

セマンティック Web と記述論理

兼岩 憲

国立情報学研究所 情報学基礎研究系

kaneiwa@nii.ac.jp

Talk outline

1. セマンティック Web
2. 記述論理の入門
 - 研究の背景
 - 構文，意味論と知識ベース（A-Box, T-Box）
 - 表現力の縮小・拡張
 - T-BoxとA-Boxの推論タスク
 - 推論アルゴリズムと計算量
3. オントロジーのための記述論理
4. OWL から記述論理への変換(例)
5. まとめ

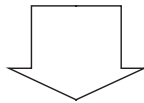
セマンティック Web

従来の Web 処理

- 情報の共有・表示 (HTML)
- テキスト処理・検索

高度な Web 処理

Web コンテンツを処理するためコンピュータが Web 情報の意味構造を解読できるようにする。



- 意味検索，情報抽出 (発見)
- データ統合
- 知的エージェント

メタデータとオントロジーと推論

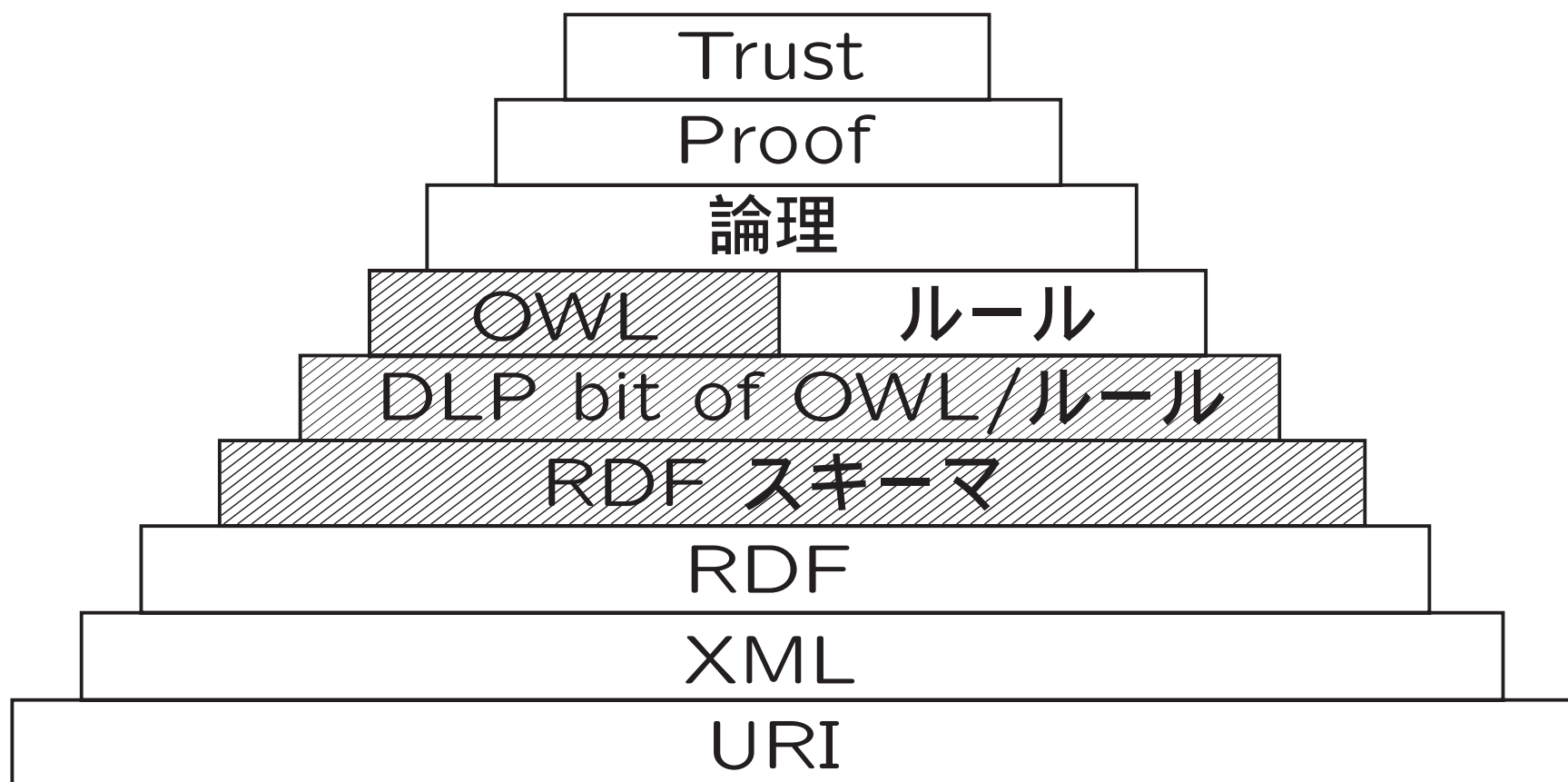
意味構造を記述するには：

- メタデータ (例えば「太郎」の年齢，住所など)
 - RDF (Resource Document Framework)
 - RDFスキーマ (RDFの語彙定義)
- オントロジー
 - Webオントロジー言語 OWL

意味構造から知的な処理を行うには：

- 推論・エージェント
 - 人工知能の技術

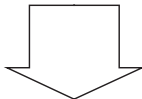
セマンティック Web 層



OWL と記述論理

OWL の3つの下位言語

- OWL Lite : フレーム言語に似た構文 , 実装向き
- OWL DL : クラスの論理表現 , プロパティ値の基数 , 個体クラス
- OWL Full : OWL DL + RDF による自由な記述

