

RDF と RDF スキーマの推論

Reasoning about RDF and RDF Schema

兼岩 憲
Ken Kaneiwa

岩手大学工学部電気電子・情報システム工学科
Department of Electrical Engineering and Computer Science, Iwate University.
kaneiwa@cis.iwate-u.ac.jp, <http://www.ai.cis.iwate-u.ac.jp/>

keywords: Resource Description Framework (RDF), RDF Schema (RDFS), Semantic Web, Deduction System

Summary

This survey paper describes the syntax and semantics for Resource Description Framework (RDF) and RDF Schema (RDFS). A deduction system for RDF and RDFS is explained as sound and complete for the RDF and RDFS semantics. To develop such a system, we introduce a minimal RDFS fragment, obtained by restricting the syntax of RDF statements, which is simple and provides us with a computationally efficient reasoning algorithm. We also sketch some approaches to an extension of the RDF language and deduction system by involving some Web Ontology Language (OWL) vocabulary.

1. はじめに

RDF (Resource Description Framework) [Lassila 99] は、セマンティック Web 技術の 1 つとして Linked Data やメタデータの記述に用いられる意味データ形式である。RDF は URI (Uniform Resource Identifier) [Berners-Lee 05] で名付けられたリソースの関係を記述 (モデル化) する枠組みを与え、RDF データによって Web 上の情報リソースに対して計算機が解釈可能なメタデータを提供する。

本稿では、RDF と RDF スキーマを深く理解するために、RDF データがどのように意味解釈され、その結果どのような推論をもたらすか説明する。RDF は主語、述語と目的語の三つ組による RDF トリプルの集まり (RDF グラフと呼ばれる) から構成され、RDF トリプルによる言明は主語と目的語の 2 項述語 (プロパティ) から単純に解釈される。RDF 言明の (2 項述語としての) 単純な解釈に加えて、RDF と RDF スキーマの語彙 (`rdf:type`, `rdfs:subClassOf`, `rdfs:subPropertyOf` など) には、リソースをクラスやプロパティとして解釈した意味論が定義される。そのような意味論に基づき、RDF と RDF スキーマに関する推論規則 (RDF entailment rules) が設計されている。

このような RDF と RDF スキーマの意味論や推論システムは、Hays [Hayes 04] によって W3C 公式の仕様書に書かれている。それに基づき、セマンティック Web の専門書 [神崎 05, Allemang 08, Segaran 09] が数多く出版され、RDF の記述方法や言語構文が解説されている。一方、RDF と RDF スキーマの推論システムはセマンティック Web の国際会議 ISWC やジャーナル J.WS の発表論文において現在でも盛んに改良・拡張がなされている。同様

に、RDF の推論技術を改良したり実際に意味処理システムを開発したりするには、RDF と RDF スキーマの意味論や推論システムの理解が不可欠となる。しかしながら、それらを詳しく説明している解説は少ない。

そこで本稿では、RDF と RDF スキーマの意味論や推論システムに焦点を当て、意味論に基づいた完全な推論システムを説明する。さらに実装を考えた最新の技術として、計算量的に優れた RDF 推論 (即ち、より少ない時間で推論できる) を目指した最小 RDF グラフを紹介する。その後、OWL (Web Ontology Language) の語彙を用いて RDF と RDF スキーマの言語を拡張するとともに、それぞれの OWL 語彙の意味づけによってもたらされる推論システムの強化についても説明する。

本稿の構成は、以下の通りである。2 章では、RDF と RDF スキーマの基本を理解する準備として、RDF と RDF スキーマの構文と意味論を説明する。3 章では、RDF と RDF スキーマの完全な推論システムを紹介し、その後、制限された最小 RDF グラフとその効率的な推論アルゴリズムを説明する。4 章では、RDF と RDF スキーマの拡張として、OWL 語彙を導入した RDF グラフの意味論とその推論規則について紹介する。最後に、5 章で本稿のまとめを述べる。

2. RDF 言語

2.1 RDF とは?

セマンティック Web では、Web や現実世界の情報リソース (概念、人や組織、イベントなど) を URI で名前付けして識別する。RDF は、それらのリソース間の関係を主語、述語と目的語の三つ組 (RDF トリプルという)

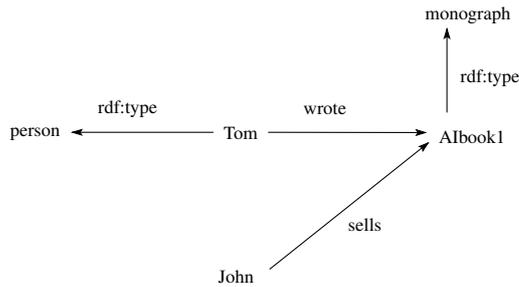


図1 RDF グラフによるインスタンス記述例

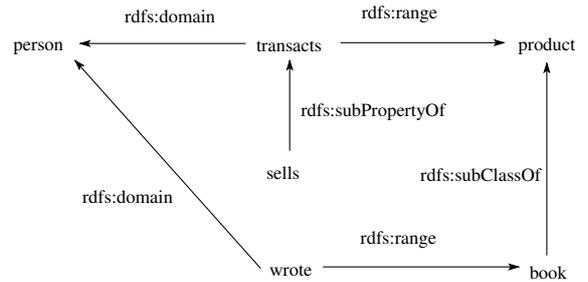


図2 RDF グラフによるオントロジー記述例

によって関係づける枠組みである．最初に，例を用いて RDF の記述を簡単に説明する．

〔例 2・1〕 以下は，主語 Tom，述語 `rdf:type`，目的語 `person` により構成された RDF トリプル例である．

```
:Tom rdf:type :person.
```

これは，Tom のタイプ (型) が `person` であることを意味する．言い換えると，Tom は `person` クラスのインスタンス (もしくはオブジェクト) であることを示す．この RDF トリプルは，Web 上のリソース Tom のタイプ (型) を宣言するメタデータと言える．

このような RDF トリプルを複数集めた RDF 言明の集合により，主語と目的語をノード，述語をエッジとしたグラフ構造 (RDF グラフ [Carroll 05]) が構成される．例えば図 1 は，上記の例を含んだ RDF トリプル集合によって記述された RDF グラフである．リソース `AIbook1` は専門書 (*monograph*) のインスタンスであり，ジョンはその本を販売している．また，トムは人間 (*person*) のインスタンスであり，本の著者 (*wrote* の主語) である．よって図 1 は，リソース間の関係とリソースのタイプ付けを行っているメタデータと言える．

