時間的遺伝性を用いたオカレンス論理

吉岡 卓，　東条 敏
北陸先端科学技術大学院大学情報科学研究科

1　はじめに

人間は言語表現における複数の出来事に対して，それらの時間的な関係を構造的に推定し把握することができる．我々は論理的なアプローチから，出来事間にみられる時間的な遺伝的関係を定義し，形式言語をもととした論理プログラミングシステムの構築を目的とする．

次章では，出来事の間にみられる時間的関係を示す．2 章においてソート階層と時間的遺伝性の概念を含む言語を提案し，その言語に対する統語論と意味論を与える．3 章では, 形式言語をもととした論理プログラミングシステムについて説明し，我々の計算機システムによる推論例を示す．最後に我々の理論におけるいくつかの論点を示す.

2　出来事と遺伝性の論理

一般に論理学における述語は静的な特性すなわち項の性質を表すが，我々は述語を出来事として定義する．各出来事は状況に依存していることより，状況自体にトークン [Barwise 96] と呼ばれる指標を与え，トークン間の関係として優先関係 $\prec$ と包含関係 $\lhd$ を定義する．特に本稿ではトークンの解釈を時間的な配置とした意味論を定義する．本稿では時制オペレーターは扱わない代わりにトークン間における時間的関係を比較する．述語を出来事として用いるため，他の方法によって静的な特性を表現する必要がある．そのような理由から型階層論理 [Aït-Kaci et al 86, Kaneiwa 99] を我々の論理に統合させる．型階層論理では，すべてのオブジェクトはソートとして分類されている．Beierle [Beierle 92] は因果関係の形式化において，ソートを単項述語として表現するような知識表現システムを提案している．ソートと単項述語は同様の表現力を持つことからサブソート宣言 $s_1 \sqsubseteq s_2$ を一般的な含意関係 $s_1(x) \rightarrow s_2(x)$ と見なす．また我々は各出来事に対応する時間的区間が存在している [Benthem 91] と仮定する．すなわち時間区間は出来事の集合として定義され，時間区間の関係もまた出来事間の関係 [Kamp 79] と同様に定義される．

2.1　構文

定義 1 (シグネチャ) 言語 OL_1 のアルファベットは以下の語彙からなる．

O_c オブジェクト定数の集合，S ソート記号の集合，
O_v オブジェクト変数の集合，P 述語記号の集合，
T_c トークン定数の集合，T_v トークン変数の集合
　さらに以下の記号を用いる．
: トークンのデリミタ，　$\neg, \vee, \wedge, \Rightarrow$ 論理結合子，
/ ソート代入，　$\sqsubseteq$ ソートの包摂関係，
$\lhd$ トークンの包含関係，$\prec$ トークンの優先関係，
$^{\vee}, ^{\wedge}, ^{>}, ^{<}$ 遺伝的性質のマーカー
　なお，括弧と句読点は必要に応じて加える．

我々は $a, b, c, \cdots$ をオブジェクト定数，$z_1, z_2, \cdots$ をオブジェクト変数，$s_1, s_2, \cdots$ をソート記号，$e_1, e_2, \cdots$ をトークン定数，そして $x_1, x_2, \cdots$ をトークン変数として用いる．

定義 2 (下位包摂関係) ソート間には半順序関係がある．$s_1 \sqsubseteq s_2$ に対して，s_2 は s_1 を包含するという．'$\sqsubseteq$' は反射的推移的関係である．

定義 3 (ソート階層) ソートの集合 S の任意の要素 s_i, s_j に対して，$s_i \sqcap s_j$ と $s_i \sqcup s_j$ を定義することができ，それぞれ s_i と s_j の積集合，和集合と呼ばれる．特に $\top = \sqcup S$ を最大ソート，$\bot = \sqcap S$ を最小ソートとよぶ．すなわち S はラティスを形成する．

定義 4 (引数集合)　$p \in P$ に対して，関数 A_{rg} はソートからなる述語の引数を呼び出す．すなわち $A_{rg}(p) \subseteq S$．2 字組 $\langle p, A_{rg}(p) \rangle$ を述語宣言と呼ぶ．

