

タンジブルなプログラミングツール Logi-CUBE への推論機能の実装と評価

Implementation and Evaluation of Inference Function in the Tangible Programming Tool Logi-CUBE

宮崎 勇希^{*1}, 本吉 達郎^{*2}, 布施 陽太郎^{*2}, ミヤグマルドラム ビルグウンマ^{*2}, 澤井 圭^{*2},
 増田 寛之^{*2}, 高野 博史^{*2}, 高木 昇^{*2},
 Yuki MIYAZAKI^{*1}, Tatsuo MOTOYOSHI^{*2}, Yotaro FUSE^{*2}, Bilguunmaa MYAGMARDULAM^{*2}, Kei SAWAI^{*2},
 Hiroyuki MASUTA^{*2}, Hironobu TAKANO^{*2}, Noboru TAKAGI^{*2}

^{*1} 富山県立大学大学院工学研究科

^{*1} Graduate School of Engineering, Toyama Prefectural University

^{*2} 富山県立大学情報工学部

^{*2} Faculty of Information Engineering, Toyama Prefectural University

Email: u454024@st.pu-toyama.ac.jp

あらまし：論理型プログラミング言語は全体構造の把握が難しく扱いにくいいため、ユーザが学習しやすいアプリケーションがいくつか開発されている。しかし、情報機器の操作リテラシーが低い初学者や視覚障がい者の使用を想定したタンジブルなツールは見受けられない。そこで、論理型プログラミングの学習が可能なタンジブルなツール Logi-CUBE の試作機を作成した。本稿では試作機への推論機能の実装、およびその評価について報告する。

キーワード：論理的思考力, タンジブルインタフェース, 情報教育

1. はじめに

近年、初等中等教育での情報教育の機会が増加しており、論理的思考力もその1つとして挙げられる。

一般的に、ユーザにとってプログラムを識別しやすい教材には、Tangible User Interface が用いられる。しかし、論理型プログラミング言語を用いた学習ツールにおいて、情報機器の操作リテラシーが高くない初学者や視覚障がい者の使用を想定したタンジブルなツールは見受けられない。よって、本研究の目的を、論理型プログラミング言語の学習が可能なタンジブルなツール Logi-CUBE の開発とした。タンジブルなツールとして、P-CUB シリーズ⁽¹⁾を基盤として開発に取り組む。

本稿では、Logi-CUBE の試作機への推論機能の実装、およびその評価について報告する。

2. Logi-CUBE

Logi-CUBE は、プログラミングブロックをプログラムマットに並べることによって論理型プログラムを構築し、質問に対する回答や知識を音声で出力できる。本研究では、Prolog⁽²⁾を参考にしてインタフェースを作成した。

2.1 システム概要

Fig.1 に Logi-CUBE の試作機のシステム構成を示す。Logi-CUBE は知識・質問の入力を行うプログラムマット、述語・変数の情報が登録されているプログラミングブロック、ブロック配置情報を転送するための転送用スイッチ、スピーカ、および制御用 PC で構成されている。

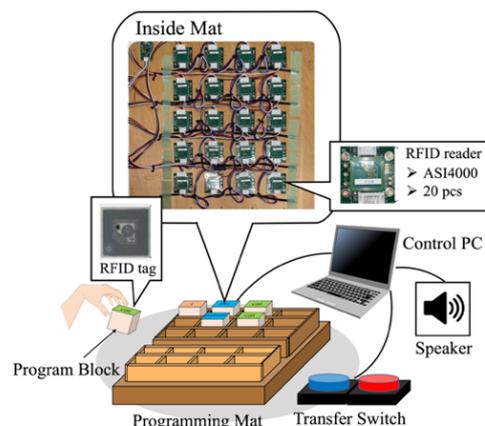


Fig. 1 System Configuration of Logi-CUBE

2.2 制御システム

制御システムは「RFID 制御による知識・質問認識システム」(以下 RCS)、「データベース制御システム」(以下 DCS) によって構成されている。制御システムの構成を Fig.2 に示す。RCS は RFID タグからブロックの配置状況を取得し、txt 形式で保存する。DCS は txt 形式で保存された知識に対して推論規則を適用し、新たな知識を作成したうえで、回答生成プログラムに送信する。質問に対しては知識と比較を行い、回答を生成・音声出力する。

3. Logi-CUBE に実装した機能

Logi-CUBE に実装した機能は、推論機能と検索機能の2種類である。

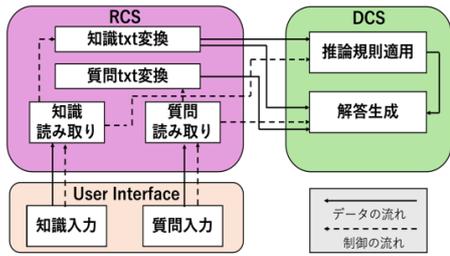


Fig. 2 System Configuration of Software

3.1 推論機能を用いたプログラム

プログラムマットの知識部の1行目に「わたし」「すき」「どうぶつ」、2行目に「いぬ」「いこーる」「どうぶつ」、質問部に「わたし」「すき」「いぬ」を配置するプログラムの場合、「すき」「いこーる」は述語ブロック、「わたし」「どうぶつ」「いぬ」は変数ブロックである。Fig.3にブロックの配置情報、知識・質問内容、および推論規則を示す。

