知識工学 II (第 3 回)

二宮 崇 (ninomiya@cs.ehime-u.ac.jp)

知識表現 (12)

§12.2.1 物理的構成

PartOf: あるオブジェクトが別のオブジェクトの部分である。例: ドアとドアノブ

推移的: $PartOf(x, y) \land PartOf(y, z) \Rightarrow PartOf(x, z)$

反射的: $PartOf(x, x)$

PartOf(Bucharest, Romania)

PartOf(Romania, EasternEurope)

PartOf(EasternEurope, Europe)

PartOf(Europe, Earth)

⇒PartOf(Bucharest, Earth)

〇合成オブジェクト

合成オブジェクトのカテゴリは、部品間の構造的関係によって定義

例: 2 足歩行する動物は胴体に 2 本の足がくっついている

$$
\begin{aligned}
Biped(a) \Rightarrow\ & \exists l_1, l_2, b\ \ Leg(l_1) \land Leg(l_2) \land Body(b) \land PartOf(l_1, a) \land PartOf(l_2, a) \land \\
& PartOf(b, a) \land Attached(l_1, b) \land Attached(l_2, b) \land \\
& l_1 \neq l_2 \land [\forall l_3\ \ Leg(l_3) \land PartOf(l_3, a) \Rightarrow (l_3 = l_1 \lor l_3 = l_2)]
\end{aligned}
$$

○まとまり(bunch)

集合を部分(PartOf)関係として持つ合成オブジェクト

例: BunchOf({$Apple_1$, $Apple_2$, $Apple_3$})

BunchOf({x})=x となることに注意

∀x　x∈s　⇒ PartOf(x, BunchOf(s))が常に成り立つ。BunchOf(s)はこの条件を満たす最小オブジェクト。

∀y [∀x　x∈s　⇒ PartOf(x, y)]⇒PartOf(BunchOf(s), y)

(BunchOf(s)は s の全ての要素を部品として持つ任意のオブジェクトに対して、その部分(PartOf)になっていなければならない。このように一定条件を満たす最小のものとして定義する仕方を「論理的最小化」と呼ぶ。)

○計測

測定値: オブジェクトが持つ、高さ、質量、コストなどの値。

ある 1.5 インチの線分を L_1 とすると、

Length(L_1)=Inches(1.5)=Centimeters(3.81)

と記述される。

異なる単位の変換: Centimeters(2.54×d)=Inches(d)

オブジェクトの記述 (あるバスケットボールのオブジェクト, $Basketball_{12}$)

　Diameter($Basketball_{12}$)=Inches(9.5)

　ListPrice($Basketball_{12}$)=$(19)

適当な尺度がない測定値: 演習の難しさ、デザートのおいしさ、詩の美しさなど

→順序づけ

「Norvig の演習は Russell の演習よりも難しい」

$e_1 \in Exercises \wedge e_2 \in Exercises \wedge Wrote(Norvig, e_1) \wedge Wrote(Russell, e_2) \Rightarrow$

$Difficulty(e_1) > Difficulty(e_2)$

c.f. 定性物理

〇物質とオブジェクト

物質 (stuff): 個別のオブジェクトへの分割ができそうにないもの。例: バター、水、エネルギー。可算名詞⇔集合名詞。

物質の分割: $x \in Butter \wedge PartOf(y, x) \Rightarrow y \in Butter$

内在的属性: オブジェクトの属性ではなく、実体に備わる属性。比重、沸点、色など。$x \in Butter \Rightarrow MeltingPoint(x, Centigrade(30))$

外在的属性: オブジェクトの属性。分割されると保持されない。重さ、長さ、形など。

§12.3 事象

変量(fluent): ある状況から次の状況の間で変化するオブジェクトあるいは述語

不変関数、述語 (atemporal/eternal functions and predicates): ある状況から次の状況の間で変化しない関数あるいは述語

変量の例: オブジェクト At(Shankar, Berkeley) ※述語ではない

事象計算: ある時点において成り立ち、計算により時間の間隔において推論が行われる。

事象計算の公理

T(f, t): 変量 f は時点 t において真

Happens(e, i): 事象 e が時間区間 i において起きている

Initiates(e, f, t): 事象 e は時点 t において変量 f を真に保ち始める

Terminates(e, f, t): 事象 e は時点 t において変量 f が真であることを終える

Clipped(f, i): 変量 f は時間区間 i のどこかで真であることを終える

Restored(f, i): 変量 f は時間区間 i のどこかで真となる

時間区間 $i=(t_1, t_2)$ 時点 t_1 から t_2 の間

$Happens(e, i) \Leftrightarrow Extent(e)=i$

事象の例

$E_1 \in Flyings \wedge Flyer(E_1, Shankar) \wedge Origin(E_1, SF) \wedge Destination(E_1, DC)$

