法的推論のための論理型イベント言語

兼岩 憲[†] 東条 敏[††]

法的推論システムの実現には，法令文のルール知識に加えて判例を描写するイベント（一回性の事象）のための表現手段が必要である．特に，判例の中身は複数のイベントによって展開されており，動的な知識表現とその論理的な推論を法的推論システムに取り込まなければならない．イベントは一回性・一時性を持った動作あるいはアクションであり，静的なプロパティと対比される概念である．これまでイベントの概念は，オントロジ，論理学，言語学，人工知能，演繹データベースなど様々な研究分野で扱われてきているが，各アプローチとも法的推論でのイベント記述の多様な側面を同時にとらえるには不十分であった．本論文では，法的推論で現れる語彙や記述を例に用い，イベントの量化，ソート階層およびイベント間の合成と排他性を導入した知識表現とその論理型言語（イベント論理と呼ぶ）を提案する．この言語は，順序ソート付き二階述語論理に準じた論理的な表現によってイベントを定数，ソート，述語および変数と見なし，イベント言明のための知識表現と推論を可能にする．さらに，法的推論システムを実現する推論メカニズムの基盤を与えるために，イベント論理に対するソート付きのタブロー計算を設計して，その反駁推論による質問応答システムを構築する．

An Logical Event-langauge for Legal Reasoning

Ken Kaneiwa[†] and Satoshi Tojo[††]

In order to implement a legal reasoning system, we have to provide a method that represents the suitable descriptions of events in addition to legal concepts and rules. An event, as opposed to an atemporal property, has its own time and location, and happens once and for all. Although the notion of events has been found in the researches of ontology, logic, linguistics, artificial intelligence and deductive databases, each approach does not seem to capture the various aspects of events represented in legal reasoning. In this paper, we propose an event logic with such expressions as quantification over events, event sort-hierarchy and composition and disjointness of events, based on some examples on event descriptions in legal knowledge. This logic is a variant of order-sorted second-order logic formalized by regarding events as constants, sorts, predicates and variables, which provides knowledge representation and reasoning for event assertions. In order to obtain a query-answering system, we present a sorted tableau calculus for refutation of event formulas in the logic.

1. はじめに

法的推論システムを実現するうえで，法令文や判例をどのように記述して，そこからどうやって結論を導くかは，最も本質的な課題である．法的推論の研究は人工知能の応用分野に位置づけられていることもあり，論理プログラミング言語を拡張して知識表現や推論が実現されている．その中でも法的推論システム New HELIC-II [15),16)] は，2 タイプの型階層を持つ論理型言語を備えており，法律知識の柔軟な記述を可能にしている．型階層ではそれまでの名詞的な概念階層に加えて動詞的な概念階層を導入することで，異なる抽象度を持つ動詞的な概念（たとえば，「暴行する」は「殴る」より抽象的である）を使って，刑法の事件記述に現れる様々な語彙（動詞）のイベント描写に有用である．

しかし New HELIC-II では名詞表現に加えて動詞表現を導入した拡張が行われたが，動詞表現が表すイベントそのものの特性を論理型言語に組み込むまでには至っていない．イベントは一回性・一時性を持った動作あるいはアクションであり，静的なプロパティと対比される概念である．そのようなイベント表現を扱うには，現実世界での行為や変化を描写しなければならないという難しさがある．実際，イベントの概念を扱う

† 情報通信研究機構知識創成コミュニケーション研究センター
Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology
†† 北陸先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Japan Advanced Institute of Science and Technology

ために数多くの研究がなされてきた．Davidson[6),18)] は，イベントに関する言明（以降，イベント言明という）を述語論理式で表し，イベント識別子 e を付加して各々のイベントを区別している．たとえば，論理式 $(\exists e)Kicked\,(Shem, Shaun, e)$ は「セムはシェーンを蹴った」というイベント e を表す．また言語学的な考察において，イベント[23)] は自然言語文を論理的に表現・説明するときの便利な見方としてとらえられている．Allen ら[2)] は，イベントは現実世界に存在する概念というよりもむしろ変化のパターンを適切に分類する便利な手段であると述べている．Sowa[21)] は，プロセス内の分離したステップに現れる変化としてイベントを定義している．一方 Landman[14)] は，イベントの集合と，イベント上の時間順序（precedence），包含（inclusion）および重複（overlap）の関係からなるイベント構造を形式化している．ソフトウェア科学のアプローチでも，アクティブ・オブジェクト指向データベースにおいてイベントの概念が用いられている[7),8)]．データベースシステム SAMOS[8)] では，複雑なイベントを定義するイベント詳述（event specification）のための機能を備えている．さらに，Galton ら[7)] は，知識表現とデータベースの異なるアプローチで提案されたイベントの定義を融合する試みを行った．

しかしこれまでのアプローチでは，法的推論に出現するようなイベントの描写を十分にサポートできるとはいいがたい．その理由は 2 つある．1 つは，多くのアプローチではイベントの持つ限られた性質にしか触れていない．実際，イベントの概念はその利用方法によってインスタンスや変数であったり，ソートや述語であったり違った側面を見せる．もう 1 つは，法的推論システムを実現することを想定していないので，イベント概念は議論されても，イベントの表現と推論を目的とした論理型言語の構文，意味論と推論メカニズムが提案されていない場合が多いことである．

本研究では，順序ソート論理を拡張して，法的推論の（刑法や道路交通法における）事件記述を例にとってイベント表現の多様性をサポートする方法を提案する．いい換えれば，その拡張は次にあげるイベントの異なる性質を考慮しなければならない．

述語としてのイベント：述語論理式「殴る (太郎, 花子)」が，太郎が花子を殴るというイベント言明を表しているとする．このとき，述語「殴る」は 1 つのイベントを示すことに使われる．このようなタイプの述語は，静的なプロパティを示す述語（たとえば，述語論理式「凶暴 (太郎)」の「凶暴」）とは明らかに区別されるべきである．

イベントの構造的な性質：「凶器で刺す」，「襲う」，「行為」の表現がそれぞれイベントを示すとき，それらの意味は構造的な関係をもたらす．たとえば，イベント「凶器で刺す」は「襲う」と「行為」よりも具体的であり，そのイベント「襲う」は「行為」より具体的で「凶器で刺す」より抽象的である．このような解釈はイベントを型と見なしており，その型関係は「凶器で刺す $\sqsubseteq$ 襲う $\sqsubseteq$ 行為」により表すことが可能である．

イベントの量化，変数およびインスタンス：文献 6), 17) では，イベント上の変数や量化が使われており，

$$(\forall e)\ \text{凶器で刺す}\ (e, \text{人}) \rightarrow (\exists e')\ \text{負傷する}\ (e', \text{人})$$

