

Linked Data を用いた理論グリッドの構築 -公理的集合論の形式理論フラグメントを用いた事例研究

佐久間 亜都武[†] 後藤 祐一[†] 高 宏彪[†] 程 京徳[†]

埼玉大学 理工学研究科[†]

1. はじめに

自動定理発見は計算機に定理を自動的に証明させるのではなく、計算機に未知の定理を自動的に発見させることである。自動定理発見手法のためのインフラストラクチャとして、理論グリッドおよびその理論的基礎が提案された[3]。理論グリッドはさまざまな論理体系に基づく、さまざまな形式理論の部分集合群を組織化し、管理する環境である。理論グリッドは管理している形式理論の部分集合を利用するためのプロトコルとインターフェースを、自動定理発見の試行者および応用プログラムに提供する。しかし、理論グリッドは未だ構築されていない。

本研究では、データを Web 上で相互にリンク付けして公開できる仕組みである Linked Data を用いた理論グリッドの構築法を提案した。また、公理的集合論の形式理論の部分集合を格納する理論グリッドのプロトタイプを構築した。

2. 前向き推論を用いた自動定理発見と理論グリッド

自動定理発見問題を解決するために、前向き推論を用いた自動定理発見手法が提案され、研究されている[1]。前向き推論とは与えられた前提から新たな結論を導出するプロセスのことである。計算機に既知の前提および導出された結論を与え、推論規則を繰り返し適用させることで、前向き推論を用いた自動定理発見を行うことができる。提案された自動定理発見手法では、推論規則、前提となる論理式の集合、処理の終了条件を入力として受け取り、終了条件を満たすまで、前提および前提から導出される論理式に推論規則を繰り返し適用し続けるプログラムである、前向き推論エンジンが使われている。自動定理発見に用いる前向き推論エンジンとして、汎用前向き推論エンジン FreeEnCal[2]が提案され、開発されており、FreeEnCal は、論理式、推論規則、論理結合子や様相演算子の入れ子の

度合い、除去規則と呼ばれる二重否定などの冗長な表現を省略するための規則を入力として受け取り、論理定理フラグメントと形式理論フラグメントを出力する。

論理定理フラグメントとはある論理体系の論理定理の集合の部分集合である。また、ある論理体系に基づいてある分野で成り立つ定理(経験定理)から導出された定理の集合を形式理論という。形式理論フラグメントは形式理論の部分集合である。前向き推論を用いた自動定理発見においては、論理結合子の入れ子の度合いを用いて両フラグメントを生成している。両フラグメントの定義については文献[2]を参照のこと。

前向き推論を用いた自動定理発見では、前提が有限個でも無限に結論が導出される可能性があるため、定理の導出は漸進的に行われる。論理定理フラグメントを例に挙げると、論理結合子の入れ子の度合いの小さいものから導出し、導出に用いる入れ子の度合いを徐々に大きくしていく。そのため自動定理発見の試行結果間には包含関係が存在する。

理論グリッドは、様々な論理体系に基づく様々な形式理論を組織化し、管理する環境である[1]。自動定理発見の試行者は理論グリッドから前向き推論の前提となる定理の集合を受け取り、導出された定理の集合を理論グリッドに格納する。理論グリッドは前向き推論の出力結果を包含関係に基づいて体系的に格納や検索ができる必要がある。また理論グリッドでは分散的に自動定理発見を行える環境が想定されている。そのため、各分野ごとに行われている自動定理発見の結果を、新たに自動定理発見を始める人が利用できる必要がある。

3. Linked Data

Linked Data は、構造化されたデータを Web 上で相互にリンク付けして、それらを公開できる仕組みを提供する実践的方法である。Linked Data はメタデータの表現方法の枠組みである Resource Description Framework (RDF)を用いて実現される。RDF ではリソースの関係を主語、述語、目的語の三つ組で表すことができ、データ公開者は述語を用いて主語と目的語の関係を

Construction of a Theory Grid with Linked Data - A Case Study in Axiomatic Set Theory

