

木構造プログラミングシステムの試作

都倉信樹(大阪電気通信大学)

ある教育用プログラミングシステムの試作報告である。このシステムは

1. プログラムを木構造として、入力・編集・保存する。
2. 編集系、実行系に加えて、代書系(DNCL, Python等のプログラムを生成する)を作る。

アルゴリズムから具体的プログラムへの橋渡しの教育実践での利用が期待できる。

1. はじめに

プログラミング教育に関して、多くのシステムが作られ、教育実践も本格化しつつある。本研究は中学高校から大学へのアルゴリズムやプログラミング教育の支援を目指しており、プログラムを木構造と捉えて扱う。

2. 木構造の編集

木構造は親ノードのもとに子ノードがいくつかぶら下がるという再帰的な構造である。試作ではWindows等のファイルディレクトリの表示に使われるTreeViewを使っている。ノードを1つずつ木に加えて行ってプログラムを作れるが、言語機能ごとに部分木を追加するという形で編集するのが楽である。編集時の状況を図1に示す。

たとえば、while という構成要素を選択すると、それを表す部分木が、選択されたノードに「接ぎ木」される。whileの継続条件は、別途右のテキストボックスの式のところに記入して設定する。こうして、言語の提供する機能に対応する部分木を接ぎ木して、名前や式、その他の情報を設定してプログラムを完成する。出来た木構造プログラムは、簡単な木表現法[2]で外部テキストファイルに保存したり、取り出して木構造を再現出来る。

3. 式と制御構造の分離

while(E) do S のように本来、式と実行文は複合しているが、式はひとつのテキストとして扱い、式以外は木の形に入力する。そしてそれぞれの構成要素の特性に合わせて順次解釈実行される。再帰アルゴリズムなどを

学習する必要から、スタックによる実現を可視化する。

4. 言語テンプレート

実はこの処理系は特定の言語に固着していない。言語テンプレートに、言語の構成要素や親子関係をやはり木構造で別に定義する。システム起動時に選択されたテンプレートで、扱う言語構造機能が決まる。このテンプレートを作ることは言語の設計・定義に相当し、プログラム言語の設計の教育にも使えるが、これは大学レベルになろう。

式は具体構文を要求するが、制御構造は抽象構文レベルでの記述になり、細かい構文より実質的内容:意味に意識を近づけたい。

5. 木構造プログラミングの発想

このシステムの発想は[1]に遡る。当時ソフトウェア工学で保守の困難が大きな問題とされていたので、プログラムとドキュメントの乖離をいかに縮めるかという意識で、木構造でプログラムを作り、必要なノードに説明を書き加える方法を提案した。その考えはここでも継承している。当時も木プログラムから具体構文にしたがうプログラムを生成する「代書系」の構想をもっていた。テンプレートの中に具体的構文への変換法を記述することで、木構造プログラムから、具体的言語でのプログラムを生成する。いくつかの代書系を検討する事で、言語の違いがどう見えてくるかが興味がある。

6. 編集時の画面例

図1は試作システムのプログラム編集時の画面例である。メニューの下はシステムからの通知で、その下に4つのタブページがある。

"An Educational Programming System Using Tree Representation" by Nobuki TOKURA (Osaka Electro-Communication University)