 形式的な基盤

記述論理

- 概念言語の構文と意味論
- 推論システム (RACER[Haarslev2001] など)
- 決定可能性と計算量

OWLユーザーに記述論理は必要か？

記述論理を学ぶことで：

- 論理的な記述の理解を助ける
- 論理的な推論を予想できる
- 論理的にクラスの矛盾や同一性を判断できる

オントロジーを構築する上で：

- 少なくとも記述論理のような論理的な意味をイメージする必要がある

2. 記述論理の入門：研究の背景

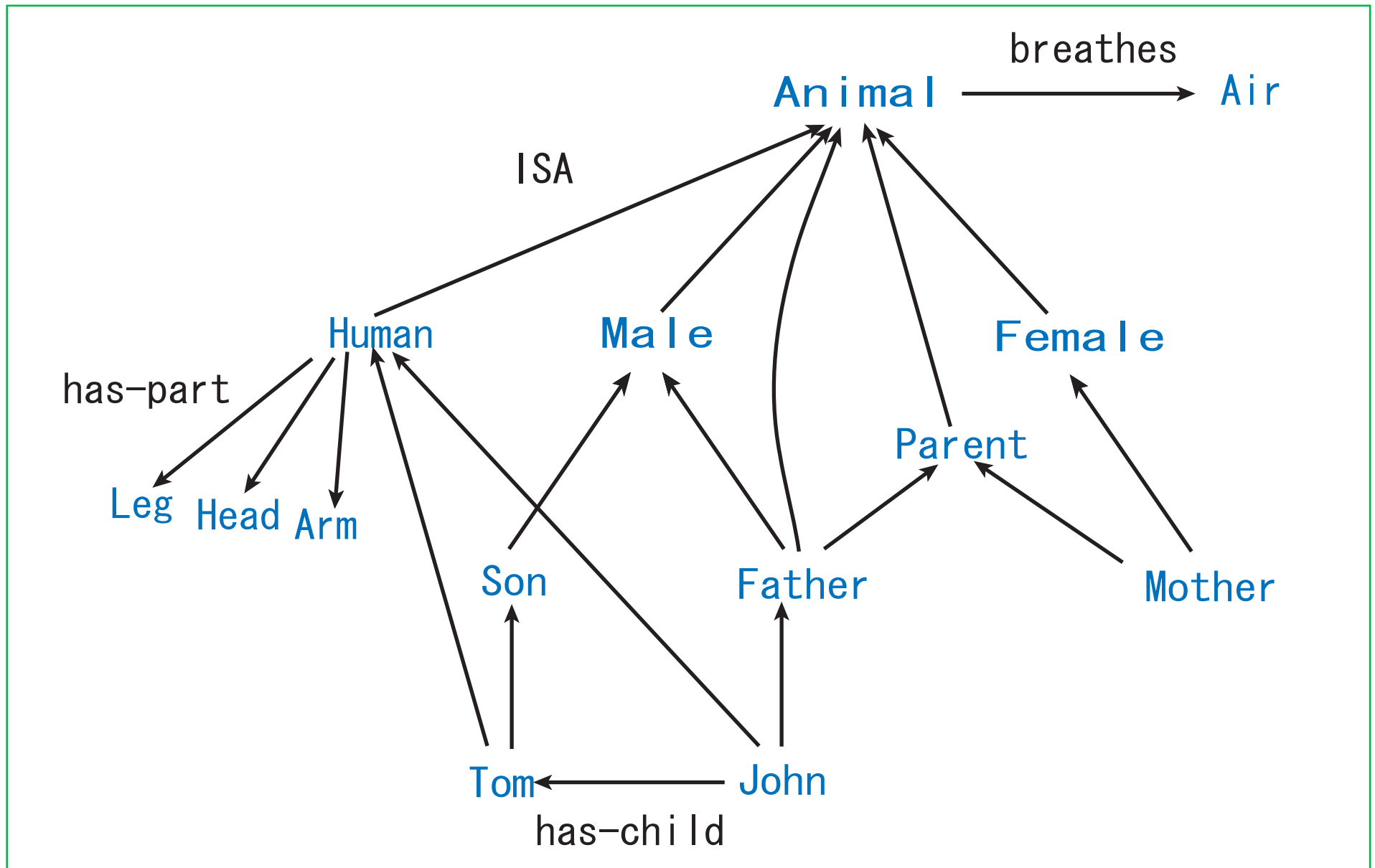
人工知能と知識表現の研究

- Semantic networks [Quillian 1968]
- Frames [Minsky 1975]
- KL-ONE [Brachman 1979, 1985]
- KRYPTON [Brachman et.al. 1983]

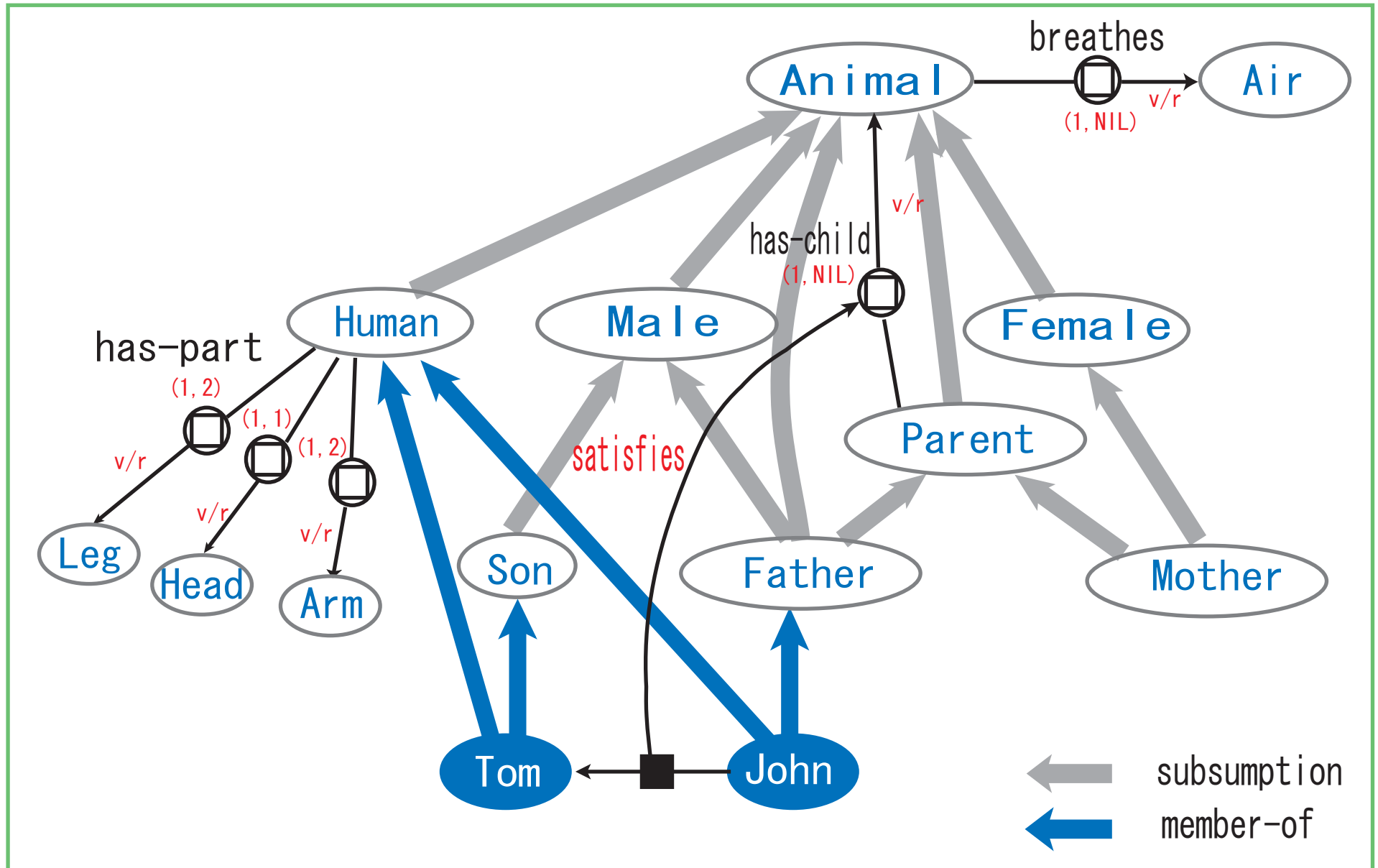
型表現をもつ論理・言語

- 順序ソート論理 [Oberschelp 1962, Walther 1988]
- 型付き論理プログラミング
- オブジェクト指向演繹言語

Semantic networks (Quillian 1968)