のようにイベント規則「凶器で人を刺すというイベントが存在するとき，人が負傷するというイベントも存在する」を記述できる．このようなイベント変数に代入される特定のイベントは，イベントインスタンスとして認識される．

イベントに含まれる時間と場所：各イベント言明には，そのイベントが生じた特定の時間と場所の情報が含まれる．たとえば,「目撃する (犯人)」（犯人を目撃する）には

$$(\exists t)(\exists l)\ \text{目撃する}\ (\text{犯人}, t{:}tim, l{:}loc)$$

（ある時間 t にある場所 l で犯人を目撃する）という暗黙の情報が含まれている．

イベントの合成と排他性：2 つのイベント言明「運転する (太郎，車)」（太郎は車を運転する）と「話す (太郎, 携帯電話)」（太郎は携帯電話で話す）が同じ時間と場所で起こったとする．そのとき，それらが合成されたイベント「運転する $\sqcap$ 話す (太郎)」（太郎は話しながら運転する）が成り立つ．また合成に反して，同時に起こりえない 2 つのイベント間の関係があり，これをイベントが互いに排他的であるという．たとえば，裁判などにおいて同じ事柄について「証言する (太郎, 事実 1)」と「黙秘する (太郎, 事実 1)」は同時に起こりえない．

以上のような法律での判例記述に含まれるイベントの概念・特性を論理型言語の中に取り込んで，その推論メカニズムを提案する．そのイベント表現の多様性に対処するために，順序ソート論理や二階述語論理の表現を応用する．本論文で提案するイベントセマンティクスは法的推論を当初の目的として設計開発されている．よって法的推論での有用性が最も大きいが，それに限らずイベントに関して広く利用できる意味記述でもある．

本論文の構成は次のとおりである．2 章では，順序ソート論理について基本的な概念を導入する．3 章は，法的推論の判例記述で現れるイベント表現が多様な論

理的表現をもたらすことを述べて，順序ソート論理の言語を拡張することでそれを表現する方法を提示する．4 章では，順序ソート論理に基づいて，イベントの表現を導入した言語を定義し，その意味論は順序ソート論理のソートモデルを変形して定義する．5 章では，述語引数の推論規則と，イベントの量化，合成および排他性に関する推論規則を加えたソート代入付きのタブロー法を提案する．6 章では，先行研究との比較を行うとともに本言語の法的推論への有用性について触れる．最後に 7 章で，本研究のまとめと今後の課題について述べる．

2. 準　　備

本章では，ソート階層を持つ一階述語倫理（すなわち，順序ソート論理[10),19),20)]）の基本的な概念を導入する．

定義 2.1（ソート言語） ソート付き一階言語は，ソート記号 $s_1, s_2, \ldots$ の集合 $\mathcal{S}$，関数記号 $f_1, f_2, \ldots$ の集合 $\mathcal{F}$，述語記号 $p_1, p_2, \ldots$ の集合 $\mathcal{P}$，論理結合子 $\wedge$，$\vee$，$\rightarrow$，$\neg$ および量化記号 $\exists$，$\forall$ を含む．特に，$\mathcal{F}_n$ は n 引数関数記号の集合を示しており，$\mathcal{P}_n$ は n 引数述語記号の集合を示す．$\mathcal{V}_s$ は，ソート s の変数 $x{:}s, y{:}s, \ldots$ の集合，$\mathcal{V} = \bigcup_{s \in \mathcal{S}} \mathcal{V}_s$ はすべてのソートに対する変数の集合を示している．

定義 2.2（ソートシグネチャ） ソート付き一階言語のシグネチャ（ソートシグネチャと呼ぶ）は，以下を満たすような組 $\Sigma = (\mathcal{S}, \mathcal{F}, \mathcal{P}, D_{\mathcal{S}})$ である．ここで，$D_{\mathcal{S}}$ はソート宣言の集合である．

（1） $(\mathcal{S}, \sqsubseteq_{\mathcal{S}})$ は，最大ソート $\bot_{\mathcal{S}}$ と最小ソート $\top_{\mathcal{S}}$ を含んだソートの半順序集合である．

（2） $f \in \mathcal{F}_n$ のとき，関数のソート宣言 $f{:}\langle s_1, \ldots, s_n, s\rangle \in D_{\mathcal{S}}$ が存在する．

（3） $p \in \mathcal{P}_n$ のとき，述語のソート宣言 $p{:}\langle s_1, \ldots, s_n\rangle \in D_{\mathcal{S}}$ が存在する．

関数のソート宣言 $f{:}\langle s_1, \ldots, s_n, s\rangle$ により，n 引数関数 f はソート $s_1, \ldots, s_n$ に属する n 個の要素組からソート s の要素への写像と宣言している．また，述語のソート宣言 $p{:}\langle s_1, \ldots, s_n\rangle$ により，n 引数述語 p はソート $s_1, \ldots, s_n$ に属する n 個の要素組に対する述語と宣言している．ソート記号 s，s' に対して，$s \sqcap s'$ と $s \sqcup s'$ はぞれぞれ s と s' の下限（greatest lower bound）と上限（least upper bound）を示している．

定義 2.3（ソート項） ソートシグネチャ $\Sigma = (\mathcal{S}, \mathcal{F}, \mathcal{P}, D_{\mathcal{S}})$ が与えられたとする．そのとき，ソート s の項は以下のように帰納的に定義される．

（1） すべてのソート変数 $x{:}s$ は，ソート s の項である．

（2） $f \in \mathcal{F}_n$ が $\langle s_1, \ldots, s_n, s\rangle \in D_{\mathcal{S}}$ であるような関数，$r_1{:}s_1, \ldots, r_n{:}s_n$ がそれぞれソート $s_1, \ldots, s_n$ の項であるとき，$f(r_1{:}s_1, \ldots, r_n{:}s_n){:}s$ はソート s の項である．

定義 2.4（ソート論理式） ソートシグネチャ $\Sigma = (\mathcal{S}, \mathcal{F}, \mathcal{P}, D_{\mathcal{S}})$ が与えられたとする．そのとき，ソート論理式は以下のように帰納的に定義される．

（1） $p \in \mathcal{P}_n$ が $p{:}\langle s_1, \ldots, s_n\rangle \in D_{\mathcal{S}}$ であるような n 項述語，$r_1{:}s_1, \ldots, r_n{:}s_n$ がそれぞれソート $s_1, \ldots, s_n$ の項であるとき，$p(r_1{:}s_1, \ldots, r_n{:}s_n)$ はソート論理式である．

（2） A と B がソート論理式であるとき，$\neg A$，$A \wedge B$，$A \vee B$，$A \rightarrow B$，$(\forall x)A[x{:}s]$ および $(\exists x)A[x{:}s]$ はソート論理式である．

