ファジィ知識表現システムに基づくWebオントロジー記述言語

森谷 俊洋 伊藤 貴康
東北大学大学院情報科学研究科

1 はじめに

ファジィ制約を備えた知識表現システムKRS-FZ[1]は，オントロジーを記述し，推論を行えるシステムである．KRS-FZを基にSemantic Web[2]のオントロジー層を構築するWebオントロジー記述言語KRS-FZ-webをRDF Schema[3]の拡張として設計し，処理系を実現した．Webオントロジー記述言語には，DAMLプログラムで開発されたDAML+OIL[4]やW3Cによって策定が進められているOWL[5]があるが，KRS-FZ-webは概念と関係の定義，事実の記述，曖昧な知識の扱いが可能なファジィ制約機能などのKRS-FZが持つ記述と非単調推論機能に加えて，用語のインポート機能とエクスポート機能からなるWeb向きの機能を備えている．新たに導入したエクスポート機能は用語をインポートできるサイトを制限する機能であり，Webにおける用語の公開と隠蔽を実現する．Webオントロジーのファジィ記述と非単調推論機能を有するKRS-FZ-webシステムの概要について報告する．

2 KRS-FZ-web言語の概要

KRS-FZ を基に RDF Schema[3]の拡張として設計された KRS-FZ-web 言語は，概念と関係の定義，具体的な事実の記述，ファジィ制約機能などに加えて，用語のインポートやエクスポートの機能を備えたWebオントロジー記述言語である．

図 1にKRS-FZ-web言語のクラス階層図を与えた．接頭辞がkrsfzであるクラスはKRS-FZ-web言語のXML名前空間であるhttp://www.ito.ecei.tohoku.ac.jp/KRSFZ-nsにおいて定義されている．RDF Schemaにおいてクラスを表すrdf:Classのサブクラスに概念を表すkrsfz:Conceptを，プロパティを表すrdf:Propertyのサブクラスに概念間の関係を表すkrsfz:Relationを各々定義し，リソースのクラスrdfs:Resourceのサブクラスに式のクラスkrsfz:Expressionとファジィ制約において使用されるメンバシップ関数のクラスkrsfz:MembershipFunctionを定義した．krsfz:Expressionのサブクラスは，連言を表すkrsfz:And，選言を表すkrsfz:Or，否定を表すkrsfz:Not，制約表現を表すkrsfz:Constraintからなる．krsfz:Constraintは，確定制約を表すkrsfz:DefinitiveConstraintとファジィ制約を表すkrsfz:FuzzyConstraintからなり，各々のサブクラスには具体的な制約表現を定義した．

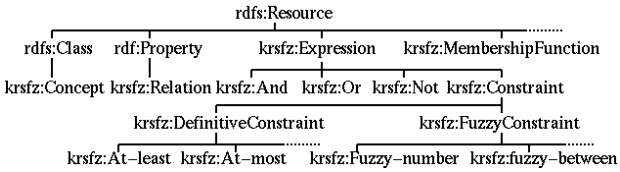

図 1 KRS-FZ-web言語のクラス階層図

KRS-FZ-web 言語は，XML によって<rdf:RDF>タグと</rdf:RDF>タグの間に記述される．

2.1 概念の定義

概念の定義は次のように記述される：

```
<krsfz:Concept c-id> export
  <c-pred>c-expr</c-pred> attr
</krsfz:Concept>
```

*c-id*は概念の名前である．概念 *c-id*は述語 *c-pred*によって式 *c-expr*であると定義され，解釈は属性 *attr* によって決定される．*c-id* の定義は，エクスポート機能 *export* によって指定されたサイトのみがインポートできる．エクスポート機能とインポート機能については 2.5 節で述べる．

