Vol.2018-CE-144 No.9
2018/3/17

プログラミング学習環境「Bit Arrow」における採点支援機能

堀越将之[1,a)]　長島和平[2]　長慎也[1]　兼宗進[3]　並木美太郎[2]

概要：著者らは，Web ブラウザを用いてプログラミング学習が可能な環境「Bit Arrow」を開発している．Bit Arrow は，プログラミング学習の授業での利用も想定されており，学習者と教員のそれぞれを支援するための機能が提供されている．しかし，教員を支援するための機能の一つである採点支援については，十分な機能が提供されていなかった．そこで授業実践を行った教員の意見を参考に機能の拡張を行っている．例えば，出力結果による出力判定機能や，学生のプログラムを教員の指定した条件により評価を与える機能を開発している．本稿では，採点支援機能の設計と今後の拡張について述べる．

Scoring support system for Bit Arrow

Horikoshi Masayuki[1,a)]　Nagashima Kazuhei[2]　Cho Shinya[1]　Kanemune Susumu[3]
Namiki Mitaro[2]

1. はじめに

著者らは，Web ブラウザを用いてプログラミング学習が可能な環境「Bit Arrow」を開発している．これまで Bit Arrow では，プログラミング学習の授業での利用を想定し，学習者が学習に取り組みやすい環境づくりのために様々な機能を提供してきた．例えば，文法エラーのメッセージと発生場所の表示や，プログラムを短く書けるようにするライブラリの提供などである．また，教員向けの支援機能も提供されており，クラス管理やプログラムの配布，学習者の進捗を把握することなどが可能である．

一方で，プログラミング教育において教員の負担となりやすい事項の一つである採点については，十分な機能が提供できていなかった．本稿では，Bit Arrow で提供する採点支援機能についてと，本稿執筆段階で構想している今後の拡張について述べる．

[1] 明星大学
Meisei University, Japan
[2] 東京農工大学
Tokyo University of Agriculture and Tecchnology, Japan
[3] 大阪電気通信大学
Osaka Electro-Communication University, Japan
[a)] 17mj005@stu.meisei-u.ac.jp

2. 教員支援機能

2020 年度以降，プログラミング教育の必修化が決定している [1]．プログラミング教育の必修化により，今まで以上にプログラミングに興味を持つ学習者の増加が見込まれる．学習者の増加は喜ばしいことではあるが，それに伴い教員にかかる負担も大きく増加することが懸念される．そこで Bit Arrow では教員を支援するために，クラス管理やプログラム配布などの機能が提供されてきた．プログラム配布機能では，**図 1** のように教員が設定した課題のテンプレートとなるファイルを各学習者のワークスペース内に配布することが可能で，そのファイルを編集して提出させるような運用が行える [2]．

一方で，採点について支援するための機能は十分な提供が行えていなかった．採点にかかる時間は学習者や課題の数に比例して増加し，学習者の能力によっても左右される．そこで，コンピュータによる採点支援機能を導入することで，教員の採点にかかる負担を軽減することができ，学習者も素早く正誤判定を知ることができると考える．