図 2 は，RDF スキーマを用いてクラスやプロパティについて記述した RDF グラフであり，1 つのオントロジーを記述している．プロパティ *sells* は *transacts* のサブプロパティであり，プロパティ (2 項述語) *transacts* のドメイン (定義域) とレンジ (値域) はそれぞれクラス *person* と *product* である．また，プロパティ (2 項述語) *wrote* のドメイン (定義域) とレンジ (値域) はそれぞれクラス *person* と *book* である．このように，図 1 が主にインスタンスの記述であったのに対して，図 2 はクラスとプロパティに対する意味構造 (オントロジー)*1 の記述である．

2.2 RDF と RDF スキーマの構文

本節では，RDF 言語の構文を形式的に定義する．RDF は，URI 参照の集合，空ノードの集合，およびリテラルの集合から構成される．これらの集合をそれぞれ URI 参照の集合 U ，空ノードの集合 B ，およびリテラルの集合 L で表す．空ノードは，「あるリソース」を意味する変数

*1 オントロジー言語 OWL を用いなくても，単純なオントロジーならば RDF スキーマで記述可能である．

(存在量化された変数) の働きをした無名リソースである．リテラルは，文字列で表されたプレーンリテラルと固定型 (整数型や日付型など) の値を表す型付きリテラルがある． $s \in U \cup B$ ， $p \in U$ ， $o \in U \cup B \cup L$ のとき，RDF トリプルは以下の三つ組で表される．

```
s p o.
```

即ち，RDF トリプルは $(U \cup B) \times U \times (U \cup B \cup L)$ の要素と言える．RDF トリプルが空ノードを含まないとき，それを基底トリプルという．本稿では，必要に応じて RDF トリプルを (s, p, o) で表すこととする．RDF グラフ G は，RDF トリプルの集合 $\{(s_1, p_1, o_1), \dots, (s_n, p_n, o_n)\}$ として定義される．故に，リソース間の関係を記述する図 1 と図 2 は，RDF トリプルの有限集合により構成された RDF グラフである．RDF グラフ G のすべての部分集合 $G' (\subseteq G)$ は， G の部分グラフである．

さらに，RDF にはリソースに関するオントロジーを記述するために RDF スキーマと呼ばれる基本語彙があらかじめ用意されている．以下は，その中でも意味論と推論において重要な働きをする語彙である．

```
s rdfs:subClassOf o.
s rdfs:subPropertyOf o.
s rdfs:domain o.
s rdfs:range o.
```

最初の 2 つは，それぞれ s が o のサブクラスと s が o のサブプロパティであることを示す．3 つ目と 4 つ目はプロパティ s のドメインとレンジを o で宣言する．RDF スキーマの語彙は，名前空間 `rdfs` の接頭辞が付与された URI 参照の集合である (即ち， U の部分集合である)．尚，意味論や推論システムの定義では `rdfs:subClassOf`，`rdfs:subPropertyOf`，`rdf:type`，`rdfs:domain` および `rdfs:range` をそれぞれ *sc*，*sp*，*type*，*dom* および *range* のように簡潔に表す．

2.3 RDF と RDF スキーマの意味論

文献 [Hayes 04, Muñoz 09] に基づいて，RDF の意味論を定義する．ある RDF の語彙 $V (\subseteq U \cup L)$ に対する解釈 \mathcal{I} は，以下より構成される組 $\mathcal{I} = (IR, IP, IC, Ext, CExt, Lit, I)$ である．

- IR : (空でない) リソースの集合
- IP : プロパティの集合
- IC : クラスの集合 (但し, $IC \subseteq IR$)
- Ext : プロパティの解釈関数 $IP \rightarrow 2^{IR \times IR}$
- $CExt$: クラスの解釈関数 $IC \rightarrow 2^{IR}$
- Lit : リテラル値の集合 (但し, $Lit \subseteq IR$)
- I : リソースとリテラルの解釈関数 $(U \cup L) \cap V \rightarrow IR \cup IP$

例えば, RDF トリプル $(book1, title, A1book1)$ の意味論を考える. ある解釈 \mathcal{I} が与えられたとする. そのとき, 主語の $I(book1)$ と目的語の $I(A1book1)$ に IR の要素, 述語の $I(title)$ に IP の要素が割り当てられる. さらに, $title$ をプロパティとして解釈するために $Ext(I(title))$ にはリソースの 2 項述語 (即ち, $IR \times IR$ の部分集合) が割り当てられる. このとき, $(I(book1), I(A1book1)) \in Ext(I(title))$ が成り立つならば, 解釈 \mathcal{I} においてこの RDF トリプルは真である.

次に, 解釈 \mathcal{I} を用いて RDF グラフ G のモデルを定義する. まず, G 内の空ノード (変数) にリソースを割り当てる関数 $A: B \rightarrow IR$ (変数割り当てと呼ぶ) を導入する. A により, G 内のすべてのノード x に対し, $x \in U \cup L$ のとき $I_A(x) = I(x)$ として, $x \in B$ のとき $I_A(x) = A(x)$ とする. ある変数割り当て A が存在して, G 内のすべての RDF トリプル (s, p, o) に対して, 解釈 \mathcal{I} が $I(p) \in IP$ かつ $(I_A(s), I_A(o)) \in Ext(I(p))$ を満たし, RDF と RDF スキーマの語彙に関する以下の条件を満たすとき, \mathcal{I} は G のモデルであるという.

- タイプ, クラスとプロパティの意味条件
 - $x \in CExt(y)$ ならば, かつそのときに限り $(x, y) \in Ext(I(type))$
 - $(x, y) \in Ext(I(type))$ ならば, $y \in IC$
 - $(x, y) \in Ext(I(dom))$ ならば, $x \in IP$ かつ $y \in IC$
 - $(x, y) \in Ext(I(range))$ ならば, $x \in IP$ かつ $y \in IC$
- ドメインとレンジの意味条件
 - $(x, y) \in Ext(I(dom))$ かつ $(u, v) \in Ext(x)$ ならば, $u \in CExt(y)$
 - $(x, y) \in Ext(I(range))$ かつ $(u, v) \in Ext(x)$ ならば, $v \in CExt(y)$
- サブクラスの意味条件
 - $(x, y) \in Ext(I(sc))$ のとき, $CExt(x) \subseteq CExt(y)$
 - $Ext(I(sc))$ は IC の下で推移的かつ反射的
- サブプロパティの意味条件
 - $(x, y) \in Ext(I(sp))$ のとき, $Ext(x) \subseteq Ext(y)$
 - $Ext(I(sp))$ は IP の下で推移的かつ反射的

2 つの RDF グラフ G と H に対して, G のすべてのモデルが H のモデルであるとき, G は H を含意するといひ, 含意関係 ($G \models H$ で表す) が成り立つ.