定義 2.5 ソートシグネチャ Σ に対するソートモデル（Σ-モデルと呼ぶ）は，空でない集合 U_o と $\mathcal{S} \cup \mathcal{F} \cup \mathcal{P}$ を定義域とする解釈関数 I の対 $M = (U_o, I)$ であり，以下の条件を満たす．

（1） $I(s) \subseteq U_o$（特に，$I(\bot_{\mathcal{S}}) = \emptyset$ かつ $I(\top_{\mathcal{S}}) = U_o$），

（2） $s_i \sqsubseteq_{\mathcal{S}} s_j \in \Sigma$ のとき，$I(s_i) \subseteq I(s_j)$，

（3） $f{:}\langle s_1, \ldots s_n, s\rangle \in D_{\mathcal{S}}$ のとき，$I(f) : I(s_1) \times \cdots \times I(s_n) \rightarrow I(s)$，

（4） $p{:}\langle s_1, \ldots s_n\rangle \in D_{\mathcal{S}}$ のとき，$I(p) \subseteq I(s_1) \times \cdots \times I(s_n)$．

3. イベントの論理的表現

最初に，法的推論の判例で現れる言明を例に用いて，イベントがもたらす様々な論理的な表現を分析する．その分析を基にして，イベントの多様性を記述するのに，順序ソート論理の知識表現を応用する．その際に，イベント知識に適した言語を設計するために，どのようにインスタンス，ソート，変数や述語などの表現を拡張していくかを議論する．

3.1 インスタンス，ソート，変数および量化

1 章で述べたように，n 引数述語 p と個体 $o_1, \ldots, o_n$ によって，述語論理式

$$p(o_1, \ldots, o_n)$$

は 1 つのイベント e_i を表すことができる．このとき，この論理式 $p(o_1, \ldots, o_n)$ はイベントインスタンス e_i（または，イベント識別子）を詳細に記述したものと見なせる．この記法は，個体のプロパティを示したり，個体間の関係を表したりするような述語の用法とは区別されるべきである．しかし意味論では，いずれの用法も述語は個体の組の集合として定義される．

ここで述べたイベントインスタンスは，一回性の言明を表しているにすぎない．したがって，もし述語論理式「襲う (太郎, 人)」（太郎が人を襲う）がイベントインスタンス e_1 を描写しているならば，同様な描写を持つイベントインスタンスは以下のようにいくらでも存在するといえる．

$e_1, e_2, \ldots, e_n \in$ 襲う (太郎, 人)

このような考えから，個体間の関係としての述語「襲う」は，もう 1 つ別にイベントインスタンスに対する型の役割を持つ[3]．それは，イベント上でのソートや述語（イベントソートとイベント述語と呼ぶ）である．たとえば，「e_1:襲う」はイベントソート「襲う」に属するイベントインスタンス e_1 を表し，「襲う (e_1)」はイベントインスタンス e_1 がイベント述語「襲う」の性質を持つことを表す．これは，個体上のソートおよび述語の記述方法をイベントの概念に応用すれば，自然に導かれる発想である．これに関連して，Kaneiwa と Tojo[11),12)] は，個体に対するソートと，イベントに対するソートの 2 つのソート階層の形式化を行っている．ここでイベントソートの集合を $\mathcal{ES}$ で表す．このイベントソートを使ってイベントの構造的な性質が説明できる．「凶器で刺す」，「襲う」，「行為」をそれぞれイベントソートとする．そのとき，

凶器で刺す $\sqsubseteq_E$ 襲う $\sqsubseteq_E$ 行為

は，ソート間の階層関係を宣言する．よってイベントのソート階層は，対 $(\mathcal{ES}, \sqsubseteq_E)$ で構成される．各イベントソートは，先ほどの議論にあるようにイベントインスタンスの集まりを意味する．したがって，「襲う (e_1)」が成り立つとき，イベントソートの階層関係により「行為 (e_1)」が導かれる．ここでさらに，イベント e_1 の詳細な記述が述語論理式「襲う (太郎, 人)」である場合，イベント e_1 の別の記述「行為 (太郎, 人)」が階層関係により導かれる．このように，表現「襲う」は個体の 2 項述語としてだけでなく，イベントのソートや述語としても利用される．

こうして，イベントに関しても個体表現と同様に，インスタンス，ソートや述語が存在することが説明された．そこで E_1 をイベントソート，e_1 をイベントインスタンス，$o_1, \ldots, o_n$ を個体表現としたとき，次のようにイベント言明を記述する．

$e_1{:}E_1(o_1, \ldots, o_n)$

この表現は，イベントインスタンス e_1 がイベントソート E_1 に属し，さらに e_1 の詳細な記述が $E_1(o_1, \ldots, o_n)$ であることを意味している．たとえば，「e_1 : 襲う (太郎, 人)」によって「太郎が人を襲う」を示す 1 つのイベントインスタンス e_1 が成り立つ．これは状況理論[3),4)] のトークン（token）と型（type）の関係：

$e_1{:}\langle\!\langle E_1, o_1, \cdots, o_n \rangle\!\rangle$

と同一のものであり，e_1 は個別のイベントを指し，それが《　》内に示される型のイベントであるという言明に対応する．しかし我々の表記は述語☆E_1 自体もイベントソートとして他のソートと関係を持ち，かつ構成要素の各 o_i もソート階層上の要素であることから，状況理論における表記をより詳述化したものになっている．

さらにイベントを含んだ一般的な規則を記述する場合に，イベントの変数および量化による拡張が必要となる．この実現には，二階述語論理の述語変数を応用する．イベントインスタンスは述語定数に相当し，イベント変数☆☆は述語変数に相当すると考える．イベント言明 $e_1{:}E_1(o_1, \ldots, o_n)$ は一般化されてイベント変数による言明 $X{:}E_1(o_1, \ldots, o_n)$ を得る．このイベント変数を量化すると，$(\forall X)X{:}E_1(o_1, \ldots, o_n)$ は「すべてのイベント $X{:}E_1$ に対して，個体の組 $(o_1, \ldots, o_n)$ が成り立つ」を意味する．これらのイベント変数とその量化を使って，次のような規則を記述できる．

$(\forall X)X$:凶器で刺す (人) $\rightarrow (\exists Y)Y$:負傷する (人)

この意味は，「凶器で人を刺すというイベントが存在するとき，人が負傷するというイベントも存在する」である．厳密に述べると「凶器で人を刺すすべてのイベント X に対して，人が負傷するというイベント Y が存在する」を示している．

3.2　一回性，合成および排他性

続いて，各イベント言明が時間と場所の情報を暗黙的にあるいは明示的に備えていることを考える．あるイベント言明 $e_1{:}E_1(o_1, \ldots, o_n)$ が成り立つとする．このとき，1 章で述べたように，通常そのイベントが生じた時間と場所が存在するはずである．時間と場所を t と l で表したとき，それは以下のように，述語引数として付加することができる．

