text{False}$
- ⑦ 論理演算における真偽値の性質に基づく簡単化：
 例) $\text{True} \wedge \phi \Leftrightarrow \phi$, $\text{False} \wedge \phi \Leftrightarrow \text{False}$, $\text{True} \vee \phi \Leftrightarrow \text{True}$, $\text{False} \vee \phi \Leftrightarrow \phi$
- ⑧ リスト間の等価性を要素間の等価性で表す変換：
 $[x_1, x_2, x_3, \dots] = [y_1, y_2, y_3, \dots] \Leftrightarrow x_1 = y_1 \wedge x_2 = y_2 \wedge x_3 = y_3 \wedge \dots$

ある変換の結果、他の変換が初めて適用可能になるケースが多数存在するため、これらの変換はくり返し試みられ、いずれの変換も適用不可能になった時点で簡単化処理は終了する。

自然言語から機械的に生成された論理式では、2つのフレーズに対する意味表現の論理的な関連付けが、論理結合子のみでなく束縛変数を介して行われることが多い。このため、図1に示すように、論理式には、解釈の過程で機械的に生成された束縛変数(図において__genXYZのような名前が付けられた変数)が多数含まれる。これらの束縛変数は生成された論理式が冗長になる大きな原因となっており、上記の規則①はこのような束縛変数を消去し、より意味を把握しやすい論理表示を得る上で重要である。

図形の幾何学的配置やその性質に関するテキストを機械的に論理式に変換した場合、例えば「三角形があり...」のようなフレーズはおおむね「任意の3点を頂点とする三角形があり...」という最も一般的な状況に対応する論理表現に変換される。しかし、例えば図形の間相対的な位置関係のみが問われる問題は非常に多く、そのような場合、登場するいくつかの点についてはその位置を特定の点、例えば原点に固定して翻訳しても元のテキストで述べられた論理的内容は損なわれない[1, 2]。このような場合を検出し、もとの論理式と等価でかつより変数の数が少ない論理式を得る処理が上記の③である(図2)。現在のシステムでは、論理式で表現された内容が平行移動および回転変換で不変となる場合を検出し、いくつかの変数値を固定する処理を行っている(図2の最初の矢印)。これをさらに進め、より特殊な場合についてのみ考慮しても問題がないような(一般性が失われない)状況(例えば「三角形の三本の中線は一点で交わる」という命題を証明する場合の図2の2番目の矢印に対応する変換)を検出し、より多くの変数を削除することも考えられるが、このような変換はユーザが解釈を確認する目的に対しては必ずしも貢献しない。これは問題を自動的に解くことが目的である場合の論理式の簡単化と、ユーザによる解釈内容の確認を目的とする本研究における論理式の変換との大きな違いである。

