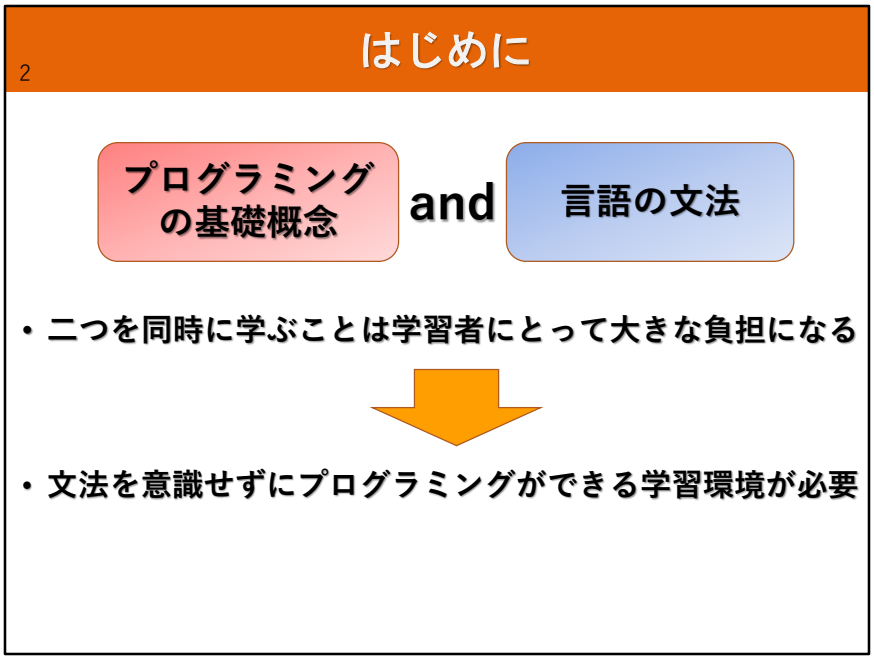
Blocklyを用いた多言語対応の
プログラミング学習支援環境の開発

香川研究室
14T239 佐野裕也

「Blocklyを用いた多言語対応のプログラミング学習支援環境の開発」と題しまして、
香川研究室の佐野が発表させていただきます。



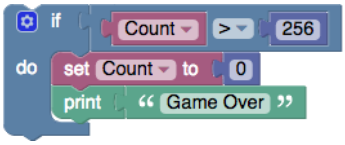
まずはじめに、研究の背景を説明させていただきます。
プログラミング学習者は、プログラミングの基礎概念と言語の文法を同時に学習しなければなりません。
これは、学習者にとって大きな負担になってしまいます。

そこで、文法を意識せずにプログラミングができる学習環境が必要になります。

3

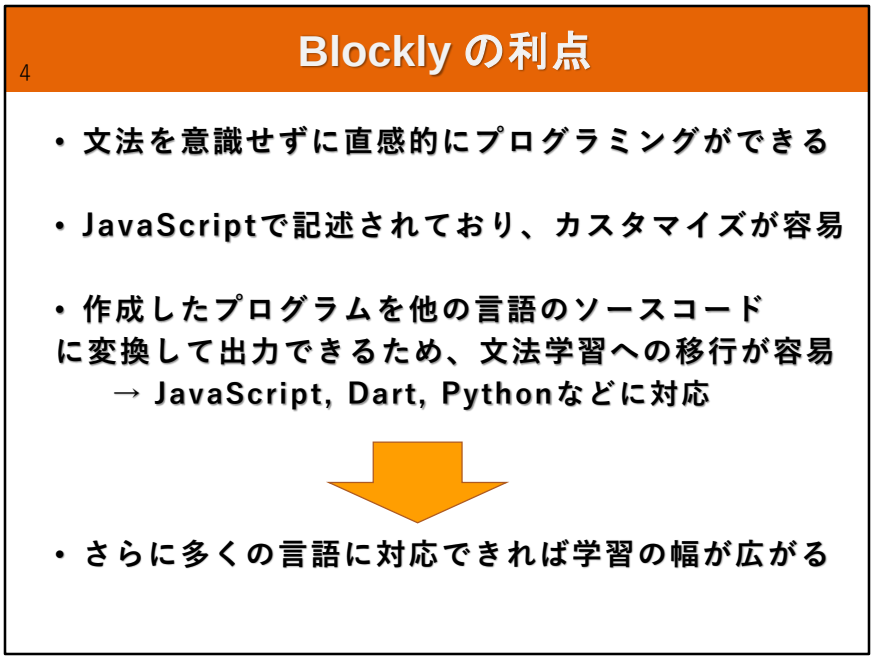
Blocklyとは

- Googleで開発されているグラフィカルな Webベースシステムのプログラミングエディタ
- ブラウザ上のブロックをドラッグ＆ドロップでつなぎ 合わせることでプログラミングを行うことができる