このことを実現するための，採点支援機能の実装について次に述べる．

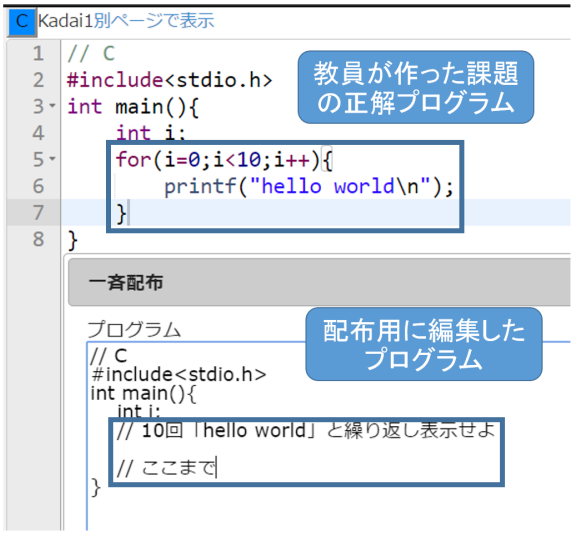

図 1　プログラム配布ダイアログ

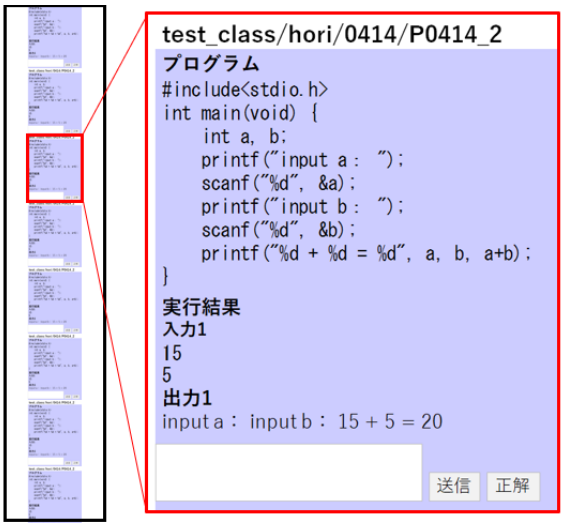

図 2　採点フォーム

3. 各機能の実装

これまでに開発した採点支援機能は，C 言語のプログラミング学習を想定して設計している．Bit Arrow では C 言語だけでなく，教育用 JavaScript やドリトルといった複数の言語を扱える．今後，採点支援機能もこれらの言語に対応できるよう開発を進める予定である．

3.1 採点フォーム

本節で述べる採点フォームの機能については以前に発表 [2] を行っているが，3.2 節で述べる新たに実装した機能の基本となるため改めて説明する．

プログラムの採点を行うには，学習者のプログラムを教員が確認して採点を行い，場合によっては学習者にフィードバックを与える必要がある．これを各学習者に対して行うのは教員にとって非常に負担となる．そこで，各学習者が編集したプログラムを課題ごとにまとめて実行し，図 2 のような採点フォームが生成される仕掛けを用意した．この採点フォームは一つの課題につき，一つの Web ページとして学習者の人数分生成され，一覧表示できる (図 2 左)．一つの採点フォーム (図 2 右) には学習者の提出したファイルのパス，学習者が提出したソースプログラム，教員側で設定した入力，その出力結果，教員が採点結果を記入するためのコメント欄が設けられている．教員側で設定する入力は現実装段階では，Bit Arrow のサーバ側のファイルに直接書き込む必要がある．採点結果のコメントは，そのまま学習者にフィードバックされるため，間違っている箇所やヒントなどを個別に指摘することも可能である．また，正解のときはボタン一つで採点できる．

現在の採点支援機能の構造を図 3 に示す．現在の開発段階では，採点フォームを出力するコマンドが実装されている．このコマンドは指定された課題の名称（プロジェクト名およびファイル名）に対して，各学生の該当するファイルについて順番に gcc コンパイラに読み込ませ，図 2 の採点フォームを一覧表示した Web ページを作成する．

Bit Arrow では独自の実装により，ドリトルや JavaScript だけでなく C 言語でもグラフィックスやアニメーションが利用可能であるが，この採点フォームはそれらのグラフィックスを出力結果とする課題にも対応している．グラフィックスの出力結果は図 4 のようなサムネイルとして表示され，これをクリックすることでアニメーションの詳細な動作を確認可能にしている．

3.2 出力判定機能

プログラミングの授業では，時間内に複数の問題を学習者に解かせるコンテスト形式で授業を行う場合がある．この時，教員がすべての解答を目視で採点することは困難であり，採点にかかる時間にムラができ公平性を保てないため，採点フォームを一覧表示させるだけではこの形式の授業を支援することができない．そこで，解答の正否をプログラムの実行結果により判定し，採点を支援する機能を開発した．この機能を出力判定機能とする．

出力判定機能はあらかじめ設定した正解と学生のプログラムの実行結果を比較して，完全一致判定により採点を行う機能である．この機能は採点フォームの拡張機能として開発している．出力判定機能を利用するには，プログラムに与える二組のテストケースをあらかじめ設定しておく必要がある．採点結果により，採点フォームのコメント欄に自動で結果に応じたコメントを書き込む．学習者に提示されるコメントは次のようなものを用意している．