KL-ONE [Brachman 1979,1985]



記述論理とは？(1)

概念言語 (concept language)

概念の構成要素

概念名 : *Man, Human, Male*

ロール名 : *has-child*

個体名 : *John*

論理結合子 : $\neg, \sqcap, \sqcup$

量化子 : $\forall, \exists$

構成

複雑な概念

Human $\sqcap$ Male

Human $\sqcap \exists has-child. Human$

記述

知識ベース *KB*

T-Box $\mathcal{T}$

概念階層

A-Box $\mathcal{A}$

言明的な知識

適用

推論アルゴリズム

記述論理とは？(2)

特徴

- 推論が決定可能である
- 言語を拡張・縮小して表現力を分類
- 表現力と計算量のトレードオフを分析
- 開世界仮説(OWA)に基づく

望まれる性質

- 知識ベースの推論：EXPTIMEまで
(多項式時間でなくてもOK?)

記述論理の構文と意味論

概念言語 (Concept language)

CN : $\top, \perp$ を含む概念名 A の集合 (e.g. Human)

RN : ロール名 P の集合 (e.g. hasChild)

IN : 個体名 a, b の集合 (e.g. John)

■ 解釈 $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$

$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ (特に $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \perp^{\mathcal{I}} = \emptyset$)

$P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$

$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$

■ $\mathcal{AL}$ 概念 [Schmidt-Schauß 1991]

$$C, D \longrightarrow A \mid \top \mid \perp \mid \neg A \mid C \sqcap D \mid \forall P.C \mid \exists P.\top$$

■ $\mathcal{ALC}$ 概念

$$C, D \longrightarrow A \mid \top \mid \perp \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall P.C \mid \exists P.C$$

■ $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} - C^{\mathcal{I}}$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

Example

$Animal \sqcap Male$ 「雄の動物」

$Human \sqcap \neg(Rich \sqcup Poor)$

「金持ちでも貧乏でもない人」

■ $(\forall P.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall e[(a, e) \in P^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}]\}$

Example

$\forall hasChild.Male$

「子供をもつならば，すべて雄である個体」

$Human \sqcap \forall hasChild.\perp$ 「子供を持たない人」
 ($= Human \sqcap \neg \exists hasChild.\top$)

■ $(\exists P.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists e[(a, e) \in P^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}]\}$

Example

$\exists hasPart.Leg \sqcap Animal$ 「足をもった動物」

$\exists be-carried-by.(Vehicle \sqcap \exists hasWheel.\top \sqcap White)$

「車輪をもつ白い乗り物によって運ばれている個体」

知識ベース (T-Box, A-Box)

知識ベース $KB = (\mathcal{T}, \mathcal{A})$

■ T-Box $\mathcal{T}$ (terminological knowledge)

$C \equiv D$: 概念等式 (concept equation)

$C \sqsubseteq D$: 概念包含 (concept inclusion)

$A \equiv C$: 概念定義 (concept definition)

$A \sqsubseteq C$: プリミティブ概念包含 (primitive concept inclusion)

■ A-Box $\mathcal{A}$ (assertional knowledge)