定義 5 (述語表現) 述語表現は述語と複数の引数からなる．各引数はソートに置換されている．

$$p(o_1/s_1, o_2/s_2, \cdots, o_n/s_n)$$

ここで p は述語記号，$o_1, o_2, \cdots, o_n$ はオブジェクト定数もしくは変数，$s_1, s_2, \cdots, s_n$ はソートである．我々はギリシャ文字 $\phi, \psi, \cdots$ を述語として用いる．

定義 6 (包含関係，優先関係) トークン間には次の半順序関係がある．$e_1 \prec e_2$ に対して，e_1 は e_2 に優先するといい，$e_1 \lhd e_2$ に対して，e_2 は e_1 を含むという．ここで '$\prec$' と '$\lhd$' は推移的性質を持つ．

定義 7 (遺伝関係) 遺伝関係に対して以下の記法を導入する．

$e^{\wedge}{:}\phi \quad e^{\wedge} \in \{e' | e' \rhd e\}$

$e^{\vee}{:}\phi \quad e^{\vee} \in \{e' | e' \lhd e\}$

$e^{>}{:}\phi \quad e^{>} \in \{e' | e' \ (\succ e \ or \ \exists e_1, e_2 (e_3 \lhd e, e_1 \lhd e', e_2 \lhd e, e_2 \not\lhd e', e_2 \prec e_1))\}$

$e^{<}{:}\phi \quad e^{<} \in \{e' | e' \ (\prec e \ or \ \exists e_1, e_2 (e_1 \lhd e, e_1 \lhd e', e_2 \lhd e, e_2 \not\lhd e', e_1 \prec e_2))\}$

次の遺伝性マーカー $_{\wedge, \vee, >, <}$ をそれぞれ *up, down, right, left* とよぶ．これらの意味については次章にて示す．我々は遺伝性マーカー付きのトークン変数の集合 $\{x_i^* \mid x_i \in T_v\}$ を T_v^* とあらわし，全てのトークン変数の集合を $T_v^+ = T_v \cup \bigcup_{* \in \{\wedge, \vee, >, <\}} T_v^*$ と定義する．ここで $*$ は任意の遺伝性マーカーをあらわす．

定義 8 (区間と原始論理式) e をトークン，ϕ を述語表現とするとき，$e{:}\phi$ は原始論理式である．また次の形式 $e^{\wedge}{:}\phi$, $e^{\vee}{:}\phi$, $e^{>}{:}\phi$, $e^{<}{:}\phi$ も言語 OL_1 の原始論理式である．上述の論理式各々について，e はトークン変数として置き換えることができる．

2.2 オカレンス意味論

我々はトークンの意味を時間的範囲とする．すなわちここではただ時間の範囲の集合があり，それらは優先関係と包含関係といった半順序をもつことのみを想定する．我々の意味論において，時間範囲の集合 U_t を導入し各トークンは U_t の要素として解釈される．もしイベントが起こったならば，それは時間範囲に対応するものとして固定される．

定義 9 (構造) OL_1 に対する構造 M は組 $\langle U, \{R_{p_i}\}, [\![\ ]\!] \rangle$ で定義される．ここで，