- 入力テストで二つとも正解した場合
  正解のプログラムを書けたことを伝える旨のコメントを表示する．
- 入力テストで一つだけ正解した場合

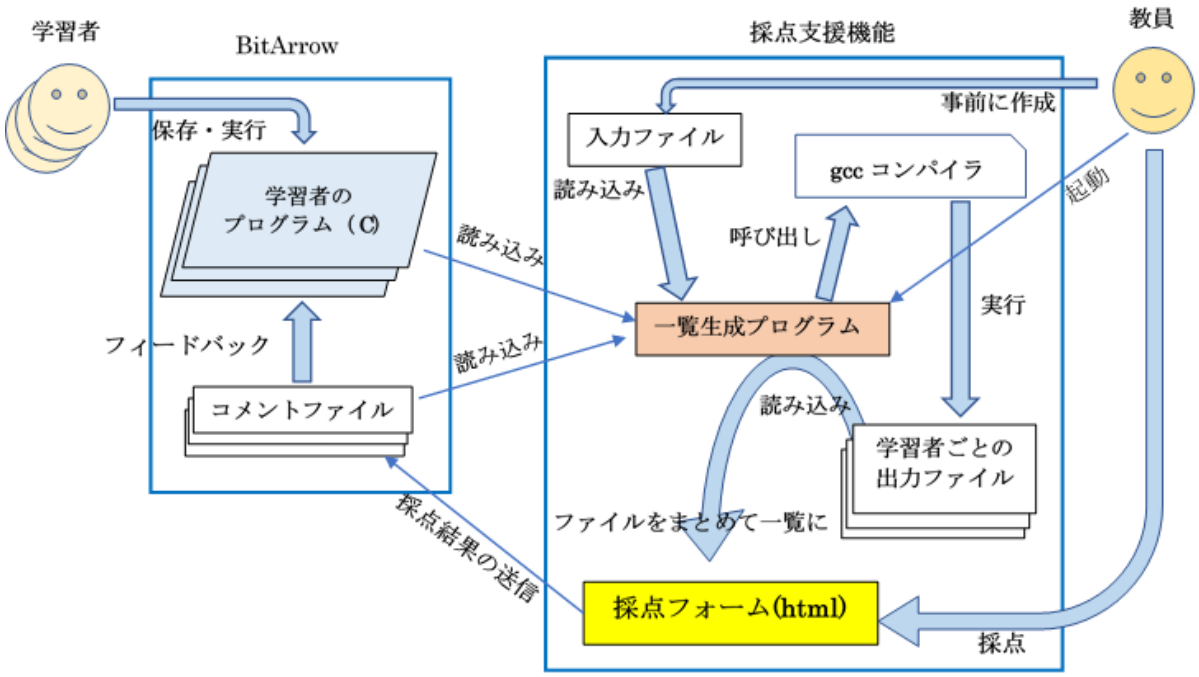

図 3　採点支援機能の構造

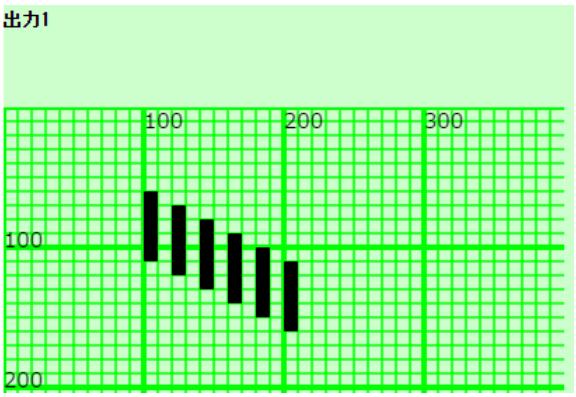

図 4　グラフィックスのサムネイル

学習者に提示している特定の入力に対してのみ対応するプログラムを書いたとして，他のテストケースに対応できるようなプログラムを書くように指示するコメントを表示する．

- 入力テストに一つも正解できなかった場合
  入力テストの入力値を一つだけ公開し，それに対する出力例をコメントで表示することで問題の意図を理解させる．
- 明らかに間違いである場合
  採点を行わず，コメントも表示されない．この明らかに間違いである場合というのは，プログラム配布機能を利用して学習者に配布されたが，編集が加えられていない状態のプログラムを実行した場合や，コンパイルエラーで実行不可だった場合を想定している．

出力判定機能はコンソール上に文字を出力するタイプの課題だけでなく，グラフィックス表示させる課題の正否判定も行える．これは，出力された画像を画像解析による一致判定を行っているのではなく，Bit Arrow がグラフィックスを表示する仕組み [3] を利用して判定を行っている．Bit Arrow でのグラフィックス表示は，**図 5** のようにソースプログラムをグラフィックス表示用の中間言語に変換し，その中間言語を読み込むことで達成している．出力判定機能では，正解となる実行結果と学生のプログラムの実行結果，それぞれの中間言語を比較することで正否を判定している．ただし，アニメーションさせる課題は中間言語の比較による判定が困難であるため，出力判定機能の対象外とする．