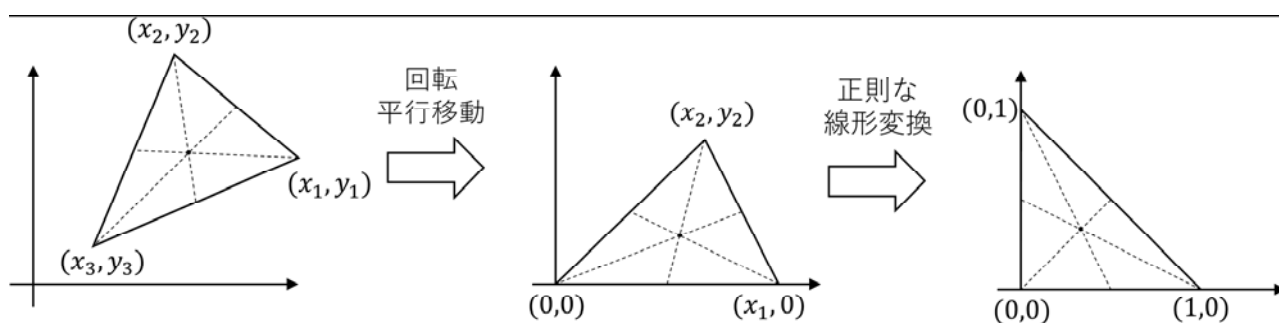


図 2 : 幾何的不変性を利用した論理式の簡単化

3 図表を含んだ答案テキストの自動生成

3-1 背景と目的

自然言語を用いて述べられた状況や要求を解析した結果をユーザに対してグラフィカルに提示し、システムによる解釈をユーザがすばやく確認することを可能にする技術は、ユーザの要求が複雑になるにつれてより重要となる。言葉で定義された図形を適切な説明とともに図として表示するタイプの数学問題は、ユーザ要求の解釈結果をグラフィカルに提示する課題のうちもっともシンプルな場合とみなせる。本研究ではその

ような背景のもと、数学問題に対し図示を含む答案テキストを自動的に生成する技術の開発を行った。

3-2 方法

図3に、以下の問題に対してシステムが生成した答案の例を示す。

原点を中心とする半径1の円板をC、(1, 0)を中心とする半径1の円板をDとする。
 さらに、CとDの共通部分をEとする。
 (1) Eを図示せよ。
 (2) Eの面積を求めよ。

言語解釈の結果に基づき、最終的に論理式で定義される図形を、言葉による説明と共に適切に表示するためには、以下のことを考慮する必要がある：

- ① 表示する図形をどのように説明するか
- ② 対象となる図形のうちのどの部分を表示するか

まず、①の言葉による図形の説明は、おうむがえしではない方法、すなわち、ユーザ入力そのものとは異なった方法で行う必要がある。これはシステムによる入力の解釈が正しいかどうかをユーザがチェックするという目的から当然である。一つの方法は、言語解釈の結果として得られる、図形を定める論理式を可能な限り簡単化したのちにユーザに提示することである。しかし、この論理式には一般に量子子などが含まれ、すばやく解釈することは必ずしも容易でない。そこで、本研究では、図形を定める論理式を前節で述べた方法で簡単化したものに加え、さらに量子子消去アルゴリズム[3]を適用することで得られる、方程式および不等式のブール結合として図形を表した式を表示する方式を検討した(図3の答案の最初の式および2番目の式。ただしこの問題では最初に提示した式は既に量子子を含まない形になっている)。

次に、②の、図形のどの範囲を表示するかを決定する問題について説明する。これを決定するには、まず図形がどの方向に有界かを自動的に調べる必要がある。例えば図3で示した図形はx軸の正負・y軸の正負の4方向全てについて有界である。しかし、例えば表示する図形が $y = x^2$ のグラフのような場合、図形はy軸の負の方向についてのみ有界(すなわちグラフ上の点のy座標は全て0以上)で、残りの3方向については有界でない。現在のシステムでは、図形が有界である方向については、数式処理によって求めたその上限あるいは下限を用いて表示範囲を決定し、有界でない方向についてはあらかじめ定めた最大表示範囲を用いて表示範囲を決定している。有界でない場合の表示範囲の決定については今後より詳細な検討が必要である。

4 人にとって理解しやすい発見的な規則に基づく演繹システム

4-1 背景と目的

デジタル・アシスタントによる情報処理をさらに高度なものとするために、ユーザの要求にもとづき演繹を行い、その結果に従って種々の情報サービスを統合的に用いる仕組みは有効だと考えられる。この場合、演繹の結果をユーザに提示し、その演繹の過程および結論がユーザの意図に合致したものであることを確認

描く領域に属する点の座標を (x, y) と置くと、問いの条件は次の一階論理式と同値になる：

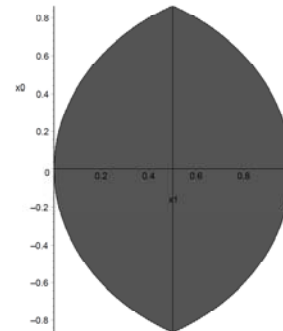
$$(y^2 + x^2 \leq 1 \wedge y^2 + (1-x)^2 \leq 1).$$