1. $U = U_o \cup U_t$: 空でない集合 *(*すなわち M の世界*)* ただし U_o は個体の集合であり，U_t は $U_o \cap U_t = \emptyset$ を満たす時間範囲の集合である．
2. $\{R_{p_i}\}$: 関係 $R_{p_i} \subseteq U_t \otimes [\![s_1]\!] \otimes \cdots \otimes [\![s_n]\!]$ の集合，ただし $p_i \in P$，$A_{rg}(p_i) = \{s_1, s_2, \cdots, s_n\}$.
3. $[\![\ ]\!]$: 以下を満たす関数である．
   (a) $c \in O_c$ に対して $[\![c]\!] \in U_o$,
   (b) $s \in S$ に対して $[\![s]\!] \subseteq U_o$,
   (c) $s_1 \sqsubseteq s_2$ $(\in R_s)$ に対して $[\![s_1]\!] \subseteq [\![s_2]\!]$,
   (d) $e \in E_c$ に対して $[\![e]\!]$ は U_t の要素であり *(*すなわち $[\![e]\!] = i$ ただし $i \in U_t$*)*,
   $i.\ [\![e^{\wedge}]\!] \supset [\![e]\!]$, $ii.\ [\![e^{\vee}]\!] \subset [\![e]\!]$
   $iii.\ [\![e^{>}]\!] > [\![e]\!]$, $iv.\ [\![e^{<}]\!] < [\![e]\!]$
   (e) $[\![\lhd]\!] = \{([\![e]\!], [\![e']\!]) \mid [\![e]\!] \subset [\![e']\!]\}$,
   $[\![\prec]\!] = \{([\![e]\!], [\![e']\!]) \mid [\![e]\!] < [\![e']\!]\}$,
   (f) $p \in P$ に対して $[\![p]\!] \subseteq [\![s_1]\!] \otimes \cdots \otimes [\![s_n]\!]$ かつ $A_{rg}(p) = \{s_1, s_2, \cdots, s_n\}$
   ここで '$\otimes$' はカルテジアン積である．

'$<$' は区間の優先関係である．我々はオブジェクトとトークンそれぞれに対して，M における変数割り当てを導入する．オブジェクト変数割り当てとは関数 $\alpha_o{:}O_v \to U_o$ である．トークン変数割り当てとは関数 $\alpha_t{:}T_v^+ \to U_t$ であり，$\alpha_t(x) \in U_t$, $\alpha_t(x^{\wedge}) = \{y \in U_t \mid (y, \alpha_t(x)) \in [\![\lhd]\!]\}$, $\alpha_t(x^{\vee}) = \{y \in U_t \mid (\alpha_t(x), y) \in [\![\lhd]\!]\}$, $\alpha_t(x^{<}) = \{y \in U_t \mid (y, \alpha_t(x)) \in [\![\prec]\!]\}$, $\alpha_t(x^{>}) = \{y \in U_t \mid (\alpha_t(x), y) \in [\![\prec]\!]\}$ を満たす．ここで $y(\in U_t)$ は区間変数をあらわす． これらの割り当てを用いてオブジェクトとトークン変数を含んだ解釈を次のように定義する．

定義 10 (解釈) 解釈 $\mathcal{I}$ は組 $\langle M, \alpha_o, \alpha_t \rangle$ で定義される．$[\![\ ]\!]_{(\alpha_o, \alpha_t)}$ の意味は以下のように定義される．

1. $[\![c/s]\!]_{(\alpha_o, \alpha_t)} = [\![c]\!]$ ただし $[\![c]\!] \in [\![s]\!]$
2. $[\![z/s]\!]_{(\alpha_o, \alpha_t)} = \alpha_o(z)$ ただし $\alpha_o(z) \in [\![s]\!]$
3. $[\![e^{\wedge}]\!]_{(\alpha_o, \alpha_t)} = [\![e^{\wedge}]\!]$, $[\![e^{\vee}]\!]_{(\alpha_o, \alpha_t)} = [\![e^{\vee}]\!]$, $[\![e^{>}]\!]_{(\alpha_o, \alpha_t)} = [\![e^{>}]\!]$, $[\![e^{<}]\!]_{(\alpha_o, \alpha_t)} = [\![e^{<}]\!]$
4. $[\![e^{\wedge}]\!]_{(\alpha_o, \alpha_t)} = \{\alpha_t(x^{\wedge})\}$, $[\![e^{\vee}]\!]_{(\alpha_o, \alpha_t)} = \{\alpha_t(x^{\vee})\}$, $[\![e^{>}]\!]_{(\alpha_o, \alpha_t)} = \{\alpha_t(x^{>})\}$, $[\![e^{<}]\!]_{(\alpha_o, \alpha_t)} = \{\alpha_t(x^{<})\}$