出力判定機能で正否判定を行うとき，一見正解の出力のように思われるものでも，一つの命令で描けるグラフィックスを二つ以上の命令を利用して描こうとしている場合や同座標に連続してグラフィックスが描かれている場合，実行画面上に表示されない座標にグラフィックスが描かれている場合などに間違いであると判定される．ただし，コンソール上に文字を出力するタイプの完全一致判定とは少し異なり，複数のグラフィックスを描く場合に，順番については考慮されない．例えば，図 4 のグラフィックスを出力させるような課題を判定しようとしたときに，左から順に出力した場合と右から順に出力した場合のどちらも正解と判定される．

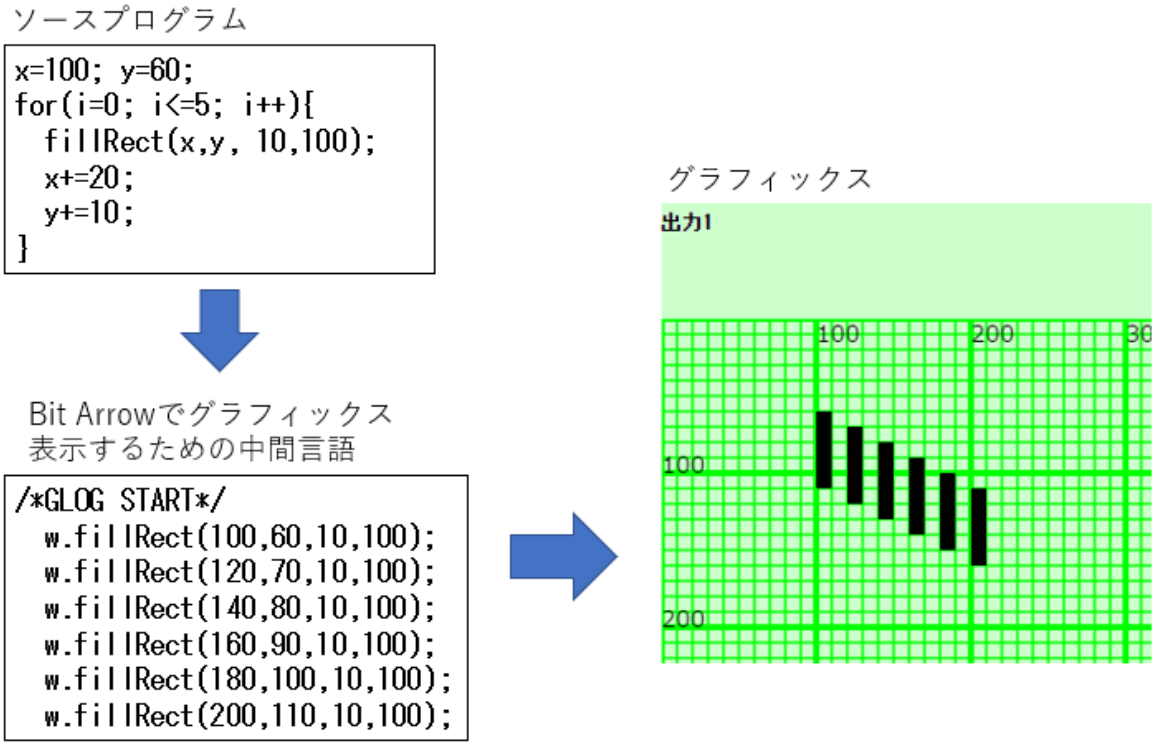

図 5　グラフィックス表示の仕組み

4. 評価

今回実装を行った採点支援機能の試験運用を行った．試験運用は，昨年の情報科学系学部 2 年生を対象に開講されるプログラミングの授業で行われ，40 人程度の学習者が被験者であった．この授業では実例を通して，C 言語の関数，ポインタ，構造体を中心に，プログラミングの技法を学ぶことを目的としていた．全 15 回の授業を行い，提出された課題は基本的に採点フォームを用いて手動で採点した．最終回のみコンテスト形式で多数の課題を学習者に出し，提出された課題を出力判定機能を用いて採点した．授業中，一部トラブルが発生したが現在は対応済みである．更なる授業実践については 5 章で述べる開発中の拡張機能と合わせて，来年度より開講される同授業内で行う予定である．

実践の回数は十分でないが，試験運用に協力していただいた教員から出力判定機能についての意見や要望をいただいたため，次に示す．

- テストケースをサーバ上に用意する必要があるため面倒
- 現状では，三つ以上のテストケースを与えられない
- 学習者が保存したファイルをすべて採点フォームに表示するため，それらが作業中で未完成なのか，完成しているのか区別がつかず，コメントをつけるタイミングが難しい