c-id，*c-pred*，*c-expr*，*attr*は次のように記述される．

```
c-id ::= rdf:about="uri" | rdf:ID="name"
c-pred ::= krsfz:is|krsfz:implies|krsfz:defaults
c-expr ::= <rdf:Description c-id/>
         | <krsfz:And>has-c-expr*</krsfz:And>
         | <krsfz:Or>has-c-expr*</krsfz:Or>
         | <krsfz:Not>has-c-expr</krsfz:Not>
         | constraint-expr
has-c-expr ::= <krsfz:hasOperand c-reso/>
    |<krsfz:hasOperand>c-expr</krsfz:hasOperand>
c-reso ::= rdf:resource="uri"
attr ::= λ
  | <krsfz:characteristic rdf:resource= ns #cw/>
ns ::= http://www.ito.ecei.tohoku.ac.jp/KRSFZ-ns
cw ::= strict-closed-world | closed-world
```

rdf:about="uri"は任意のURIであるuriを名前とし，rdf:ID="name"は記述された文書のURIの末尾にnameを結合したURIを名前とする．

述語 *c-pred*は3つある．krsfz:isはタグの内容に記述された式と等しいこと，krsfz:implies は式を含むこと，krsfz:defaults は概念が適用された時点で式が知識ベースに矛盾しない場合には式と等しいことを各々表す述語である．

式*c-expr*において，<rdf:Description c-id/>は他の概念c-idを表す．論理演算子には，連言(krsfz:And)，選言(krsfz:Or)，論理的否定

Web Ontology Language Based on Fuzzy Knowledge Representation System
Toshihiro Moriya, Takayasu Ito
Graduate School of Information Sciences, Tohoku University

表 1　KRS-FZ-web 言語における確定制約表現

種類	構文	解釈
個数	<At-least num="*I* ">*oR*</At-least>,　*oR* ::= <onRelation rdf:resource="*R* "/> <At-most num="*I* ">*oR*</At-most> <Exactly num="*I* ">*oR*</Exactly> <Between min="I_1" max="I_2">*oR*</Between>	$R(i_1)\wedge...\wedge R(i_n) \wedge n \geqq I$ $R(i_1)\wedge...\wedge R(i_n) \wedge n \leqq I$ $R(i_1)\wedge...\wedge R(i_n) \wedge n=I$ $R(i_1)\wedge...\wedge R(i_n) \wedge n\in[I_1, I_2]$
領域	<All>*oR hC*</All>,　　*hC* ::= <hasConcept rdf:resource="*C* "/> <Some>*oR hC*</Some> <The>*oR hC*</The>	$\forall x\ (R(x)\wedge x\in C)$ $\exists x\ (R(x)\wedge x\in C)$ $(R(x)\wedge x\in C)$となる xが唯一存在する
値	<Filled-by>*oR*<hasValue rdf:resource="*i* "/></Filled-by> <*rel* hasValue="*N* ">*oR*</*rel*>,　*rel* ::= morethan|lessthan|more=|less=|=|/= <Region><hasBag><rdf:Bag>$ri_1...ri_n$</rdf:Bag></hasBag></Region> *ri* ::= <rdf:li><EachRegion min="N_1" max="N_2">*oR*</EachRegion></rdf:li>	$R(i)$ $R(x)\wedge(x\ rel\ N)$となる x $ri_1(x_1)\wedge...\wedge ri_n(x_n)$となる $(x_1,...,x_n)$ $ri(x) = R(x) \wedge x\in[N_1,N_2]$

I：非負整数，*R*：関係を表す URI，*C*：概念を表す URI，*i*：事例を表す URI，*N*：実数．各語の接頭辞 krsfz は省略している．

(krsfz:Not)があり，それぞれオペランドをkrs-fz:hasOperandによって記述する．*constraint-expr*は制約表現であり，2.4 節において述べる．

概念の解釈は属性*attr*で指定する．*attr*に*cw*が指定された場合には，定義した概念は閉世界仮説で解釈される．*cw* は 2 つあり，strict-closed-worldでは文書に記述された知識に限定した推論が行われ，closed-worldでは(後述の)インポート機能によって他サイトから導入した知識も考慮される．*attr* が空である場合(すなわちλである場合)には開世界仮説で解釈される．