3. RDF と RDF スキーマの推論システム

前章で RDF の構文と意味論を定義したが, 本章では RDF グラフに対する推論システムについて説明する.

3.1 完全な推論システム

RDF と RDF スキーマの推論システムは, RDF グラフから演繹的に成り立つ RDF トリプルを導く. 推論システムの特徴は, 単なる検索ではなく明示されていない関係を導出する能力にある. セマンティック Web の RDF クエリ言語 SPARQL (SPARQL Protocol and RDF Query Language) [Prud'hommeaux 08] は RDF グラフから検索を行うが, 明示されていない RDF トリプルを導出できないので推論システムが別に必要となる.

例を用いて, RDF と RDF スキーマの推論とはどういうものかを説明していく.

〔例 3.1〕 図 1 と図 2 のうち次の RDF トリプルから構成した RDF グラフを G' とする.

```
:transacts rdfs:domain :person.
:sells rdfs:subPropertyOf :transacts.
:John :sells :A1book1.
```

このとき, この RDF グラフ G' から以下の RDF トリプルを新たに推論できる.

```
:John rdf:type :person.
```

この帰結は, ジョンが人間のインスタンスであることを意味する. この推論では, 3 つ目の RDF トリプルによりジョンが $A1book1$ を販売 ($sells$) しており, 2 つ目の RDF トリプルにより $sells$ の上位プロパティが $transacts$ であることから, ジョンはその本を取引 ($transacts$) していることが導かれる. さらに, 1 つ目の RDF トリプルにより $transacts$ のドメインが $person$ であるので, ジョンのタイプ (型) は $person$ であると導かれる.

このように, 検索を超えた推論を行うために W3C 仕様書の RDF 意味論では, 推論規則 (RDF entailment rules) が規定されている. しかしその後の研究 [Marin 04, Horst 05] で, 推論規則が不完全であると指摘されている. 言い換えると, そのままでは導けない推論結果が存在する. その問題を解決すべく, Horst ら [Horst 05] は RDF トリプルの述語に変数 (即ち, 2 項述語の変数) を許した一般化 RDF グラフ (generalized RDF graph) を定義して完全な推論を実現した. 一般化 RDF グラフでは, RDF トリプルは $(U \cup B) \times (U \cup B) \times (U \cup B \cup L)$ の要素となる. また, Muñoz ら [Muñoz 09] は, 新たな推論規則を追加して完全性の成り立つシステムを定式化した. この定式化では, 数学的にシンプルにするため RDF グラフを一般化して, RDF トリプルを $(U \cup B \cup L) \times U \times (U \cup B \cup L)$ の要素としている. 即ち, RDF トリプルの主語にリテラルを許している.

表 1 空ノードに関する推論規則

シンプル規則
G_i
↓
$G_{i+1} = G_i[A/Z]$
$A \in \mathbf{U} \cup \mathbf{B} \cup \mathbf{L}$ かつ $Z \in \mathbf{B} - G_i$

本稿では, Muñoz らの完全な推論システムを紹介する. 推論規則は, 表 1 から表 4 により与えられる. 尚, 推論規則内では任意のリソース, 空ノード, およびリテラルを表すために, 記号 A, B, C, X, Y, Z を用いる. こららの推論規則は, 筆者が Muñoz らの推論規則から冗長な記述を除いて簡潔にわかりやすく書き直したものである.

表 1 は, RDF トリプルの主語と目的語に現れる URI 参照, 空ノード, またはリテラルを新しい空ノードへ変換する規則である. Z を RDF グラフ G_i に含まれない空ノード ($Z \in \mathbf{B} - G_i$ で表す) とする. そのとき, 主語または目的語として G_i に出現する同じ名前の A をいくつか (任意の数) 空ノード Z へ一括置換する操作を $G_i[A/Z]$ で表す. 従って, シンプル規則は RDF グラフ G_i から $G_{i+1} = G_i[A/Z]$ を導く規則である. このとき, RDF トリプルの述語 A は空ノード Z へ置換できない.

表 2 は, サブクラスとサブプロパティに対する推論規則である. 例えば, サブクラス規則 (a) は, (A, sc, B) と (B, sc, C) が G_i に含まれているとき, A が B のサブクラス, B が C のサブクラスなので, A が C のサブクラスであることから, (A, sc, C) が G_i に含まれないならばそれを G_i に追加して G_{i+1} を導く. 表 3 は, インスタンスのタイプを導く規則である. 例えば, タイプ規則 (a) は, $(A, dom, B), (C, sp, A), (X, C, Y)$ が G_i に含まれているとき, A のドメインが B, C が A のサブプロパティ, プロパティ C の主語と述語が X と Y なので, X のタイプが B であることから, $(X, type, B)$ が G_i に含まれないならばそれを G_i に追加して G_{i+1} を導く.

表 4 は, サブクラスとサブプロパティの反射性に関する推論規則である. 例えば, サブクラスの反射性規則は, $(A, sc, B), (B, sc, A), (X, dom, A), (X, range, A)$, または $(X, type, A)$ のいずれかが G_i に含まれているとき, A がクラスであると言えるので, (A, sc, A) が G_i に含まれないならばそれを G_i に追加して G_{i+1} を導く.

ここで, ターゲットの RDF データから上記の推論規則を用いてどのように推論するか考える. 1 つの RDF グラフ G が与えられたとき, 次の手順で推論を行う. 最初に, G に RDF スキーマの公理 $(sp, sp, sp), (sc, sp, sc), (type, sp, type), (dom, sp, dom)$ と $(range, sp, range)$ を追加した集合を初期グラフ G_0 とする. G_0 を初期グラフとして表 1 から表 4 の推論規則を適用していくことで, 演繹的な推論が可能となる. 厳密には, G の初期グラフ G_0 から推論規則を 0 回以上適用して得られた $G_n (n \geq 0)$

が存在するとき, G_n の部分グラフ $H (\subseteq G_n)$ は G から導出可能であるといい, $G \vdash H$ で表す.

次のように, RDF 意味論において RDF 推論システムの完全性が成り立つことが証明できる.

[定理 3.1] (RDF 推論システムの完全性) 有限の RDF グラフ G と H が与えられているとする. $G \vdash H$ のとき, かつそのときに限り $G \models H$ である.