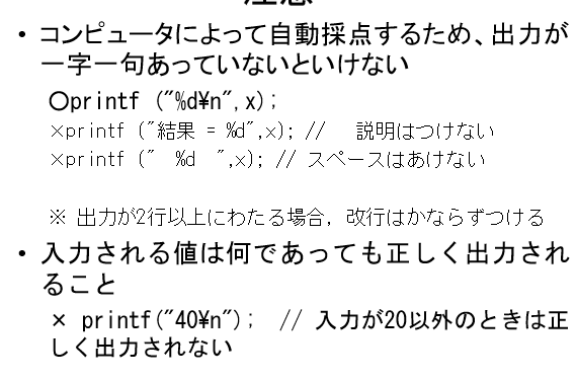

図 6　出力判定機能の注意書き

- 教員が指定した命令や構文が使われているか検索し，使われていない場合は不正解としたい
- プログラムの本質があっている場合，説明文などが書かれていても正解としたい

これらの意見について対応できるように拡張を行う予定であり，現在既に設計と実装を進めている．

5. 今後の採点支援機能の拡張

本稿執筆までに行った試験運用では，出力判定機能が正解と出力の完全に一致するプログラムのみを正解とするため，学習者には次の図 6 のような注意書きを提示した．

試験運用時のように，出力が結果のみであるプログラムを書かせるのは，表現の幅を狭めたり，あとからプログラ

ムを見返したときに出力の意図が分かりにくいなど，プログラミングの授業としては望ましいものではない．可能な限り避けたいが，現システムの都合上，このようなプログラムを学習者に書かせてしまっている．また，出力形式を厳密に指定すると，出題されたプログラムの本質とは違う部分で時間がかかる恐れがある．一方で，出力形式に自由度を持たせるとコンピュータのみによる採点は困難になる．そこで，採点支援機能の更なる拡張として，コメント支援機能を考案中である．

コメント支援機能では，出力形式に自由度を持たせて出題し，その中でも明らかに間違いと思われるものや，正解である確率が高い出力について自動的に評価コメントを作成し，採点フォームに書き込むことができる．教員は，書き込まれたコメントをもとに採点結果の修正を行うことで，一から採点する手間をへらす．

このとき，自動的に評価コメントを作成するための基準は，教員が出題時に設定するが，その基準を素早く記述するための言語（以下，採点基準言語と呼ぶ）を考案した．また，出題を Bit Arrow のプログラム編集画面から行える機能や，学生側が明示的に提出を行える機能を追加中である．

5.1 課題作成機能

採点フォームを出力するには，対象となる課題に対応する入力を設定する必要があるが，4 章で述べた試験運用の時点では，入力となるデータを作成するためのユーザインタフェースが存在していなかったため，入力データをサーバ側のファイルに直接書き込む必要があり，作業が煩雑であった．そこで，Bit Arrow のエディタ内で，課題を作成するための仕組みを作成中である．

Bit Arrow には教員としてログインする機能があり，教員でログインした場合のみ，課題を管理するメニューが追加される．このメニューを開くと，**図 7** のようなダイアログが表示され，課題を追加・編集することができる．左側のダイアログは，新しい課題を追加したり，これまでに出題した課題の一覧を表示・編集したりすることができる．右側のダイアログは，それぞれの課題を採点するときに，コメントを自動的につけるための基準を採点基準言語で記述する欄である．ここで記述した採点基準は，該当する課題の採点フォームを生成したときに，採点基準欄に自動的に書き込まれる．また，採点フォーム上で採点基準を編集すると，採点基準が保存される．

5.2 課題提出機能

4 章で述べた試験運用の時点では，前述したプログラム配布機能を用いて指定したファイル名で課題を提出させる運用を行った．その際，各学習者が該当するプログラムの保存をしたことを「提出」と見なしていたため，教員からの意見にあった通り，完成していないプログラムも採点フォームに表示されてしまっていた．

そこで，提出する動作を学習者に明示的に行わせるよう，**図 8** のような課題提出のためのダイアログを Bit Arrow に新たに追加し，このダイアログで明示的に提出を行ったものだけが採点フォームに出力されるようにする予定である．

5.3 コメント支援機能

コメント支援機能の主な働きは，採点フォームに出力されている学生のプログラムやその実行結果から，教員の指定する命令や構文が使われているか，期待通りの出力ができているかを検索し，該当した学習者のコメント欄に指定したコメントを書き込むことである．そして，コメントを書き込んだ採点フォームを教員に提示する．

