

競技プログラミングコンテストサイト Codeforces における プリプロセッサマクロの利用実態調査の試み

神取 大貴¹ 平野 学¹ 藤原 賢二^{1,a)}

概要: 競技プログラミングコンテスト参加者の中には、回答ソースコード作成時間を短くするためにマクロや関数などのコードスニペットを用意している者が存在する。本稿では、競技プログラミングコンテストの一つである Codeforces に提出された、C++ソースコード中のマクロに着目した3つのリサーチクエスチョンを立て、それらの調査手法を提案する。

1. はじめに

現在、多種多様なプログラミングコンテストが開催されており、コンテストの一形態として競技プログラミングコンテストが多く開催されている。競技プログラミングコンテストでは、与えられたいくつかの問題を制限時間内に他の参加者よりも速く解くことを競う。コンテストの多くはオンラインで開催されており、コンテストを取りまとめることを目的としたプログラミングコンテストサイトがいくつか存在する。代表的なサイトとして Codeforces や AtCoder などが存在し、これらのサイト間では問題の傾向や参加者のコミュニティなどの差異がある。

先に述べたとおり、競技プログラミングコンテストにおいては問題を閲覧し、内容を理解したうえで適切な回答ソースコードを作成するまでの時間が重要となる。そのため、回答ソースコードの作成時間をできる限り短くするために汎用的に使用可能なソースコードの断片（コードスニペット）をあらかじめ用意し、テンプレートとして活用している参加者が存在する。我々が実施した事前調査において、Codeforces の上位参加者からランダムに抽出した50名の回答ソースコードを対象に分析したところ、41名が何らかのコードスニペットを用いてコンテストに参加していることが確認できた。

コンテストの上位参加者らがどのようなコードスニペットを作成し、活用しているのかを明らかにすることは、初級者らがより多くの問題を効率良く解くための手助けになると考える。近年においては、プログラミング初学者ら

が自学自習のためにプログラミングコンテストを利用することもあり、競技プログラミングコンテスト初級者らの支援をする意義は大きいと考える。そこで、本研究では Codeforces におけるコンテスト参加者のコードスニペットについて調査するための手法を提案する。

具体的には、Codeforces を対象に、コンテスト参加者らがどのようなコードスニペットを活用しているのかについて定量的な調査を行う。Codeforces や AtCoder などでは、回答を作成するプログラミング言語を特に制限していない。本研究では Codeforces の回答データの約90%を占めるC++を分析対象とする [1]。C++を対象とする場合、コンテスト参加者らが使用するコードスニペットはプリプロセッサを用いたマクロと、関数に大きく分けられる。本研究ではそのうちマクロを分析の対象として以下の調査を行う。

- RQ1.** コンテスト参加者らはマクロをどの程度テンプレートとして利用しているか？
- RQ2.** コンテスト参加者らが利用しているマクロはどの程度効果があるのか？
- RQ3.** 上位コンテスト参加者らが利用している共通のマクロは存在するか？

本稿では、これらのリサーチクエスチョンに対する調査手法を提案する。

2. 調査対象

本研究では堤らが収集した Codeforces に提出された1,644,636件の回答データを調査対象とする [1]。このデータセットは2016年5月19日から2016年11月15日までに Codeforces に提出された回答データを含んでいる。Codeforces ではコンテストという単位で問題セットが作ら

¹ 豊田工業高等専門学校
National Institute of Technology, Toyota College, Eisei-cho
2-1, Toyota, Aichi, Japan
^{a)} fujiwara@toyota-ct.ac.jp

れており、堤らのデータセットからは、どのコンテストのどの問題に対して参加者が投稿したソースコードであるかを確認することができる。

3. プリプロセッサマクロに関する情報の収集

堤らのデータセットからは参加者が回答したソースコードを確認できるがマクロに関する直接的な情報は含まれていない。そのため、各回答ソースコードに対して構文解析を行い、マクロの定義と展開に関する情報を収集する。ここで、本研究で扱う C++ のプリプロセッサマクロについて説明する。プリプロセッサマクロは以下の構文からなる。

```
#define <識別子> <トークン列>  
#define <識別子>(<引数リスト>) <トークン列>
```