定義 11 (充足可能性) $\mathcal{I} = \langle M, \alpha_o, \alpha_t \rangle$ は解釈であり，F は論理式であるとする．充足可能性 $\mathcal{I} \models F$ は次のように定義される*:*

1. $\mathcal{I} \models e{:}p(o_1/s_1, \cdots, o_n/s_n)$ *iff* $\langle rel \rangle \in [\![p]\!]$ かつ $i = [\![e]\!]_{(\alpha_o, \alpha_t)}$ に対して $\langle i, rel \rangle \in R_p$．ここで rel は列 $[\![o_1/s_1]\!]_{(\alpha_o, \alpha_t)}, \cdots, [\![o_n/s_n]\!]_{(\alpha_o, \alpha_t)}$ である，
2. $\mathcal{I} \models e^*{:}p(o_1/s_1, \cdots, o_n/s_n)$ *iff* $\langle rel \rangle \in [\![p]\!]$ かつ $i \in [\![e^*]\!]_{(\alpha_o, \alpha_t)}$ に対して $\langle i, rel \rangle \in R_p$．ここで $*$ は 遺伝的性質のマーカーをあらわす，

3. $\mathcal{I} \models \neg A$ *iff* $\mathcal{I} \not\models A$,
4. $\mathcal{I} \models A \vee B$ *iff* $\mathcal{I} \models A$ *or* $\mathcal{I} \models B$,
5. $\mathcal{I} \models (\forall z/s)A$ *iff* すべての $d \in [\![s]\!]$ に対して $\mathcal{I}' \models A$. ただし $\mathcal{I}' = \langle M, \alpha_o[z \mapsto d], \alpha_t \rangle$,
6. $\mathcal{I} \models (\forall x)A$ *iff* すべての $y \in U_t$ に対して $\mathcal{I}' \models A$. ただし $\mathcal{I}' = \langle M, \alpha_o, \alpha_t[x \mapsto y] \rangle$.

原始論理式 $e{:}\,\phi$, $e^{\wedge}{:}\,\phi$, $e^{\vee}{:}\,\phi$, $e^{>}{:}\,\phi$, $e^{<}{:}\,\phi$, に対する真理値は，与えられたプログラムと呼ばれる真である節の集合から操作的に推論される．次章においてこの推論ルールについて説明する．

3 論理プログラミングシステム

我々は標準的なホーン節計算にもとづいた，オカレンス論理の推論システムを提案する．ここでリテラルとは 2.1 章における原始論理式であり，節とは複数のリテラルの論理和である．このとき正のリテラルは多くとも 1 回現れる．

3.1 オブジェクトとソート

以下のルールにおいて，G はリテラルの列，単一化子は '[ ]' であり 'A/B' は 'B' が 'A' で置換されることを示している．

ルール 1 (オブジェクトインスタンス化)

$$\frac{\texttt{?-}\ G', e{:}\,p(z/s) \qquad e{:}\,p(c/s) \Leftarrow G}{\texttt{?-}\ (G', G)[c/z]}$$

$$\frac{\texttt{?-}\ G', e{:}\,p(c/s) \qquad e{:}\,p(z/s) \Leftarrow G}{\texttt{?-}\ (G', G)[c/z]}$$

ルール 1 はオブジェクト変数がインスタンスを作成するときに推論の結果として用いられる．上部の右側はファクト，上部の左側はクエリに対応する．

ルール 2 (ソート特殊化)

$$\frac{\texttt{?-}\ G', e{:}\,p(c/s_2) \qquad e{:}\,p(c/s_1) \Leftarrow G}{\texttt{?-}\ (G', G)[s_1/s_2]}\ (s_1 \sqsubseteq s_2).$$