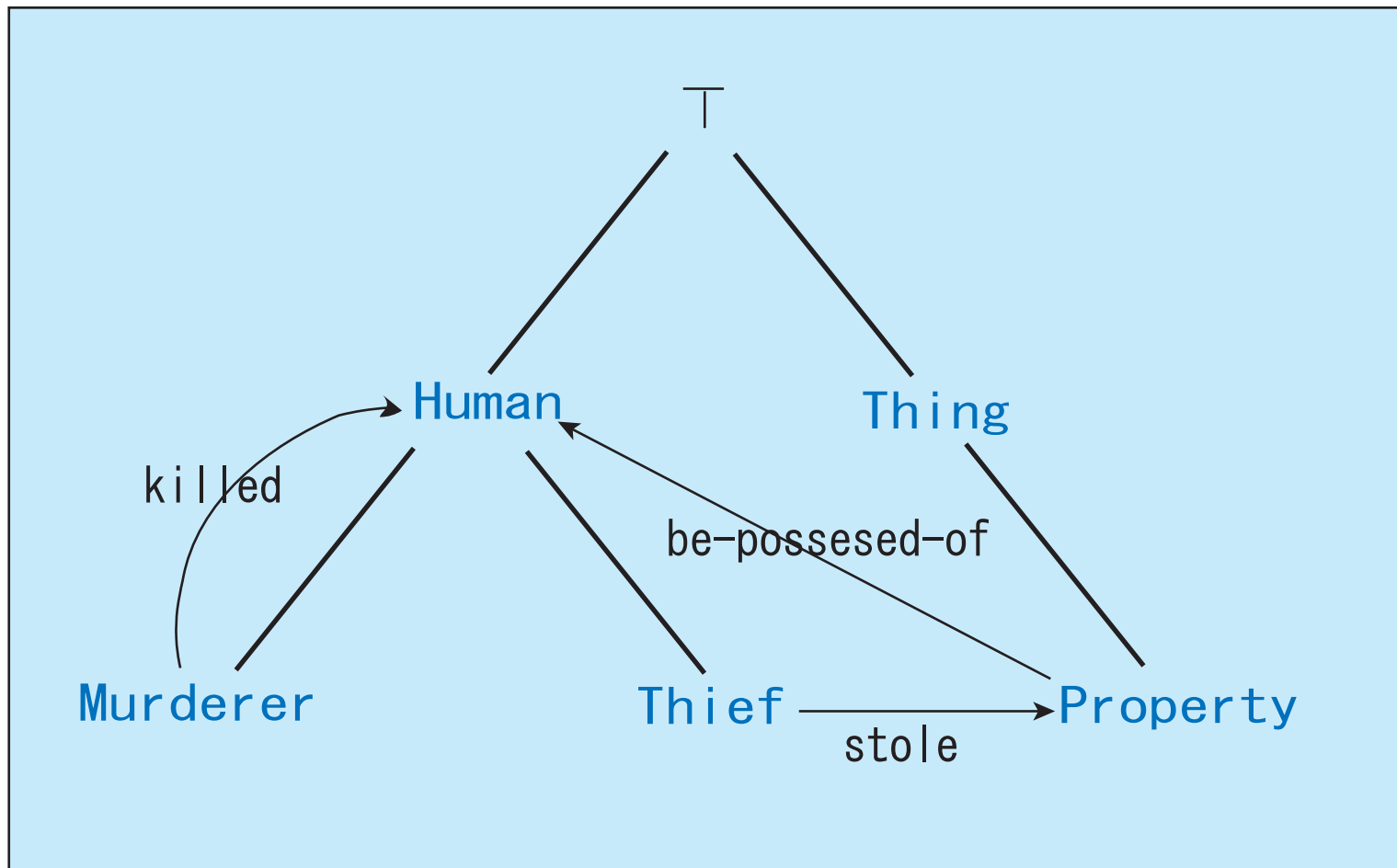
$C(a)$: 概念のインスタンス

$P(a, b)$: ロールのインスタンス

Example

$\top$ -Box $\mathcal{T}_1 = \{$
 $Property \equiv \forall be\text{-possessed-of}. Human \sqcap Thing,$
 $Murderer \equiv \exists killed.Human \sqcap Human,$
 $Thief \equiv \exists stole.Property \sqcap Human$
 $\}$

概念階層：



概念言語の縮小・拡張

- $\mathcal{FL}^-$ 言語ファミリー: $\mathcal{FL}^-[\mathcal{E}][\mathcal{U}][\mathcal{N}][\mathcal{Q}][\mathcal{R}][\mathcal{I}][\mathcal{O}][\mathcal{B}]$

$$\mathcal{FL}^-: C \sqcap D + \forall P.C + \exists P.\top$$

$$\mathcal{FL}: \mathcal{FL}^- + R|_C$$

- $\mathcal{AL}$ 言語ファミリー: $\mathcal{AL}[\mathcal{E}][\mathcal{U}][\mathcal{N}][\mathcal{Q}][\mathcal{R}][\mathcal{I}][\mathcal{O}][\mathcal{B}]$

$$\mathcal{AL}: \mathcal{FL}^- + \perp, \top + \neg A$$

$$\mathcal{ALE}: \mathcal{AL} + \exists P.C(\mathcal{E})$$

$$\mathcal{ALU}: \mathcal{AL} + C \sqcup D(\mathcal{U})$$

$$\mathcal{ALEU}: \mathcal{ALC}$$

■ 制限付き数量限定 ($\mathcal{Q}$), 制限なし数量限定 ($\mathcal{N}$)

$$\geq nR.C, \geq nR.\top$$

$$\leq nR.C, \leq nR.\top$$

$$\{a \in \Delta^{\mathcal{I}} \mid \#\{e \in \Delta^{\mathcal{I}} \mid (a, e) \in R^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\} \geq n\}$$

$$\{a \in \Delta^{\mathcal{I}} \mid \#\{e \in \Delta^{\mathcal{I}} \mid (a, e) \in R^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\} \leq n\}$$

Example

$$\geq 2 \text{ hasCar.Red}$$

「少なくとも2台の赤い車を所有する個体」

■ ロール積 ($\mathcal{R}$), 逆ロール ($\mathcal{I}$)

$$P_1 \sqcap P_2$$

$$P^-$$

$$P_1^{\mathcal{I}} \cap P_2^{\mathcal{I}}$$

$$\{(a, e) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (e, a) \in P^{\mathcal{I}}\}$$

Example

$\exists hasChild^- . Rich$ 「金持ちの親をもつ個体」

■ ONE-OF($\mathcal{O}$) , FILLS($\mathcal{B}$)

$\{a_1, \dots, a_n\}$

$R: a$

$\{a_1^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\}$

$\{e \in \Delta^{\mathcal{I}} \mid (e, a^{\mathcal{I}}) \in R^{\mathcal{I}}\}$

Example

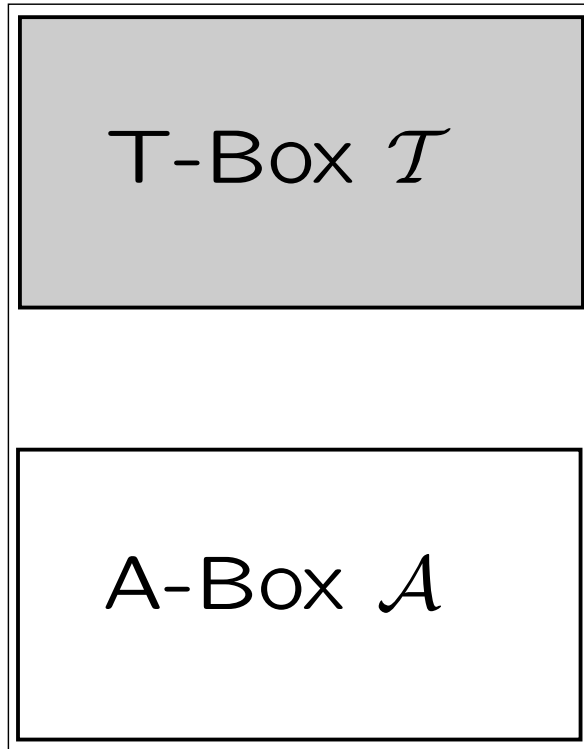
$\{John, Mary, Tom, Bob\}$

$hasChild: Tom$ 「トムを子供にもつ親」

$(= \exists hasChild. \{Tom\})$

T-Boxの推論タスク

知識ベース KB



- ← (1) 概念の充足可能性 $\mathcal{I} \models C$
- ← (2) 概念間の包摂 $\mathcal{I} \models C \sqsubseteq D$
- ← (3) 概念間の同値 $\mathcal{I} \models C \equiv D$

