

Description Logicsについて

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

- DL 研究の背景
- DL 言語の構文 , 意味論と知識ベース (A-Box, T-Box)
- DL 言語の縮小・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
- オントロジー記述のための DL
- 最近の DL 研究とまとめ

DL 研究の背景

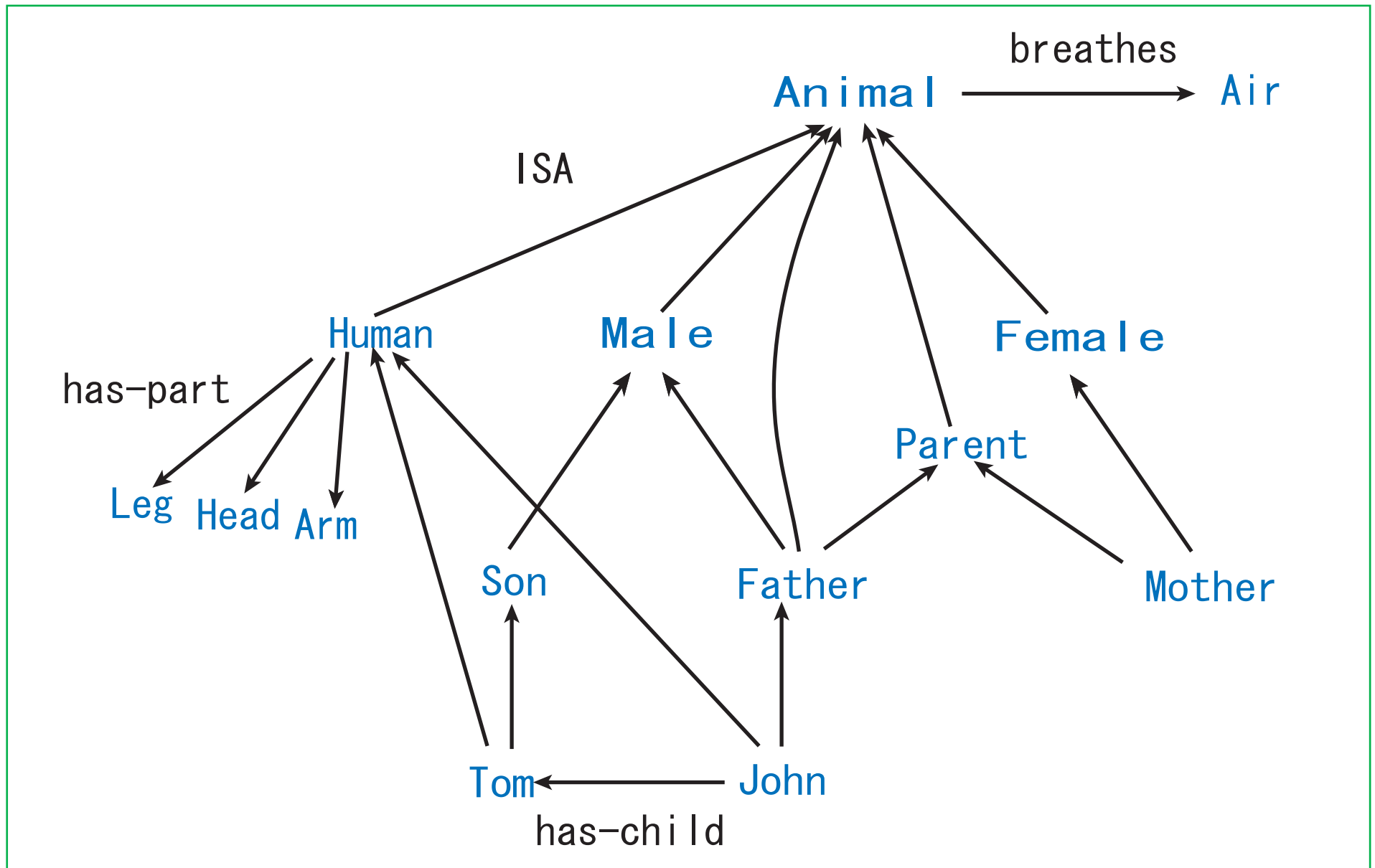
人工知能と知識表現の研究

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

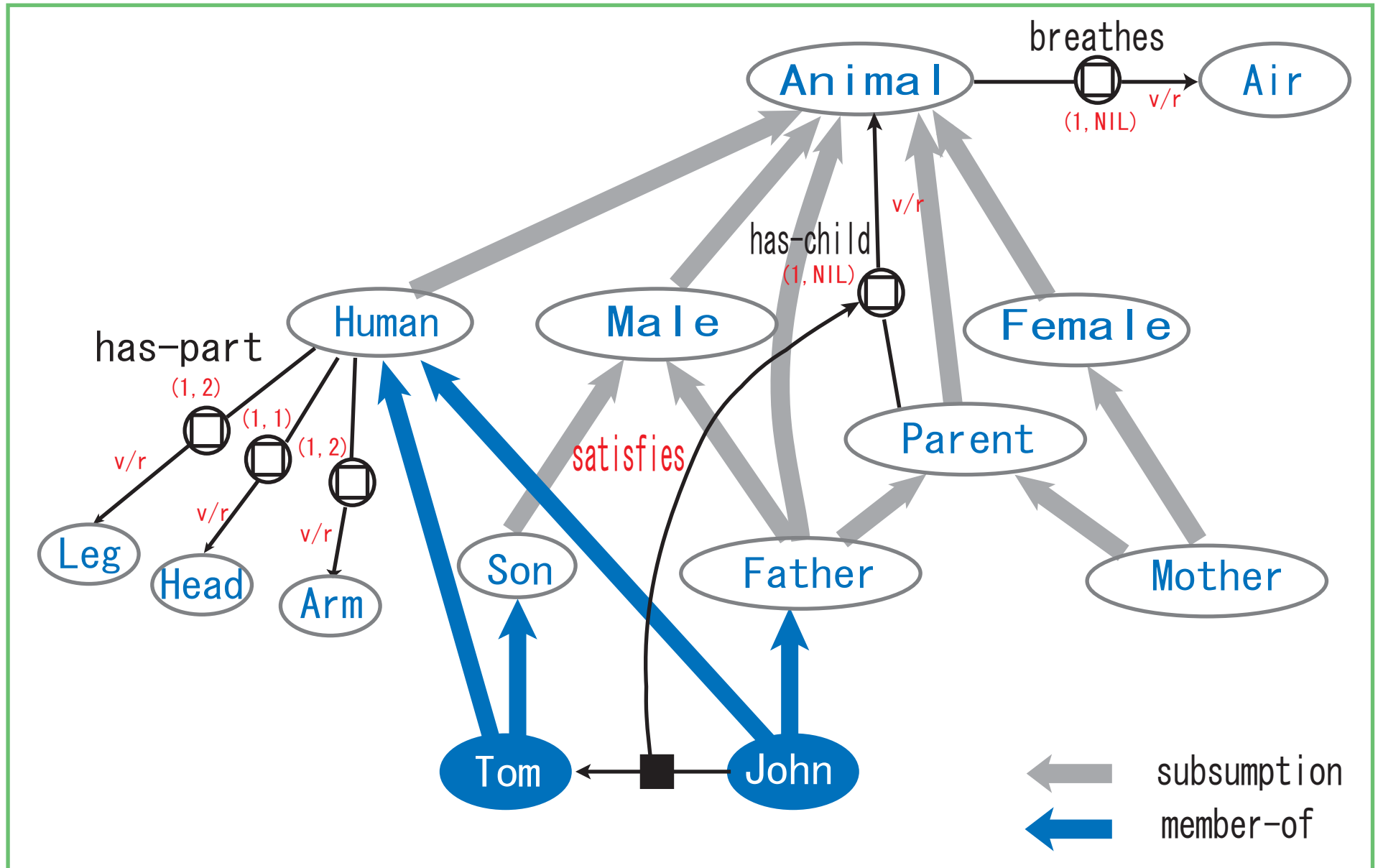
型表現をもつ論理・言語

- 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]



Description Logicとは？

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 $\sqcap \exists has-child. Human$

知識ベース *KB*

T-Box $\mathcal{T}$

概念階層

A-Box $\mathcal{A}$

言明的な知識

記述

適用

推論アルゴリズム

DL 言語の構文と意味論

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. has-child)