例えば，「男子学生（MaleStudent）は男性（Male）であり，かつ，学生（Student）である」という概念の定義は次のように記述できる．

```
<krsfz:Concept rdf:ID="MaleStudent">
 <krsfz:is>
  <krsfz:And>
    <krsfz:hasOperand rdf:resource="#Male"/>
    <krsfz:hasOperand rdf:resource="#Student"/>
  </krsfz:And>
 </krsfz:is>
</krsfz:Concept>
```

2.2　関係の定義

関係の定義は次のように記述される：

```
<krsfz:Relation r-id> export
  <r-pred>r-expr r-domain r-range</r-pred> attr
</krsfz:Relation>
```

関係*r-id*は述語*r-pred*，式*r-expr*，定義域*r-domain*，値域*r-range*を用いて定義され，属性*attr*によって解釈が決定される．*r-pred*,*r-expr*,*r-domain*，r-*range*は次のように記述される．

```
r-pred ::= krsfz:is | krsfz:implies
r-expr ::= <rdf:Description r-id/>
         | <krsfz:And>has-r-expr*</krsfz:And>
has-r-expr ::= <krsfz:hasOperand r-reso/>
     |<krsfz:hasOperand>r-expr</krsfz:hasOperand>
r-domain ::= <krsfz:Domain c-id/>
r-range ::= <krsfz:Range c-id/>
```

例えば，「友人(friend)」関係であり，値域が「男性(Male)」である「男友達（malefriend）」は次のように記述できる．

```
<krsfz:Relation rdf:ID="malefriend">
  <krsfz:is>
    <rdf:Description rdf:about="#friend"/>
    <krsfz:Range rdf:about="#Male"/>
  </krsfz:is>
</krsfz:Relation>
```

2.3　具体的な事実の記述

具体的な事実は次のように記述される：

```
<rdf:Description i-id> export
  prop1 ... propn
</rdf:Description>
```

事例を表すリソース *i-id* について，命題 $prop_1,...,prop_n$ の連言を事実として記述する．命題 *prop* は次のように記述される．

```
prop ::=<rdf:type c-reso/> | <r-name reso/>
       |<krsfz:hasConstraint>
           constraint-expr</krsfz:hasConstraint>
       |<krsfz:hit-degree>
         <krsfz:ConceptDegree krsfz:degree="N01">
          <krsfz:concept c-reso/>
         </krsfz:ConceptDegree>
        </krsfz:hit-degree>
```

rdf:type は概念 *c-reso* に属することを指定し，*r-name* は関係 *r-name* の値に *reso* を指定する．例えば，「太郎（taro）は学生(Student)であり，悟郎(goro)の友人(friend）である」という事実は次のように記述できる．

```
<rdf:Description rdf:ID="taro">
  <rdf:type rdf:resource="#Student"/>
  <friend rdf:resource="#goro"/>
</rdf:Description>
```

krsfz:hasConstraint は制約表現 *constraint-expr* の成立を述べる．krsfz:hit-degree は 2.4 節で述べるファジィ制約のための構文であり，概念 *c-reso* の適合度に N_{01} を指定する．

2.4　KRS-FZ-web言語における知識の制約表現

制約表現はリソースに対する修飾子であり，確定制約とファジィ制約からなる．

確定制約は，個数制約，領域制約，値制約からなる．**表 1** に KRS-FZ-web 言語において使用される確定制約を与えた．リソース *i* と *R* の関係にあるリソースの集合を I_vとすると，個数制約