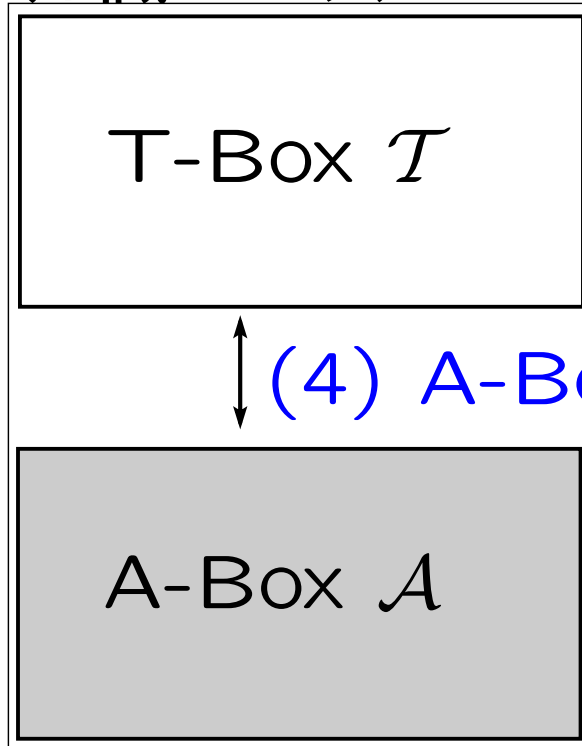
(1) $\mathcal{I} \models \mathcal{T}$ かつ $C^{\mathcal{I}} \neq \emptyset$ を判定

(2) $\mathcal{I} \models \mathcal{T}$ となるすべての $\mathcal{I}$ に対して, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ を判定

(3) $\mathcal{I} \models \mathcal{T}$ となるすべての $\mathcal{I}$ に対して, $C^{\mathcal{I}} = D^{\mathcal{I}}$ を判定

A-Boxの推論タスク

知識ベース KB



← (1) 概念の充足可能性 $\mathcal{I} \models C$

← (2) 概念間の包摂 $\mathcal{I} \models C \sqsubseteq D$

← (3) 概念間の同値 $\mathcal{I} \models C \equiv D$

↕ (4) A-Box $\mathcal{A}$ とT-Box $\mathcal{T}$ の無矛盾性

← (5) 概念のインスタンス $\mathcal{A} \models C(a)$

(4) $\mathcal{I} \models \mathcal{A}$ かつ $\mathcal{I} \models \mathcal{T}$ を判定

(5) $\mathcal{I} \models \mathcal{A}$ となるすべての $\mathcal{I}$ に対して, $C(a)$ を判定

推論アルゴリズム

■ 代表的な推論方式

- 構造比較アルゴリズム ($\mathcal{FL}$ など) $\Leftarrow$ 構文的な方法
- タブロー法に基づいたアルゴリズム ($\mathcal{ALC}$ など) $\Leftarrow$ 意味論的な方法

■ タブロー法による推論 (制約システム)

- 制約 : $x : C, (x, y) : P, x \neq y$
- clash : $\{x : \perp\}, \{x : A, x : \neg A\}$
- 完全な制約集合 S_i : これ以上完備化規則を適用できない

■ 制約集合 S_i の完備化規則

$\sqcap$ -規則

$$S_i \longrightarrow S_i \cup \{x : C_1, x : C_2\}$$

if $x : C_1 \sqcap C_2 \in S_i$ かつ $\{x : C_1, x : C_2\} \not\subseteq S_i$

$\sqcup$ -規則

$$S_i \longrightarrow S_i \cup \{x : C_1\}, S_i \cup \{x : C_2\}$$

if $x : C_1 \sqcup C_2 \in S_i$ かつ $\{x : C_1, x : C_2\} \cap S_i = \emptyset$

$\exists$ -規則

$$S_i \longrightarrow S_i \cup \{y : C, (x, y) : P\}$$

if $x : \exists P.C \in S_i$ かつ $\{z \mid z : C, (x, z) : P \in S_i\} \neq \emptyset$
(但し, $y \notin V(S_i)$)

$\forall$ -規則

$$S_i \longrightarrow S_i \cup \{y : C\}$$

if $x : \forall P.C, (x, y) : P \in S_i$ かつ $y : C \notin S_i$

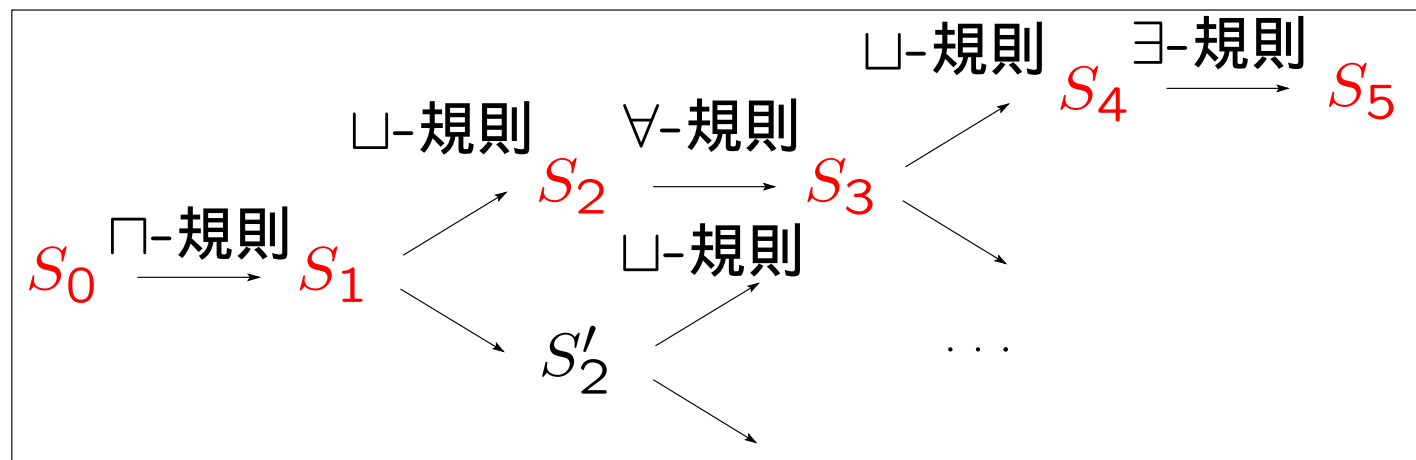
■ 概念 C の充足可能性アルゴリズム ($\emptyset \models C$)

1. C を NNF (Normal Negation Form) に変換

($\neg\neg A_1 \sqcap \neg(A_2 \sqcup A_3)$ から $A_1 \sqcap (\neg A_2 \sqcap \neg A_3)$)