これは以下と同値である：

$$((x = 0 \wedge y = 0) \vee (0 < x \wedge x \leq \frac{1}{2} \wedge -\sqrt{2x-x^2} \leq y \wedge y \leq \sqrt{2x-x^2})) \vee (\frac{1}{2} < x \wedge x < 1 \wedge -\sqrt{1-x^2} \leq y \wedge y \leq \sqrt{1-x^2}) \vee (x = 1 \wedge y = 0).$$

これを図示すれば図1のようになる：

図1:



そのつぎに求める実数を

$$x_{gen4}$$

と置くと、問いの条件は次の一階論理式と同値になる：

$$x_{gen4} = \frac{1}{6} (4\pi - 3\sqrt{3}).$$

よって答は以下の通り：

$$x_{gen4} = \frac{1}{6} (4\pi - 3\sqrt{3}).$$

図3：グラフィクスを含む答案の例

図3の答案の最初の式および2番目の式。ただしこの問題では最初に提示した式は既に量子子を含まない形になっている。

する必要がある。しかし、現在主流の自動演繹の技術は例えば論理記号の導入・消去、あるいは融合原理による節の導出のように、非常にプリミティブな操作を組み合わせる推論を実現している。そのため、これらの方法による演繹の過程は（例えば通常の数学の証明にみられるような）人間による推論過程の説明と比べると、個々のステップは単純だがそれらの長大な系列となっており、非常に可読性が低い。

さらに、初等幾何のように原理的に証明アルゴリズムが存在する一部の領域を除くと、実用上重要でかつ完全な証明体系が存在する領域はほとんど存在しない[4]。数学においては整数問題がこのように一般的な解法が存在しない領域の典型である。このような領域において、機械による演繹の過程をユーザが理解するための1つの方法は、演繹によって処理可能な入力は頻出タイプのものに限定する代わりに、ひとつひとつの推論ステップを人が理解しやすいものとし、理解可能な手順の積み重ねとして演繹過程を提示することであろう。本研究では、このような考察のもと、完全な一般的解法が存在しない整数問題をターゲットとして、論理的な等価性は厳密に保つが発見的な論理式の書き換えを行う規則を集めた演繹システムを開発し、現実に行えると期待されるレベルの演繹課題であるセンター試験の整数問題に対してどの程度有効かを調査した。

4-2 方法

まずセンター試験およびセンター試験形式模試の過去問から整数に関する問題を収集し、システム開発時に参照する開発用データとシステムの性能検証に用いるテスト用データに分割した。次に開発用データを分析し、解答を導出する過程に現れる推論ステップを観察し、いわゆる「解法」に近い高レベル（しかし適用範囲は限定的）な書き換え規則と、頻出する低レベルな規則とともに収集した。ここでいう書き換え規則とは、問題を表現する論理式に対し、それと等価でより単純な形になるよう変形を行うもので、一般的な形式的証明体系における推論規則が前提から帰結へ一方の含意のみ成立するよう定式化されることが多いのに比べると、より限定的な形式になる。このような形式を選択した理由は、人にとって理解しやすい演繹過程の説明を生成することを念頭に置いていることと共に、単に証明を行うのみでなく、問題の答え（未知数）を演繹を通じて得ることが目的であるためである。図4に多元一次不定方程式の問題から収集した書き換え規則の例を示す。番号1～4は比較的高レベルの書き換え規則、番号7～13は低レベルな書き換え規則の例となっている。