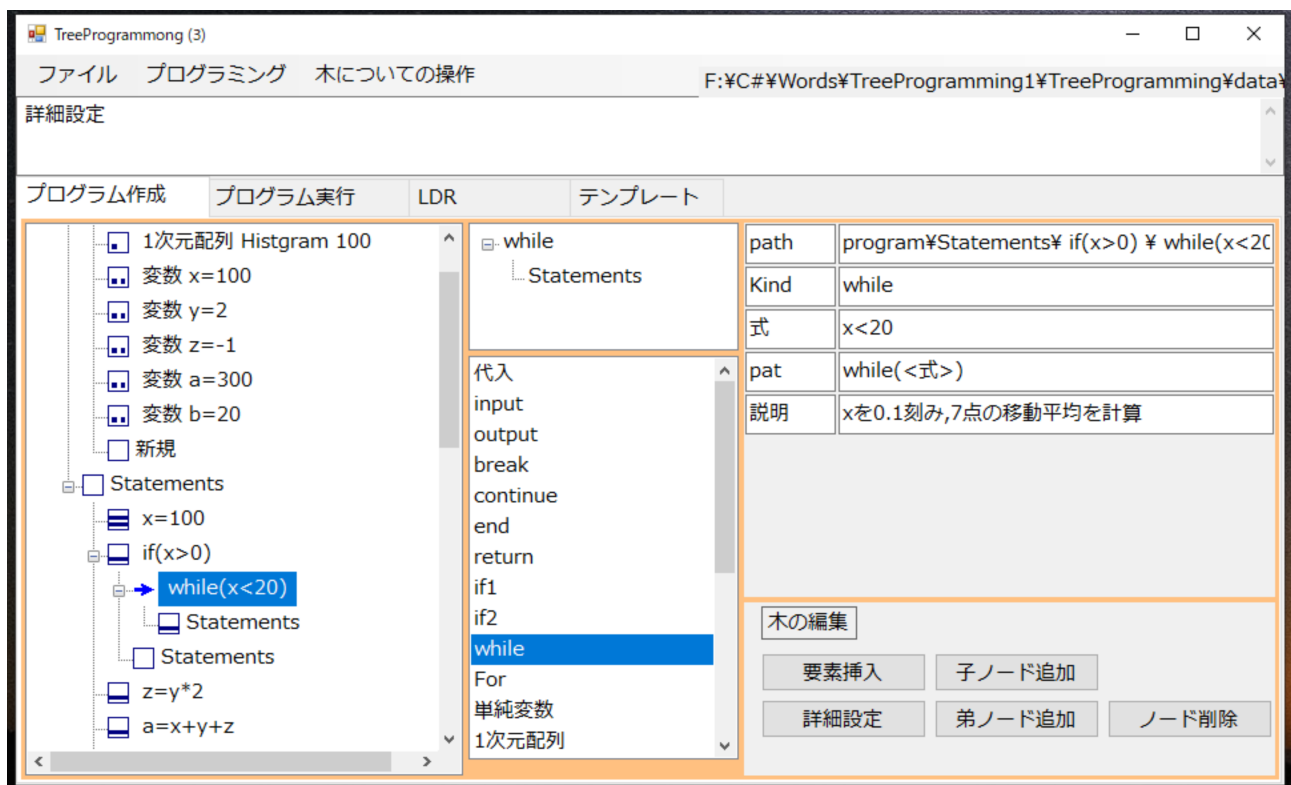


図1 試作システムの編集時の画面例 (while を挿入後, 詳細設定をしようとしている)

「プログラム作成」タブページはL,C,R3つの領域に分割され, 左のLが編集中のプログラム木であり, 中央Cは上下にペインC1,C2に分かれる. C2に挿入可能な構成要素のリストを表示する. その1つを選択すると, その木構造がC1に表示される. Rの上ペインR1には選択中のノードの詳細情報を表示する. Rの下ペインR2には, 編集用のボタンがある.

「要素挿入」ボタンをクリックすると, Lの選択されたノード(青い→)にR1の木が「接ぎ木」される. 接ぎ木された部分木の詳細な指定は, R1にノードの詳細情報を用意し, R2で「詳細設定」をクリックすると設定できる. これを繰り返してプログラムを完成する.

現在のテンプレートでは, path に現在見ているノードのフルパスを, Kind に構成要素の種別を表す名前 while を表示している. 「式」では, 繰返ししの継続条件の式をテキストとして記入する. pat には, どのような形でノードを表示するかを指定している. pat など開発者向けの情報を試作段階では表示しているが, 必要がないときは隠す. 「説明」はこの構成要素で「何を行うか」を記入し, のちに作り込むコードとの整合性をとることを求め

ている. 他にテンプレートには, patに似た記述で, 代書の仕方を示すなど, いくつかの記述を行う. テンプレートも木構造で表現しており, プログラム同様に編集できる.

7. 実行はインタプリティブに行う

スタックを使い, 式は逆ポーランド記法に変換し, 評価する. 関数呼び出しもスタックで処理する. デバッグ用に実行ログを生成する. テンプレートにループ不変式の記述などを含める可能性もある.

現状は実行までの基本機能がテストできた段階であり, さらなる作り込みが必要である.

引用文献

- [1]Y.Nakamoto, M.Iwamoto, M.Hori, K.Hagihara and N. Tokura: An Editor for Documentation in Π -system to Support Software Development and Maintenance, 6th Int'l Conf. on Software Eng. 330-339, 1982.
- [2]都倉:ある単純な木表現法の応用について, FIT2019, 6D-7 (Sept. 2019).