2. 制約集合 $S_0 = \{x : NNF(C)\}$ を完備化規則に適用

3. 制約集合の有限列 $S_0, S_1, S_2, \dots, S_n$ を得る



4. ある完全な S_n が clash を含まない $\Rightarrow C$ は充足可能

すべての完全な S_n が clash を含む $\Rightarrow C$ は充足不可能

充足可能性問題への還元

■ タブロー法による充足可能性の判定アルゴリズムを他の推論タスクに適用できる．

- 概念間の包摂 $\mathcal{T} \models C \sqsubseteq D \Leftrightarrow C' \sqcap \neg D'$ が充足不可能
- 概念間の同値 $\mathcal{T} \models C \equiv D \Leftrightarrow (C' \sqcap \neg D') \sqcup (\neg C' \sqcap D')$ が充足不可能
- A-Box $\mathcal{A}$ と T-Box $\mathcal{T}$ の無矛盾性 $\Leftrightarrow \mathcal{A}'$ が充足可能
- 概念のインスタンス $\mathcal{A} \models C(a) \Leftrightarrow \mathcal{A} \cup \{\neg C(a)\}$ が充足不可能

Example $\mathcal{T}_1 \models Thief \sqsubseteq Human$ の判定

$\mathcal{T} \models C \sqsubseteq D$ の判定方法

$\mathcal{T} \models C \sqsubseteq D$	$\stackrel{(1)}{\Leftrightarrow}$	$\emptyset \models C' \sqsubseteq D'$
	$\stackrel{(2),(3)}{\Leftrightarrow}$	$C' \sqcap \neg D'$ は充足不可能

(1) $Property \equiv \forall be\text{-}possessed\text{-}of.Human \sqcap Thing$,
 $Thief \equiv \exists stole.Property \sqcap Human$ を展開する .

$$Thief \sqsubseteq Human$$



$$\begin{aligned} &\exists stole.(\forall be\text{-}possessed\text{-}of.Human \sqcap Thing) \sqcap Human \\ &\sqsubseteq Human \end{aligned}$$

(2) $C' \sqsubseteq D'$ を $C' \sqcap \neg D'$ の形にする .

$$\begin{aligned} &(\exists stole.(\forall be\text{-}possessed\text{-}of.Human \sqcap Thing) \sqcap Human) \\ &\sqcap \neg Human \end{aligned}$$

(3) $C' \sqcap \neg D'$ をタブロー法に適用する .

$$S_0 = \{ x : (\exists sto. (\forall pos. H \sqcap T) \sqcap H) \sqcap \neg H \}$$

↓ $\sqcap$ -規則

$$S_1 = S_0 \cup \{ x : \exists sto. (\forall pos. H \sqcap T) \sqcap H, x : \neg H \}$$

↓ $\sqcap$ -規則

$$S_2 = S_1 \cup \{ x : \exists sto. (\forall pos. H \sqcap T), x : H \}$$

↓ $\exists$ -規則

$$S_3 = S_2 \cup \{ (x, y) : sto, y : \forall pos. H \sqcap T \}$$

↓ $\sqcap$ -規則

$$S_4 = S_3 \cup \{ y : \forall pos. H, y : T \}$$

S_4 は完全であり , $\{x : H, x : \neg H\}$ を含む $\Rightarrow$ 充足不可能

故に , $\mathcal{T}_1 \models Thief \sqsubseteq Human$

3. オントロジーのための記述論理 [Horrocks 1999]

■ 推移的ロール ($\mathcal{S}$) , 関数的ロール ($\mathcal{F}$) , ロール階層 ($\mathcal{H}$)

$$P \in RN^+$$

$$P \in FN$$

$$P_1 \sqsubseteq P_2$$

$$P^{\mathcal{I}} = (P^{\mathcal{I}})^+$$

$$(a, e) \in P^{\mathcal{I}} \wedge (a, e') \in P^{\mathcal{I}} \Rightarrow e = e'$$

$$P_1^{\mathcal{I}} \subseteq P_2^{\mathcal{I}}$$

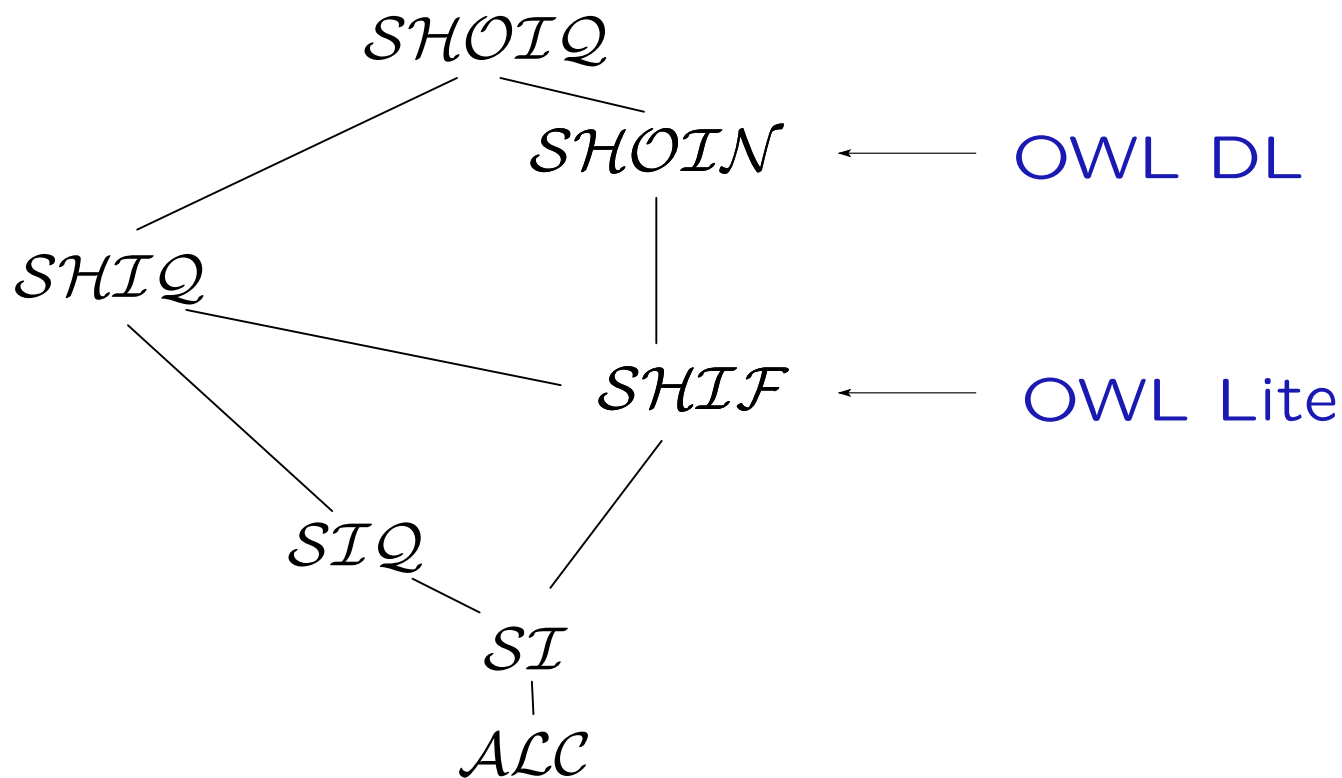
Example

- 推移的なロール : *bigger-than*
- 関数的なロール : *father*
- ロール階層 : *hasDaughter* $\sqsubseteq$ *hasChild*