以上の RDF 推論システムを用いて, RDF グラフからの推論例を説明する.

[例 3.2] 例 3.1 の RDF グラフ G' に, 以下の RDF トリプルを追加したものを G'' とする.

```
:Tom :wrote :AIbook1.
:Tom rdf:type :person.
:AIbook1 rdf:type :monograph.
:transacts rdfs:range :product.
:book rdfs:subClassOf :product.
:wrote rdfs:domain :person.
:wrote rdfs:range :book.
```

これにより, RDF グラフ G'' は図 1 と図 2 の RDF 言明すべてを表す.

前処理として, G'' に RDF スキーマの公理を追加して初期グラフ G_0 を得る. その後に, G_0 に推論規則を適用すると図 3 のような推論が可能である. 最初に, G_0 には $(transacts, dom, person), (sells, sp, transacts)$ および $(John, sells, AIbook1)$ を含むのでタイプ規則 (a) の適用条件が成り立ち, G_0 に $(John, type, person)$ を追加して G_1 へ拡張される. 次に, G_1 は $(sells, sp, transacts)$ と $(John, sells, AIbook1)$ を含むのでサブプロパティ規則 (b) の適用条件が成り立ち, $(John, transacts, AIbook1)$ を追加して G_2 へ拡張される. ここで G_2 は $(Tom, wrote, AIbook1)$ を含むことからプロパティの反射性規則により, $(wrote, sp, wrote)$ を追加して G_3 へ拡張される. 続いて, G_3 は $(wrote, range, book), (wrote, sp, wrote)$ と $(Tom, wrote, AIbook1)$ を含むのでタイプ規則 (b) の適用条件が成り立つことから, G_3 に $(AIbook1, type, book)$ を追加して G_4 へ拡張される. 最後のステップでは, G_4 は $(book, sc, product)$ と $(AIbook1, type, book)$ を含むのでサブクラス規則 (b) の適用条件が成り立ち, G_4 に $(AIbook1, type, product)$ を追加して G_5 へ拡張される.

3.2 最小 RDF 推論システム

セマンティック Web の実現には意味処理に伴うオーバーヘッドが問題となるので, 大規模な RDF データの高速処理が不可欠である. そのため大規模データから推論するときに, 計算量的に優れた推論システムが求められる. そこで, 次のような RDF の表現が制限された RDF グラフを説明する.

【定義 3.1】(最小 RDF) RDF トリプル (s, p, o) が基底トリプルであり, 主語 s と目的語 o いずれにも RDF ス

表 2 サブクラス関係とサブプロパティ関係に関する推論規則 (1)

サブクラス規則 (a)	サブクラス規則 (b)
G_i ↓ $G_{i+1} = G_i \cup \{(A, sc, C)\}$	G_i ↓ $G_{i+1} = G_i \cup \{(\mathcal{X}, type, B)\}$
$\{(A, sc, B), (B, sc, C)\} \subseteq G_i$ かつ $(A, sc, C) \notin G_i$ のとき	$\{(A, sc, B), (\mathcal{X}, type, A)\} \subseteq G_i$ かつ $(\mathcal{X}, type, B) \notin G_i$ のとき
サブプロパティ規則 (a)	サブプロパティ規則 (b)
G_i ↓ $G_{i+1} = G_i \cup \{(A, sp, C)\}$	G_i ↓ $G_{i+1} = G_i \cup \{(\mathcal{X}, B, \mathcal{Y})\}$
$\{(A, sp, B), (B, sp, C)\} \subseteq G_i$ かつ $(A, sp, C) \notin G_i$ のとき	$\{(A, sp, B), (\mathcal{X}, A, \mathcal{Y})\} \subseteq G_i$ かつ $(\mathcal{X}, B, \mathcal{Y}) \notin G_i$ のとき (但し, B は空ノードでない)

表 3 インスタンスのタイプに関する推論規則

タイプ規則 (a)	タイプ規則 (b)
G_i ↓ $G_{i+1} = G_i \cup \{(\mathcal{X}, type, B)\}$	G_i ↓ $G_{i+1} = G_i \cup \{(\mathcal{Y}, type, B)\}$
$\{(A, dom, B), (C, sp, A), (\mathcal{X}, C, \mathcal{Y})\} \subseteq G_i$ かつ $(\mathcal{X}, type, B) \notin G_i$ のとき	$\{(A, range, B), (C, sp, A), (\mathcal{X}, C, \mathcal{Y})\} \subseteq G_i$ かつ $(\mathcal{Y}, type, B) \notin G_i$ のとき

表 4 サブクラス関係とサブプロパティ関係に関する推論規則 (2)

サブクラスの反射性規則
G_i ↓ $G_{i+1} = G_i \cup \{(A, sc, A)\}$
$(A, sc, B), (B, sc, A), (\mathcal{X}, dom, A), (\mathcal{X}, range, A)$, または $(\mathcal{X}, type, A) \in G_i$, かつ $(A, sc, A) \notin G_i$ のとき
サブプロパティの反射性規則
G_i ↓ $G_{i+1} = G_i \cup \{(A, sp, A)\}$
$(A, sp, B), (B, sp, A), (\mathcal{X}, A, \mathcal{Y}), (A, dom, \mathcal{X})$, または $(A, range, \mathcal{X}) \in G_i$, かつ $(A, sp, A) \notin G_i$ のとき

G_0
 \Downarrow タイプ規則 (a)
 $G_1 = G_0 \cup \{(John, type, person)\}$
 \Downarrow サブプロパティ規則 (b)
 $G_2 = G_1 \cup \{(John, transacts, AIbook1)\}$
 \Downarrow サブプロパティの反射性規則
 $G_3 = G_2 \cup \{(wrote, sp, wrote)\}$
 \Downarrow タイプ規則 (b)
 $G_4 = G_3 \cup \{(AIbook1, type, book)\}$
 \Downarrow サブクラス規則 (b)
 $G_5 = G_4 \cup \{(AIbook1, type, product)\}$

図3 RDF グラフからの RDF 推論例

キーマの語彙 $sc, sp, type, dom$ および $range$ が用いられていないならば、それを最小 RDF トリプルと呼ぶ。

(sp, p, sp) のように主語や目的語に RDF スキーマの語彙が現れると意味論的に複雑になり、推論の探索空間も膨大になってしまう。それを改善するために、最小 RDF トリプルではそうした危険な表現を禁止している*2。ここで、最小 RDF トリプルのみ集合からなる RDF グラフを最小 RDF グラフと呼ぶ。

