

デザインレシピ内で使用される目的文の作成のための ブロックプログラミング環境

東京工業大学 情報理工学院 数理・計算科学系
19B30563 酒井大我
指導教員 増原英彦教授

1 はじめに

目的文とは、プログラマが関数の動作を、具体的な実装を見ずとも理解できるように書かれた自然言語の文章のことである。ソースコードと自然言語を対応づけるように目的文を設計することで、関数内の変数や再帰呼び出しに対する自然言語の説明に利用できる。また、目的文の一部箇所を具体的な値に置き換えることによって、特定の状況に対する目的や動作の説明にも利用できる。

本研究では、学習者の作成する目的文を改善することを目標に、プログラミングによる問題解決の手順を示したデザインレシピ [1] 内での目的文の作成と活用にもフォーカスしたシステムを提案する。また、これらの機能をデザインレシピに沿ったブロック型プログラミング学習支援環境 Mio [2] に組み込む。提案しているシステムでは、デザインレシピ内で利用される目的文の作成を支援する。また、再帰呼び出しや変数に対する自然言語の説明や、パターンマッチの各パターンで求めるべき具体的な値に対応する自然言語の説明を提供する。テスト例の作成も自然言語の説明を用いて補助する。

2 目的文作成のためのプログラミング環境

提案する環境の概観を図 1 に示す。この環境では、問題文から問題の構成要素を取り出し、ブロックを組み合わせることで、目的文を作成する。このとき、引数情報を問題の構成要素に結びつけることにより、入出力の情報しか含まれない目的文を書くことを防ぎ、シグネチャとの違いを明確にする。

学習者はプログラミング演習を行う際、以下のステップに従い、目的文の作成を行う。

■問題文の整理 学習者は、問題文から問題の構成要素を取り出し、各構成要素に対してプロパティや子要素を入力する。また、構成要素やプロパティが引数であるかを入力する。

図 1 は、点列の各 2 点間の距離から、それぞれの点の原点からの距離を求める関数 distance の例である。学習者は最初に左側のウィンドウを用いて、問題を構成する要素である「点列」「点」を選択する。すると、選択した各構成要

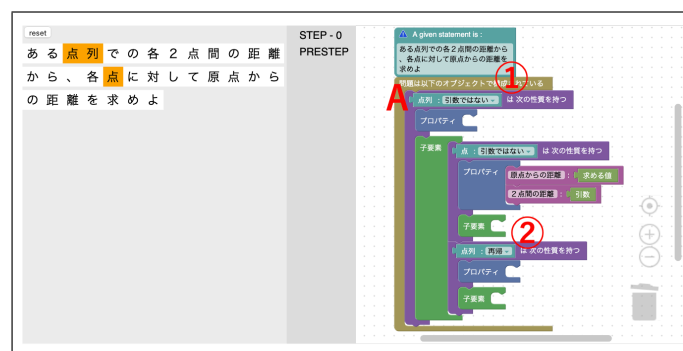


図 1 作成した環境の概観

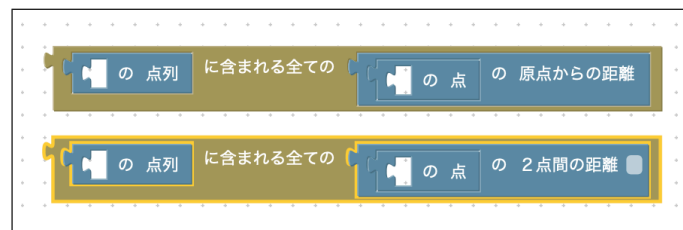


図 2 生成されるブロック

素から図 1 での点列ブロックのようにブロックが生成される。学習者は、点列ブロック (図 1-A) の子要素に、点と点列ブロックを入力する。このとき、点列は引数ではないとブロックの付加情報 (図 1-1) に設定し、また子要素に持つ点列は再帰構造になるため、子要素の付加情報 (図 1-2) を再帰的であると設定する。そして、点はプロパティとして原点からの距離と 2 点間の距離を持ち、それぞれ求める値と引数であると設定する。

■目的文の作成 問題文の整理が完了すると、図 2 のようなブロックが生成される。学習者は、図 2 のブロックと図 3 の文章構造を表すブロックを組み合わせ、目的文を作成する。また、前ステップで引数としたブロックに適切な引数名を入力する。

distance 関数の目的文では、引数名として segments を入力する。完成した目的文は図 4 の通りである。

3 デザインレシピでの目的文の活用

前節で作成した目的文を Mio で活用する。Mio は、目的文の変形により、関数の求める値を表すブロックを作成する。また、各引数や再帰呼び出しに対して、目的文から生成した自然言語の説明を提供する。学習者は、提供されたブ

