3G3-4

合成に関して閉じているトピックマップ問い合わせ言語の提案と実装
A Topic Maps query language closed under composition

小暮 陽太[*1] 篠埜 功[*2] 木村 昌臣[*2]
Yota Kogure Isao Sasano Masaomi Kimura

[*1] 芝浦工業大学大学院 [*2] 芝浦工業大学
Graduate School, Shibaura Institute of Technology Shibaura Institute of Technology

Topic Maps is a standardized data model for the representation and interchange of knowledge, with an emphasis on the findability of information. Although we need query language to retrieve data from Topic Maps, there is no Topic Maps query language closed under composition. In this paper, we propose a topic maps query language closed under composition.

1. はじめに

近年, 我々は膨大な量の情報を扱い, かつその量は爆発的に増加を続けている. そのため, 必要な情報の発見が困難になるという問題がある. そこで, 情報の間や情報リソースなどを関係に基づいて結びつけたリンク構造によって, 情報を効率的に管理するトピックマップ技術が注目を集めている[1][2]. トピックマップは主に任意の概念を指す「トピック」, トピック間の関係を表す「関連」, トピック間の関連におけるトピックの役割を表す「関連役割」, そしてトピックと Web 等の情報リソースをしめす「出現」によって構成され, これらの構成要素をトピックマップ構成物と呼ぶ. トピックマップではトピック間を関連や関連役割で結びつけることでリンク構造を形成している.

トピックマップのデータモデルに基づいて構成されたデータをデータベースとみなしたとき, 求める情報を的確に得るための問い合わせ言語が必要であるが, この言語仕様はいまだに正式に決まっていない. 一方, 現在多く使われているトピックマップ問い合わせ言語として tolog があげられる[3]. tolog はトピックマップに対する問い合わせ結果を表構造として出力する問い合わせ言語である. 問い合わせ結果が表構造であるため, tolog によって得られた問い合わせ結果は再度 tolog による問い合わせの入力として用いることができない. すなわち, 問い合わせをあるトピックマップを変換する写像とみなしたとき, tolog は合成において閉じていない. しかし, 問い合わせ言語において入出力のデータモデルが統一されていることは, より複雑な問い合わせを行うための重要な要件である.

そこで本研究では, トピックマップのデータモデルに基づいて構成されたデータに対する, 合成に関して閉じているトピックマップ問い合わせ言語の提案と実装を行った.

2. 提案言語の概要

合成に関して閉じている問い合わせ言語であるために, 提案言語の出力はトピックマップのデータモデルに従っている必要がある. また, トピックマップ構成物の名前や構成物同士のリンク構造を指定して問い合わせができる必要がある. そこで本研究では提案言語を, トピックマップ構成物を表す式, 関数宣言, トピックマップ構成物やそのつながりを指定する条件式などをもとに構成される, トピックマップを表す式として定義する.

連絡先: 小暮 陽太, 芝浦工業大学大学院理工学研究科, m110054@shibaura-it.ac.jp

2.1 トピックマップ構成物を表す式

提案言語では指定された名前や型を持つトピックマップ構成物を *item* で表す. **topic (** *name* **)**は指定された名前や型を持つトピック, **role (** *type* **)**は指定された型を持つ関連役割, **assoc (** *type* **)**は指定された型を持つ関連をそれぞれ表す. このとき名前や型として_が指定されたとき, 指定された種類のすべてのトピックマップ構成物を表す. *item* とそれを表す変数を *node* および *pnode* で表現する. *pnode* 式での *v@item* は *item* であらわされたトピックマップ構成物を指定された変数 *v* に束縛する.

2.2 関数宣言

提案言語では **let** 式の中で関数を宣言できる. *decl* はひとつの関数宣言を表し, *decls* は関数宣言の並びを表す. 関数宣言は関数名, 引数となる *node*, および関数の実行結果として返されるトピックマップを表す式からなる.

2.3 条件式

関数宣言では条件式によってユーザの指定するトピックマップを返す問い合わせを可能としている. 条件式はひとつの条件を表す *cond*, 条件の並びの論理積を表す *conds* で構成される. *node* 間を~でつないで表すことでトピックマップ内のリンク構造を指定することができる.

2.4 トピックマップを表す式

提案言語ではトピックマップを式 *e* で表す. *v* は変数に束縛されたトピックマップを, *node* (~ *node*)+は指定されたトピックマップ構成物の間をつないだトピックマップを表す. *f* **(** [*node* **(,** *node*)*] **)**は宣言された関数の適用によって得られるトピックマップを, **let** *decls* **in** *e* **end** は *decls* で宣言された関数の適用を含む式 *e* によって表されるトピックマップを表す. *file* はそのファイルパスで指定されるファイルが持つトピックマップを, **IN** e_1 e_2はe_1の表すトピックマップを対象とするe_2の問い合わせを行った結果としてのトピックマップをそれぞれ表す.

2.5 提案言語の拡張 BNF 記法による表記

提案言語は以下の拡張 BNF 記法で定義される式 *e* である.

e ::= **let** *decls* **in** *e* **end;**
| *v;*
| *node* (~ *node*)+;
| *f* **(** [*node* **(,** *node*)*] **);**
| *file* ;
| **IN** *e e*;

decls ::= *decl decls;* | ε;
decl ::= **function** *f*([*pnode* (**,** *pnode*)*] **)** = *e* [**where** *conds*];
conds ::= cond [**AND** conds];
cond ::= *pnode*; | *path* ~ *path*;
node ::= *v*; | *item;*
pnode ::= *v* [**@** *item*]; | *item;*
item ::= **topic (** *name* **);**
| **role (** *type* **);**
| **assoc (** *type* **);**
name ::= *string* **:** *string*; | *string* **: _**;
| **_ :** *string*; | **_ : _**;
type ::= *string*; | **_**;

3. 提案言語を実装したプロトタイプの作成

提案言語を実装したプロトタイプシステムを作成した. プロトタイプはトピックマップデータベース管理システムである TOME[4] 上で動作し, 問い合わせに対してトピックマップを出力する.