https://code.google.com/p/blockly/

その例として、Blocklyというシステムが挙げられます。
Blocklyとは、Googleが提供するグラフィカルなWebベースシステムのプログラミングエディタです。
ブラウザ上のブロックを、マウスや手で直接タッチしてドラッグ＆ドロップでつなぎ合わせることでプログラミングを行うことができます。

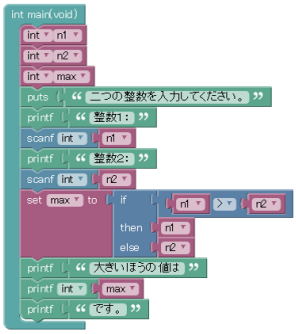


Blocklyを用いることで、文法を意識せずに直感的にプログラミングができます。
JavaScriptで記述されており、カスタマイズが容易にできます。
作成したプログラムを他の言語のソースコードに変換して出力できるため、プログラミングの概念の学習後、文法の学習への移行がしやすくなります。
現在は、JavaScript, Dart, Pythonのなどへのサポートが行われていますが、さらに多くの言語に対応することができれば学習の幅が広がると考えられます。

5

尾崎の研究

- BlocklyをC言語、Flex言語に対応
- プログラミング入門者が対象
- 文法を意識せずにC言語を学ぶことができる



13G454 尾崎陽一（2014年度修士論文）
「Webベースグラフィカルプログラミングエディタを用いた円滑な移行が可能なC言語学習支援環境の開発」

Blocklyによる過去の研究の一例として、尾崎の研究が挙げられます。
BlocklyをC言語、Flex言語に対応させたもので、システムの対象者がプログラミング入門者です。
このシステムによって文法を意識せずにC言語を学ぶことができます。

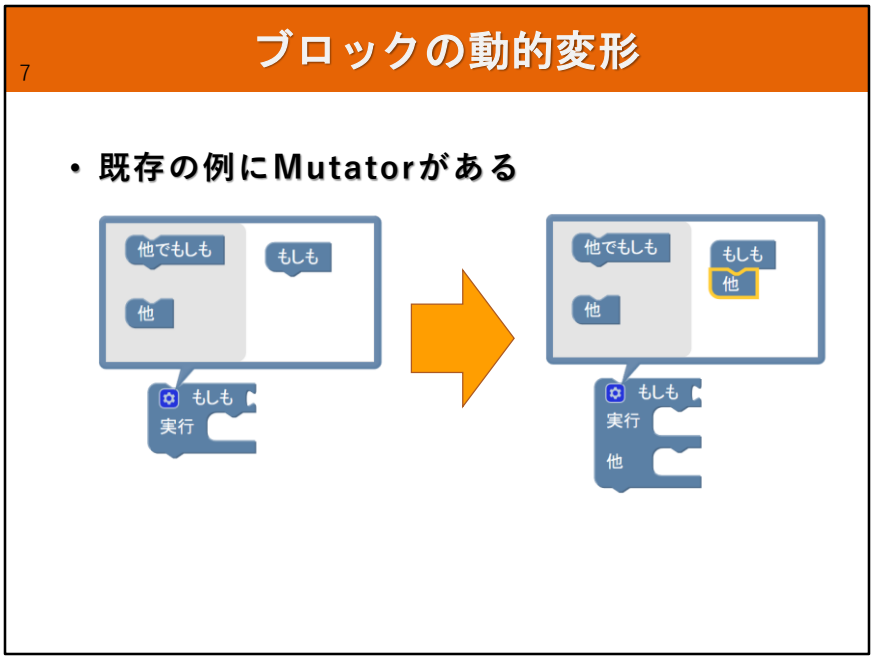
先行研究の問題点

6

- **大学の講義で学習する言語に対応しきれていない**
  →多言語化が必要
- **ブロックの動的変形が限られている**
  →柔軟性のあるプログラミング言語が
  ブロックの形状によって制約

しかし、先行研究には、次のような問題点が挙げられます。
まず、Blocklyが、大学の講義で学習する言語すべてに対応していないことです。
そこで、システムの多言語化が必要であると考えました。

