RDF データに対する Shape Expression Schema の妥当性検証アルゴリズム

藤永　健†　　鈴木　伸崇††
† 筑波大学情報学群知識情報・図書館学類　　†† 筑波大学図書館情報メディア系

1. はじめに

近年 RDF データに対するスキーマ言語として，Shape Expression Schema[1]（以下，ShEx）が提案されている．ShEx は従来の RDF スキーマよりもデータ構造を厳密に定義することが可能である．ShEx に対する妥当性検証手法も考案されているが[2]，これはデータ全体を主記憶に収めて処理することを前提としている．本稿では，より大きな RDF データでの検証も可能とするため，メモリ消費量を抑えた妥当性検証アルゴリズムを提案する．

2. ShEx

ShEx は3次組 $S=(\Sigma,\Gamma,\delta)$と表される．ここで，Σはラベルの集合，Γは型の集合，δは型を定義する規則関数である．型の例を示す．

$$\delta(t1) = a :: t2 || b :: t3$$

ここで t1,t2,t3 は型，a,bはラベル，||は無順序の連結演算子である．ShEx ではノードが（単一の型ではなく）型の集合をもつことを許している(multi-type semantics)．各ノードに割り当てられる型の集合を表す関数をλとする．

$G=(V,E)$をグラフとする．各ノード $n \in V$ と n に割り当てられた各型 $t \in \lambda(n)$が $\text{fl-out-lab-type}^{\lambda}{}_{G}(n) \cap \delta(t) = \emptyset$ を満たすとき，G は S に妥当であるという．ここで，$\text{fl-out-lab-type}^{\lambda}{}_{G}(n)$は出力ラベルと接続先ノードの型の組み合わせを平坦化したものである．

3. 提案手法

提案アルゴリズムの概要を次に示す．

> Input: グラフ $G=(V,E)$, ShEx $S=(\Sigma,\Gamma,\delta)$
> Output: ノードに対する型の割り当て λ
> 1.各ノード $n \in V$ に対して $\lambda(n)=\Gamma$ とする．ただし，中間ノード n に対してのみ$\lambda(n)$を主記憶に保持し，葉ノードは主記憶に載せない．
> 2.λが変化する限り，以下を繰り返す
> - G をシーケンシャルに読み，Refine を実行する
> 3.λを返して終了

ここで，Refine とは以下の式で表される処理である．

$$|\text{Refine}(\lambda(n))| = \{t \in \lambda(n) \mid \text{fl-out-lab-type}^{\lambda}{}_{G}(n) \cap \delta(t) = \emptyset\}$$

Refine を行うごとに，$\lambda(n)$のサイズは単調に減少する．この処理をλが収束するまで繰り返す．

4. 評価実験

SP2Bench[3]で生成した 200MB から 1000MB の 5 つの RDF データを使い，提案アルゴリズムを使用して妥当性検証を行った際の処理時間とメモリ消費量を測定した．A は提案アルゴリズム，B は in memory 版である．

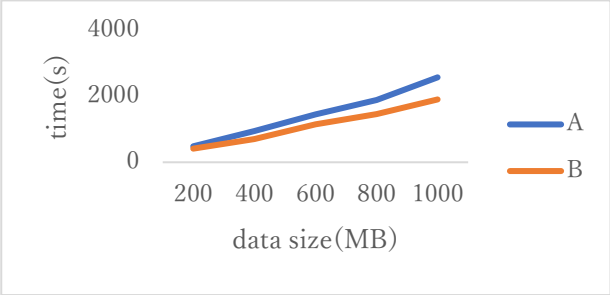

図 1. 処理時間

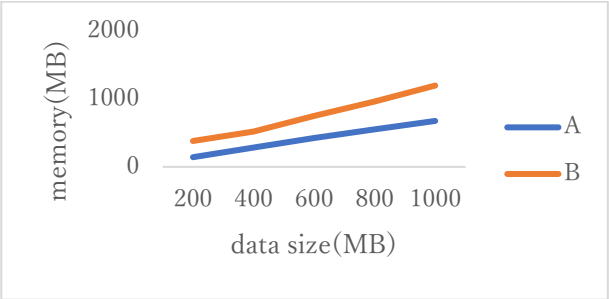

図 2. メモリ消費量

提案アルゴリズムの処理時間は線形であり，in memory 版と比較して若干の増加に留まっている．一方，提案アルゴリズムのメモリ消費量は in memory 版の 7 割程度に抑えられている．

5. 今後の課題

今後は主記憶を超えるグラフデータに対しても妥当性検証が行えるようにする．また，Refine のアルゴリズムを見直し，速度の向上を目指す．

参考文献

[1] World Wide Web Consortium (W3C).“Shape Expressions (ShEx)”. http://shex.io/shex-primer/
[2] S. Staworko, et al.“Complexity and Expressiveness of ShExfor RDF”. ICDT 2015, pp.195-211.
[3] M. Schmidt, et al. “SP2Bench: a SPARQL Performance Benchmark,” ICDE 2009, pp. 371-393.

Copyright © 2018 IEICE