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」

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

Example

$\forall has-child.Male$

「子供をもつならば，すべて雄である個体」

$Human \sqcap \forall has-child.\perp$ 「子供を持たない人」

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

Example

$\exists has-part.Leg \sqcap Animal$ 「足をもった動物」

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

「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)

DL 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)

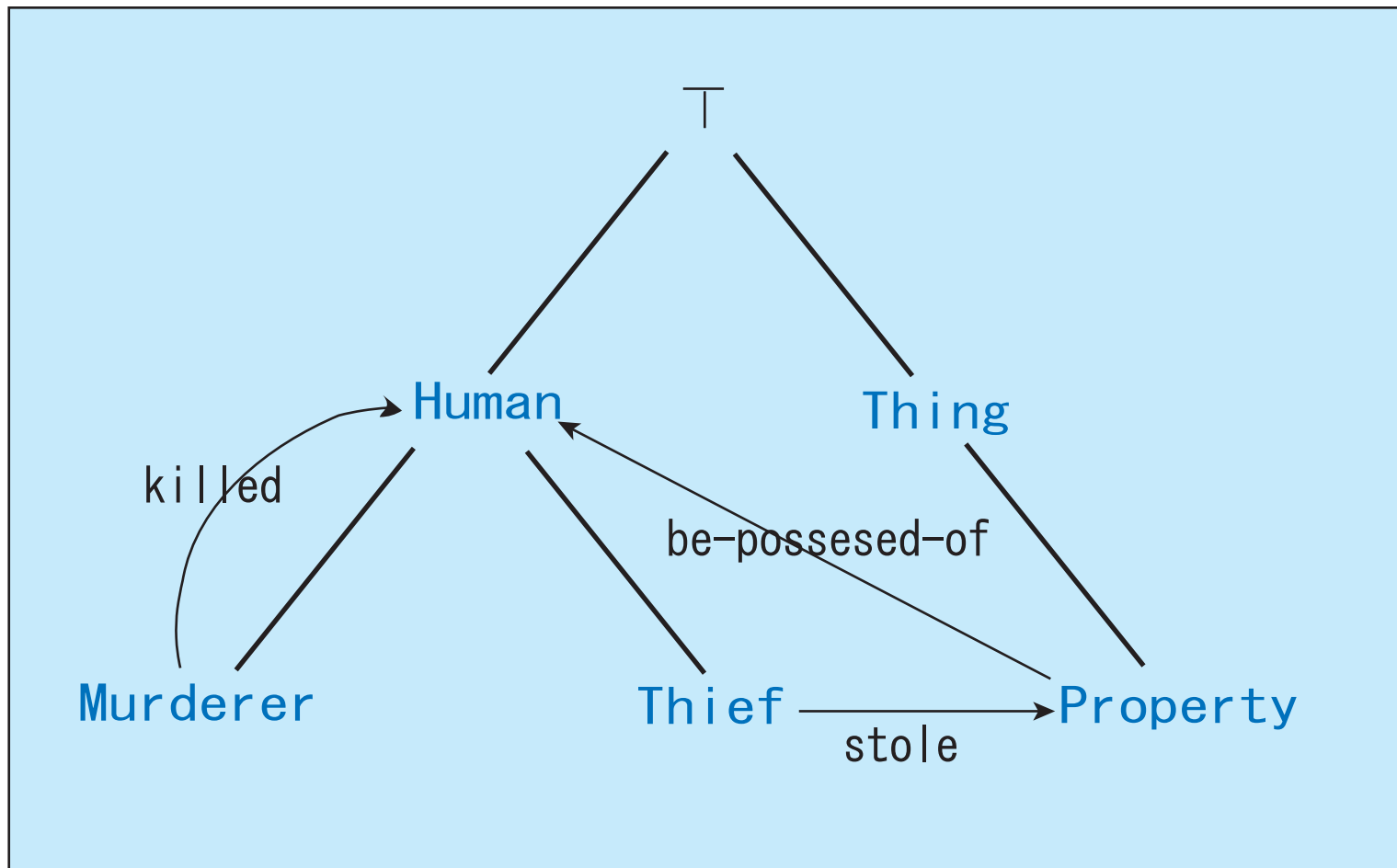
$C(a)$: 概念のインスタンス

$P(a, b)$: ロールのインスタンス

Example

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

概念階層：



DL 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 ($\mathcal{N}$)