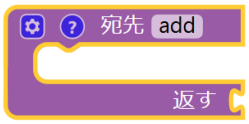
次に、ブロックの形の動的変形が限られているということです。
これでは、柔軟性のあるプログラミング言語をブロックの形状によって制約されてしまうことになります。



ブロックを動的に変形できる既存の機能にMutatorが挙げられます。
この、「もしも、実行」ブロックの左上の歯車のマークを押すと、その近くにこのようなふきだしが現れます。
このふきだしの左側のグレーの部分の２つブロックのいづれかを右のブロックの「もしも」ブロックに結合すると、「もしも、実行」ブロックの形状が変化するという仕組みです。

ブロックの多言語化

8




```
1  deleteOne :: Integer -> [Integer] -> [Integer]
2  deleteOne _ [] = []
3  deleteOne n (x:xs) = if n == x then xs else x : deleteOne n xs
```

- Haskellの関数によるパターンマッチングは既存のブロックでは対応できない
  →Haskellのカリー化も非対応

- 多言語化を行うために、動的変形による新たなブロックを定義しなければならない

既存のJavascriptのシステムにはこのような関数ブロックが実装されています。こちらは、Haskellのコードで、ある関数によるパターンマッチングをおこなっているものです。
このソースコードを、関数ブロックで表現しようとしても、JavaScriptの関数ブロックはパターンマッチングに対応していないため、Blockly上で表現することができません。
そこで、Blocklyの多言語化を行うために、動的変形によるパターンマッチングができる新たなブロックを実装しなければなりません。

研究方針

9

- 多言語化として C、Haskell、Flex に対するシステムの拡張を行った
- 動的変形機能を含んだ新たなブロックを定義
- システムの対象者は、
  プログラミング初心者～中級者

ここから、本研究で行った実装について説明させていただきます。

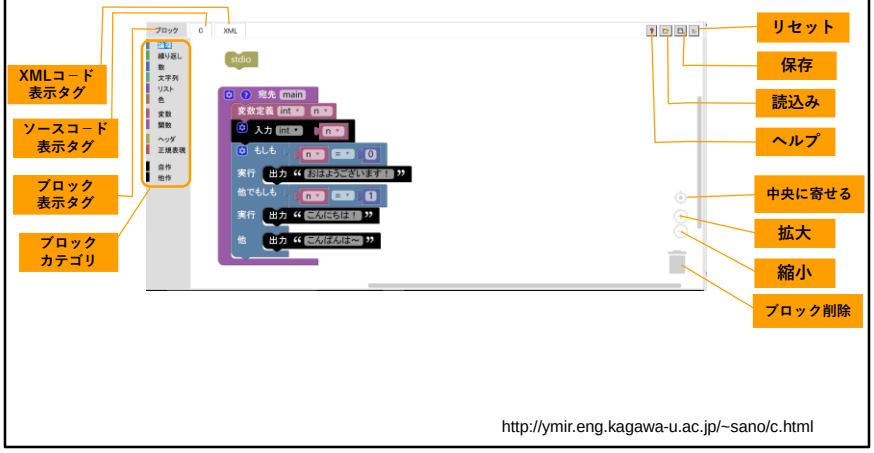
本研究では、BlocklyのC、Haskell、Flexに対するシステムの拡張を行いました。

その際に、動的変形機能を含んだ新たなブロックを定義しました。

システムの対象者は、プログラミング初心者から中級者です。

10

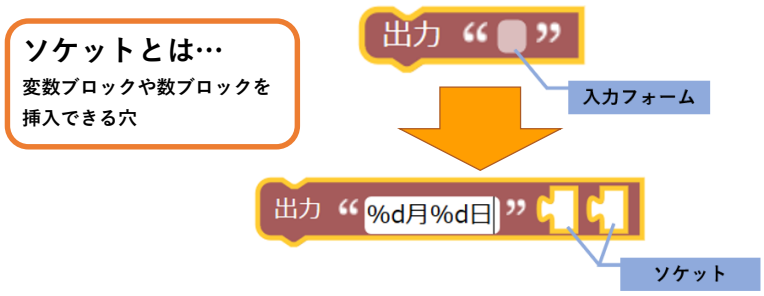
システムの概略図



こちらが、システムの概要図です。
このシステムには、３つのタグが表示されていて、ブロックタグ、ソースコードタグ、XMLコードタグがあります。
ブロックタグを押して、新しいブロックを取り出します。
ブロックを組み立てたら、ソースコードタグを押してソースコードを確認することができます。