$e_1{:}E_1(o_1, \ldots, o_n, t, l)$

時間と場所が不明確な場合や省略されている場合には，$e_1{:}E_1(o_1, \ldots, o_n)$ が，そのまま $(\exists x)(\exists y)e_1{:}E_1(o_1, \ldots, o_n, x{:}tim, y{:}loc)$ を暗黙に示していると考えるべきである．これは，$e_1{:}E_1(o_1, \ldots, o_n)$ と $e_2{:}E_1(o_1, \ldots, o_n)$ のように 2 つのイベント e_1，e_2 の描写が同じだが，それらが別々のイベントで，時間と場所が一致しない

☆ 状況理論における関係（relation）．

☆☆ 汎化 (∀) されたイベント変数は状況理論でいうところのパラメトリック・インフォン（parametric infon）

$\{\dot{E} | \dot{E}{:}\langle\!\langle E_1, o_1, \cdots, o_n \rangle\!\rangle\}$

と同様のものである．

場合に対応できる．たとえば，「e_1:目撃する (犯人)」と「e_2:目撃する (犯人)」のイベントでは，描写「犯人を目撃する」は同じだがイベントインスタンス e_1，e_2 は異なっており，「e_1: 目撃する (犯人, t_1, l_1)」と「e_2:目撃する (犯人, t_2, l_2)」のようにそれぞれ違う時間と場所で起こるイベントかもしれない．

各イベントの時間と場所を明記することを用いて，2 つのイベントの演算子や関係を扱う．本論文では特に，2 つのイベントからの合成演算子と 2 つのイベントの排他性関係を考える．以下のように 2 つのイベント言明が成り立っているとする．

e_1:$E_1(o_1, \ldots, o_n, t_1, l_1)$

e_2:$E_2(o'_1, \ldots, o'_m, t_2, l_2)$

このとき，2 つのイベントを合成するための条件は，イベントを構成する個体が同じであり，時間と場所も一致することである．すなわち，$o_1 = o'_1, \ldots, o_n = o'_m$ ($n = m$) および $t_1 = t_2$，$l_1 = l_2$ である．それにより得られた合成イベントは以下のように表される．

$e_1 \circ e_2$:$E_1 \sqcap E_2(o_1, \ldots, o_n, t_1, l_1)$

たとえば，2 つのイベント「e_1:不法侵入する (太郎, t, l)」と「e_2:窃盗する (太郎, t, l)」が成り立つとき，「$e_1 \circ e_2$:不法侵入する $\sqcap$ 窃盗する (太郎, t, l)」（太郎が不法侵入すると同時に窃盗する）という合成イベントが得られる．

しかし，それぞれのイベント言明に時間と場所が明示されていないときは，このようにはいかない．その場合，少なくとも 2 つのイベントが同じ時間と場所で起きていることを以下のイベント間の関係によって表す．

e_1:$E_1 \approx_t e_2$:E_2

e_1:$E_1 \approx_l e_2$:E_2

前者はイベント e_1，e_2 が同じ時間に起きていることを宣言し，後者は同じ場所で起きていることを宣言している．さらに，イベント間の重要な関係として排他性を導入する．その関係は 2 つのイベントソートに属しているイベントインスタンスが時間的に同時に成り立たないことを示しており，その性質はイベントインスタンスではなくイベントソートの種類に依存する．E_1，E_2 をイベントソートとすると，排他性は以下のように 2 つのイベントソート間の関係で宣言される．

$E_1 ||_E E_2$

たとえば，「証言する」と「黙秘する」がそれぞれイベントソートであるとき，ソート間の排他性

証言する $||_E$ 黙秘する

が成り立つ．ただし，イベントの排他性がいえても，時間さえ違えば同じ場所で成り立ってもよい．たとえば，2 つのイベント「e_1:証言する (太郎, t_1, l)」と「e_2:黙秘する (太郎, t_2, l)」は，場所 l において，太郎は時間 t_1 では証言していたが，時間 t_2 では黙秘していたことを表している．以上で述べたイベントの合成には，このイベントの排他性が成り立っていない制約が加えられる．先の例の「不法侵入しながら窃盗する」は「不法侵入する」と「窃盗する」が排他的でないので合成可能であるが，「証言する」と「黙秘する」は排他的なので「証言と黙秘を同時に行う」ようなイベントの合成は適切ではない．

イベントの合成には，時間と場所が一致するほかに，イベントを描写する個体が同じでなければならないという条件がある．しかしこの条件は，イベントの描写で論理式の引数構成が異なるような，多くの種類のイベント言明の合成を妨げてしまう．たとえば，1 章で述べた「e_1:運転する (太郎, 車)」と「e_2:話す (太郎, 携帯電話)」の合成を考えたときに，2 つ目の引数「車」と「携帯電話」が一致しないので，先ほど述べた合成の条件にはあてはまらない．これを解決するのに Kaneiwa と Tojo によって提案されている述語の引数操作[11] を取り入れる．その手順は，まず 2 つのイベントから引数を削除してより情報量の少ないイベント言明を導く．以上の 2 つのイベントの第 2 引数を削除すると，「e_1:運転する (太郎)」（太郎は運転している）と「e_2:話す (太郎)」（太郎は話している）が導かれる．これにより，「運転する」と「話す」が排他的でなく，「e_1:運転する $\approx_t$ e_2:話す」と「e_1:運転する $\approx_l$ e_2:話す」であるときに，「運転する$\sqcap$話す (太郎)」（太郎は話しながら運転している）のようなイベントの合成が可能になる．

以上に加えて，イベント間の時間的な関係を記述するために，包含関係 $\lhd_t$ と時間の順序関係 $\prec_t$ を導入する☆．これまで述べたイベントの記法に基づいて，次のように表す．

e_1:$E_1 \lhd_t e_2$:E_2

e_1:$E_1 \prec_t e_2$:E_2

前者がイベント e_2 がイベント e_1 を時間的に包含していることを示す．後者はイベント e_1 がイベント e_2 より時間的に先行していることを示す．たとえば，「e_1:出血する $\lhd_t$ e_2:負傷する」は，負傷している間に出血していることを示す．すなわち，「負傷しているイベント e_2 は，出血しているイベント e_1 を時間的に包含している」ことを示している．

☆ 加えて場所に対する包含関係，時間関係には overlapping や隣接などが考えられる．しかし本論文では，時間の包含関係と順序関係以外は法的推論においては重要ではなく，言語の形式化が無用に複雑になることを避けるために扱わない．

4. イベント記述の論理

本章では，個体上のソートと，イベント上の量化，ソート，合成および排他性を備えたイベント論理の構文と意味論を定義する．

定義 4.1（ソートイベント言語） ソート付きイベント言語は，イベントインスタンス記号 $e_1, e_2, \ldots$ の集合 $\mathcal{E}$，イベントソート記号 $E_1, E_2, \ldots$ の集合 $\mathcal{ES}$，イベントの 2 項述語記号 $\lhd_t, \prec_t, \approx_t, \approx_l$，イベントの 2 項関数記号 $\circ$（イベント合成），およびソート付き一階言語の記号を含んでいる．