上の例で 1 行目の形式では単純に特定の識別子を別のトークン列に変換する。一方、2 行目の形式では引数を指定することができ、トークン列の一部を引数で与えたトークンに置き換えることができる。これらのマクロの定義と、マクロが実際に展開された箇所の情報を収集することで Codeforces におけるマクロの利用実態を調査する。

4. 調査手法

本節では 1 節で述べた 3 つのリサーチクエスチョンに対して、それぞれどのように調査を行うのかを説明する。

4.1 RQ1. コンテスト参加者らはマクロをどの程度テンプレートとして利用しているか？

テンプレートとして利用されているマクロは、マクロとして定義されても実際には展開されていないことがあると考えられる。そこで、ある参加者 u のソースコード s 中で定義されているが、一度も展開されていないマクロの割合を未使用マクロ率 $R_{u,s}$ として以下のように定義する。

$$R_{u,s} = 1 - \frac{|m_{u,s}|}{|M_{u,s}|}$$

ここで、 $M_{u,s}$ はある参加者 u が提出したソースコード s で定義されているマクロ、 $m_{u,s}$ は同様に展開されているマクロの集合を表す。各参加者が提出したソースコードから未使用マクロ率を算出し、Codeforces 参加者がマクロをどの程度テンプレートとして利用しているかを調査する。

4.2 RQ2. コンテスト参加者らが利用しているマクロはどの程度効果があるのか？

コンテスト参加者らがマクロをテンプレートとして利用する目的の一つとして、競技中のタイピング数を減らす目的があると考えられる。そこで RQ2 では、提出されたソースコードと、そのソースコードのマクロを展開した場合の文字数およびトークン数を比較することで、マクロによりど

の程度ソースコードを短くすることができるのかを調査する。なお、これらの値を計測する際にテンプレートとして定義されたマクロや関数を含めるのは適切ではないと考える。そこで、main 関数と実行時に呼び出される関数のみを対象として文字数およびトークン数を計測する。

4.3 RQ3. 上位コンテスト参加者らが利用している共通のマクロは存在するか？

RQ3 では、上位コンテスト参加者らのソースコード中に定義されているマクロを抽出し、参加者間で共通して定義されるマクロがあるかを調査する。そのために、共通のマクロを上位参加者の何割が使用しているのかを調査する。本研究では、異なる参加者が定義したマクロにおいて、識別子とトークン列が互いに一致するものを共通のマクロとして扱う。定義されたマクロに引数リストが存在する場合は、仮引数名の差異については扱わず、引数の個数が同一であれば共通のマクロとして扱う。

共通のマクロを集計する際、同一の参加者が同じトークン列に対して、提出したソースコード間で別々の識別子をマクロ名として与えている場合が考えられる。そのため、マクロをトークン列ごとに集計し、名前として設定された識別子の割合を計算する。例えば、ある参加者が全体で 100 件のソースコードを提出し、70 件のソースコードでは `#define fi first`、30 件のソースコードでは `#define X first` とマクロを定義していた場合を考える。この場合、参加者の定義したマクロのうち、トークン列 `first` に対する識別子の割合として、 fi が 0.7、 X が 0.3 となるように計算する。この計算を参加者ごとの提出したソースコードに対して行い、得られた割合を識別子、トークン列ごとに合計した値を集計する。そして、この集計結果が高いマクロが、共通のマクロであると考えられる。

5. おわりに

本稿では競技プログラミングコンテストサイトである Codeforces において、プリプロセッサマクロがどのように利用されているのかを調査するために、3 つのリサーチクエスチョンを設定し、それらの調査手法を提案した。今後は、実際に堤らのデータセットからマクロに関する情報を収集し、マクロの利用実態を調査する予定である。

謝辞 本研究の一部は JSPS 科研費 19K14337 の助成を受けた。また、多くの助言を頂いた名古屋大学の吉田則裕准教授と福岡工業大学の戸田航史准教授に深謝致します。

参考文献

- [1] 堤祥吾, 吉田則裕, 崔恩潯, 井上克郎: プログラミングコンテスト参加者を対象とした編集作業の特徴調査, 情報処理学会研究報告, Vol. 2017-SE-197, No. 6, pp. 1-8 (2017).