以下では、最小 RDF グラフ G から 1 つの RDF トリプル (s, p, o) を導出可能か調べるアルゴリズムを説明する。

§1 最小 RDF グラフの推論アルゴリズム

RDF グラフ G と RDF トリプル (s, p, o) を入力したとき、以下のルールを実行して、ルール内に明記された要素が G に含まれるならば、 G から (s, p, o) が導出可能であるという ($G \vdash_m (s, p, o)$ と表す)。但し、 $n \geq 0$ とする。

- (1) (s, p, o) が (A, dom, B) または $(A, range, B)$ のとき、 (s, p, o) が G に含まれるか調べる。
- (2) (s, p, o) が (A, sc, B) のとき、 A から B へのパス $(A, sc, A_1), (A_1, sc, A_2), \dots, (A_n, sc, B)$ (A は B のサブクラスである) が G に存在するか調べる。
- (3) (s, p, o) が (A, sp, B) のとき、 A から B へのパス $(A, sp, A_1), (A_1, sp, A_2), \dots, (A_n, sp, B)$ (A は B のサブプロパティである) が G に存在するか調べる。
- (4) (s, p, o) が $(A, type, B)$ のとき、 G 内に $(A_1, sc, A_2), \dots, (A_n, sc, B)$ (A_1 は B のサブクラスである) が含まれ、以下のいずれかが含まれるか調べる。
 - (A, C_1, \mathcal{Y}) ((A, \mathcal{Y}) は C_1 のインスタンスである) かつ $(C_1, sp, C_2), \dots, (C_n, sp, C_{n+1}), (C_{n+1}, dom, A_1)$ (C_1 はドメイン A_1 をもつ C_{n+1} のサブプロパティである)
 - (\mathcal{Y}, C_1, A) ((\mathcal{Y}, A) は C_1 のインスタンスである) かつ $(C_1, sp, C_2), \dots, (C_n, sp, C_{n+1}), (C_{n+1}, range, A_1)$ (C_1 はレンジ A_1 をもつ C_{n+1} のサブプロパティである)

- $(A, type, A_1)$ (A は A_1 のインスタンスである)
- (5) (s, p, o) が (1)-(4) 以外のとき、 (s, p, o) が G に含まれるか調べる。含まれないとき、 G 内にある (s, A_1, o) が存在し、 $(A_1, sp, A_2), \dots, (A_n, sp, p)$ (A_1 は p のサブプロパティである) が含まれるか調べる。

以上のルールにおいて、 $A = B$ のときサブクラス関係 sc またはサブプロパティ関係 sp による A から B へのパスは自明のため RDF グラフ G に含まれているものとする。

【例 3.3】最小 RDF グラフの推論アルゴリズムを使って、前節の RDF グラフ G'' から図 3 のように以下を導く例を説明する。

$$G'' \vdash_m (AIbook1, type, product)$$

アルゴリズムを見ると、RDF グラフ G'' がルール 4 を満たす RDF トリプルを含めばこの結論が導かれることが分かる。ルール 4 は、 (s, p, o) が $(AIbook1, type, product)$ のとき、 G'' 内に $(book, sc, product)$ が含まれ、 $(Tom, wrote, AIbook1)$ と $(wrote, range, book)$ が含まれることから満たされる。言い換えると、 $book$ は $product$ のサブクラス、 $(Tom, AIbook1)$ はプロパティ $wrote$ のインスタンス、 $wrote$ のレンジは $book$ である。

このように、サブクラスやサブプロパティを辿って条件が成り立つか調べればよいので、前節の RDF 推論システム (推論規則を何度も適用しなければならない方法) に比べて単純で実装が容易である。その上、次のように最小 RDF グラフに限定して推論の完全性が成り立つ。

[定理 3.2] (最小 RDF 推論システムの完全性) 有限の最小 RDF グラフ G と H が与えられているとする。 $G \vdash_m H$ のとき、かつそのときに限り $G \models H$ である。

最小 RDF 推論システムの優位性として、推論の計算量について述べる。文献 [Horst 05] によると、通常の RDF グラフ G と H に対して含意関係 $G \models H$ の判定には以下の計算量が必要である。

[定理 3.3] (RDF 推論の計算量) 有限の RDF グラフ G と H に対して、含意関係 $G \models H$ の判定は NP-完全である。RDF グラフ G が空ノードを含まないならば、含意関係 $G \models H$ の判定は多項式時間で計算可能である。

この計算量を減らすために RDF グラフを単純化した結果、最小 RDF グラフのスキーマ推論は上記アルゴリズムを用いて以下の少ない計算量で実行可能である。

[定理 3.4] (最小 RDF 推論の計算量) 有限の最小 RDF グラフ G と H に対して、導出可能性問題 $G \vdash_m H$ は $O(|H| \cdot |G| \log |G|)$ で計算可能である。

尚、 $|H|$ と $|G|$ は最小 RDF グラフ G と H の大きさを構成要素数で表している。

4. RDF 言語の拡張

4.1 OWL 語彙の導入

前章では RDF と RDF スキーマの推論が NP や多項式時間で計算できて、最小 RDF グラフに制限すればよ

*2 文献 [Bruijn 07] では、危険な表現を排除するために RDF スキーマ語彙の標準利用に限定している。

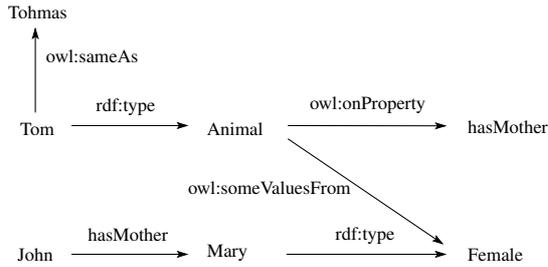


図4 OWL 語彙を用いた RDF グラフの記述例

り少ない計算量 $O(|H| \cdot |G| \log |G|)$ で実行できることを述べた。本節では、RDF 言語に Web オントロジー言語 OWL (Web Ontology Language) [Patel-Schneider 04] の語彙を導入する。図4は、OWL 語彙を用いてより表現力の高いオントロジーを記述した RDF グラフである。ここでは、OWL 語彙を一部追加しても推論の計算量を本質的に増やさない枠組み [Horst 05] を紹介する。言い換えれば、OWL Full を制限したサブセット言語の中でも計算量的に優れた言語である。注目すべきは、記述論理 [Baader 07, 兼岩 09] に基づいた OWL Lite や OWL DL のサブセット言語ではなく、(OWL DL では書けない) RDF 言語の自由な表現を残していることである。