11 C言語のシステムで新たに実装したブロック

- 動的変形機能を実装した入力・出力ブロック
- 入力フォームの%の数を検出して、
  動的変形機能でソケットの数が増える



本システムでいくつかのブロックを新たに実装しました。
まず、動的変形機能を実装した入力・出力ブロックです。
このブロックは、C言語のシステムで用意されています。
入力フォームで“%”の数を検出して、その数だけソケットの数を動的変形機能で増やしています。
検出のタイミングは、入力フォームの中身に変化があるごとに行われます。

12 Haskell言語のシステムで新たに実装したブロック①

- 関数定義ブロックと関数呼び出しブロック
- 関数呼び出しブロックは、関数ブロックを右クリックしてメニューで取り出す

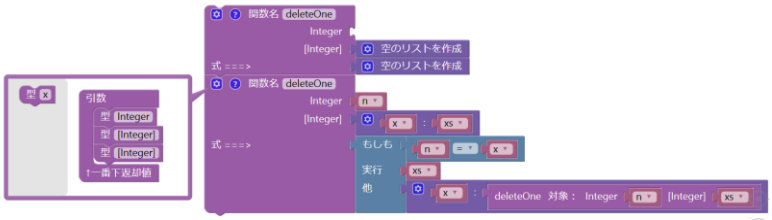
Haskell言語のシステムで新たに関数定義ブロックと関数呼び出しブロックを定義しました。

関数定義ブロックでは、Mutator機能でブロックを動的に変形させることができます。

関数呼び出しブロックは、関数ブロックを右クリックしてコンテクストメニューで取り出すことができます。

ブロックの接続例

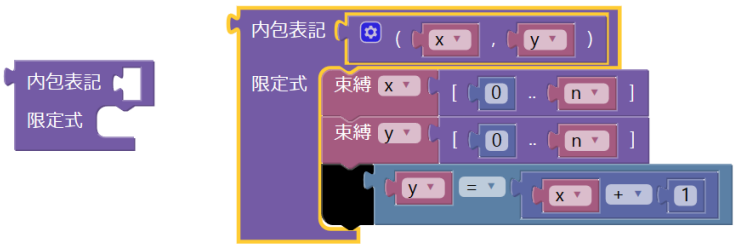
```
1  deleteOne :: Integer -> [Integer] -> [Integer]
2  deleteOne _ [] = []
3  deleteOne n (x:xs) = if n == x then xs else x : deleteOne n xs
```



そして、こちらがさきほどのHaskellのソースコードで提示したものをBlocklyで表現したものです。
関数定義ブロックの仮引数の数と関数呼び出しブロックの実引数のソケットの数は動的に対応できるようになっています。

14 Haskell言語のシステムで新たに実装したブロック②

- リスト内包表記ブロック
- ソースコードは、リストの中に1つの式と複数の限定式で記述される



Haskell言語のシステムのリストカテゴリーで新たに内包表記ブロックを定義しました。
図の左が、リスト内包表記ブロックに何も接続していない初期状態で、右がリスト内包表記ブロックの接続例です。
リスト内包表記ブロックは、ソケットに１つの式と複数の限定式を挿入します。

15

評価方法

- 学部生3名と院生2名を対象に評価を行った
- 評価方法は、以下の評価項目に自由に回答する形式で行った
  - 操作方法は直感的に分かったか
  - 使ったブロックとそのブロックの評価

本研究室の学部生3 名と院生2 名を対象に、実際にシステムを使用してもらい、その後以下の評価項目に自由に回答する形式で行いました。

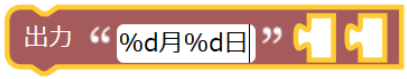
16 評価結果「操作方法は直感的に分かったか」

- 配列以外の項目では、ブロックのカスタマイズが直感的に分かった(C言語のシステム)
- どういう風につなげて良いかとかどう関数を作ったらいいかとかが分からなかった(Haskell言語のシステム)

「操作方法は直感的に分かったか」という項目では、C言語のシステムで「配列以外の項目では、ブロックのカスタマイズが直感的に分かった」など、肯定的な回答が見られました。ただし、C言語の配列は改善の必要があります。一方、Haskell言語のシステムでは「どういう風につなげて良いかとかどう関数を作ったらいいかとかが分からなかった」などといった否定的な回答が見られました。