また, ネットワーク可視化ソフトである Pajek の形式のデータを出力することで結果のトピックマップを可視化する機能を持つ.

4. 実験・考察

本実験では図 1 に示すトピック数 174 個, 関連数 864 個のポケモンについてのトピックマップを対象として用いる. なお, 本稿ではトピックマップを表すグラフにおいてトピックを黒のノード, 関連を赤のノード, 関連役割をエッジで表現する.

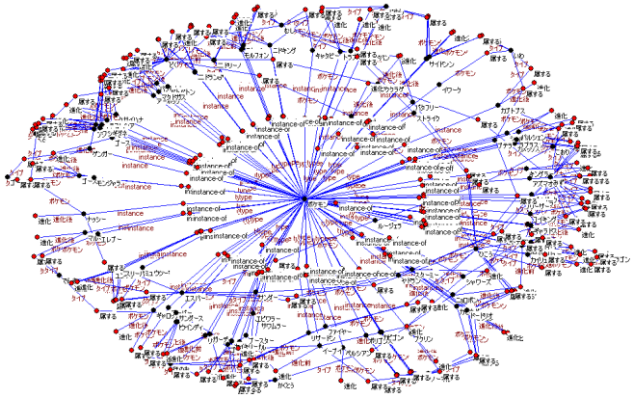

図 1 ポケモントピックマップ

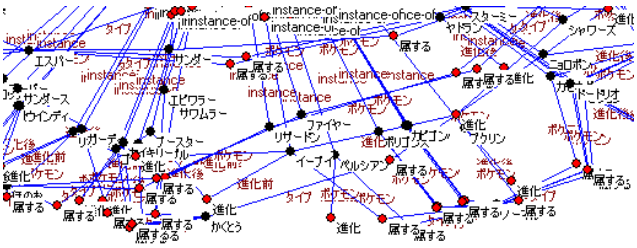

図 2 トピック「イーブイ」を中心に拡大したポケモントピックマップ

図 2 のように, このトピックマップは複数の種類の関連を持つため一見してトピック間の関係の把握が困難である. そこで本実験ではまず対象のトピックマップに対し"進化"という関連のみでつながれたトピックを取り出す以下のような問い合わせを行った.

IN pokemonTM
let function *f*()=v_1 ~ v_2 ~ v_3 ~ v_4 ~ v_5
where v_1**@topic**(_:'ポケモン') ~ v_2**@role**(_)
~ v_3**@assoc**('進化') ~ v_4**@role**(_) ~ v_5**@topic**(_:'ポケモン')
in *f*() **end**

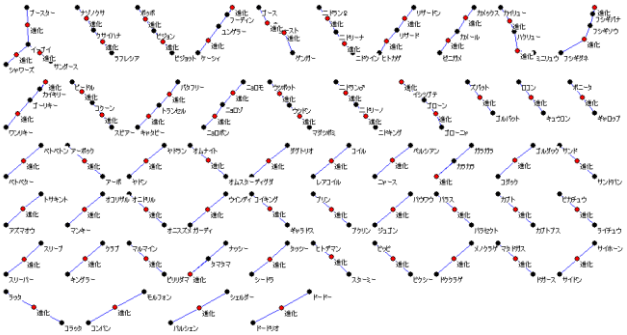

図 3 出力結果

図 4 トピック「イーブイ」を中心に拡大した出力結果

対象のポケモントピックマップに対して前述の問い合わせを行ったところ図 3 のような部分トピックマップを得た. また, 図 4 に示すように元のトピックマップで関係の把握が困難だったトピックについても明瞭に関係を把握可能であった.

またこの図 3 の示す, 上記の問い合わせ結果として得られたトピックマップに対し進化先が存在するポケモンのみからなるトピックマップを得る以下の問い合わせを行った.

IN *f*()
let function *g*()=v_1
where v_1**@topic**(_:'ポケモン') ~ **role**('進化前')
~ **assoc**('進化') ~ **role**(_) ~ **topic**(_:'ポケモン')
in *g*() **end**

この問い合わせにより, 進化先を持ち型がポケモンである 70 のトピックからなるトピックマップを得た.

このように提案言語は合成に関して閉じており, tolog ではできない問い合わせが可能である.

5. おわりに

本研究では, トピックマップに対する問い合わせ結果に対して再度問い合わせが可能な, 合成に関して閉じているトピックマップ問い合わせ言語の提案および実装を行った. また, 提案言語が既存言語ではできない問い合わせが可能なことを具体例を用いて示した.

今後の展望として, 複数のトピックマップの和集合をとる構文などを追加し, 提案言語の拡張を行う. また大規模なトピックマップへの適用し, 提案言語の有効性や効率について検証する.

参考文献

[1] ISO/IEC 13250-2:Information Technology – Topic Maps – Part 2 : Data Model. International Organization for Standardization, 2006.
[2] 内藤求: トピックマップ入門, 東京電機大学出版局, 2006
[3] Ontpia: tolog – A Topic Maps Query Language, Charting the Topic Maps Research and Applications Landscape Lecture Notes in Computer Science, Volume 3873/2006.
[4] 栗原優樹, 木村昌臣: トピックマップデータベースへの問い合わせ最適化手法の検討(第 2 報), 電気情報通信学会 2011 総合大会, 2011