コメント支援機能の完成予想図を**図 9** に示す．教員が学習者のプログラムの採点基準となる条件と評価コメントを記述すると，採点基準に従い学習者のコメント欄にコメントが自動的に書き込まれる．ここで生成される採点フォームは，課題作成機能で与えられる採点基準を利用して生成されているが，採点基準で入力値を与えたり，採点基準を編集したりすることで採点フォームの内容を更新することができる．この採点基準設定欄は次に示す**図 10** の採点基準言語に基づき作成している．

採点基準言語を利用すると，学習者のプログラムのソースコードや出力からパターンマッチを行うことができる．パターンマッチとは，指定した条件で検索を行い，条件に一致した場合に指定した評価コメントを書き込むことである．採点基準言語の仕様は次のとおりである．

- セクション
  一つのセクションは，パターンマッチを行う対象の指定を行頭に書き，その後パターンマッチ行を一つ以上指定したものである．パターンマッチを行う対象はソースコードまたは出力である．出力を対象とする場合は入力を与えることができ，入力に対応する出力を対象にパターンマッチを行う．また，異なる入力ごとに別セクションを作成できる．ソースコードを対象とする場合は “P” と書き，出力を対象とする場合は “I” と書く．I の後ろにはスペース区切りで入力値 *value* を指定できる．
- パターンマッチ行
  検索を行う条件であるパターンと，パターンを検出した場合に書き込むコメントを記す．
  - *pattern* に文字列を記述し，その文字列がプログラムや出力の中に含まれるか検索する．*pattern* の先頭と末尾がスラッシュの場合，正規表現でパターンマッチを行う．パターンの先頭に “not” と記述すると指定した条件が含まれないもののパターンマッチを行う．パターンの先頭に “*” と記述すると，パターンが検出された場合，同セクション内でのこれ以降の