$\mathcal{EV}_{\langle s_1,\ldots,s_n\rangle}$ は，述語宣言 $\langle s_1,\ldots,s_n\rangle$ 上のイベント変数 $X, Y, \ldots$ の集合である．$\mathcal{EV}$ は，すべての述語宣言上のイベント変数の集合を示す．

定義 4.2（イベントのソート階層） 排他性を含んだイベントのソート階層は，イベントソートの集合 $\mathcal{ES}$（ただし，$\bot_E$，$\top_E \in \mathcal{ES}$），$\mathcal{ES}$ 上の半順序集合 $\sqsubseteq_E$（イベントのサブソート関係）と $\mathcal{ES}$ 上の 2 項関係 $||_E$（イベントの排他性関係）からなる組 $(\mathcal{ES}, \sqsubseteq_E, ||_E)$ である．

イベントソート E，E' に対して，$E \sqcap E'$ と $E \sqcup E'$ はぞれぞれ E と E' の下限（greatest lower bound）と上限（least upper bound）を示している．本論文では，知識ベースの構築のしやすさからイベントのソート階層が必ずしも束（lattice）であることを仮定しない．したがって推論の過程でイベントソート E，E' の上限や下限が存在しない場合は，$E \sqcap E'$ と $E \sqcup E'$ と示して人工的な上限と下限を用意する．排他性関係 $E||_E E'$ は，イベントソート E，E' に含まれるイベントが同時に生じないことを意味している．

定義 4.3（イベントシグネチャ） ソートシグネチャ $\Sigma = (\mathcal{S}, \mathcal{F}, \mathcal{E}, D_S)$ が与えられているとする．このとき，ソート付きのイベント言語のソートシグネチャ（イベントシグネチャと呼ぶ）は，以下を満たすような組 $\Sigma_{ev} = (\mathcal{S}, \mathcal{F}, \mathcal{E}, \mathcal{ES}, \mathcal{T}, \mathcal{L}, D)$ である．ここで，D_E はイベントインスタンスのソート宣言からなる集合である．

(1) $\mathcal{T}$ は，時間記号 $t_1, t_2, \ldots$ の集合，
(2) $\mathcal{L}$ は，場所記号 $l_1, l_2, \ldots$ の集合，
(3) $\prec_t$（時間の順序関係）は，$\mathcal{T}$ 上の順序関係，
(4) $\lhd_t$（時間の包含関係）は，$\mathcal{T}$ 上の半順序関係，
(5) $e \in \mathcal{E}$，$E \in \mathcal{ES}$ のとき，$|e| = (t, l)$，$e{:}E \in D_E$ および $e{:}\langle s_1,\ldots,s_n\rangle \in D_S$，
(6) $D = D_S \cup D_E$．

イベントシグネチャのソート宣言 D は，ソートシグネチャ（定義 2.2）のソート宣言 D_S にイベントインスタンスに関する宣言 D_E を加えたものである．イベントシグネチャでは，$e{:}E \in D_E$ によって各イベントインスタンス e がイベントソート E のインスタンスとして宣言され，同時に $e{:}\langle s_1,\ldots,s_n\rangle \in D_S$ を満たす n 引数述語とも見なされる．したがって，イベントインスタンスは 2 つの異なった種類のソート宣言によって制限される．1 つはイベントソート（イベントの集まり）に属することであり，もう 1 つは n 項述語（個体間の関係）に属することである．$|e|$ は，イベント e が生じている時間と場所を示している．イベントシグネチャ Σ_{ev} で $|e| = (t, l)$ が宣言されているとき，$|e|_t = t$ かつ $|e|_l = l$ である．さらに，イベント変数 X には特定の時間や場所が決まっていないので，時間と場所の変数 x_t，y_l を導入して $|X| = (x_t, y_l)$ と定義する．次のイベント項の定義において，関数 $|\ |$ はイベント合成へ拡張される．

ソート付きイベント言語の表現（イベント項とイベント論理式）を定義する．

定義 4.4（イベント項） イベントシグネチャ $\Sigma_{ev} = (\mathcal{S}, \mathcal{F}, \mathcal{E}, \mathcal{ES}, \mathcal{T}, \mathcal{L}, D)$ が与えられているとする．このとき，イベント項 $et{:}E$ の集合は次のように帰納的に定義される．

(1) $e{:}E \in D_E$ かつ $e{:}\langle s_1,\ldots,s_n\rangle \in D_S$ のとき，イベントインスタンス $e{:}E$ は $\langle s_1,\ldots,s_n\rangle$ のイベント項である．
(2) $X \in \mathcal{EV}_{\langle s_1,\ldots,s_n\rangle}$ のとき，イベント変数 $X{:}E$ は $\langle s_1,\ldots,s_n\rangle$ のイベント項である．
(3) $et_1{:}E_1$ と $et_2{:}E_2$ がそれぞれ $\langle s_1,\ldots,s_n\rangle$ と $\langle s_1,\ldots,s_m\rangle$ $(n \le m)$ のイベント項，$|et_1|$ と $|et_2|$ が単一化可能であるとき，$et_1 \circ et_2{:}E_1 \sqcap E_2$ は $\langle s_1,\ldots,s_n\rangle$ のイベント項であり，$|et_1 \circ et_2| = mgu(|et_1|, |et_2|)$☆である．

項目 (1)，(2) ではイベントインスタンス $e{:}E$ とイベント変数 $X{:}E$ によるアトミックなイベント項を定義し，項目 (3) では $et_1{:}E_1$ と $et_2{:}E_2$ がそれぞれイベント項であるとき合成したイベント項を定義している．この合成されたイベント項 $et_1 \circ et_2{:}E_1 \sqcap E_2$ は，単一化可能な 2 つのイベント項を引数としたある種の関数といえる．もし 2 つのイベント項が単一化可能でないならば，その合成は未定義といえる．さらにイベント項に関する n 引数関数を導入することも可能だが，複雑さを避けるために本論文では導入しない．

☆ $|et_1|$ と $|et_2|$ の最汎単一化を示している．$|et_i|$ は時間（定数 t または変数 x_t）と場所（定数 l または変数 y_l）の組を表す．したがって，その最汎単一化は両方が変数ならば同一となるための変数のリネーミングであり，一方が定数ならもう一方の変数にその定数を代入することである．

イベント論理式は，2 種類の原子論理式（イベント述語論理式とイベント項の関係），論理結合子および一階と二階の量化子によって構成される．

定義 4.5（イベント論理式） イベントシグネチャ $\Sigma_{ev} = (\mathcal{S}, \mathcal{F}, \mathcal{E}, \mathcal{ES}, \mathcal{T}, \mathcal{L}, D)$ が与えられたとする．イベント論理式の集合は以下のように帰納的に定義される．

