擬似コードから考える 自然言語を活かしたプログラミング言語

高野 志歩[1,a)]　田村 みゆ[2]　富岡 真由[2]　秋信 有花[2]　倉光 君郎[1,b)]

概要：大学入試センターが公表した大学入学共通テスト（情報）のサンプル問題において擬似コードが用いられたことから，擬似コードによるプログラミング教育は，今後高校生の間に広がる可能性がある．擬似コードは，自然言語による理解しやすさもあり，プログラミング教育に導入しやすいという利点がある一方で，実際に動くコードを書くことなくプログラミング学習を済ましてしまうケースが増える懸念もある．本稿では，自然言語によるプログラムの理解のしやすさを活かし，かつ段階的にプログラミングを覚えることのできる新しいプログラミング言語処理系 Samoyed を提案する．Samoyed の試作を通して，プログラミングに自然言語を取り込む際の多くの技術課題が明らかになった．本発表では，Samoyed の構想と試作開発の状況を報告したい．

1. はじめに

社会活動全般のデジタル化が進み，中学校や高等学校でもプログラミングを学ぶ機会が拡大している．しかし，どのプログラミング言語を最初に教えるべきかという問いは，情報教育の古くて答えのないテーマである．同時に，教員側の準備や能力的な課題もあり，今後も容易に統一が取られるものではない．

中高生の学ぶプログラミング言語が統一されない中，公平性を保ちながらプログラミング能力や理解度を確かめるため，大学入試センター試験「情報関係基礎」では，大学入試センター独自の日本語表記の疑似言語 (DNCL[1]) が設計され，利用されてきた．今後，「情報」科目の導入が予定されている大学入学共通テストにおいても，DNCL の仕様に実用プログラミング言語の良いところを取り入れ，一部表記を改めた擬似コードが使用される見込みである．

このように，自然言語による擬似コードは，公平性を保ちながら，プログラミング能力や理解度を確かめることができる手段として，試験問題などで採用されている．また，自然言語に由来する理解しやすさもあり，プログラミング教育に導入しやすいという利点もある．しかし，少し懸念されるのが，受験対策として擬似コードの暗記に終始して，実際に動くコードを書くことなくプログラミング学習を済ますケースが増えることである．

本研究の動機は，中高生にプログラミングの楽しさを学んでもらうため，擬似コードを動作させて実行可能なプログラミング言語処理系の開発である．ただし，目標とするのは，単に DNCL 処理系ではない．自然言語によるプログラムの理解のしやすさを活かしながら，プログラミングの概念や動作原理の理解が得られるような処理系の創出である．

本稿では，Samoyed と名付けた，新しいプログラミング言語処理系の構想と試作開発の状況を報告する．Samoyed は，自然言語による擬似コードを Python などのプログラミング言語に変換し，実行できるようにする．ただし，先に述べたとおり，単に DNCL 風の擬似コードに対応させるのが目的ではない．むしろ，構文を新たに覚える必要がなく，自由に記述可能な言語処理系である．そして，形式的な構文ルールを少しずつ学び，より正確なプログラムの動作を可能にすることで，プログラミングを段階的に学べることを目指していきたい．

本稿の残りの構成は以下のとおりである．2 節では，現在教育現場で利用されている中高生向けの擬似コードを外観する．3 節では，Samoyed の開発目標を述べる．4 節では，Samoyed の試作と技術的なチャレンジを考える．5 節では，関連研究を外観する．6 節では，本稿を総括し，今後の展望をまとめる．

[1] 日本女子大学理学部数物科学科
Department of Mathematical and Physical Sciences, Japan Women's University, 2-8-1 Mejirodai, Bunkyo-ku, Tokyo 112–8681, Japan
[2] 日本女子大学大学院理学研究科数理・物性構造科学専攻
Graduate School of Science Division of Mathematical and Physical Sciences, Japan Women's University, 2-8-1 Mejirodai, Bunkyo-ku, Tokyo 112–8681, Japan
[a)] m1816053ts@ug.jwu.ac.jp
[b)] kuramitsuk@fc.jwu.ac.jp

```
Angoubun = ["p", "y", "e", "b", ...]
配列変数Hirabunを初期化する
hukugousuu = 26 -
i を 0から要素数(Angoubun)-1まで1ずつ増やしながら：
    bangou = 差分()
    もし bangou != -1 ならば：
        もし bangou + hukugousuu <= 25 ならば：
            Hirabun[i] = 文字(bangou + hukugousuu)
        そうでなければ:
            Hirabun[i] = 文字(bangou + hukugousuu - 26
    そうでなければ:
        Hirabun[i] =
表示する(Hirabun)
```