表 2　KRS-FZ-web 言語におけるファジィ制約表現

種類	構文	解釈
個数	<Fuzzy-number>*hv hF oR*</Fuzzy-number> *hv* ::= <hasVhedge rdf:resource="*h* "/> *hF* ::= <hasMemFunction rdf:resource="*F* "/> *oR* ::= <onRelation rdf:resouce="*R* "/>	$R(i_1)\wedge\ldots\wedge R(i_n) \wedge F_h(n)\geqq T$
	<Fuzzy-between>*hv*$_1$ *hF*$_1$ *hv*$_2$ *hF*$_2$ *oR*</Fuzzy-between>	$R(i_1)\wedge\ldots\wedge R(i_n) \wedge (\forall y_1,y_2\ F_{h1}(y_1)\geqq T\wedge F_{h2}(y_2)\geqq T\wedge\ n\in[y_1,y_2])$
領域	<Fuzzy-all>*oR hv hF hC*</Fuzzy-all> *hC* ::= <hasConcept rdf:resource="*C* "/>	$\forall x\ (R(x)\wedge\ F_h(C_m(x))\geqq T)$
	<Fuzzy-some>*oR hv hF hC*</Fuzzy-some>	$\exists x\ (R(x)\wedge\ F_h(C_m(x))\geqq T)$
	<Fuzzy-the>*oR hv hF hC*</Fuzzy-the>	$(R(x)\wedge F_h(C_m(x))\geqq T)$となる xが唯一存在する
値	<Filled-by>*oR hv hF*</Filled-by>	$R(x)\wedge x\in f\text{-}val$ となる $x, f\text{-}val=\{x\mid F_h(x)\geqq T\}$
	<*rel*>*oR hv hF*</*rel*>,　*rel* ::= morethan|lessthan|more=|less=| =|/=	$R(x)\wedge(\forall y\ y\in f\text{-}val\ \wedge\ (x\ rel\ y))$となる x
	<Region><hasBag><rdf:Bag>*fri*$_1$...*fri*$_n$</rdf:Bag></hasBag></Region>	$fri_1(x_1)\wedge\ldots\wedge fri_n(x_n)$となる$(x_1,\ldots,x_n)$
	fri ::=<rdf:li><EachRegion>*oR hv*$_1$ *hF*$_1$ *hv*$_2$ *hF*$_2$</EachRegion></rdf:li>	$fri(x) = R(x) \wedge (\forall y_1,y_2\ F_{h1}(y_1)\geqq T \wedge F_{h2}(y_2)\geqq T \wedge\ x\in[y_1,y_2])$

h：言語ヘッジを表す URI，*F*：メンバシップ関数を表す URI，*R*：関係を表す URI，*T*：しきい値，*C*：概念を表す URI
F_h：言語ヘッジ *h* に修飾されたメンバシップ関数，C_m：概念 *C* の適合度を返す関数．　各語の接頭辞 krsfz は省略している．

は I_V の要素数$|I_V|$を，領域制約は I_V の要素が属する概念を，値制約は I_Vの各要素を制約する．

記述するリソースが曖昧な性質を持つ場合，KRS-FZ-web 言語ではファジィ制約を用いて扱うことができる．ファジィ制約の導入と共に，概念の適合度[0,1]を各リソースに与えており，適合度がしきい値 $T\in(0,1]$以上のリソースはその概念に属すると判断される．しきい値 *T* は <rdf:Description rdf:about="" krsfz:set-threshold="*T*"/>によって与えられる．

表 2 に KRS-FZ-web 言語において使用されるファジィ制約を与えた．リソース *i* と *R* の関係にあるリソースの集合を I_V とすると，ファジィ個数制約は I_V の要素数$|I_V|$，ファジィ領域制約は I_V の要素に対する概念の適合度，値制約は I_V の各要素が，言語ヘッジ *h* によって修飾されたメンバシップ関数 F_h が与える数値となる．メンバシップ関数の定義は次のように記述される：

```
<krsfz:MembershipFunction m-id> export
  <krsfz:function>f-body</krsfz:function>
</krsfz:MembershipFunction>
f-body ::= <krsfz:bell 4ps/>|<krsfz:l-trpzd 4ps/>
  |<krsfz:r-trpzd 4ps/>|<krsfz:m-trpzd 4ps 56ps/>
4ps ::= krsfz:p1="N" krsfz:p2="N"
        krsfz:p3="N" krsfz:p4="N"
56ps ::= krsfz:p5="N" krsfz:p6="N"
```

bell，l-trpzd，r-trpzd，m-trzpd は**図 2**の各関数に相当する．言語ヘッジ *h* は KRS-FZ-web 言語の XML 名前空間で定義された次の５つがある．

```
h ::= very | more-or-less | not
    | not-very | not-more-or-less
```