以下は、プロパティの性質のうち関数性、逆関数性、対称性および推移性を記述する OWL 語彙である。

```
owl:FunctionalProperty
owl:InverseFunctionalProperty
owl:SymmetricProperty
owl:TransitiveProperty
```

これらの語彙は、*fun-p*、*inv-fun-p*、*sym-p* と *tra-p* で簡潔に表す。

〔例4.1〕 RDF トリプル (*hasFather*, *type*, *fun-p*) は、リソース *hasFather* が関数的プロパティであることを表現している。

これら OWL 語彙に対して、解釈 I が RDF グラフのモデルとなる条件に以下を追加する。

- $x \in CExt(I(fun-p))$ かつ $(u, v_1), (u, v_2) \in Ext(x)$ ならば、 $(v_1, v_2) \in Ext(I(same))$
- $x \in CExt(I(inv-fun-p))$ かつ $(u_1, v), (u_2, v) \in Ext(x)$ ならば、 $(u_1, u_2) \in Ext(I(same))$
- $x \in CExt(I(sym-p))$ かつ $(u, v) \in Ext(x)$ ならば、 $(v, u) \in Ext(x)$
- $x \in CExt(I(tra-p))$ かつ $(u, v), (v, w) \in Ext(x)$ ならば、 $(u, w) \in Ext(x)$

この意味論は、リソース間の同一性を示す OWL 語彙 `owl:sameAs` (簡潔に *same* で表す) を用いて、関数的・逆関数的・対称的・推移的プロパティを適切に意味づけている。

以上の OWL 語彙に対する意味論により、表5のようなプロパティの性質に関する推論規則が導入される。例えば、

プロパティ規則 (a) では、RDF グラフ G_i が $(A, type, fun-p)$ 、 (Z, A, X) と (Z, A, Y) を含むとき、プロパティ A は関数的であるので、 $(X, same, Y)$ を含まないならばそれを加えて G_i を G_{i+1} へ拡張する。言い換えると、プロパティ A のすべての主語 Z に対して、一意の目的語 ($X = Y$) が決まる。

以下は、同一性、クラス間関係とプロパティ間関係を記述する OWL 語彙である。

```
owl:sameAs
owl:inverseOf
owl:equivalentClass
owl:equivalentProperty
```

これらの語彙は、*same*、*inv*、*equiv-c* と *equiv-p* で簡潔に表す。

〔例4.2〕 図4左上に示すように、RDF トリプル (*Tom*, *same*, *Tohmas*) はプロパティ *same* により2つの名前 *Tom* と *Tohmas* が同一リソースであることを表現している。

同様に OWL 語彙に対して、解釈 I が RDF グラフのモデルとなる条件に以下を追加する。

- $x \in IC$ ($x \in IP$) かつ $(x, y) \in Ext(I(same))$ ならば、 $(x, y) \in Ext(I(sc))$ ($(x, y) \in Ext(I(sp))$)
- $x \in IP$ 、 $(u_1, v_1) \in Ext(x)$ 、 $(u_1, u_2) \in Ext(I(same))$ 、 $(v_1, v_2) \in Ext(I(same))$ ならば、 $(u_2, v_2) \in Ext(x)$
- $Ext(I(same))$ は IR の下で同値関係
- $(x, y) \in Ext(I(inv))$ ならば、 $(u, v) \in Ext(x)$ のときかつそのときに限り $(v, u) \in Ext(y)$
- $(x, y) \in Ext(I(equiv-c))$ のとき、かつそのときに限り、 $(x, y) \in Ext(I(sc))$ かつ $(y, x) \in Ext(I(sc))$
- $(x, y) \in Ext(I(equiv-p))$ のとき、かつそのときに限り、 $(x, y) \in Ext(I(sp))$ かつ $(y, x) \in Ext(I(sp))$

表6は、サブクラス関係、サブプロパティ関係と同値関係に関する推論規則である(尚、*inv* に関する推論は表5のプロパティ規則 (c) で導入されている)。サブクラス規則 (c) とサブプロパティ規則 (c) は、同一性 *same*、および同値関係 *equiv-c* と *equiv-p* の記述から導出される推論規則である。また、同値関係規則 (a)(b) は、サブクラス関係 *sc* とサブプロパティ関係 *sp* から導かれる推論規則である。例えば、同値関係規則 (a) では、RDF グラフ G_i が (A, sc, B) と (B, sc, A) を含むとき、プロパティ A と B は同値なので、 $(A, equiv-c, B)$ を含まないならばそれを加えて G_i を G_{i+1} へ拡張する。

〔例4.3〕 図5は、OWL 語彙の推論規則を適用した例を示す。前処理として、図4の RDF グラフに RDF スキーマの公理を追加して初期グラフ G_0 を得る。最初に、 G_0 は $(Tom, type, Animal)$ を含むのでプロパティ規則 (b) の適用条件が成り立ち、 G_0 に $(Animal, same, Animal)$ を追加して G_1 を得る。次に、 G_1 は $(Tom, same, Tohmas)$ 、 $(Animal, same, Animal)$ と $(Tom, type, Animal)$ を含

表 5 プロパティの性質に関する推論規則

プロパティ規則 (a)	プロパティ規則 (b)	プロパティ規則 (c)
G_i ↓ $G_{i+1} = G_i \cup \{(X, same, Y)\}$	G_i ↓ $G_{i+1} = G_i \cup \{(X, same, X)\}$	G_i ↓ $G_{i+1} = G_i \cup \{(X, A, Y)\}$
$\{(A, type, fun-p), (Z, A, X), (Z, A, Y)\},$ $\{(A, type, inv-fun-p), (X, A, Z), (Y, A, Z)\},$ $\{(X, same, Z), (Z, same, Y)\},$ $\{(Y, same, X)\}$ のどれかを G_i が含み $(X, same, Y) \notin G_i$ のとき	$(X, A, Y) \in G_i$ または $(Y, A, X) \in G_i$, かつ $(X, same, X) \notin G_i$ のとき	$\{(A, inv, B), (Y, B, X)\},$ $\{(B, inv, A), (Y, B, X)\},$ $\{(A, type, sym-p), (Y, A, X)\},$ $\{(A, type, tra-p), (X, A, Z), (Z, A, Y)\},$ $\{(Z_1, same, X), (Z_2, same, Y),$ $(Z_1, A, Z_2)\}$ のどれかを G_i が含み $(X, A, Y) \notin G_i$ のとき