$E_1 \in Flyings(Shankar, SF, DC)$

$\forall f,t, e, t_1,t_2 \quad Happens(e, (t_1,t_2)) \wedge Initiates(e, f, t_1) \wedge \neg Clipped(f, (t_1, t)) \wedge (t_1 < t) \Rightarrow T(f, t)$

$\forall f,t, e, t_1,t_2 \quad Happens(e, (t_1,t_2)) \wedge Terminates(e, f, t_1) \wedge \neg Restored(f, (t_1, t)) \wedge (t_1 < t) \Rightarrow \neg T(f, t)$

$\forall f, t_1, t_2\ Clipped(f, (t_1, t_2)) \Leftrightarrow \exists e,t,t_3\ Happens(e, (t, t_3)) \wedge t_1 \leqq t < t_2 \wedge Terminates(e, f, t)$

$\forall f, t_1, t_2\ Restored(f, (t_1, t_2)) \Leftrightarrow \exists e, t, t_3\ Happens(e, (t, t_3)) \wedge t_1 \leqq t < t_2 \wedge Initiates(e, f, t)$

$\forall f, t_1, t_2 \quad T(f, (t_1, t_2)) \Leftrightarrow [\forall t\ (t_1 \leqq t < t_2) \Rightarrow T(f,t)]$

§12.3.1 プロセス

プロセス(process)カテゴリ: 始まりがあって、途中があって、終わりがある事象カテゴリ。プロセスのどの副区間も、また同じプロセスカテゴリの要素となる。流動性事象 (liquid event) カテゴリともいう。離散事象 (discrete event) v.s. 流動性事象。

$\forall e, t_1, t_2, t_3, t_4$

$(e \in Processes) \land Happens(e, (t_1, t_4)) \land (t_1 < t_2 < t_3 < t_4) \Rightarrow Happens(e, (t_2, t_3))$

§12.3.2 時間区間 (time intervals)

時間区間は、瞬間(moment)と幅のある時間区間(extended interval)に分けられる。

Partition({Moments, ExtendedIntervals}, Intervals)

i∈Moments ⇔ Duration(i)=Seconds(0)

Time: 瞬間の絶対時間を返す関数

Interval(i)⇒Duration(i)=(Time(End(i)) – Time(Begin(i)))

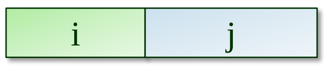
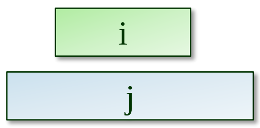
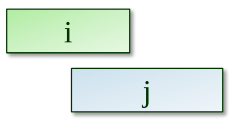
Time(Begin(AD1900))=Seconds(0)

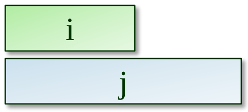
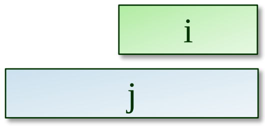
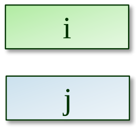
Time(Begin(AD2001))=Seconds(3187324800)

Time(End(AD2001))=Seconds(3218860800)

Duration(AD2001)=Seconds(31536000)

○Allen による時間区間の定義

Meet(i, j) ⇔ End(i)=Begin(j)	
Before(i, j) ⇔ End(i) < Begin(j) After(j, i) ⇔ Before(i,j)	
During(i, j) ⇔ Begin(j) < Begin(i) < End(i) < End(j)	
Overlap(i, j) ⇔ Begin(i) < Begin(j) < End(i) < End(j)	

