

ソースコードの振る舞い制約と 機能の理解を指向した学習支援環境の提案

Behavioral Constraints in Source Code and Proposal for a Learning Support Environment Oriented to Understanding Functionality

松為 泰生^{*1}, 前田 新太郎^{*1}, 古池 謙人^{*1}, 東本 崇仁^{*2}
Taiki MATSUI^{*1}, Shintaro MAEDA^{*1}, Kento KOIKE^{*1}, Takahito TOMOTO^{*2}

^{*1}東京工芸大学大学院工学研究科

^{*1}Graduate School of Engineering, Tokyo Polytechnic University

^{*2}東京工芸大学工学部

^{*2}Faculty of Engineering, Tokyo Polytechnic University

Email: m2265013@st.t-kougei.ac.jp

あらまし: プログラミング学習においてソースコードから変数の振る舞いを確認し、その振る舞いから機能を解釈する学習は、ソースコードと機能の関係を理解するうえで有効だと思われる。そのうえ、機能の解釈には、学習者に変数がどのように規則的に振る舞うか、すなわち振る舞いの制約について思考させることが求められる。そこで本研究では、ソースコード及び変数の初期値と最終値の結果を学習者に提示する。そして、振る舞いの制約とその振る舞いの制約からソースコードがどのような機能として解釈できるのかを学習者に思考させる手法を提案する。

キーワード: ソースコード, 振る舞い, 振る舞い制約, 機能

1. はじめに

プログラミングにおいてソースコードの機能を理解するには、ソースコードを1行ずつ理解するのではなく、意味のあるまとまりごとにソースコードを理解することが重要である。例として $c = a$, $a = b$, $b = c$ という3行のソースコードについて、ソースコードが持つ機能を考えてみる。このとき1行ずつ機能を解釈すると、各変数を代入していると解釈することができる。しかし、3行で機能を解釈すると変数 a と b の値を交換すると解釈することができる。

また、ソースコードから機能を解釈するには振る舞いを理解することが重要である。振る舞いとは、ソースコードの実行によって得られる状態遷移のことであり、本研究では特に変数に焦点を当て、変数の初期状態と最終状態の差分、つまり変数の初期値と最終値の差分として定義する。振る舞いを理解するには、ひとつのソースコードに対して複数のデータセット(変数の初期値の組み合わせ)を与え、データセットそれぞれの値の変化を確認する必要がある。例えば、`if (b < a), swap a and b` という2行のソースコードでは、 a に4の値、 b に3の値を入れたデータセット1と、 a に3の値、 b に4の値を入れたデータセット2ではソースコードは異なる振る舞いをする。データセット1では、入力値の a の値は b の値よりも大きいので、`if` 文の条件を満たし、変数 a の値と b の値は入れ替えられる。データセット2では、入力値の a の値は b の値よりも小さいので、`if` 文の条件を満たさず、最終値では a の値と b の値は変化しない。つまり、この一連のソースコードは、変数 a と b の値のうち大きい値を変数 b の最終値とし、小さい値を変数 a の最終値とすることになる。換言すれば、

変数 a と b は互いの数値により制約を受けた最終値になるといえる。このとき、変数 a と b を昇順に並べ替えるという機能として解釈できる。このように、学習者にソースコードが持つ機能を解釈させるには、変数がどのような制約で振る舞うか、振る舞いの制約(以下、振る舞い制約)を思考することが求められる。

学習者に振る舞い制約を思考させるには、複数の振る舞いを観察させ、ソースコードがどのような制約で振る舞うか理解させる必要がある。そこでソースコード以外の物質において振る舞いと機能の概念を深く追求し、振る舞いモデルから機能モデルを表現するために笹島ら⁽¹⁾が開発した Function and Behavior Representation Language(以下、FBRL)を活用する。FBRLは機能モデルを表現するために Functional Topping(以下、FT)を用いて振る舞いモデルを解釈する。このFTにある解釈時に注目するポートの「P-Focus」の概念を活用することで学習者は複数のケースから初期値と最終値にはどのような規則性があるのか理解し、振る舞い制約を思考することができる。と考える。

本研究では、学習者がソースコードの振る舞いから振る舞い制約を思考させ、振る舞い制約からソースコードの機能を解釈する活動を提案する。

2. 振る舞いと機能の関係性

笹島らは振る舞いを「時間とともに変化する対象の状態遷移であり、利用者が必要とする対象のシミュレーションの結果」と定義した。また、対象の状態を表すパラメータの集合や、その間に成り立つ制約式などを振る舞いモデルと呼んでいる。ある部品をシステムに組み込む場合、その部品に対して意図された望ましい状態があり、これが目標である。その「目