(1) $et{:}E$ が $\langle s_1, \ldots, s_m \rangle$ のイベント項であり，$r_1{:}s_1, \ldots, r_n{:}s_n$ $(n \leq m)$ がそれぞれソート $s_1, \ldots, s_n$ の項であるとき，$et{:}E(r_1{:}s_1, \ldots, r_n{:}s_n)$ はイベント論理式である．

(2) $et_1{:}E_1$ と $et_2{:}E_2$ がイベント項であるとき，$et_1{:}E_1 \prec_t et_2{:}E_2$，$et_1{:}E_1 \lhd_t et_2{:}E_2$，$et_1{:}E_1 \approx_t et_2{:}E_2$ および $et_1{:}E_1 \approx_l et_2{:}E_2$ は，イベント論理式である．

(3) A と B がイベント論理式であるとき，$\neg A$，$A \wedge B$，$A \vee B$，$A \rightarrow B$，$(\forall x)A[x{:}s]$ および $(\exists x)A[x{:}s]$ はイベント論理式である．

(4) A がイベント論理式であるとき，$(\forall X)A[X{:}E]$ と $(\exists X)A[X{:}E]$ はイベント論理式である．

関係 $et_1{:}E_1 \approx_t et_2{:}E_2$ と $et_1{:}E_1 \approx_l et_2{:}E_2$ は，$et_1{:}E_1 \approx et_2{:}E_2$ によって略記される．

ソート付きのイベント言語の意味論は，イベントシグネチャ Σ_{ev} の条件によって制約されたソートモデルによって定義される．$\vec{d} = (d_1, \ldots, d_n, t, l)$ は，個体の全体集合 U_o の要素 d_i と時間記号 t と場所記号 l からなる組を表す．$sb(\vec{d}) = \{(d_1, \ldots, d_i, t, l) \mid 1 \leq i \leq n\}$ は，5 章で定義する引数操作（削除）の推論に対応して意味論を定義するのに用いる．たとえば，「盗む (太郎, 次郎, 財布)」（太郎が次郎から財布を盗む）を解釈するために，時間記号 t と場所記号 l を付加した要素の組 (太郎, 次郎, 財布, t, l) を基にして解釈を行う．この組を sb に適用すると以下の集合が得られる．

$$sb(\text{太郎}, \text{次郎}, \text{財布}, t, l) = \{(\text{太郎}, \text{次郎}, \text{財布}, t, l), (\text{太郎}, \text{次郎}, t, l), (\text{太郎}, t, l)\}$$

さらに，場所情報は無視して時間記号だけを比較するために関数 $arg\text{-}tim(sb(\vec{d})) = \{(d_1, \ldots, d_i, t) \mid (d_1, \ldots, d_i, t, l) \in sb(\vec{d})\}$ を定義する．上記の例を適用すると以下のように場所記号が取り除かれた集合が得られる．

$$arg\text{-}tim(sb(\text{太郎}, \text{次郎}, \text{財布}, t, l)) = \{(\text{太郎}, \text{次郎}, \text{財布}, t), (\text{太郎}, \text{次郎}, t), (\text{太郎}, t)\}$$

定義 4.6 イベントシグネチャ Σ_{ev} のソートモデル（Σ_{ev}-モデルと呼ぶ）は，空でない集合 U_o，U_t，U_l（ただし，$U_o \cap U_t \cap U_l = \emptyset$）と $\mathcal{S} \cup \mathcal{F} \cup \mathcal{E} \cup \mathcal{ES} \cup \mathcal{T} \cup \mathcal{L} \cup \{\prec_t, \lhd_t\}$ を定義域とする解釈関数 I^+ からなる組 $M^+ = (U_o, U_t, U_l, I^+)$ であり，以下の条件を満たす．

(1) $M = (U_o, I)$ は，$I \subseteq I^+$ であるような Σ-モデル，

(2) $(U_t, I^+(\prec_t)) = (\mathcal{T}, \prec_t)$，

(3) $(U_t, I^+(\lhd_t)) = (\mathcal{L}, \lhd_t)$，

(4) $I^+(\circ)(sb(\vec{d_1}), sb(\vec{d_2})) = sb(\vec{d_1}) \cap sb(\vec{d_2})$，

(5) $I^+(E) \subseteq \{sb(\vec{d}) \mid \vec{d} \in \bigcup_{\{s_1,\ldots,s_n\} \subseteq \mathcal{S}} I^+(s_1) \times \cdots \times I^+(s_n) \times U_t \times U_l\}$，

(6) $E_1 \sqsubseteq_E E_2$ のとき，$I^+(E_1) \subseteq I^+(E_2)$，

(7) $I^+(\bot_E) = \emptyset$，

(8) $I^+(E_1 \sqcap E_2) = (I^+(E_1) \cap I^+(E_2)) \cup \{I^+(\circ)(sb(\vec{d_1}), sb(\vec{d_2})) \mid sb(\vec{d_1}) \in I^+(E_1), sb(\vec{d_2}) \in I^+(E_2)\}$，

(9) $E_1 ||_E E_2$ のとき，すべての $sb(\vec{d_1}) \in I^+(E_1)$ および $sb(\vec{d_2}) \in I^+(E_2)$ に対して，$arg\text{-}tim(sb(\vec{d_1})) \cap arg\text{-}tim(sb(\vec{d_2})) = \emptyset$，

(10) $e{:}E \in D_E$ のとき，$I^+(e) \in I^+(E)$，

(11) $|e| = (t, l)$ および $e{:}\langle s_1, \ldots, s_n \rangle \in D$ に対して，$I^+(e) = sb(\vec{d})$ $(\vec{d} \in I^+(s_1) \times \cdots \times I^+(s_n) \times \{I^+(t)\} \times \{I^+(l)\})$．

定義 4.6 では，3 章で説明したイベントの合成にともなう述語引数の調整やイベントの排他性が同時間でのイベントを禁止する条件が意味付けられている．条件 (4) では，2 つのイベントそれぞれの引数が順序を保って短いほうに一致するように定義されている．たとえば，関数 sb から得られた以下の 2 つの集合を考える．

$$sb(\text{太郎}, \text{車}, t, l) = \{(\text{太郎}, \text{車}, t, l), (\text{太郎}, t, l)\}$$
$$sb(\text{太郎}, \text{携帯電話}, t, l) = \{(\text{太郎}, \text{携帯電話}, t, l), (\text{太郎}, t, l)\}$$

これらを合成すると，積集合から以下の集合が得られる．

$$sb(\text{太郎}, \text{車}, t, l) \cap sb(\text{太郎}, \text{携帯電話}, t, l) = \{(\text{太郎}, t, l)\}$$