■ SI 言語ファミリー : $SI[\mathcal{H}][\mathcal{Q}][\mathcal{N}][\mathcal{F}][\mathcal{I}][\mathcal{O}]$

$SI = \mathcal{ALC} + \text{推移的ルール}(S) + \text{逆ルール}(I)$

SI 言語の階層



4. OWL から記述論理への変換：個体クラス

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Tom" />
      </owl:oneOf>
    </owl:Class>
    <owl:Class>
      <owl:oneOf rdf:parseType="Collection">
        <owl:Thing rdf:about="#Tom" />
        <owl:Thing rdf:about="#John" />
      </owl:oneOf>
    </owl:Class>
  </owl:intersectionOf>
</owl:Class>
```

$\{\text{Tom}\} \sqcap \{\text{Tom}, \text{John}\}$

2つのプロパティ表現

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasChild" />  
  <owl:allValuesFrom rdf:resource="#Male" />  
</owl:Restriction>
```

$\forall hasChild.Male$

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasPart" />  
  <owl:someValuesFrom rdf:resource="#Leg" />  
</owl:Restriction>
```

$\exists hasPart.Leg$

プロパティ 値

```
<owl:Restriction>  
  <owl:onProperty rdf:resource="#hasChild" />  
  <owl:hasValue rdf:resource="#Tom" />  
</owl:Restriction>
```

$\exists hasChild.\{Tom\}$

プロパティ 値の基数宣言

```
<owl:intersectionOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#hasChild" />
    <owl:maxCardinality rdf:datatype="&xsd;
      nonNegativeInteger">5</owl:maxCardinality>
  </owl:Restriction>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#hasChild" />
    <owl:minCardinality rdf:datatype="&xsd;
      nonNegativeInteger">2</owl:minCardinality>
  </owl:Restriction>
</owl:intersectionOf>
```

$\leq 5hasChild.\top \sqcap \geq 2hasChild.\top$

クラスの等価関係

```
<owl:Class rdf:ID="Murderer">
  <owl:equivalentClass>
    <owl:intersectionOf>
      <owl:Class>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#killed"/>
          <owl:someValuesFrom rdf:resource="#Human"/>
        </owl:Restriction>
      </owl:Class>
      <owl:Class>
        <owl:Class rdf:about="#Human"/>
      </owl:Class>
    </owl:intersectionOf>
  </owl:equivalentClass>
</owl:Class>
```

$Murderer \equiv \exists killed.Human \sqcap Human$

クラスの否定表現と排他関係

```
<owl:Class rdf:ID="Wife">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:complementOf>
        <owl:Class rdf:about="#Male"/>
      </owl:complementOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
```

$Wife \sqsubseteq \neg Male$

```
<owl:Class rdf:about="#Man">
  <owl:disjointWith rdf:resource="#Woman"/>
</owl:Class>
```

$Man \sqsubseteq \neg Woman$

オブジェクトプロパティと和クラス

```
<owl:ObjectProperty rdf:ID="hasBankAccount">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Person"/>
        <owl:Class rdf:about="#Corporation"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
```

$\exists hasBankAccount.\top \sqsubseteq Person \sqcup Corporation$

サブプロパティ関係

```
<owl:ObjectProperty rdf:ID="hasMother">  
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>  
</owl:ObjectProperty>
```

hasMother $\sqsubseteq$ hasParent

```
<owl:ObjectProperty rdf:ID="hasChild">  
  <owl:inverseOf rdf:resource="#hasParent"/>  
</owl:ObjectProperty>
```

hasChild $\equiv$ hasParent⁻

関数的オブジェクトプロパティ