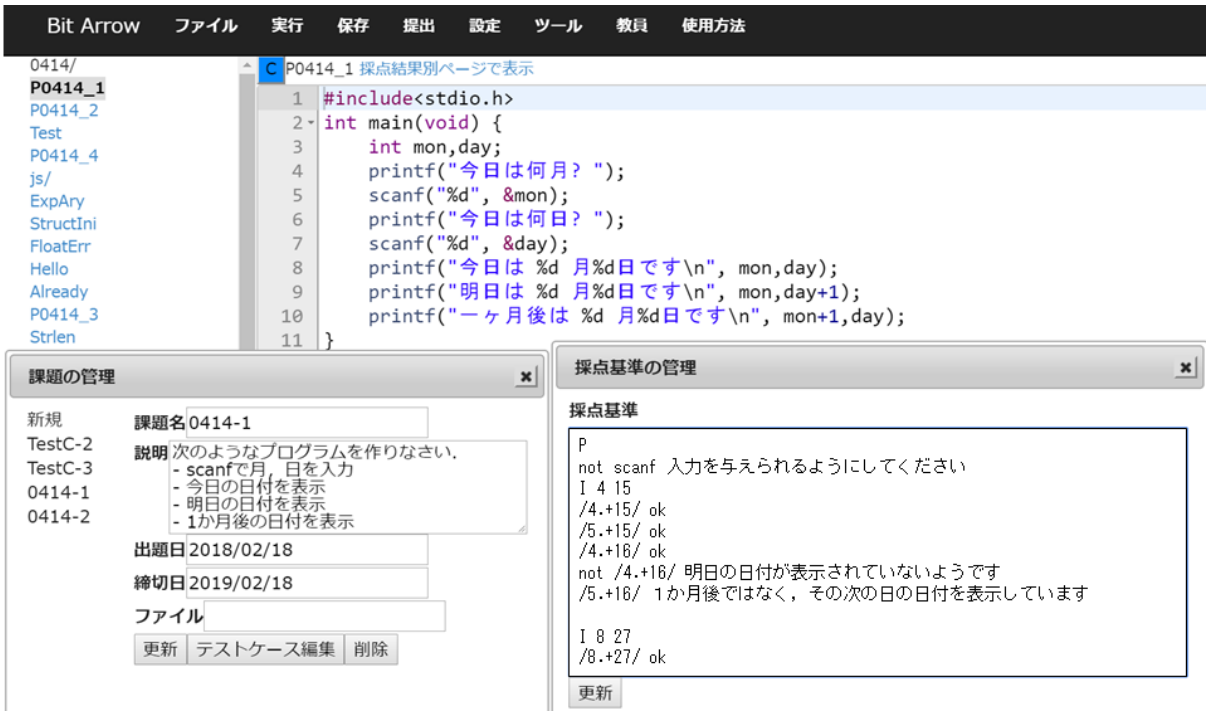

図 7　課題作成機能

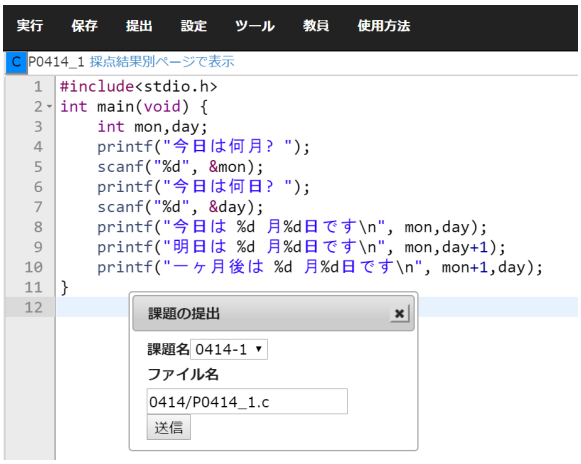

図 8　課題提出機能

パターンマッチを行わない．

– パターンを検出したときに *comment* に書かれた内容を学習者のコメント欄に書き込む．*comment* に “ok” が指定されると，セクション内で ok の指定されたパターンマッチ行すべてがパターンを検出し，かつそれ以外のパターンマッチ行がいずれもパターンを検出しなかった場合に “pass” というステータスを与える．セクション内で ok が指定されたパターンマッチ行が一つもないときは，すべてのパターンマッチ行がパターンを検出しなかった場合に pass ステータスが与えられる．pass ステータスを持つセクションの数に応じて，最終結果に「正解」や「再提出」などの汎用コメントを表示する．

例えば，次の**図 11** のような課題の採点でコメント支援

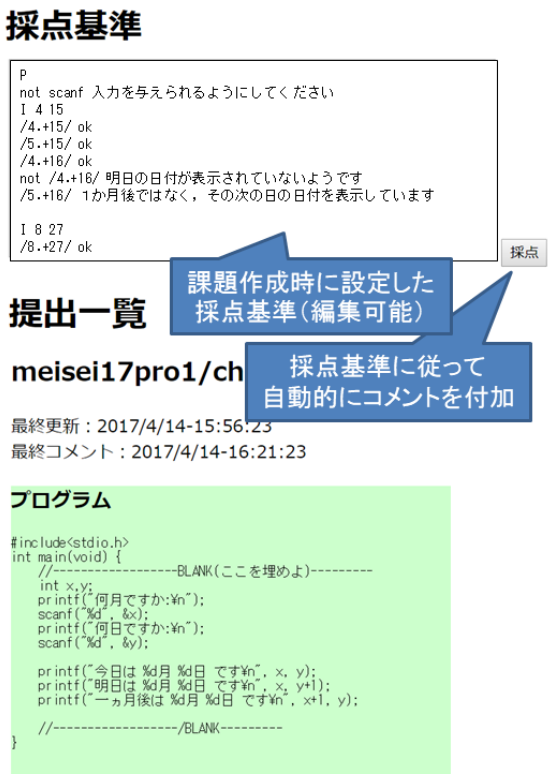

図 9　コメント支援機能の採点基準設定欄（完成予想図）

機能を利用する場合，**図 12** に示すような採点基準を書く．図 12 は次のような意味を持つ．

- 学習者のソースコードに `scanf` が含まれていなければ，「入力を与えられるようにしてください」というコメントを書き込む．
- 入力値 4 15 に対して，4 月 15 日，4 月 16 日，5 月 15

```
P
pattern comment
pattern comment

I value value
pattern comment

I value value
pattern comment
pattern comment
```

図 10　採点基準言語の構文入力方法

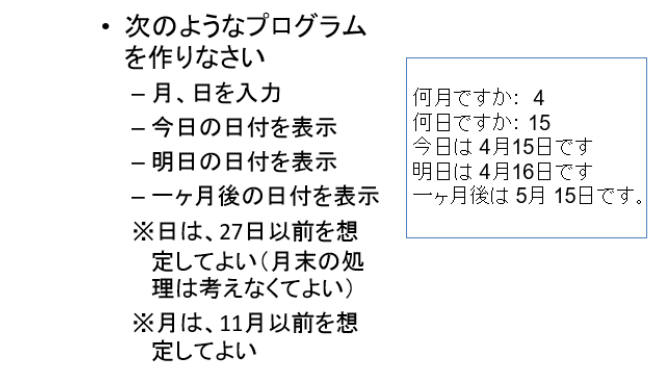

図 11　プログラミングの授業で出題された課題の例

```
P
not scanf 入力を与えられるようにしてください

I 4 15
/4.+15/ ok
/5.+15/ ok
/4.+16/ ok
not /4.+16/ 明日の日付が表示されていないようです
/5.+16/ 1 か月後ではなく，その次の日の日付を表示しています

I 8 27
/8.+27/ ok
/8.+28/ ok
/9.+27/ ok
not /8.+27/ 明日の日付が表示されていないようです
/9.+28/ 1 か月後ではなく，その次の日の日付を表示しています
```