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

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

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

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

Example

$$\geq 2 \text{ has-car.Red}$$

「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 ($\mathcal{I}$)

$$P_1 \sqcap P_2$$

$$P^-$$

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

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

Example

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

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

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

$R: a$

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

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

Example

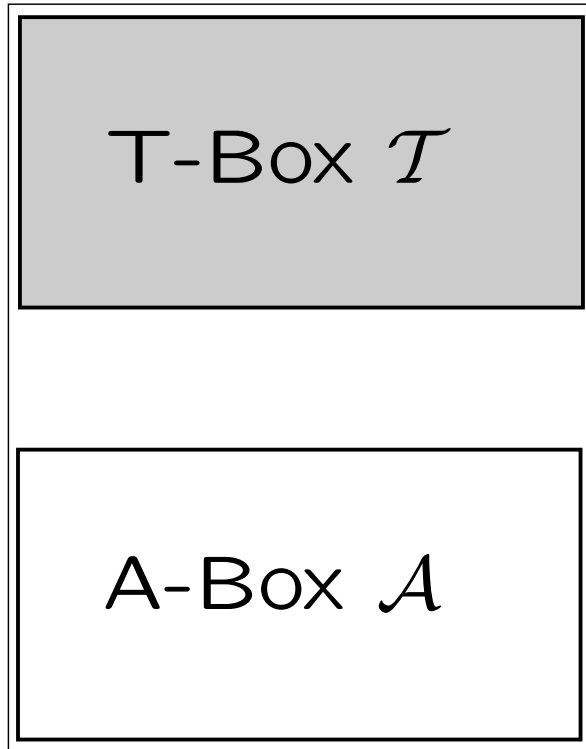
$\{John, Mary, Tom, Bob\}$

$has-child: Tom$ 「トムを子供にもつ親」

$(= \exists has-child. \{Tom\})$

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 $\mathcal{I} \models C \equiv D$

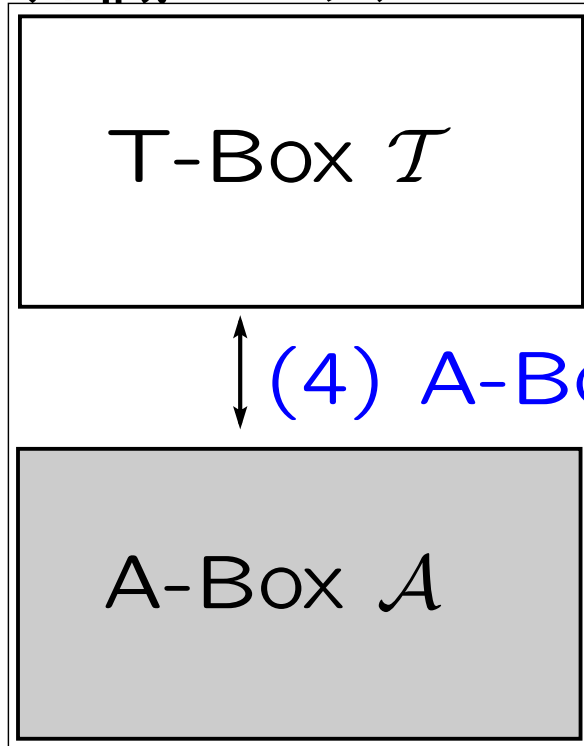
(1) $\mathcal{I} \models \mathcal{T}$ かつ $C^{\mathcal{I}} \neq \emptyset$ を判定