```
<owl:ObjectProperty rdf:ID="husband">  
  <rdf:type          rdf:resource="&owl;  
                                FunctionalProperty" />  
  <rdfs:domain rdf:resource="#Woman" />  
  <rdfs:range  rdf:resource="#Man"  />  
</owl:ObjectProperty>
```

$\top \sqsubseteq \leq 1 \textit{husband} . \top$

$\exists \textit{husband} . \top \sqsubseteq \textit{Woman}$

$\top \sqsubseteq \forall \textit{husband} . \textit{Man}$

OWLの推論とは？

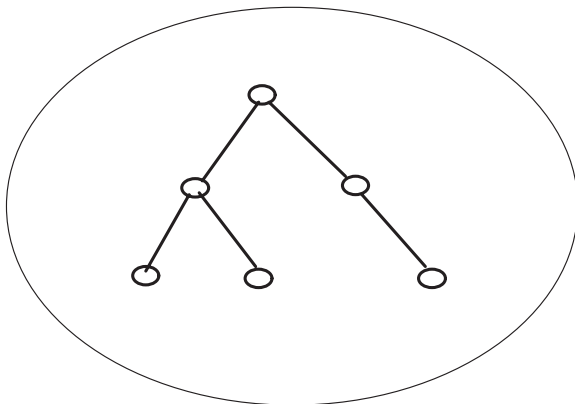
OWLオントロジーは、クラス/個体/プロパティ公理の有限集合である。

$$\mathcal{O} = \{\text{公理1, 公理2, ...}\}$$

オントロジー含意(entailment)とは？

$$\mathcal{O} \models \{\text{公理}\}$$

オントロジー $\mathcal{O}$



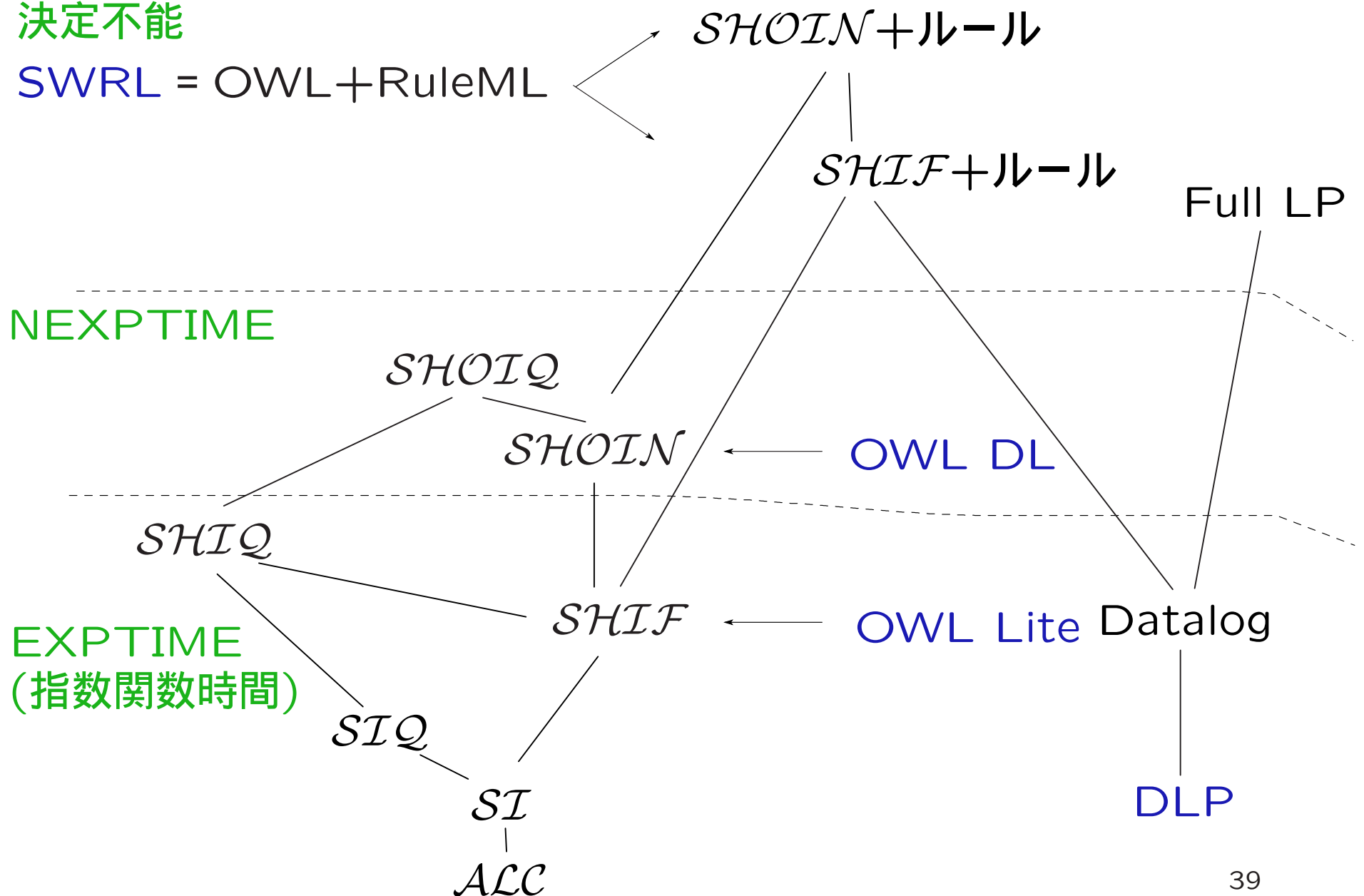
$\models \{\text{SubClassOf(ライオン 動物)}\}$

$\models \{\text{Individual(太郎 type(人間))}\}$

OWLと記述論理の計算量

決定不能

SWRL = OWL + RuleML



5. まとめ

記述論理の特徴・優位性

- 概念(または概念階層)の記述に特化して,一階述語論理にはない豊富な表現
- 形式的な論理体系: 構文,意味論および推論の健全性・完全性
 - ⇒ オントロジー記述言語の理論的基盤
- 計算量に関する徹底的な研究
- 計算可能な推論タスクとそのアルゴリズム
 - ⇒ オントロジーの推論システムが実装可能
- 表現力と計算量のトレードオフへの取り組み

記述論理の解説論文・解説本(1)

- (1) Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider P. F.: *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
- (2) Donini, F. D., Lenzerini, M., Nardi, D., and Schaerf, A.: Reasoning in Description Logic, in Brewka, G. ed., *Principles of Knowledge Representation*, CSLI Publications, FoLLI (1996).
- (3) Calvanese, D., Giacomo, G. D., Lenzerini, M., and Nardi, D.: Reasoning in Expressive Description Logics, in *Handbook of Automated Reasoning*, pp. 1581–1634 (2001).
- (4) Kusters, R.: *Non-standard inferences in description logics*, LNCS 2100, Springer-Verlag (2001).
- (5) Horrocks, I. and Patel-Schneider, P. F.: Reducing OWL entailment to description logic satisfiability, *J. of Web Semantics*, (2004).

記述論理の解説論文・解説本(2)

- (1) 兼岩憲, 佐藤健: レクチャーシリーズ「哲学とAIにおける対象世界モデリング」
〔第6回〕DL: Description Logics, 人工知能学会誌, Vol. 18, No. 1(2003),
pp. 73–82.
- (2) 兼岩憲: 特集号<セマンティック Web と計算機科学>「OWL の推論とその
計算量」(解説論文), コンピュータソフトウェア, 10月号, 2005.

■ データ型 d とデータ型ルール T

$\forall T.d$

$\exists T.d$

$\{v_1, \dots, v_n\}$

$\{a \in \Delta^{\mathcal{I}} \mid \forall e[(a, e) \in T^{\mathcal{I}} \rightarrow e \in d^{\mathcal{D}}]\}$

$\{a \in \Delta^{\mathcal{I}} \mid \exists e[(a, e) \in T^{\mathcal{I}} \wedge e \in d^{\mathcal{D}}]\}$

$\{v_1^{\mathcal{D}}, \dots, v_n^{\mathcal{D}}\}$

Example

- $\forall age.Nat$ は「年齢をもつならば、それは自然数である
個体」
- $\{1991, 2003, 2004\}$

■ データ型ロールの数量限定

$$\geq nT$$

$$\leq nT$$

$$\{a \in \Delta^{\mathcal{I}} \mid \#\{e \in \Delta_{\mathbf{D}}^{\mathcal{I}} \mid (a, e) \in T^{\mathcal{I}}\} \geq n\}$$

$$\{a \in \Delta^{\mathcal{I}} \mid \#\{e \in \Delta_{\mathbf{D}}^{\mathcal{I}} \mid (a, e) \in T^{\mathcal{I}}\} \leq n\}$$

Example

- $\geq 2stringName$ 「2つ以上名前(文字列)をもつ個体」

■ 知識ベース T-Box $\mathcal{T}$

$$P_1 \sqsubseteq P_2, T_1 \sqsubseteq T_2 : \quad \text{ロール階層}$$

■ 知識ベース A-Box $\mathcal{A}$

$$T(a, v) : \quad \text{データ型ロールのインスタンス}$$