プログラムマットの同行に配置された述語ブロックと変数ブロックは1つの知識・質問として認識され、知識「私は動物が好きです」「犬は動物です」、質問「私は犬が好きですか」とRCSで認識されデータベース(以下DB)に転送される。DBに蓄積された知識を推論規則「xはyが好きです、かつzはyです、ならばxはzが好きです」を適用し、新たな知識「私は犬が好きです」を生成する。質問に適合する知識がDBに存在し、知識と質問の内容が一致するため、DCSは回答「はい」を出力する。

3.2 検索機能を用いたプログラム

3.1節で説明したブロック配置に加え、質問部の2行目に「わたし」「すき」「はてな」を加えるプログラムの場合、知識は「私は動物が好きです」「犬は動物です」「私は犬が好きです」である。質問は「私は何が好きですか」とRCSで認識されDBに転送される。このとき、質問部の1行目は考慮されない。DCSは、知識と質問を比較し、「?」と一致する変数が存在するため、回答「動物」「犬」を出力する。

4. システム評価および結果

ブロックを配置した際、RFIDリーダがRFIDタグの情報を取得できているかを検証した。また、所望のスイッチを押した際、RFIDシステムと制御システムが連携し、3章の機能が実装されているかを検証した。

ブロックを配置した際、RFIDリーダがRFIDタグの情報を取得していることが確認できた。知識転送スイッチの入力があった場合、制御システムはRCSを起動し、ブロック情報をtxt形式で保存した。その後、制御システムはDCSを起動し、知識を推論規則に適用させ新たな知識を作成した。質問転送スイッチの入力があった場合、知識転送スイッチと同様に質問内容をtxt形式で保存した。制御システムはDCS

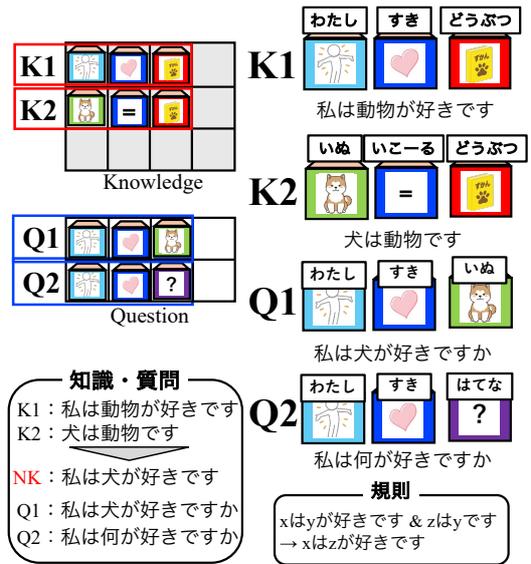


Fig. 3 Block Placement

を起動し、質問と知識の比較を行い、回答を生成した。Fig.4にシステム評価結果を示す。

ブロック配置情報を取得

ID	名前	状態	位置	時刻	名前
R00	E00401509E403994	OK	---	15:55:24.455	i
R01	E00401509E418A68	OK	---	15:55:24.458	like
R02	E00401509E403103	OK	---	15:55:24.477	cat
R03	E00401509E4038A6	OK	---	15:55:24.568	cat
R04	E00401509E7766F8	OK	---	15:55:24.564	is_are
R05	E00401509E405404	OK	---	15:55:24.585	dog

出力結果

```

Prolog Outp...
Question_1: like(i,dog)
Question_2:
Knowledge_1: like(i,cat)
Knowledge_2: is_are(cat,dog)
Knowledge_3:
New Knowledge: like(i,dog)
No: like(i,cat) and like(i,dog) don't match.
No: is_are(cat,dog) and like(i,dog) don't match.
Yes: like(i,dog) and like(i,dog) match.
No: and like(i,dog) don't match.
    
```

新たな知識

回答

ブロック配置

Fig. 4 System Evaluation Results

5. まとめ

本研究では、論理型プログラミング言語の学習が可能なタンジブルなツール Logi-CUBE の開発を行い、試作機への推論機能を実装、およびその評価に取り組んだ。Logi-CUBEに実装した2種類の機能が所望の動きを行っていることが確認できた。

今後は、試作機を用いた評価実験、インタフェースデザインの選定を行う。

参考文献

- (1) 本吉達郎, 掛橋駿, 小柳健一, 大島徹, 増田寛之, 川上浩司, ブロック型プログラミングツール P-CUBE の学習初期段階における有用性の検証, 日本知能情報ファジィ学会誌, Vol. 27, No. 6, pp.909-920, (2015).
- (2) 小川東:「Prolog による論理プログラミング入門」, 啓学出版, pp.2-20, 1990.