表 6 サブクラス関係, サブプロパティ関係と同値関係に関する推論規則

サブクラス規則 (c)	サブプロパティ規則 (c)
G_i ↓ $G_{i+1} = G_i \cup \{(A, sc, B)\}$	G_i ↓ $G_{i+1} = G_i \cup \{(A, sp, B)\}$
$\{(A, same, B), (X, type, A)\} \subseteq G_i$, $(A, equiv-c, B) \in G_i$, または $(B, equiv-c, A) \in G_i$, かつ $(A, sc, B) \notin G_i$ のとき	$\{(A, same, B), (X, A, Y)\} \subseteq G_i$, $(A, equiv-p, B) \in G_i$, または $(B, equiv-p, A) \in G_i$, かつ $(A, sp, B) \notin G_i$ のとき
同値関係規則 (a)	同値関係規則 (b)
G_i ↓ $G_{i+1} = G_i \cup \{(A, equiv-c, B)\}$	G_i ↓ $G_{i+1} = G_i \cup \{(A, equiv-p, B)\}$
$\{(A, sc, B), (B, sc, A)\} \subseteq G_i$ かつ $(A, equiv-c, B) \notin G_i$ のとき	$\{(A, sp, B), (B, sp, A)\} \subseteq G_i$ かつ $(A, equiv-c, B) \notin G_i$ のとき

G_0
↓ プロパティ規則 (b)
 $G_1 = G_0 \cup \{(Animal, same, Animal)\}$
↓ プロパティ規則 (c)
 $G_2 = G_1 \cup \{(Tohmas, type, Animal)\}$
↓ 制約規則 (a)
 $G_3 = G_2 \cup \{(John, type, Animal)\}$

図 5 RDF グラフからの OWL 推論例

みプロパティ規則 (c) の適用条件が成り立つことから, $(Tohmas, type, Animal)$ を追加して G_2 へ拡張される.

以下は, プロパティの制約(クラスの属性表現に相当する)を記述する OWL 語彙である.

```
owl:onProperty
owl:hasValue
owl:someValuesFrom
owl:allValuesFrom
```

これらの語彙は, $on-p$, $value$, $some$ と all で簡潔に表す.

【例 4.4】 図 4 右側のように, RDF トリプル $(Animal, on-p, hasMother)$ と $(Animal, some, Female)$ は, クラス $Animal$ のすべてのインスタンス x がある母親インスタンス y をもち $((x, y)$ はプロパティ $hasMother$ のインスタンスである), その母親インスタンス y のタイプは $Female$ であることを表現する.

プロパティによってクラスを制限するこれらの OWL 語彙に対して, 解釈 I が RDF グラフのモデルとなる条件に以下を追加する.

- $(x, y_1) \in Ext(I(on-p))$ かつ $(x, v) \in Ext(I(value))$ ならば, $u \in CExt(x)$ のときかつそのときに限り $(u, v) \in Ext(y_1)$
- $(x, y_1) \in Ext(I(on-p))$, $(x, y_2) \in Ext(I(some))$, $(u, v) \in Ext(y_1)$, $v \in CExt(y_2)$ ならば, $u \in CExt(x)$
- $(x, y_1) \in Ext(I(on-p))$, $(x, y_2) \in Ext(I(all))$, $u \in CExt(x)$, かつ $(u, v) \in Ext(y_1)$ ならば, $v \in CExt(y_2)$

この意味論では, $(x, y_1) \in Ext(I(on-p))$ のとき, OWL 語彙 $on-p$ によりクラス x に対してプロパティ y_1 の制約が与えられる. その際に, $value$, $some$ (存在量化) と all (全称量化) のいずれかによりプロパティ上のインスタンスが量化される. 表 7 は, プロパティ制約に関する推論規則である. 例えば, 制約規則 (a) では, RDF グラフ G_i が $(A, on-p, B)$, $(A, value, Y)$ と (X, B, Y) を含むとき, プロパティ B を介して Y を目的語にもつ X のタイプが A に決まる. 従って, $(X, type, A)$ を含まないならばそれを加えて G_i を G_{i+1} へ拡張する. 言い換えると, $(A, on-p, B)$, $(A, value, Y)$ と (X, B, Y) がすべて成り立つような X がクラス A のインスタンスとなる. この制約規則 (a) では, RDF グラフ G_i が $(A, on-p, B)$, $(A, some, Y)$, (X, B, Z) と $(Z, type, Y)$ をすべて含む場合も同じ結論が導かれる.

〔例 4・5〕 図 5 の後半は、表 7 の推論規則を適用した例である。ここで、 G_0 から推論した G_2 は図 4 の RDF グラフを拡張した結果なので、 $(Animal, on-p, hasMother)$ 、 $(Animal, some, Female)$ 、 $(John, hasMother, Mary)$ と $(Mary, type, Animal)$ を含んでいる。よって制約規則 (a) の適用条件が成り立ち、 $(John, type, Animal)$ を追加して G_3 へ拡張される。

4.2 その他のアプローチ

本稿では詳しくは紹介しないが、セマンティック Web 研究では意味データ処理を強化するために RDF 言語の様々な拡張が提案されている。例えば、RDF 言語に時間表現を導入した時間 RDF グラフが研究されている [Gutierrez 07]。また、RDF に不足している否定表現を加えた言語も提案されている [Analyti 05]。さらに、RDF グラフは高階な記述を可能にしてしまうことから、OWL Full まで表現力を上げると決定不能に陥る [Motik 07]。そこで、高階表現に対する意味論や推論の決定可能性 (決定不能性) が証明されたり、決定可能を保持した安全な高階言語が提案されたりしている [Pan 07]。

これらの拡張から分かるように RDF は 1 つのデータ記述言語としての広がりと可能性を備えており、意味データの標準フォーマットとなっている。尚、RDF 言語のその他の拡張に関心のある読者は関連文献を参照されたい。

5. おわりに

本稿では、セマンティック Web におけるメタデータやオントロジーの基盤技術として RDF と RDF スキーマの意味論と推論システムを解説した。特に、RDF 推論システムの実装・開発に役立つように、分かりやすく簡潔に書き直して推論規則を紹介した。また、RDF 言語の拡張として OWL 語彙の限定的な導入により、記述論理に基づく推論を避けて RDF 本来のライトウェイトな推論を可能にする実用的な推論方法を紹介した。今後セマンティック Web の実現において、少なくとも RDF と RDF スキーマがコアとなり、必要に応じて OWL 語彙が限定的に用いられることが現実的であろう。