(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 $y : C \notin S_i$

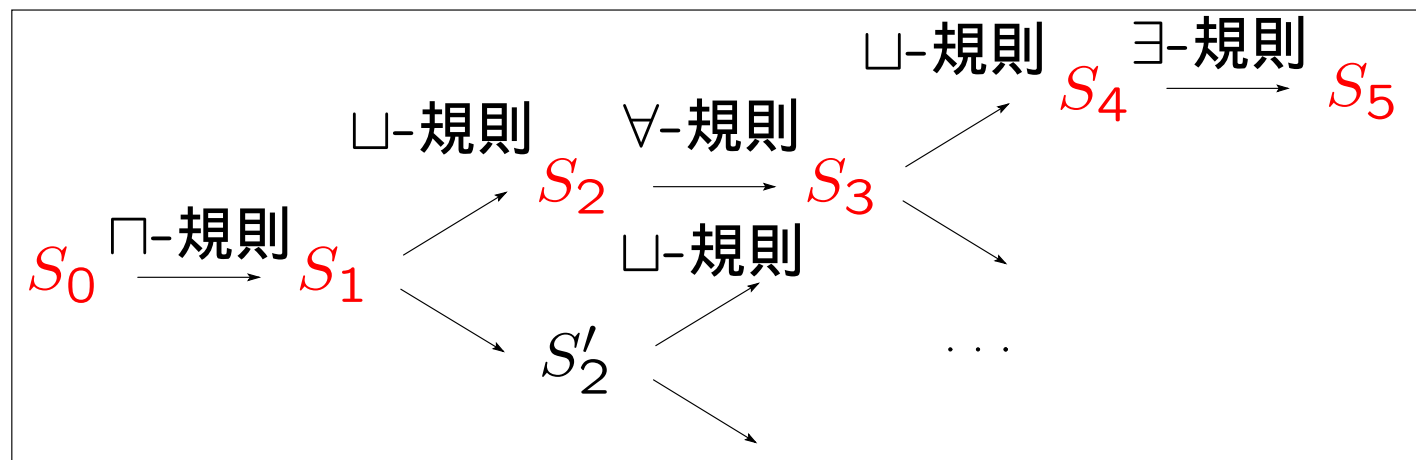
■ 概念 C の充足可能性アルゴリズム ($\emptyset \models C$)

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 .

$$\begin{array}{c} Thief \sqsubseteq Human \\ \downarrow \\ \exists stole.(\forall be\text{-}possessed\text{-}of.Human \sqcap Thing) \sqcap Human \\ \sqsubseteq Human \end{array}$$

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

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

(3) $C' \sqcap \neg D'$ をタブロー法に適用する .

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

$\downarrow \sqcap$ -規則

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

$\downarrow \sqcap$ -規則

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

$\downarrow \exists$ -規則

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

$\downarrow \sqcap$ -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 , $\mathcal{T}_1 \models Thief \sqsubseteq Human$

推論タスクの計算量

DL 言語	包摂関係 ($\emptyset \models C \sqsubseteq D$)	充足不可能性 ($\emptyset \not\models C$)
$\mathcal{FL}^-$	P	-
$\mathcal{FL}$	PSPACE	-
$\mathcal{AL}, \mathcal{ALN}$	P	P
$\mathcal{ALU}, \mathcal{ALUN}$	co-NP	co-NP
$\mathcal{ALE}, \mathcal{ALR}, \mathcal{ALER}$	NP	NP
$\mathcal{ALC}, \mathcal{ALCNR}$	PSPACE	PSPACE

オントロジー記述のためのDL

■ $SHIQ$ [Horrocks 1999]

- $ALCQI$ + 推移的ルール R^+ + 関数的ルール $\mathcal{F}$ + ルール階層 $\mathcal{H}$

Example

- 推移的なルール : *bigger-than*
- 関数的なルール : *age*
- ルール階層 : *has-daughter* $\sqsubseteq$ *has-child*

■ $SHIQ(D)$ (=OIL)

$SHIQ$ + ONE-OF($\mathcal{O}$) + Concrete Domain

■ Concrete domain(D) + n 引数述語

$\exists(u_1, \dots, u_n).P$ (但し, $u_1, \dots, u_n$ は関数的ロール)

$\{d \in \Delta^{\mathcal{I}} \mid (u_1(d), \dots, u_n(d)) \in P^D\}$

Example

$\exists(age). \geq_{20} \sqcap Human$ (但し, $\geq_{20}(x) \Leftrightarrow x \geq 20$)

「20歳以上の人」

■ FaCT(Fast Classification of Terminologies) [Horrocks 2000]

• $SHIF$, $SHIQ$ の推論システム

■ DLP(Description Logic Prover) [Patel-Schneider 1998]