図 1 大学入学共通テスト (サンプル問題)

```
n を自然数とする.
開始時点において, A[1], ..., A[n] にそれぞれ
 1 以上 n 以下の異なる自然数が1つずつ格納されているものとする.

i を n から 2 まで 1 ずつ減らしながら,
  jを1からi-1まで1ずつ増やしながら,
    もし A[j]>A[i]ならば A[j]を 1 減らす
    を繰り返す
を繰り返す
```

図 2 第 8 回科学の甲子園全国大会 (第 11 問)

2. 高校生に広がる擬似コード

自然言語による擬似コードは，旧来からアルゴリズムを記述するときに用いられてきた．プログラミング言語を使用せず，自然言語で擬似的にプログラムの内容を記述することで，誰もが直感的に処理の内容を理解することができる．また，これまで学んできたプログラミング言語の種類を問わず，誰もが理解可能な形式であるため，高い公平性が求められる中等教育の現場での活用も見込まれる．

本節では，現在の教育現場における擬似コードの活用場面とサンプルコードを外観する．

2.1 大学入学共通テスト (情報)

2021 年，大学入試センターは，令和 7 年度大学入学共通テストから「情報」を含む 7 教科 21 科目に再編すると発表し，新設される「情報」のサンプル問題を公表した．このサンプル問題の印象的な特徴は，プログラミング能力を試す問題に，図 1 に示すような擬似コードが用いられた点にある．この擬似コードは，大学入試センターの「情報関係基礎」のために設計された独自の日本語表記の疑似言語 (以下，DNCL[1]) をベースにし，実用プログラミング言語の良いところを取り入れ，一部表記を改めたものになっている．高等学校の授業で何らかのプログラミング言語を用いて実習した生徒であれば容易に理解できるものである．さらには，実際のプログラミング言語で擬似コードを見ながらプログラムを記述することも容易にできる [2], [3].

2.2 科学の甲子園 (情報)

科学の甲子園は，国立研究開発法人科学技術振興機構 (JST) が創設し，高等学校等の生徒チームを対象として，理科・数学・情報における複数分野の競技を行う取り組みである．現在，県大会，全国大会と大規模に開催されており，高校生がチャレンジする機会も多い．図 2 は，第 8 回科学の甲子園全国大会 (第 11 問) の抜粋である．大学入学共通テストのサンプル問題と細部は異なるが，擬似コードである点がよく似ている．たぶん，初期の DNCL の言語仕様から独自に派生した記法になっている．

2.3 擬似コードの位置付け

大学入学共通テストや科学の甲子園で出題される問題例から示された通り，高校生のプログラミング教育において，擬似コードに接する機会は多くなる．しかし，より注目し

たいのは，このような擬似コードに対して，現在，厳密な統一規格が示されていない点である．DNCL は、多くの擬似コードのベースとなっているが，最新版の仕様は公開されていないようである．

情報系の標準的な常識からすると，わざわざ曖昧さを残しておくのは不自然である．ここからは著者らの推測の範囲に過ぎないが，ひとつ考えられる理由として，DNCL の設計者は，DNCL を独立したプログラミング言語として学習することを嫌がっていると考えられる．つまり，プログラミング教育は，世の中に実在するプログラミング言語で実習することを望み，擬似コードはあくまでもその学習成果を確認するための必要悪とたぶん位置付けられている．実際，コンピュータ科学の専門家の大半は，Python や C 言語などの実在するプログラミング言語での実習を好み，仮に DNCL が厳密に仕様化されたとしても，その構文規則を覚えるような学習を望まないだろう．しかし，試験問題に擬似コードが取り入れられるという影響力を考えると，今後どのようにプログラミング教育が高校生に広がるかは予想しにくい．

3. 目標と設計

大学入試センターがサンプル問題を示したことで，意図する・せざるに関係なく，擬似コードによるプログラミング教育は，高校生の間に広がる可能性がある．本研究では，このような状況下でどのようなプログラミング言語が教育的に考えられるか，プログラミング言語の設計論の立場から検討を行う．