ルール 2 はソート階層を反映している．

3.2 トークンルール

オカレンスの推論の仕様について考察する．ここではトークンに対する一般的なルールを定義する．

ルール 3 (トークン特殊化)

$$(i)\quad \frac{\texttt{?-}\ G', e'{:}\,\phi \qquad e^{*}{:}\,\phi \Leftarrow G}{\texttt{?-}\ (G', G)[e^{*}/e']}\ (e' * e)$$

$$(ii)\quad \frac{\texttt{?-}\ G', e{:}\,\phi \qquad e^{*}{:}\,\phi \Leftarrow G}{\texttt{?-}\ (G', G)}$$

$$(iii)\quad \frac{\texttt{?-}\ G', e^{*}{:}\,\phi \qquad e{:}\,\phi \Leftarrow G}{\blacksquare}$$

ここで '$*$' は 定義 7 に従う．また '$\blacksquare$' はこれ以上の解決が無理であることを意味している．

ルール 4 (トークン単一化)

$$(i)\quad \frac{\texttt{?-}\ G', x{:}\,\phi \qquad e{:}\,\phi \Leftarrow G}{\texttt{?-}\ (G', G)[e/x]},$$

$$\frac{\texttt{?-}\ G', e{:}\,\phi \qquad x{:}\,\phi \Leftarrow G}{\texttt{?-}\ (G', G)[e/x]},\quad \frac{\texttt{?-}\ G', x_1{:}\,\phi \qquad x_2{:}\,\phi \Leftarrow G}{\texttt{?-}\ (G', G)[x_1/x_2]}$$

$$(ii)\quad \frac{\texttt{?-}\ G', e{:}\,\phi \qquad x^{*}{:}\,\phi \Leftarrow G}{\texttt{?-}\ (G', G)[e'/x]}\ (e' * e)$$

$$(iii)\quad \frac{\texttt{?-}\ G', e^{*}{:}\,\phi \qquad x{:}\,\phi \Leftarrow G}{\blacksquare}$$

ここで *(i)* は基本的な単一化，*(ii)* はヘッド部にある遺伝的マーカー付きのトークンがトークン定数によって置換されることを示している．*(iii)* は閉世界仮説を示している．

3.3 実例

OL_1 に対するシステムを JAVA (JDK 1.2) によって Solaris 5.7, SUN^{TM} Sparc station Ultra 5-10 上に実装した．本章では推論システムを用いた例を示す．以下の例において，簡単のために引数部分は省略する．

例 1 "鍋にいれた水が沸いたら鶏肉を茹でる．その間に香りがたつまでニンニクを炒める．"

まず初めに '**まで**' と '**間に**' を以下のように定義する．

$$\begin{cases} x^{\wedge}{:}\,boil \Leftarrow x{:}\,frizzle. & ;\text{茹でる間に炒める} \\ x^{<}{:}\,frizzle \Leftarrow x{:}\,scent. & ;\text{香りがたつまで} \end{cases}$$

この状況に対して以下のファクトと関係を宣言する．

$$\begin{cases} e_1{:}\,bubble \Leftarrow . \quad e_2{:}\,boil \Leftarrow . \quad e_3{:}\,frizzle \Leftarrow . \\ e_4{:}\,scent \Leftarrow . \quad e_3 \lhd e_2, \qquad e_1 \prec e_2 \end{cases}$$

このとき，*'boiling'* が e_4 で起こっているか否かを判定する．その為の証明過程を図 *1* に示す．図 *1* より *'boiling'* の性質は e_4 に下方遺伝されていることが分かる．

$$\dfrac{\dfrac{\dfrac{\texttt{?-}\ e_4{:}boil \quad x_1^{\wedge}{:}boil \Leftarrow x_1{:}frizzle}{\texttt{?-}\ e_4{:}frizzle} \qquad x_2^{<}{:}frizzle \Leftarrow x_2{:}scent}{\texttt{?-}\ e_4{:}scent} \qquad e_4{:}scent \Leftarrow}{\square}$$

図 1: 例 1 に対する証明

$$\dfrac{\dfrac{\dfrac{\texttt{?-}\ x_1{:}heat, x_1{:}evap \quad x_2^{\wedge}{:}evap \Leftarrow x_2{:}boil}{\texttt{?-}\ x_1{:}heat, x_1{:}boil} \qquad x_3^{>}{:}boil \Leftarrow x_3{:}heat}{\texttt{?-}\ x_1{:}heat} \qquad e_1{:}heat \Leftarrow}{\blacksquare}$$