• $\mathcal{ALCN}_{R+} + R \sqcup S + R \circ S + /C$ の推論システム

最近のDL研究(1)

■ 概念の単一化問題 [Baader 1999, 2002]

(1) $C \equiv^? D$ iff $C \equiv \sigma(D)$

(C : 概念, D : 概念パターン, σ : 代入)

(2) $C \sqsubseteq^? D$ iff $C \sqsubseteq \sigma(D)$ (iff $C \equiv^? C \sqcap D$)

Example

$\forall has-child.Male \equiv^? \forall has-child.Male \sqcap \forall has-child.X$

$\forall has-child.Male \sqsubseteq^? \forall has-child.X$

$\sigma = \{X \mapsto Animal\}$

■ 概念言語の拡張

Concrete domain(D) + Role-forming n 引数述語

$$\frac{\exists(u_1, \dots, u_k), (v_1, \dots, v_m). P}{\{(a, b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (u_1(a), \dots, u_k(a), v_1(b), \dots, v_m(b)) \in P^D\}}$$

Example

$(\exists(\textit{expenses}), (\textit{income}). >).\{John\} \sqcap Human$

「ジョンの収入より出費が多い人」

■ 様相オペレーターの導入

Temporal Description Logic [Sturm et.al 2002]

Example

$\Box Living$ (always in the future)

$\Diamond \neg Living$ (eventually), $\bigcirc Living$ (next)

$Lighting \mathcal{U} BrokenThing$ (until)

最近のDL研究(2)

■ 論理プログラミングとの融合

- $\mathcal{AL}$ -log [Donini et.al. 1998]
- CARIN [Levy 1998]
- Description Logic Programs [Grosz et.al. 2003]

Example $\mathcal{AL}$ -log

$illegal(x) \leftarrow know(x, y), sold(x, y, z) \ \& \ x : Human$
 $\square \exists sells.Alcohol, y : Minor, z : Alcohol$
 $Minor \equiv \neg Major$

■ DL 推論の最適化

● Semantic branching search

Example

$$S_i = \{x : C \sqcup D_1, x : C \sqcup D_2\}$$

$$\begin{array}{ccc}
 & \swarrow & \searrow \text{ } \sqcup\text{-規則} \\
 S_{i+1} = S_i \cup \{x : C\} & & S'_{i+1} = S_i \cup \{x : D_1, \textcolor{red}{x : \neg C}\} \\
 \downarrow & & \downarrow \text{ } \sqcup\text{-規則} \\
 \textcolor{red}{\text{clash}} & & S'_{i+2} = S'_{i+1} \cup \{x : D_2\}
 \end{array}$$

● Simplification

$$\frac{\neg C_1, \dots, \neg C_n, C_1 \sqcup \dots \sqcup C_n \sqcup D}{D}$$

Example

$$\begin{array}{c}
 S_i = \{\textcolor{red}{x : C \sqcup (D_1 \sqcap D_2)}, x : \neg D_1 \sqcup \neg D_2, \textcolor{red}{x : \neg C}\} \\
 \downarrow \text{ } \sqcup\text{-規則} \\
 S_{i+1} = S_i \cup \{x : D_1 \sqcap D_2\} \\
 \downarrow \text{ } \sqcap\text{-規則} \\
 S_{i+2} = S_{i+1} \cup \{x : D_1, x : D_2\}
 \end{array}$$