3.1 自然言語とプログラミング

まず，著者らは自らの教育経験や開発経験 [4] により，プログラミング教育における自然言語の活用は，大きな可能性があると考えている．プログラミング初学者は，プログラミング言語が自然言語と異なり，正確なルールで書かなければ解釈されないという点に納得できない場合が多い．そのため，プログラミング言語の文法習得に時間が取られ，ささやかな構文エラーでも学習のつまづきになることが多い．

また，識別子 (変数名) のわかりやすさも，古いテーマであるが重要な論点である．最近のプログラミング言語は，ASCII 文字以外の識別子を用いることができるが，日本語の識別子を活用することに否定的なエンジニアも多い．しかし，教育現場では，図 3 で示した例のように，コード中に日本語識別子を用いると理解がしやすいという学生も少なくない．

近年では，教科書でローマ字の変数名が用いられることも少なくない．実際，DNCL 系の擬似コードで使われる変数名もローマ字である．なぜ，変数名に漢字やひらがな，カタカナを使わないのかを含め，再考する余地がある．

```
def 市松模様(縦,横):
  return '#' if (縦+横)%2==0 else '.'

for 縦位置 in range(10):
  for 横位置 in range(10):
    print(市松模様(縦位置,横位置))
  print()
```

図 3 日本語識別子によるプログラム例

3.2 設計目標

我々は，自然言語によるプログラミング理解の向上を目指し，新しいプログラミング処理系の構築を目指している．基本的な設計目標は次の通りである．

- 文法 (記法) を覚えなくてよい
- 文法 (記法) を覚えると，より精密な動作が可能になる
- 覚えた文法 (記法) は，本格的なプログラミングにおいて無駄にならない

Samoyed は，我々が試作を進めるプログラミング処理系である．自然言語による擬似コードを Python などのプログラミング言語に変換し，実行できるようにする．原則，現代日本語の文法にしたがっていれば，自由な記述を認める．さらに，Python の記法を覚えると，曖昧さが取り除かれ，より正確な解釈（プログラムの実行）が可能になる．最終的に，Python のサブセットの教育言語として利用できる．

図 4 は，Samoyed における擬似コードに近い記述によるプログラミング例である．

```
一松模様は，縦位置と横位置に対し，
    縦位置＋横位置が偶数なら'#'
    そうでなければ'.'となる

縦位置を0から10まで1つずつ増やしながら:
    横位置を0から10まで1つずつ増やしながら:
        縦位置と横位置の一松模様を表示する
    改行する
```

図 4 Samoyed におけるプログラム例

4. 試作システム

Samoyed は，自然言語による自由な記述からプログラミング言語まで，シームレスな処理系を目指している．その開発は，伝統的なコンパイラ構成法に基づく技法だけでなく，自然言語処理を統合させることが求められ，技術的なチャレンジも多い．本節では，試作開発の状況と今後の見通しを述べる．

4.1 構文解析

Samoyed は，自然言語から擬似コード，Python コードと様々な構文が混在した言語となる．このような言語の構文を定義するため，我々は解析表現文法/PEGTree[5] をベースとした優先度付き統合構文解析法を採用している．

図 5 は，繰り返しに関する 3 種類の構文を，優先度付き選択を用いて定義した構文 (抜粋) である．まず，Python 構文として構文解析を試みて，次に DNCL 風構文の解析を行う．最後に，自然言語文として，解釈する．

```
ForStatement = {
    "for" Expression
    "in" Expression ":"
    Block
    #For
}

NLForIncStatement = {
    Expression "を"
    Expression "から"
    Expression "まで"
    Expression "ずつ"
    "増やしながら" ":"
    Block
}

NLStatement = {
    (!'. ' *)
}
```

図 5 PEGTree による構文定義

解析表現文法は，字句解析を用いない統合解析であるため，各構文間の字句単位の違いを気にすることなく，異種の構文を組み合わせることができる．一方，日本語文法をそのまま形式化すること [6] は難しく，現時点で形態素解析と併用が求められるなどの課題が残っている．

4.2 機械翻訳と対話システム

Samoyed は，最終的に自然言語として扱うため，プログラミング言語の構文エラーが存在しない．一方，構文エラーがないことは，従来の抽象構文木ベースのコード変換が適用しにくいことになる．我々は，日本語記述からコードに変換する手法として，ニューラル機械翻訳に基づく確率的な変換も視野にいれ，翻訳モデルの構築 [7] を進めている．ただし，機械翻訳を用いるのは，仮に技術開発が成功したとしても，プログラミング教育的に望ましいことか議論は残る．