また条件 (9) では関数 $arg\text{-}tim$ を使って，イベントの排他性が時間の同一性を排除しても場所の相違を許すように定義されている．たとえば，「証言する $||_E$ 黙秘する」のとき，

$$sb(\text{太郎}, t_1, l_1) \in I^+(\text{証言する})$$
$$sb(\text{太郎}, t_2, l_2) \in I^+(\text{黙秘する})$$

に対して，条件 (9) により以下を満たす必要があるので，$l_1 = l_2$ はかまわないが $t_1 \neq t_2$ でなければならない．

$$arg\text{-}tim(sb(\text{太郎}, t_1, l_1)) \cap arg\text{-}tim(sb(\text{太郎}, t_2, l_2)) = \{(\text{太郎}, t_1)\} \cap \{(\text{太郎}, t_2)\} = \emptyset$$

個体集合上の変数割当て α は，ソート変数から U_o の要素への写像であり，条件 $\alpha(x{:}s) \in I^+(s)$ を満たす．$I^+(E_{\langle s_1,\dots,s_n\rangle}) = I^+(E) \cap \{sb(\vec{d}) \mid \vec{d} \in \bigcup_{1\le i\le n} I^+(s_1) \times \cdots \times I^+(s_i) \times U_t \times U_l\}$ を定義する．イベント上の変数割当て Ω は，イベント変数から $\bigcup_{i\in\mathbb{N}} U_o^i \times U_t \times U_l$ の部分集合への写像であり，$X \in \mathcal{EV}_{\langle s_1,\dots,s_n\rangle}$ に対して $\Omega(X{:}E) \in I^+(E_{\langle s_1,\dots,s_n\rangle})$ を満たす．

定義 4.7 Σ_{ev}-解釈は，Σ_{ev}-モデル M^+，個体集合上の変数割当て α，およびイベント上の変数割当て Ω からなる組 $\mathcal{I} = (M^+, \alpha, \Omega)$ である．ソート項とイベント項に対する解釈 $[\![\ ]\!]_{\alpha,\Omega}$ は，以下によって定義される．

(1) $[\![x{:}s]\!]_{\alpha,\Omega} = \alpha(x{:}s)$
(2) $[\![f(r_1{:}s_1,\dots,r_n{:}s_n)]\!]_{\alpha,\Omega} = I^+(f)([\![r_1{:}s_1]\!]_{\alpha,\Omega}, \dots, [\![r_n{:}s_n]\!]_{\alpha,\Omega})$
(3) $[\![e{:}E]\!]_{\alpha,\Omega} = I^+(e)$
(4) $[\![X{:}E]\!]_{\alpha,\Omega} = \Omega(X{:}E)$
(5) $[\![et{:}\bot_E]\!]_{\alpha,\Omega} = \emptyset$
(6) $[\![et_1 \circ et_2{:}E_1 \sqcap E_2]\!]_{\alpha,\Omega} = I^+(\circ)([\![et_1{:}E_1]\!]_{\alpha,\Omega}, [\![et_2{:}E_2]\!]_{\alpha,\Omega})$

上記の定義では，項目 (1)–(2) がソート項の解釈を定義し，項目 (3)–(6) がイベント項の解釈を定義している．

$\vec{d} = (d_1,\dots,d_n,t,l)$ とする．このとき，$arg(sb(\vec{d})) = \{(d_1,\dots,d_i) \mid (d_1,\dots,d_i,t,l) \in sb(\vec{d})\}$，$tim(sb(\vec{d})) = t$ と $loc(sb(\vec{d})) = l$ を定義する．

定義 4.8 $\mathcal{I} = (M^+, \alpha, \Omega)$ を Σ_{ev}-解釈，F をイベント論理式とする．充足可能性関係 $\mathcal{I} \models F$ は，以下によって定義される．

(1) $\mathcal{I} \models et{:}E(r_1{:}s_1,\dots,r_n{:}s_n)$ iff $([\![r_1{:}s_1]\!]_{\alpha,\Omega},\dots,[\![r_n{:}s_n]\!]_{\alpha,\Omega}) \in arg([\![et{:}E]\!]_{\alpha,\Omega})$
(2) $\mathcal{I} \models et_1{:}E_1 \prec_t et_2{:}E_2$ iff $(tim([\![et_1{:}E_1]\!]_{\alpha,\Omega}), tim([\![et_2{:}E_2]\!]_{\alpha,\Omega})) \in I^+(\prec_t)$
(3) $\mathcal{I} \models et_1{:}E_1 \lhd_t et_2{:}E_2$ iff $(tim([\![et_1{:}E_1]\!]_{\alpha,\Omega}), tim([\![et_2{:}E_2]\!]_{\alpha,\Omega})) \in I^+(\lhd_t)$
(4) $\mathcal{I} \models et_1{:}E_1 \approx_t et_2{:}E_2$ iff $tim([\![et_1{:}E_1]\!]_{\alpha,\Omega}) = tim([\![et_2{:}E_2]\!]_{\alpha,\Omega})$
(5) $\mathcal{I} \models et_1{:}E_1 \approx_l et_2{:}E_2$ iff $loc([\![et_1{:}E_1]\!]_{\alpha,\Omega}) = loc([\![et_2{:}E_2]\!]_{\alpha,\Omega})$
(6) $\mathcal{I} \models \neg A$ iff $\mathcal{I} \not\models A$
(7) $\mathcal{I} \models A \wedge B$ iff $\mathcal{I} \models A$ かつ $\mathcal{I} \models B$
(8) $\mathcal{I} \models A \vee B$ iff $\mathcal{I} \models A$ または $\mathcal{I} \models B$
(9) $\mathcal{I} \models A \to B$ iff $\mathcal{I} \models A$ ならば $\mathcal{I} \models B$
(10) $\mathcal{I} \models (\forall x)A[x{:}s]$ iff すべての $d \in I^+(s)$ に対して，$\mathcal{I}[x{:}s/d] \models A$
(11) $\mathcal{I} \models (\exists x)A[x{:}s]$ iff ある $d \in I^+(s)$ に対して，$\mathcal{I}[x{:}s/d] \models A$
(12) $\mathcal{I} \models (\forall X)A[X{:}E]$ iff すべての $sb(\vec{d}) \in I^+(E_{\langle s_1,\dots,s_n\rangle})$ に対して，$\mathcal{I}\{X{:}E/sb(\vec{d})\} \models A$
(13) $\mathcal{I} \models (\exists X)A[X{:}E]$ iff ある $sb(\vec{d}) \in I^+(E_{\langle s_1,\dots,s_n\rangle})$ が存在して，$\mathcal{I}\{X{:}E/sb(\vec{d})\} \models A$