まとめ

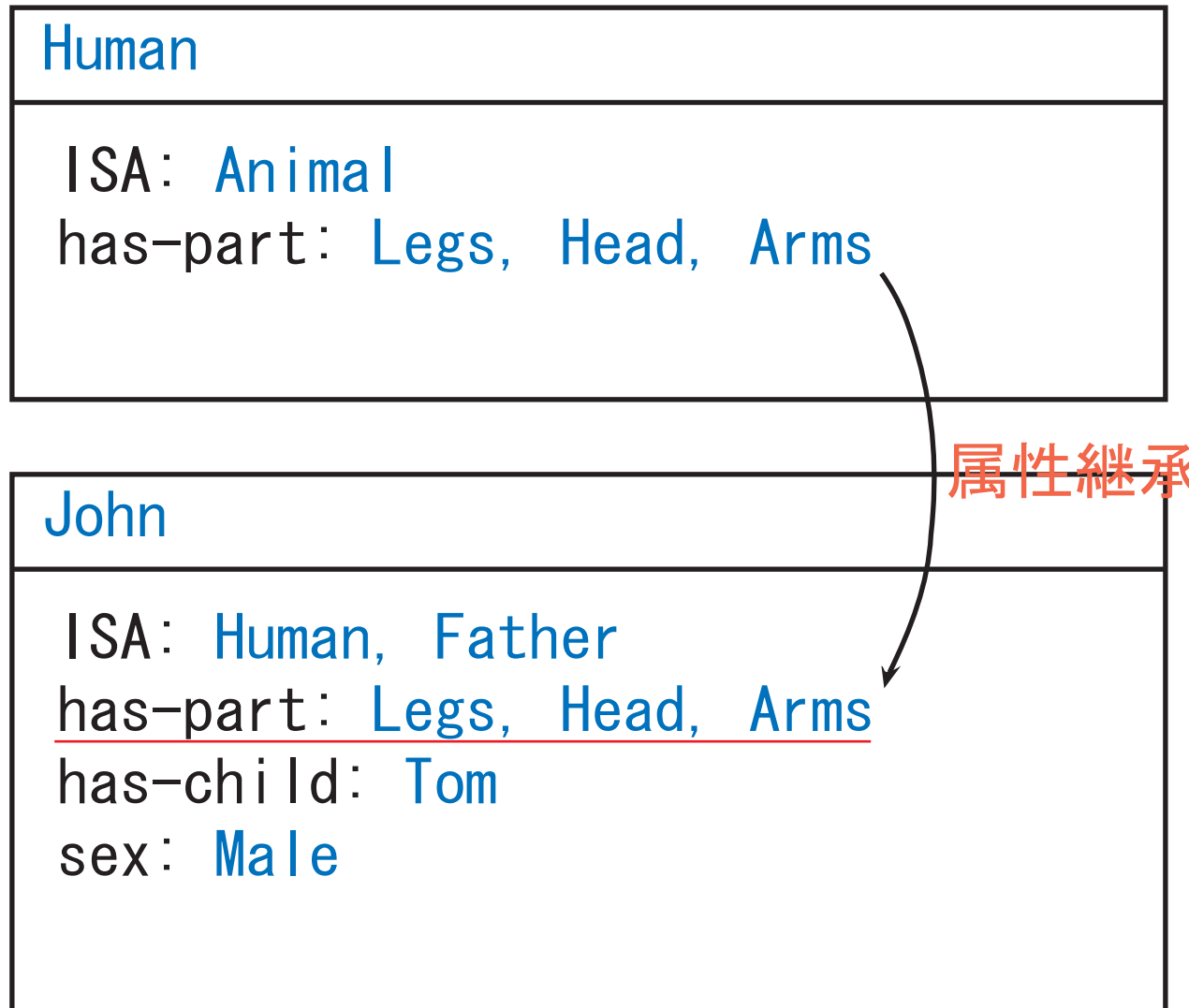
DLの特徴・優位性

- 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

DLの解説論文・解説本

- (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) 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.
- (4) Calvanese, D., Giacomo, G. D., Lenzerini, M., and Nardi, D.: Reasoning in Expressive Description Logics, in *Handbook of Automated Reasoning*, pp. 1581–1634 (2001).
- (5) Kusters, R.: *Non-standard inferences in description logics*, LNCS 2100, Springer-Verlag (2001).

Frames [Minsky 1975]



DAML+OIL と DL [Horrocks 2002]

クラス	DL 概念
complementOf C	$\neg C$
intersectionOf C1...Cn	$C_1 \sqcap \dots \sqcap C_n$
unionOf C1...Cn	$C_1 \sqcup \dots \sqcup C_n$
toClass P C	$\forall P.C$
hasClass P C	$\exists P.C$
hasValue P a	$\exists P.\{a\}$
minCardinalityQ n P C	$\geq n P.C$
maxCardinalityQ n P C	$\leq n P.C$
CardinalityQ n P C	$\geq n P.C \sqcap \leq n P.C$
oneOf a1...an	$\{a_1, \dots, a_n\}$