また，自然言語による記述は，常にプログラムとして解釈できる場合ばかりではない．コードに変換するときに必要な情報が足りなかったり，そもそもコードに変換できない記述もある．このような問題に対し，我々は，発話意図の推定を含めた自然言語の対話システムを参考に，適切な記述を促すプログラミング支援システムの創出を目指している．

プログラミングは，言語を正しく覚えて，正確に意図を記述することが前提であり，不完全で曖昧な記述に対する処理は考えられて来なかった．自由な記述，すなわち曖昧な記述も認めてしまう Samoyed に対して対話システムを導入することは，曖昧性を減らす観点から，さらには教育支援の観点からも重要な要素になると考えられる．

5. 関連研究

日本語プログラミング言語は，日本語で記述できるプログラミング言語である．母国語である日本語によって構文定義を行うことで，英語に不慣れなプログラミング初学者でも学習しやすい言語を目指している．これまでにも，さまざまな特徴をもった日本語プログラミング言語がいくつか開発されてきた．

なでしこは，オープンソースの日本語プログラミング言語である [8]．本格的なアプリ開発を行える言語機能を備えており，Microsoft Office などの外部のサービスとの連携も可能となっている．なでしこの特徴は，より日本語に近い語順でプログラムの記述が可能な点である．より日本語に近い語順で構文定義を行うことで，プログラミング特有の記述方法を覚えるコストの削減につながる．

ドリトルは，教育向けの日本語プログラミング言語である [9]．日本の中学校・高等学校の教科書等に採用されており，教育現場への普及が進んでいる．ドリトルの特徴は，ビジュアルプログラミングの要素を取り入れている点である．オブジェクトの動きをプログラムの命令として与えることで，命令の意味を直感的に理解しやすくしている．

6. むすびと展望

本稿では，大学入学共通テストに導入された擬似コードから着想を得て，自然言語によるプログラム理解を高める言語処理系の構想を示した．自然言語から Python までシームレスにプログラムの記述を可能にすることで，擬似コードと実用プログラミング言語の両者の利点を踏まえた学習につながると期待される．

参考文献

[1] 西田知博， 川合慧：未来のコンピュータ好きを育てる: 11．大学入試センター試験とプログラミング言語，情報処理， Vol. 50, No. 10, pp. 1013–1016 (2009).
[2] 中野由章：ぺた語義：大学入学共通テスト「情報」試作問題（検討用イメージ）と私感，情報処理，Vol. 62, No. 7, pp. 331–335 (2021).
[3] 水野修治：ぺた語義：大学入学共通テスト新科目「情報」〜

これまでの経緯とサンプル問題〜, 情報処理, Vol. 62, No. 7, pp. 326–330 (2021).
[4] 秋信有花，倉光君郎：自然言語記述からの近似コード生成を用いた初学者プログラミング支援，*The 4th. cross-disciplinary Workshop on Computing Systems, Infrastructures, and Programming (xSIG 2020)* (2020).
[5] Kuramitsu, K.: Nez: Practical Open Grammar Language, *Proceedings of the 2016 ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, Onward! 2016, New York, NY, USA, ACM, pp. 29–42 (online), DOI: 10.1145/2986012.2986019 (2016).
[6] 渡邉遥輔，秋信有花，若杉祐依，倉光君郎：解析表現文法を用いた日本語文法定義の試作，情報処理学会論文誌プログラミング（PRO），Vol. 13, No. 2, pp. 18–18 (2020).
[7] 秋信有花，小原百々雅，縫嶋慧深，倉光君郎：Transformer による日本語と Python コード間の機械翻訳，情報処理学会論文誌プログラミング（PRO），Vol. 14, No. 2, pp. 25–25 (2021).
[8] 酒徳峰章：日本語プログラミング言語「なでしこ」，コンピュータ ソフトウェア，Vol. 28, No. 4, pp. 23–28（オンライン），DOI: 10.11309/jssst.28.4_23 (2011).
[9] 兼宗　進，御手洗理英，中谷多哉子，福井眞吾，久野　靖：学校教育用オブジェクト指向言語「ドリトル」の設計と実装，情報処理学会論文誌プログラミング（PRO），Vol. 42, No. SIG11(PRO12), pp. 78–90 (2001).