例として「生徒数は非常に多い」という曖昧な表現を考える．この場合の「多い」は特定の要素の個数を意味し，「多い」を表すメンバシップ関数 many を例えば次のように定義する．

```
<krsfz:MembershipFunction rdf:ID="many">
  <krsfz:function>
```

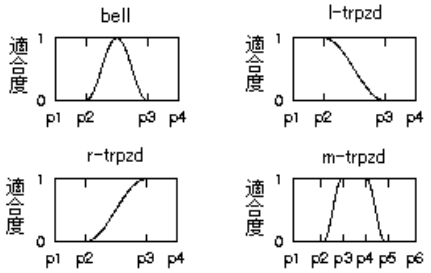

図 2 KRS-FZ-web で使用可能なメンバシップ関数

```
    <krsfz:l-trpzd krsfz:p1="0" krsfz:p2="500"
                   krsfz:p3="1000" krsfz:p4="5000"/>
  </krsfz:function>
</krsfz:MembershipFunction>
```

「非常に」に言語ヘッジ very を用いて，上述の例は次のように記述できる．

```
<krsfz:Fuzzy-number>
  <krsfz:hasVhedge rdf:resource="http://
      www.ito.ecei.tohoku.ac.jp/KRSFZ-ns#very"/>
  <krsfz:hasMemFunction rdf:resource="#many"/>
  <krsfz:onRelation rdf:resource="#hasStudent"/>
</krsfz:Fuzzy-number>
```

述語 hasStudent(x)が真となる x の数を *n*，言語ヘッジ very によって修飾されたメンバシップ関数 many を $many_{very}$，しきい値を *T* とすると，**Fuzzy-number** 構文を用いて $many_{very}(n)\geqq T$ が成立することを述べている．

2.5　インポート機能とエクスポート機能

DAML+OIL[4]は，他サイトで定義された用語を自サイトにインポートする機能を持つ．KRS-FZ-web 言語では，インポート機能に加えて，インポート可能なサイトを制限するエクスポート機能を新たに実現した．定義された用語は，エクスポートされたサイトのみがインポートを許可され，それ以外のサイトからはインポートできない．エクスポート機能 *export* は，概念や関係の定義などにおいて次のように記述される：

```
export ::= λ| open-sites | closed-sites
```

open-sites ::= $open_1$ ... $open_n$
closed-sites ::= $closed_1$ ... $closed_n$
open ::= <krsfz:open rdf:resource="*uri*"/>
closed ::= <krsfz:closed rdf:resource="*uri*"/>

*open*は指定された*uri*に用語をエクスポートし，*closed*は*uri*以外にエクスポートする．*export*が空である場合には任意のサイトにエクスポートする．次は，エクスポート機能の使用例である．

```
<krsfz:Concept rdf:ID="PassLevel"/>
  <krsfz:open rdf:resource="http://a.ac.jp/"/>
  ......
</krsfz:Concept>
```

概念「合格基準」(PassLevel)は A 大学(http://a.ac.jp/)にエクスポートされているため，A 大学のみが PassLevel の定義をインポートできる．

用語のインポートは次のように記述される：

```
<rdf:Description rdf:about="">
  <krsfz:import rdf:resource="uri "/>
</rdf:Description>
```

*uri*にある文書を調べ，自サイトにエクスポートされた用語を導入する．インポートとエクスポートの機能によりWebで用語を公開・隠蔽できる．

3 KRS-FZ-web 処理系と推論機能

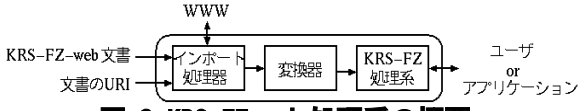

図 3 KRS-FZ-web 処理系の概要

KRS-FZ-web 言語は，KRS-FZ 処理系を基に実現された KRS-FZ-web 処理系によって処理される．KRS-FZ-web 処理系は**図 3**に与えた３つからなる．