番号	変形前 ^{*1}	変形後 ^{*1}
1 ^{*2}	$\sum_{j=1}^n i_j x_j = i$	$\exists k_1, \dots, k_{n-1}$ $\left(\bigwedge_j x_j = \sum_{l=1}^{n-1} a_l k_l + b_j \right)$ (不定方程式をパラメータ表示する)
2 ^{*3}	$\forall x \exists y \left(\bigwedge_k i_{k1} x + i_{k2} = a_{k1} y + a_{k2} \right)$	$\bigvee_d \left(\bigwedge_k a_{k1} = \frac{i_{k1}}{d} \alpha + i_{k2} \wedge \exists \alpha \left(\bigwedge_k a_{k2} = \frac{i_{k1}}{d} \alpha + i_{k2} \right) \right)$ (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}	$a i$	$\bigvee_j a = j$ j は i の因数
5 ^{*5}	$\forall \mathbf{x} (\mathbf{x} = \mathbf{a} \rightarrow \phi(\mathbf{x}))$	$\phi(\mathbf{a})$
6	$N \bmod \alpha = r \rightarrow \phi(N \operatorname{div} \alpha)$	$\forall q (N = \alpha q + r \rightarrow \phi(q))$
7	$\exists x \phi(x) \rightarrow \varphi$	$\forall x (\phi(x) \rightarrow \varphi)$
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	$(\phi_1 \wedge \phi_2) \rightarrow \phi_3$	$\phi_1 \rightarrow (\phi_2 \rightarrow \phi_3)$
11	$(\phi_1 \vee \phi_2) \rightarrow \phi_3$	$(\phi_1 \rightarrow \phi_3) \wedge (\phi_2 \rightarrow \phi_3)$
12	$\phi_1 \wedge \phi_2$	$\phi_2 \wedge \phi_1$
13	$\phi_1 \vee \phi_2$	$\phi_2 \vee \phi_1$

図 4：不定方程式問題のための論理式書き換え規則

次に収取した書き換え規則を用いる推論器を構築した。これは、問題の直訳である論理式を入力とし、可能な書き換えを幅優先で探索していくものである。この過程で、完全な証明アルゴリズムが存在するプレスバーガー算術の式が得られた場合は既存の証明器 Princess [5]をサブ推論器として用いて量子子を除去する。書き換え規則あるいは Princess での処理を通じて全ての量子子が除去できた場合、残った条件から未知数を求めることは通常容易である。探索ステップ数があらかじめ定めた上限を超えた場合は、失敗を報告して推論を終了する。

4-3 結果

構築した推論システムをセンター試験形式模試から収集したテスト用問題 10 題に対して適用した結果を表 1 に示す。比較のため、数式処理システム Mathematica の Reduce コマンドを用いた場合の結果を共に示す。この結果から、システムの得点率はまだ 3 割程度と限定的ではあるが、Mathematica の Reduce コマンドに比べおよそ 3 倍の数の問題が解けていることが分かる。

表 1：発見的な規則を用いた推論による得点率

-	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

4-4 低レベルな書き換えの抑制による推論の効率化

構築した推論システムで問題を解く際に探索の過程で実行されている書き換え規則のタイプと実行数を調べると、乗法の可換性や等式の対称性を表す低レベルな規則が非常に多数実行されていることが分かった(図 5)。システムでは高レベルの規則と低レベルの規則が区別されているため、推論過程をユーザに対して提示する目的では、高レベルの規則の適用ステップのみを示すことで、演繹過程を可読な形で提示することは可能である。しかし、演繹自体の効率化を考えた場合、多数実行されている低レベルの規則の適用を抑制することが効果的だと考えられる。そこで、論理式中の部分式に対する正規形を定義し、さらに書き換え規則の適用時に低レベルな規則の実行に相当する処理を同時に行うことで、低レベルな規則の適用を抑制することによる探索空間の削減を試みた。

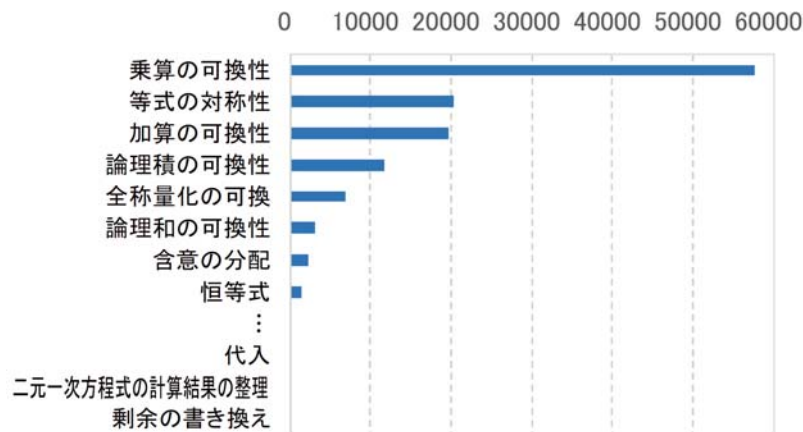


図 5：書き換え規則ごとの適用回数

図 6 に、改良した推論システム（新システム）と 4-2 節で述べた推論システム（旧システム）による、問題ごとの実行時間の比較を示す。およそ半数程度の問題で推論のための実行時間の大幅な減少が実現できたことが分かる。