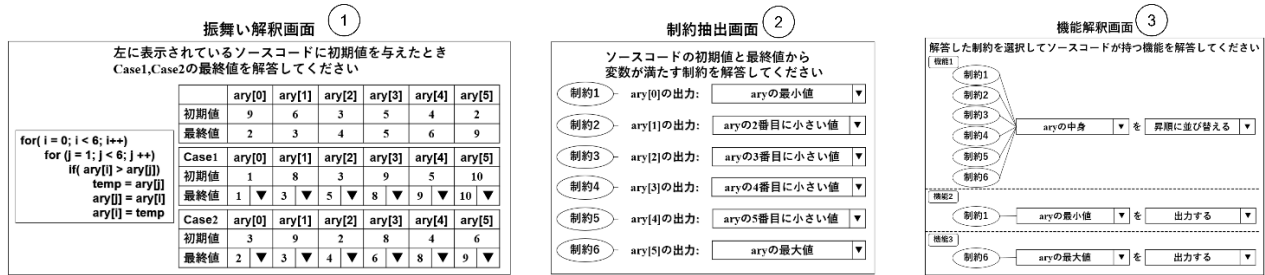


図1 提案手法をシステム化したイメージ図

標のもとで振る舞いを解釈した結果」を機能と定義している。さらに、振舞いモデルを解釈するためのFTを提案し、その一部として解釈時に注目するポート(注目する入出力)を定義している。したがって、ある部品がどのような機能を果たしているかを解釈するためには、ある振舞いにより引き起こされるパラメータ変化時に成立する制約が重要であり、どの入出力関係に着目するかが重要となる。

古池ら^(2,3)はプログラミングにおいて、一連のソースコードを「部品」として定義し、プログラミングにおける変数を含む状態の遷移を「振る舞い」、そしてその状態遷移を目標の下で解釈したものを「機能」とした。しかし、古池らは振舞いがもつ制約や、FTに基づく機能解釈については言及していない。

そこで、本研究では振舞いの制約や P-Focus の概念を導入した学習方法を提案する。

3. 提案システム

本提案システムについて図1を用いて説明する。学習者は①, ②, ③の画面の順に学習する。

まず①振舞い解釈画面から説明する。この画面では学習者にソースコードを提示し、ソースコード内の変数に初期値を与え、学習者に実行した結果である最終値を複数のケースで解答させる。つまり、この画面から、学習者はソースコードの振舞いを確認することができる。図1では学習者に提示しているソースコードとして昇順に並び替える機能を持つソースコードを表示し、ソースコード内で扱っている変数に初期値を与えて複数のケースを表示している。学習者はそれぞれのケースに対するソースコードの初期値に対して最終値を解答する。例として if 文, swap 文が表示され、if 文の条件が(a > b)の場合、与える初期値が b より a の方が小さいので、if 文の条件を満たさない。したがって if 文の条件を満たしたときとは違った最終値になる。以上のことから、学習者は複数のケースの振舞いを解答する。

次に②制約抽出画面について説明する。この画面では学習者が振舞い解釈画面で解答したソースコードの振舞いからソースコードの振舞い制約を解答する。図1では、ary[0]から ary[5]の振舞い解釈画面で解答した各変数の振舞い制約を解答する。また、バブルソート内の変数 ary[0]の初期値と最終値から解

答する制約として「ary の最小値」、ary[5]では「ary の最大値」とそれぞれの変数が持つ振舞い制約を解答することができる。

最後に③機能解釈画面について説明する。この画面では、学習者は制約抽出画面で解答した振舞い制約からソースコードの機能を解釈する画面である。図1では制約抽出画面で解答した制約から解答した制約全てを使用して「ary の中身を昇順に並び替える」という機能を解釈している。また、制約1の ary[0]の出力のみを使用して「ary の最小値を出力する」、制約6の ary[5]の出力のみを使用して「ary の最大値を出力する」という機能も解釈している。このように入出力に注目させた機能解釈を促している。

4. おわりに

本稿ではソースコードの振舞いから振舞い制約を思考させ、振舞い制約から機能を解釈する支援をする手法の提案を行った。

今後の課題として提案手法にもとづいてシステムを開発し、開発したシステムは学習効果があるのか評価を行う。また、今回は初期値と最終値のみで表現できる範囲を扱っているが、本手法が print 文などほかの範囲も扱うことができるか検討を行うことがあげられる。

謝辞

本研究の一部は JSPS 科研費 JP19H04227, JP21H03565, JP22K12322 の助成による。

参考文献

- (1) 笹島宗彦, 來村徳信, 池田満, 溝口理一郎, “機能と振舞いのオントロジーに基づく機能モデル表現言語 FBRL の開発”, 人工知能学会誌, vol.11, no.3, pp.420-431 (1996)
- (2) 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: “プログラミングの構造的理解を指向した部品の段階的拡張手法の提案と支援システムの開発・評価”, 教育システム情報学会誌, Vol. 36, No. 3, pp. 190-202 (2019)
- (3) 古池謙人, 東本崇仁, 堀口知也, 平嶋宗: “プログラミング学習における再利用性を指向した知識組織化のための知的支援: 機能・振舞い・構造の観点に基づく問題解決過程のモデル”, 人工知能学会論文誌, Vol. 35, No. 5, pp. C-J82_1-17 (2020)