- **インポート処理器**：インポート機能によって指定されたページをWWWから探し，文書のURIにエクスポートされた用語を導入する．処理効率はネットワークのトラフィックに依存する．
- **変換器**：KRS-FZ-web 言語の記述を，対応する KRS-FZ 言語の記述に変換する．**表 3** に変換器の実験結果を与えた．入力文書に記述された構文の数の増加に伴って処理時間も増大している．
- **KRS-FZ 処理系**：変換器が生成した KRS-FZ 言語の記述を知識ベースに保存し，ユーザやアプリケーションの要求に応じて推論を行う．

インポート処理器と変換器は静的に実行でき，結果をファイルに保存できる．実行時にファイルをロードすることでKRS-FZ-webの実行時の処理時間はKRS-FZ処理系の処理時間に等しくなる．

KRS-FZ-web では，KRS-FZ 処理系を基に非単調推論が行える．次はデフォルト規則の例である．

```
<krsfz:Concept rdf:ID="Polite">
  <krsfz:defaults>
    <rdf:Description rdf:resource="#Excellent"/>
  </krsfz:Concept>
<krsfz:Concept rdf:ID="Excellent">
  <krsfz:implies>
```

表 3 変換機を用いた実験結果

入力された文書に含まれる構文の数	20	40	60	80	100
変換処理時間[sec]	0.8	2.8	7.5	16	30

CPU:UltraSPARC-Ⅱ360MHz, Memory:640MB, OS:Solaris2.6

```
      <krsfz:more= hasValue="90">
        <krsfz:onRelation rdf:resource="#score"/>
      </krsfz:more=>
    </krsfz:implies>
  </krsfz:Concept>
  <rdf:Description rdf:ID="taro">
    <rdf:type rdf:resource="#Polite"/>
    <score>70</score>
  </rdf:Description>
```

概念 Polite は概念 Excellent をデフォルトに含み(krsfz:defaults)，概念 Excellent は score の値が 90 以上(krsfz:more=)という性質を持つ．一方，taro は概念 Polite に属する(rdf:type)と主張されたが，score が 70 である事実(<score>70</score>)は概念 Excellent が持つ性質に矛盾する．したがって taro は概念 Excellent に属さないと推論される．KRS-FZ-web では,デフォルト推論の他，Web 向きの機能と合わせて導入した 2 種類の閉世界仮説による暗黙の限定などの非単調推論が可能だが，これらは KRS-FZ 処理系の内部機構である分類器[1)]によって実現されている．

4 おわりに

知識表現システムKRS-FZに基づくWebオントロジー記述言語KRS-FZ-webとその処理系について報告した．KRS-FZ-web言語は，Semantic Webのオントロジー層を構築する機能として概念・関係の定義や事実の記述，曖昧な知識の扱いが可能なファジィ制約機能などのKRS-FZが備える機能とともに，用語のインポートとエクスポートを行うWeb向きの機能を備えている．新たに導入したエクスポート機能は，Webにおける用語の公開と隠蔽を実現する．KRS-FZ-web処理系はKRS-FZ処理系を用いて実現され，Web上で非単調推論が行える．KRS-FZ-web言語用Java APIの研究も行われており，KRS-FZ-webの応用を進めている．

参考文献

1) 森谷 俊洋，伊藤 貴康．ファジィ制約を備えた知識表現システムとその ISLISP による実現．情報処理学会論文誌，Vol.43，No.10，pp.3137-3152，2002.
2) Tim Berners-Lee, et al. The Semantic Web. Scientific American, 2001. http://www.sciam.com/2001/0501issue/0501berners-lee.html
3) Dan Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema, 2002. http://www.w3.org/TR/2000/WD-rdf-schema-20020430
4) DARPA Agent Markup Language (DAML). http://www.daml.org/
5) Mike Dean, et al. OWL Web Ontology Language 1.0 Reference, 2002. http://www.w3.org/TR/2002/WD-owl-ref-20020729