[†]Atomu SAKUMA, Yuichi GOTO, Hongbiao GAO, and Jingde CHENG, Saitama University

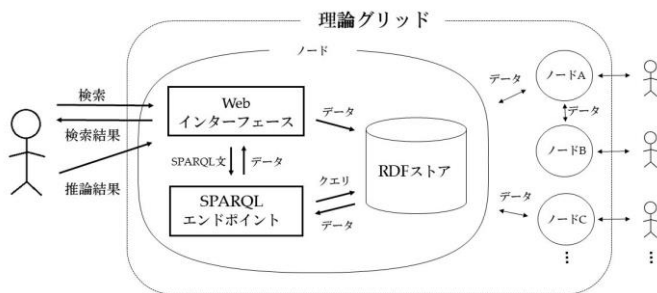


図1:理論グリッドと利用者の関係

明示することができる[4]. また、RDF データを検索するために、RDF クエリ言語 SPARQL が広く用いられている. SPARQL を用いることで RDF ストアと呼ばれる RDF 専用データベースから欲しいデータを取得することができる.

4. Linked Data を用いた理論グリッド

理論グリッドでは、自動定理発見が分散的に行われ、利用者がネットワークを通じてそれぞれが必要となるデータにアクセスすることが想定されている. Web 上で構造化データを簡易に扱える仕組みである Linked Data は、理論グリッドの想定利用法に合っている.

Linked Data を用いた理論グリッドと理論グリッドの利用者の関係を図1に示した. 利用者が直接 SPARQL を使用するのとは難しいため、Web インターフェースで SPARQL 文を生成して SPARQL エンドポイント越しに RDF ストアのデータを検索する. 理論グリッドには複数のノードが存在し、各ノードにはある分野の既知の定理の集合が包含関係に基づき格納され、ノード間の結び付きは分野間の包含関係に基づき構成される. データの共有をするために、各ノード間では格納されたデータのやり取りを行う. また、データを格納する際は自動定理発見のデータを Web インターフェースで RDF のデータへと変換した後 RDF ストアに格納される.

Linked Data ではデータの識別子として URI を使用することになっている. Linked Data を用いた理論グリッドでは、形式理論フラグメントや、自動定理発見の入力ファイルに必要なデータを管理しやすいように、「式、形式理論の部分集合、論理体系、分野、推論規則」にそれぞれ URI を付与し、リソース間の関係を定義した. また、既存の語彙ではリソース間の関係を適切に表現できないと判断したものに関して、理論グリッドで使用できる述語 Theory Grid Relations(接頭辞:tgr)を定義した.

提案された自動定理発見手法では同じ内容の論理式が他のフラグメントにも存在する可能性

があるので、無駄な ID の生成や衝突を避けるために、式の内容と語彙から得られるハッシュ関数の値を ID 付けに使用し、固有の ID を付与している.

5. 公理的集合論における事例研究

本研究では、RDF ストアおよび SPARQL エンドポイントに OpenLink Software 社の Virtuoso7.2[5]を使用した. 現在はプロトタイプとして単一のノードの SPARQL エンドポイントと RDF ストアの部分だけを作成し、SPARQL 文の生成、格納するデータの RDF への変換、RDF ストアへの格納は手作業で行っている.

格納されたフラグメントの一覧を表示し、その中から特定のフラグメントの詳細やそのフラグメントに属する式の一覧、また包含関係について検索できることを確認し、作成した理論グリッドのプロトタイプの有用性を確認した.

6. おわりに

Linked Data を用いて、自動定理発見の試行者が利用する情報を体系的に格納し検索が行える、公理的集合論における理論グリッドのプロトタイプを構築した. 今後の課題としては、未着手である Web インターフェースの開発や、公理的集合論以外の分野における自動定理発見の推論結果の格納などが挙げられる.

参考文献

[1] J. Cheng: Entailment Calculus as the Logical Basis of Automated Theorem Finding in Scientific Discovery, AAAI Technical Report SS-95-03, pp. 105-110, AAAI Press, March 1995.
 [2] J. Cheng et al.: FreeEnCal: A Forward Reasoning Engine with General-Purpose, LNAI/LNCS, Vol. 4693, pp. 444-452, Springer-Verlag, September 2007.
 [3] J. Cheng et al.: A Cooperative Grid Computing Approach to Automated Theorem Finding and Automated Problem Proposing, LNAI/LNCS, Vol. 4693, pp. 840-851, Springer-Verlag, September 2007.
 [4] 神崎 正英: セマンティック・ウェブのための RDF/OWL 入門, 森北出版株式会社, 2005年3月.
 [5] OpenLink Software:
<https://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/>