図 12　採点基準言語の入力例

日などの出力がすべて含まれていれば正解とする．正規表現を利用しているので，「月」や「日」を使っていなくてもよいものとする．

- 入力値 4 15 に対して，4 月 16 日などの出力が含まれていなければ，「明日の日付が表示されていないようです」というコメントを書き込む．ここでは省略しているが，「4 月 15 日」「5 月 15 日」が表示されないことについても，同様にコメントを書き込むことができる．

```
#include<stdio.h>
  int main(void) {
  int day,month;
  printf("何月ですか：");
  scanf("%d",&month);
  printf("何日ですか:");
  scanf("%d",&day);
  printf("今日は%d 月%d 日です \n",month,day);
  day=day+1;
  printf("明日は%d 月%d 日です \n",month,day);
  month=month+1;
  //一ヶ月と一日後を表示してしまう
  printf("一ヶ月後は%d 月%d 日です \n",month,day);
}
```

図 13　学習者の誤ったプログラム

```
●ソースコードの内容について
結果：pass
●入力 4 15 について
あなたのプログラムの出力
———
何月ですか：何日ですか:今日は 4 月 15 日です
明日は 4 月 16 日です
一ヶ月後は 5 月 16 日です
———
結果：1 か月後ではなく，その次の日の日付を表示しています
●入力 8 27 について
あなたのプログラムの出力
———
何月ですか：何日ですか:今日は 8 月 27 日です
明日は 8 月 28 日です
一ヶ月後は 9 月 28 日です
———
結果：1 か月後ではなく，その次の日の日付を表示しています
●最終結果：
再提出
```

図 14　コメント支援機能での採点例

- 入力値 4 15 に対して，5 月 16 日などの出力が含まれていれば，「1 か月後ではなく，その次の日の日付を表示しています」というコメントを書き込む．
- 入力値 8 27 に対しても，入力値 4 15 のときのようにパターンマッチを行っていく．

例えば**図 13** は，「一ヶ月後は 5 月 15 日です」と出力させないといけないところを「一ヶ月後は 5 月 16 日です」と出力するプログラムであり，これは図 11 の課題の意図に沿わないため，**図 14** のように間違いを知らせるコメントが生成される．

この機能の実装により，これまでに実装した採点支援機能よりも，柔軟かつ円滑な採点が期待できる．

なお，採点基準の設定が困難なもの，例えば出力が多いものやグラフィックスを出力結果とするもの，については，5.1 節の課題作成において模範解答のプログラムと入力を

与えると，それに対応する出力に一致するようなパターンマッチ行を自動生成して，それをもとに教員が修正を行える機能を実装する予定である．

6. まとめ

本稿では，Web ブラウザを用いてプログラミング学習が可能な環境 Bit Arrow における採点支援機能の実装と今後の拡張について述べた．

本稿執筆時までに実装が完了している出力判定機能では，完全一致による正解と学習者のプログラムの実行結果を比較することで採点支援を行っている．また，Bit Arrow で利用可能なグラフィックスを出力させるタイプの課題も，この機能による採点を可能としている．この機能を用いた授業実践は，来年度以降に執り行う予定である．

今後は採点支援機能の拡張として，課題作成や課題提出，ある程度自由度を持つ学習者のプログラムの採点を行う機能などを実装する．

謝辞 本研究は JSPS 科研費 17K00989 の助成を受けたものです．

参考文献

[1] 文部科学省：教育の情報化の推進 (online)，入手先 ⟨http://www.mext.go.jp/a_menu/shotou/zyouhou/index.htm⟩，(2018.02.20)．
[2] 長島 和平 , 堀越 将之 , 長 慎也 , 間辺 広樹 , 兼宗 進 , 並木 美太郎：プログラミング学習支援環境 Bit Arrow の教員支援機能の設計と試作，情報教育シンポジウム 2017 論文集， Vol. 2017， pp. 129-136 (2017)．
[3] 長 慎也 , 長島 和平 , 堀越 将之 , 兼宗 進 , 並木 美太郎：オンラインプログラミング環境 Bit Arrow を用いた C 言語プログラミングの授業実践，情報教育シンポジウム 2017 論文集， Vol. 2017， pp. 121-128 (2017)．