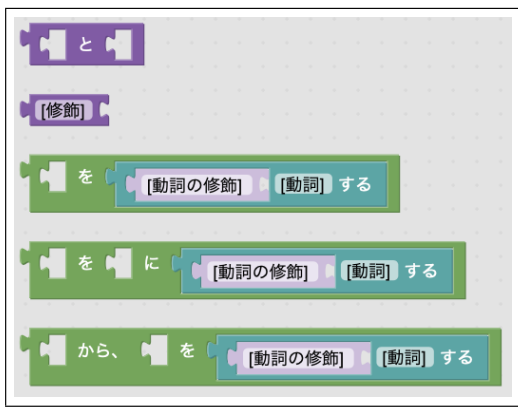


図3 文章構造を表すブロック



図4 distance の目的文

ブロックや自然言語の説明を活用しながら、デザインレシピのステップに従ってプログラムを設計する。以降、distance 関数 (図5) を用いた具体例とともに Mio の動作を説明する。

1. データ定義、データ例

データ型とデータ例を入力する。このとき、Mio でコンストラクタの各フィールドに自然言語での説明を入力する。(1) では、 x に「2点間の距離」、 xs に「残りの点列に含まれる全ての点の2点間の距離」という自然言語の説明を加える。

2. 目的文、入出力の型

前節で説明したように、ブロックで目的文を作成する。そして、入出力の型を入力する。今回作成した目的文は、求める値「segments に対して計算した点列に含まれる全ての点の原点からの距離」に変形され、以降のステップで利用される。

3. テスト例

Mio で提供される求める値のブロックを用いて、テスト例を定義する。ブロックの引数部分に具体的な値を入力し、自然言語の説明をヒントに出力例を入力する。(2) では、segments の箇所に具体的な値 List(2, 1, 3) を入力し、出力例 List(2, 3, 6) も入力する。

4. テンプレート

それぞれのパターンに対するテンプレートを求める。Mio ではテンプレートを求める際、フィールド変数や引数、再帰関数に紐づけられた自然言語の説明とそれぞれのパターンでの求める値がヒントとなる。(3) では、「Nil に対して計算した点列に含まれる全ての点の原点からの距離」と「 $x :: xs$ に対して計算した点

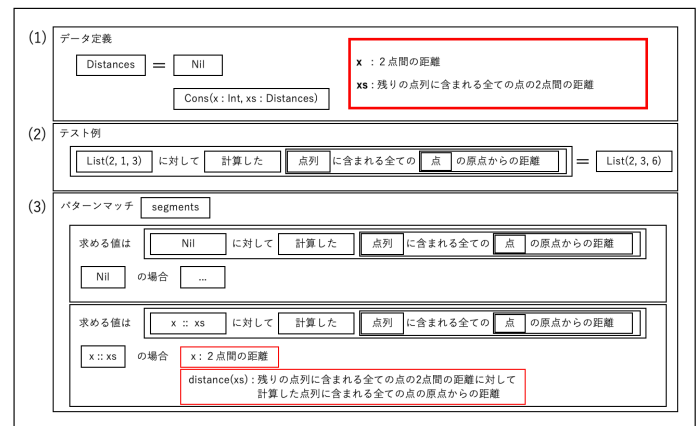


図5 distance 関数での目的文の活用例

列に含まれる全ての点の原点からの距離」が各パターンにおける求める値として、表示される。また、 x や $distance(xs)$ に対して、「2点間の距離」と「残りの点列に含まれる全ての点の2点間の距離に対して計算した点列に含まれる全ての点の原点からの距離」という自然言語の説明が表示される。

5. 関数実装とテスト

Mio で表示されるテンプレートの各ブロックに対する自然言語の説明をヒントに、求める値を計算する方法を考える。最後にテストを行う。

4 プログラミング環境の実装

本環境の実装は Blockly [3] で行う。現在、目的文作成のためのプログラミング環境を部分的に実装している。現在の実装では、構成要素とプロパティを区別している。理由としては、プロパティとして抽出された情報が、構成要素に結びつけられることなく、目的文に組み込まれることを防ぐためである。

5 まとめと今後の課題

本研究では、デザインレシピ内で利用される構造的な目的文の作成を支援するブロック型プログラミング環境の設計を行った。また、Mio 上で目的文を用いたプログラミング支援を行えることを示した。

今後の課題としては、二分木などのユーザー定義型による再帰的構造についても目的文の利用をできるようにすることが挙げられる。また、学生に対するユーザー実験を行うことによって、学習効果や学生の作った目的文がどのように変化したかを評価したいと考えている。

[1] M. Felleisen et al., “How to design programs: An introduction to computing and programming”, The MIT Press, 2001.
[2] 能勢 純弥, Mio: プログラムデザインを支援するブロック学習環境, 2022. 修士論文, 東京工業大学
[3] Blockly. <https://developers.google.com/blockly>.