図 2: 例 2 に対する証明

例 2 “水を熱し沸騰させる．その過程において水分は蒸発している．”

我々は以下の制約を定義する．

$$\begin{cases} x^{>}{:}boil \Leftarrow x{:}heat.; \text{熱した結果沸騰する} \\ x^{\wedge}{:}evaporate \Leftarrow x{:}boil.; \text{沸騰前後も蒸発する} \end{cases}$$

加えて以下のファクトと関係を定義する．

$$\begin{cases} e_1{:}heat \Leftarrow . \quad e_2{:}boil \Leftarrow . \quad e_3{:}evaporate \Leftarrow . \\ e_2 \lhd e_1, \qquad e_2 \lhd e_3 \end{cases}$$

ここで次の質問を試みる．“どの箇所で蒸発し，熱しているのか．”この時の証明過程を図 *2* に示す．

図 *2* の最初のステップにおいて，x_1 は $x_2^{\wedge}$ の要素でなければならない．また次のステップにおいて x_1 は $x_3^{>}$ の要素でもあり矛盾が生じる．ただし以下の証明によってシステムは候補 e_3 までは見つけることができる．

$$\dfrac{\dfrac{\texttt{?-}\ x_1{:}evap \quad e_3{:}evap \Leftarrow e_2{:}boil}{\texttt{?-}\ e_2{:}boil} \qquad e_2{:}boil \Leftarrow e_1{:}heat}{\square}$$

残念ながら *‘*?- $e_3{:}heat$*’* と *‘*$e_1{:}heat \Leftarrow$*’* も解くことはできない．しかし e_1 と e_3 が共通の時間 $e_1 \cap e_2$ を持つならば，それこそが冒頭の解答となることが分かる．

4 考察

我々は本稿において (i) オカレンス論理を提案し，構造の世界を従来の U から U_o と U_t へと拡張した．(ii) 加えて 2 つの時間範囲に包含関係が与えられたとき，我々はイベントの上方遺伝とステイトの下方遺伝を区別することができた．また右側遺伝と左側遺伝を表現するために，2 つの時間範囲に優先関係を導入した．(iii) 最後にオカレンスの推論を可能とするホーン節計算を示し計算機上に実装した．

参考文献

[Aït-Kaci et al 86] H. Aït-Kaci and R. Nasr., Login: A logic programming language with built-in inheritance, *Journal of Logic Programming*, 3(3):185–215, 1986.

[Barwise 96] J. Barwise, D. Gabbay, and C. Hartonas., On the logic of information flow, In J. Seligman and D. Westerstahl, editors, *Logic, Language, and Computation, vol. 1.* CSLI, Stanford University, 1996.

[Beierle 92] C. Beierle, U. Hedtsuck, U. Pletat, P.H. Schmitt, & J. Siekmann., An order-sorted logic for knowledge representation systems., *Artificial intelligence*, **55**, 149–191, 1992.

[Benthem 91] J. van Benthem., *The Logic of Time - second edition*, Kluwer Academic Press, 1991.

[Kamp 79] H. Kamp., *Some remarks on the logic of change. Part I*, In Roherer (ed.), *Time, Tense and Quantifiers*, Niemeyar, Tübingen, 1979.

[Kaneiwa 99] K. Kaneiwa and S. Tojo., Event, property, and hierarchy in order-sorted logic, In *Proceedings of ICLP '99*, pages 94–108, 1999.

[Tojo 99] S. Tojo., Event, state, and process in arrow logic, *Journal of Minds and Machines*, 9:81–103, 1999.

[Yoshioka 03] S. Yoshioka, K. Kaneiwa and S. Tojo., Occurrence Logic with Temporal Heredity In *Proceeding of the IICAI-03*, pp.1296-1309, 2003