17 評価結果 「使ったブロックとそのブロックの評価」

- %で出力変数を動的に変更できるのは良いと思った



- Haskellの関数ブロックの扱いが分かりづらかった

「欲しいブロックとそのブロックの評価」という項目では、「%で出力変数を動的に変更できるのは良いと思った」といった回答が得られ、出力ブロックの評判は良かったです。
Haskellの関数ブロックについては、扱いづらかったようです。

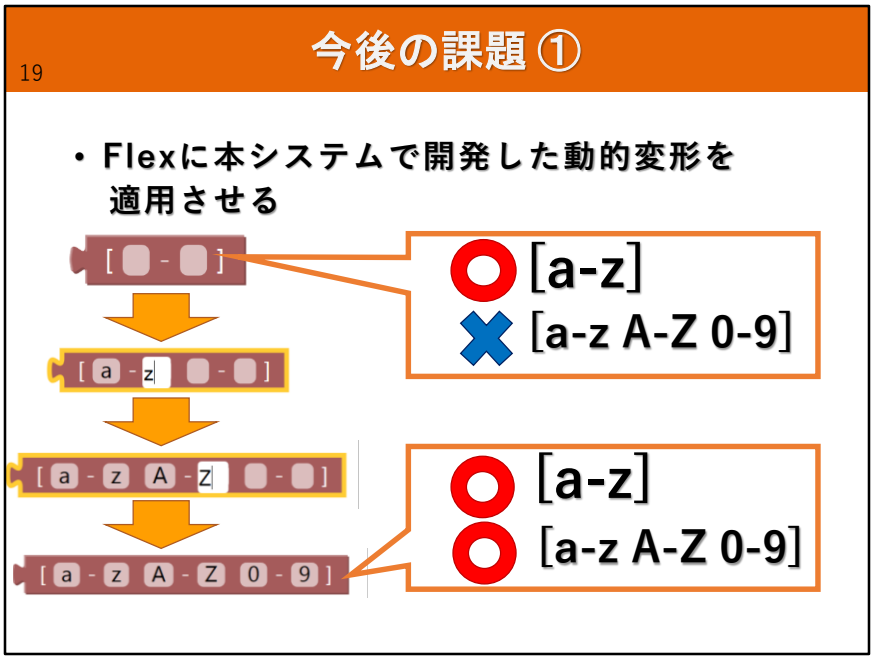
まとめ

18

Blocklyを用いて多言語対応のプログラミング学習支援環境を開発した

- **動的変形の実装で、さまざまなプログラムを組み立てることができる**
- **評価をもとにシステムを改良し、授業で効果を確認する必要がある**

まとめです。
本研究では、Blocklyを用いて多言語対応のプログラミング学習支援環境を開発しました。
その際に、ブロックの種類が多くなりすぎないように動的変形の機能の拡張を行いました。
この拡張によって、さまざまなプログラムを組み立てることができます。
しかし、実際に授業で使用してもらっていないので、今回の研究室で得られたフィードバックをもとにシステムを改良し、授業で効果を確認する必要があります。

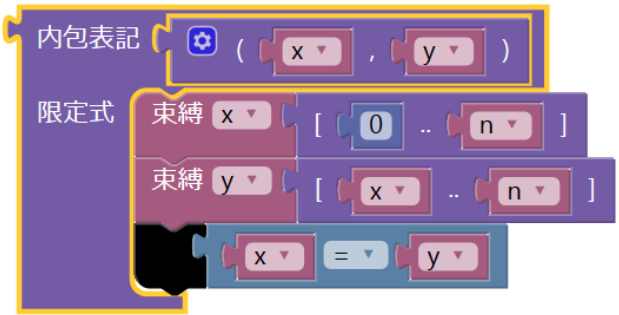


今後の課題として、Flexの正規表現に関するブロックを紹介します。
このブロックは、単体の文字クラスを表します。
複数の文字クラスを表すことができるように、このブロックの２つの入力フォームに文字を入力し終えたら、入力フォームがもう１セット追加されるような動的変形を実装したいと考えております。

20

今後の課題 ②

- ブロック接続部の改善



内包表記ブロックでは、Haskellの文法上接続できない場合があるため、接続できないときのために作られた黒色のブロックを用意しました。
このブロックは、プログラム上では何も意味を持たないものでエラーの原因にもなるため、ブロックの接続部の改善を検討します。

ご清聴ありがとうございました

ご清聴ありがとうございました