最後に、本稿により RDF と RDF スキーマのデータ表現をユーザーとして利用するだけでなく、RDF と RDF スキーマの言語と推論システムを改良・発展させる研究開発が国内でも促進されることを期待したい。

謝 辞

本稿の改善にあたり、読者から有益なコメントをいただきました。ここに感謝致します。

◇ 参 考 文 献 ◇

[Allemang 08] Allemang, D. and Hendler, J. A.: *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*, Morgan

- Kaufmann (2008)
- [Analyti 05] Analyti, A., Antoniou, G., Damásio, C. V., and Wagner, G.: Stable Model Theory for Extended RDF Ontologies, in *Proceedings of the 8th International Semantic Web Conference (ISWC 2005)*, LNCS 3729, pp. 21–36, Springer (2005)
- [Baader 07] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. eds.: *The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge (2nd Edition)* (2007)
- [Berners-Lee 05] Berners-Lee, T., Fielding, R. T., and Masinter, L.: Uniform Resource Identifier (URI): Generic Syntax, <http://labs.apache.org/webarch/uri/rfc/rfc3986.html> (2005)
- [Bruijn 07] Bruijn, de J. and Heymans, S.: Logical Foundations of (e)RDF(S): Complexity and Reasoning, in *Proceedings of the 6th International Semantic Web Conference/2nd Asian Semantic Web Conference (ISWC 2007/ASWC 2007)*, Busan, Korea, November 11-15, 2007, LNCS 4825, pp. 86–99, Springer (2007)
- [Carroll 05] Carroll, J. J., Bizer, C., Hayes, P. J., and Stickler, P.: Named graphs, *J. Web Sem.*, Vol. 3, No. 4, pp. 247–267 (2005)
- [Gutierrez 07] Gutierrez, C., Hurtado, C. A., and Vaisman, A. A.: Introducing Time into RDF, *IEEE Trans. Knowl. Data Eng.*, Vol. 19, No. 2, pp. 207–218 (2007)
- [Hayes 04] Hayes, P.: RDF Semantics, W3C Recommendation, <http://www.w3.org/TR/rdf-mt/> (2004)
- [Horst 05] Horst, ter H. J.: Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary, *J. Web Sem.*, Vol. 3, No. 2-3, pp. 79–115 (2005)
- [兼岩 09] 兼岩 憲: 記述論理と Web オントロジー言語, オーム社 (2009)
- [神崎 05] 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 (2005)
- [Lassila 99] Lassila, O. and Swick, R.: Resource Description Framework (RDF) model and syntax specification, W3C Recommendation, <http://www.w3.org/TR/PR-rdf-syntax> (1999)
- [Marin 04] Marin, D.: A formalization of RDF, Technical Report TR/DCC-2006-8, Universidad de Chile, <http://www.dcc.uchile.cl/cgutierrez/ftp/draitan.pdf> (2004)
- [Motik 07] Motik, B.: On the Properties of Metamodeling in OWL, *Journal of Logic and Computation*, Vol. 17, No. 4, pp. 617–637 (2007)
- [Muñoz 09] Muñoz, S., Pérez, J., and Gutierrez, C.: Simple and Efficient Minimal RDFS, *J. Web Sem.*, Vol. 7, No. 3, pp. 220–234 (2009)
- [Pan 07] Pan, J. and Horrocks, I.: RDFS(FA): Connecting RDF(S) and OWL DL, *IEEE Trans. Knowl. Data Eng.*, Vol. 19, No. 2, pp. 192–206 (2007)
- [Patel-Schneider 04] Patel-Schneider, P. F., Hayes, P., and Horrocks, I.: OWL Web Ontology Language Semantics and Abstract Syntax, W3C Recommendation, <http://www.w3.org/TR/2004/REC-owl-semantics-20040210/> (2004)
- [Prud'hommeaux 08] Prud'hommeaux, E. and Seaborne, A.: SPARQL Query Language for RDF, W3C Recommendation, <http://www.w3.org/TR/rdf-sparql-query/> (2008)
- [Segaran 09] Segaran, T., Evans, C., and Taylor, J.: *Programming the Semantic Web*, O'Reilly (2009)

著 者 紹 介

兼岩 憲 (正会員)

1993年～1996年富士通(株)。2001年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了。同年国立情報学研究所情報学基礎研究系助手。2006年4月～2010年9月(独)情報通信研究機構。2010年10月より岩手大学工学部電気電子・情報システム工学科准教授。情報科学博士。順序ソート論理、論理プログラミング、オントロジーとセマンティック Web、UML ダイアグラム矛盾検証の研究に従事。2005年人工知能学会論文賞、情報処理学会、電子情報通信学会、ソフトウェア科学会、ALP 各会員。

表 7 プロパティ制約に関する推論規則

制約規則 (a)	制約規則 (b)	制約規則 (c)
G_i \Downarrow $G_{i+1} = G_i \cup \{(\mathcal{X}, type, \mathcal{A})\}$	G_i \Downarrow $G_{i+1} = G_i \cup \{(\mathcal{X}, \mathcal{B}, \mathcal{Y})\}$	G_i \Downarrow $G_{i+1} = G_i \cup \{(\mathcal{Z}, type, \mathcal{Y})\}$
$(\mathcal{A}, on-p, \mathcal{B}) \in G_i$ かつ , $\{(\mathcal{A}, value, \mathcal{Y}), (\mathcal{X}, \mathcal{B}, \mathcal{Y})\} \subseteq G_i$ または $\{(\mathcal{A}, some, \mathcal{Y}), (\mathcal{X}, \mathcal{B}, \mathcal{Z}), (\mathcal{Z}, type, \mathcal{Y})\} \subseteq G_i$, かつ $(\mathcal{X}, type, \mathcal{A}) \notin G_i$ のとき	$\{(\mathcal{A}, on-p, \mathcal{B}), (\mathcal{X}, type, \mathcal{A}),$ $(\mathcal{A}, value, \mathcal{Y})\} \subseteq G_i$ かつ $(\mathcal{X}, \mathcal{B}, \mathcal{Y}) \notin G_i$ のとき	$\{(\mathcal{A}, on-p, \mathcal{B}), (\mathcal{A}, all, \mathcal{Y}),$ $(\mathcal{X}, \mathcal{B}, \mathcal{Z}), (\mathcal{X}, type, \mathcal{A})\} \subseteq G_i$ かつ $(\mathcal{Z}, type, \mathcal{Y}) \notin G_i$ のとき