プロパティ	ルール
inverseOf P	P^-
transitiveProperty P	$P \in R^+$
uniqueProperty P	$\leq 1P$
unambiguousProperty P	$\leq 1P^-$

公理	T-Box 言明
subClassOf C1 C2	$C_1 \sqsubseteq C_2$
sameClassAs C1 C2	$C_1 \equiv C_2$
subPropertyOf P1 P2	$P_1 \sqsubseteq P_2$
samePropertyAs P1 P2	$P_1 \equiv P_2$
disjointWith C1 C2	$C_1 \sqsubseteq \neg C_2$
sameIndividualAs a b	$\{a\} \equiv \{b\}$
differentIndividualFrom a b	$\{a\} \sqsubseteq \neg\{b\}$

DL 言語から述語論理への変換

$$T^x(A) = A(x)$$

$$T^x(\top) = x = x$$

$$T^x(\perp) = \neg(x = x)$$

$$T^x(\neg C) = \neg T^x(C)$$

$$T^x(C \sqcap D) = T^x(C) \wedge T^x(D)$$

$$T^x(C \sqcup D) = T^x(C) \vee T^x(D)$$

$$T^x(\forall R.C) = \forall y(T^{x,y}(R) \rightarrow T^y(C))$$

$$T^x(\exists R.C) = \exists y(T^{x,y}(R) \wedge T^y(C))$$

$$T^x(\{a_1, \dots, a_n\}) = x = a_1 \vee \dots \vee x = a_n$$

$$T^x(R : a) = \exists y(y = a \wedge T^{x,y}(R))$$

$$T^{x,y}(P) = P(x, y)$$

$$T^{x,y}(R \sqcap S) = T^{x,y}(R) \wedge T^{x,y}(S)$$

$$T^{x,y}(R^-) = T^{y,x}(R)$$

論理プログラミングとの比較

■ 言語表現

DL：論理結合子，量化子，数量限定子など

LP： n 引数述語，一階項(関数，変数)とルール表現

■ LPの計算量 [Dantsin et.al.1997]

言語	Queryの計算量
Full LP	決定不可能
Non-recursive LP	NEXPTIME
Restricted LP with functions	PSPACE
Datalog	EXPTIME
Datalog (a fixed program)	P

Negation Normal Form

- 概念名の前だけに否定をもつ形式であり，以下の等式によって変換できる．

$$\neg\neg C \equiv C$$

$$\neg C \sqcap D \equiv \neg C \sqcup \neg D$$

$$\neg C \sqcup D \equiv \neg C \sqcap \neg D$$

$$\neg\forall R.C \equiv \exists R.\neg C$$

$$\neg\exists R.C \equiv \forall R.\neg C$$

$$\neg \geq 0R.C \equiv \perp$$

$$\neg \geq (n+1)R.C \equiv \leq nR.C$$

$$\neg \leq nR.C \equiv \geq (n+1)R.C$$

■ 否定を含まない概念言語 [Brachman 1984]

$\mathcal{FL}^-$ 概念

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

$\mathcal{FL}$ 概念

$$C, D \longrightarrow A \mid C \sqcap D \mid \forall R.C \mid \exists R.\top$$

$$R \longrightarrow P \mid R|_C$$

$$\blacksquare (R|_C)^{\mathcal{I}} = \{(d, e) \in \Delta^{\mathcal{I}} \mid (d, e) \in R^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$$

Example

$$\exists has-child|_{Daughter}.\top$$

「娘をもつ個体」

■ 循環的な概念定義とT-Box

$$\begin{aligned} Employee &\equiv Worker \sqcap \\ &\quad \forall has\text{-}boss.(Employee \sqcup Employer) \end{aligned}$$

「従業員」 = 「労働者であり，上司が従業員または雇用者であるもの」

知識ベースの充足可能性

■ T-Boxの充足可能性

- $\mathcal{I} \models C \equiv D$ iff $C^{\mathcal{I}} = D^{\mathcal{I}}$
- $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$

■ A-Boxの充足可能性

- $\mathcal{I} \models C(a)$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$
- $\mathcal{I} \models P(a, b)$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$

Example A-Box $\mathcal{A}_1 = \{$
 killed(John, Mary),
 stole(John, a),
 be-possessioned-of(a, Mary) $\}$