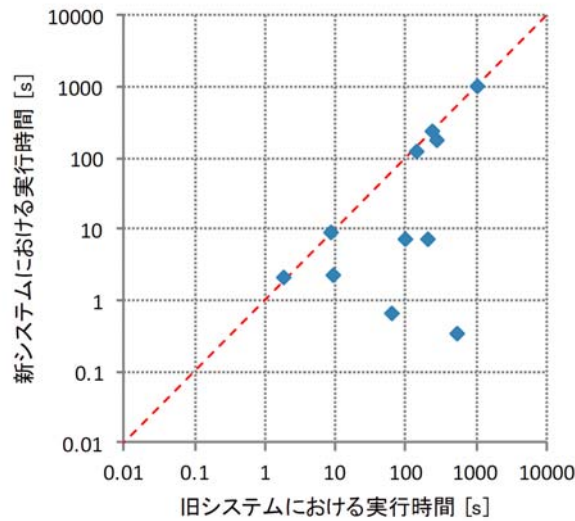


図 6：低レベル規則の抑制による効率化

5 まとめと今後の課題

本研究では、デジタル・アシスタントへの自然言語による処理要求の入力が、システムに正しく解釈されたことを、ユーザが素早くかつ正確に確認するための情報提示に向けた基礎技術の開発に取り組んだ。具体的には、① 自然言語入力の解釈を表す論理表現の簡単化、② ユーザ入力の解釈をグラフィカルに表示するための基本処理、③ システムによる演繹の過程を人にとって分かりやすいものとするための推論システムの構築、の3つについて、複雑なユーザ入力の例として自然言語で書かれた数学問題を具体的に取り上げて技術開発を行った。

本研究で行った解釈内容の簡単化・グラフィックスを交えた解釈内容の表示・システムによる演繹過程の可読化を統合したユーザインターフェースを開発・評価することが次の課題となる。また、メール作成と一斉送信・社内リソースの予約・オンラインショッピングといった種々の要素を含む、より現実的な課題の解決を要求するユーザ入力に対しても、本研究で開発した情報提示のための基礎技術を応用していくことが将来的な課題となる。

【参考文献】

- [1] Harrison, J., “Without Loss of Generality,” in Proceedings of the 22nd International Conference on Theorem Proving in Higher Order Logics, TPHOLs'09, pp. 43-59, Springer-Verlag, 2009.
- [2] Atkey, R., Johann, P., and Kennedy, A., “Abstraction and Invariance for Algebraically Indexed Types,” in Proceedings of the 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, pp.87-100, ACM, 2013.
- [3] 穴井, 横山. 「QE の計算アルゴリズムとその応用 — 数式処理による最適化」 東京大学出版会, 2011.
- [4] Tarski, A., “A decision method for elementary algebra and geometry,” University of California Press, 1951.
- [5] Rummer, P., “A constraint sequent calculus for first-order logic with linear integer arithmetic,” LNCS, vol. 5330, pp. 274–289. Springer, 2008.

〈 発 表 資 料 〉

題 名	掲載誌・学会名等	発表年月
センター試験「数学」整数問題解答システムの開発	人工知能学会全国大会（第31回）	2017年5月
Semantic Parsing of Pre-university Math Problems	Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL-2017), pp. 2131-2141	2017年7月
Can an A.I. win a medal in the mathematical olympiad? – Benchmarking mechanized mathematics on pre-university problems	AI Communications, 31(3), pp. 251-266	2018年5月
センター試験「数学」整数問題解答システムにおける探索空間の削減	人工知能学会全国大会（第32回）	2018年6月