Begins(i,j) ⇔ Begin(i)=Begin(j)	
Finishes(i,j) ⇔ End(i) = End(j)	
Equals(i,j) ⇔ Begin(i)=Begin(j) ∧ End(i)=End(j)	

§12.4 心的事象と心的オブジェクト

〇信念の形式的理論

命題的態度: Believes, Knows, Wants

〇具体化 (reification)

命題をオブジェクトに変える

CanFly(Superman) … CanFly は述語

Knows(Lois, CanFly(Superman)) … CanFly は関数

〇一階述語論理の参照透過性の問題

参照透過性: ある項をそれと等しい項に置き換えられること

Clark=Superman

CanFly(Clark)=CanFly(Superman)

次の関係が成り立ってしまう

(Superman=Clark)∧Knows(Lois, CanFly(Superman)) $\models$

Knows(Lois, CanFly(Clark))

解決策→様相論理

〇様相論理

様相論理≠一階述語論理

一階述語論理: 一つの世界 (唯一のモデルと解釈を仮定)

様相論理: 複数の世界(複数の 〃 ; 可能世界意味論)

$\mathbf{K}_A P$: A は文 P が真であると知っている

例: ボンドは誰がスパイなのか知っている

$\exists x\ \mathbf{K}_{Bond}\ Spy(x)$

例: ボンドはスパイがいることを知っている

$\mathbf{K}_{Bond}\ \exists x\ Spy(x)$

例: ロイスはクラークがスーパーマンであるかどうかは知らないかもしれないが、そのことをクラークが知っていることを彼女は知っている。

$\mathbf{K}_{Lois}[\mathbf{K}_{Clark}\ Identity(Superman, Clark) \lor \mathbf{K}_{Clark} \neg Identity(Superman, Clark)]$

〇**K** に関する公理

・$(\mathbf{K}_a P \land \mathbf{K}_a(P \Rightarrow Q)) \Rightarrow \mathbf{K}_a Q$

$\mathbf{K}_A(P \lor \neg P)$は恒真式(トートロジー)

$(\mathbf{K}_A P) \vee (\mathbf{K}_A \neg P)$は恒真式ではない

・$\mathbf{K}_a P \Rightarrow P$

・$\mathbf{K}_a P \ \Rightarrow \ \mathbf{K}_a(\mathbf{K}_a P)$

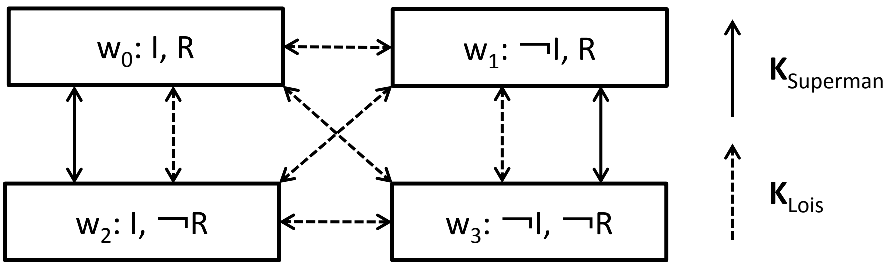
〇到達可能性 (accessibility)

Acc($\mathbf{K}_A$, w_0, w_1): $\mathbf{K}_A$ に関する世界 w_0 と世界 w_1 の 2 項関係。A が世界 w_0 について知っていることについて、世界 w_1 においても成り立つ。

世界 w において $\mathbf{K}_A P$ が真 ⇔ w から到達可能な全ての世界において P が真

例 1: スーパーマンは彼自身の正体について知っている。スーパーマンもロイスも明日の天気予報については知らない。

$\mathbf{K}_{\text{Superman}}$ Identity(Superman, Clark) $\vee$ $\mathbf{K}_{\text{Superman}} \neg$Identity(Superman, Clark)



I… スーパーマンの正体はクラークである
R…明日の天気予報は雨である

例 2: 例 1 に加えてさらにロイスは明日の天気予報について知っている。

$\mathbf{K}_{\text{Lois}}$ WeatherReport(Rain, Tommorow) $\vee$ $\mathbf{K}_{\text{Lois}}$ WeatherReport(Rain, Tommorow)