Σ_{ev}-解釈 $\mathcal{I}$ が存在して $\mathcal{I} \models F$ のとき，論理式 F は充足可能である．そうでなければ，F は充足不可能である．すべての Σ_{ev}-解釈 $\mathcal{I}$ に対して，$\mathcal{I} \models F$ ならば，F は恒真である．

5. イベント論理の反駁推論

5.1 タブロー計算

ソート付きのタブロー法に基づいて，ソート付きイベント言語の推論メカニズムを提案する．本論文では，与えられたイベント論理式の集合が矛盾するかどうかを判定する反駁推論を採用する．

順序ソート論理のタブロー規則：A，B をイベント論理式とする．

$$\frac{A \wedge B}{A; B}\ (\wedge 1) \qquad \frac{\neg(A \wedge B)}{\neg A \mid \neg B}\ (\wedge 2) \qquad \frac{A \vee B}{A \mid B}\ (\vee 1)$$

$$\frac{\neg(A \vee B)}{\neg A; \neg B}\ (\vee 2) \qquad \frac{A \to B}{\neg A \mid B}\ (\to 1) \qquad \frac{\neg(A \to B)}{A; \neg B}\ (\to 2)$$

$$\frac{\neg\neg A}{A}\ (\neg\neg)$$

A をイベント論理式，$r{:}s'$ をソート s' の項，$c{:}s'$ をソート s' の新しいインスタンスとする．$s' \sqsubseteq_S s \in \Sigma$ のとき，

$$\frac{(\forall x)A[x{:}s]}{A[r{:}s']}\ (\forall 1) \qquad \frac{\neg(\forall x)A[x{:}s]}{\neg A[c{:}s']}\ (\forall 2)$$

$$\frac{(\exists x)A[x{:}s]}{A[c{:}s']}\ (\exists 1) \qquad \frac{\neg(\exists x)A[x{:}s]}{\neg A[r{:}s']}\ (\exists 2)$$

以上の順序ソート論理のタブロー規則に加えて，イベント論理式のために導入するタブロー規則は次のとおりである．

引数操作の規則：$et{:}E$ を $\langle s_1,\dots,s_n\rangle$ のイベント項とする．

$$\frac{et{:}E(r_1{:}s_1,\dots,r_m{:}s_m)}{et{:}E(r_1{:}s_1,\dots,r_{m-1}{:}s_{m-1})}\ (a1)$$

$$\frac{\neg et{:}E(r_1{:}s_1,\dots,r_{m-1}{:}s_{m-1})}{(\forall x)\neg et{:}E(r_1{:}s_1,\dots,r_{m-1}{:}s_{m-1},x{:}s_m)}\ (a2)$$

（ただし，$1 < m \leq n$，かつ $x{:}s_m$ は新しい変数である）

イベント合成の規則：$et_1{\circ}et_2{:}E_1{\sqcap}E_2$ を $\langle s_1, \ldots, s_m\rangle$ のイベント項，$\vec{t}$ をソート項 $(n \leq m)$ の列 $r_1{:}s_1, \ldots, r_n{:}s_n$ とする．

$$\frac{et_1 \circ et_2{:}E_1 \sqcap E_2(\vec{t})}{et_1{:}E_1(\vec{t}); et_2{:}E_2(\vec{t})}\ (c1)$$

$$\frac{\neg et_1 \circ et_2{:}E_1 \sqcap E_2(\vec{t}) \quad et_1{:}E_1 \approx et_2{:}E_2}{\neg et_1{:}E_1(\vec{t}) \mid \neg et_2{:}E_2(\vec{t})}\ (c2)$$

$$\frac{et_1 \circ et_2{:}E_1 \sqcap E_2(\vec{t}) \quad \neg(et_1{:}E_1 \approx et_2{:}E_2)}{et_1 \circ et_2{:}\bot_{\scriptscriptstyle E}(\vec{t})}\ (c3)$$

イベント排他性の規則：$et_1{:}E_1$ と $et_2{:}E_2$ をそれぞれ $\langle s_1, \ldots, s_n\rangle$ と $\langle s_1, \ldots, s_m\rangle$ のイベント項とし，$\vec{t}$ をソート項（$k \leq m, n$）の列 $r_1{:}s_1, \ldots, r_k{:}s_k$ とする．イベントシグネチャ Σ_{ev} において $E_1||_{\scriptscriptstyle E}E_2$ であるとき，

$$\frac{et_1{:}E_1(\vec{t}) \quad et_1{:}E_1 \approx_t et_2{:}E_2}{\neg et_2{:}E_2(\vec{t})}\ (d1)$$

$$\frac{et_1{:}E_1(\vec{t}) \quad et_2{:}E_2(\vec{t}) \quad et_1{:}E_1 \approx_t et_2{:}E_2}{et_1 \circ et_2{:}\bot_{\scriptscriptstyle E}(\vec{t})}\ (d2)$$

イベント量化の規則：$X \in \mathcal{EV}_{\langle s_1, \ldots, s_m\rangle}$ とし，A をイベント論理式，$et{:}E'$ を $\langle s_1, \ldots, s_n\rangle$（$n \leq m$）のイベント項，$e{:}E'$ を $\langle s_1, \ldots, s_k\rangle$（$k \leq m$）の新しいイベントインスタンスとする．イベントシグネチャ Σ_{ev} において $E' \sqsubseteq_{\scriptscriptstyle E} E$ であるとき，

$$\frac{(\forall X)A[X{:}E]}{A[et{:}E']}\ (\forall X1) \qquad \frac{\neg(\forall X)A[X{:}E]}{\neg A[e{:}E']}\ (\forall X2)$$

$$\frac{(\exists X)A[X{:}E]}{A[e{:}E']}\ (\exists X1) \qquad \frac{\neg(\exists X)A[X{:}E]}{\neg A[et{:}E']}\ (\exists X2)$$

引数操作の規則は，正のイベント論理式に含まれる最後の引数を削除したり，負のイベント論理式に対してソート変数を引数として補完したりする．イベント合成の規則は，合成したイベント言明を分解して2つのイベント言明を導く．そのとき，イベント言明の時間と場所は同じであり，イベントの排他性が成り立つことはない．2つのイベントソート間の排他性が成り立つとき，イベント排他性の規則によって，イベント言明がそれとは排他的なイベント言明の否定を導く．具体的には $E_1||_{\scriptscriptstyle E}E_2$ であるにもかかわらず，2つのイベントが同じ時間 $et_1{:}E_1 \approx_t et_2{:}E_2$ であると仮定すると，片方が偽であるか（$(d1)$ 規則），矛盾を導いている（$(d2)$ 規則）．イベント量化の規則は述語変数の量化の考えに基づき，イベント変数に対してサブソートのイベント項を代入する．拡張されたタブロー法を用いて，論理式集合から論理式集合への導出は次のように定義される．

定義 5.1（導出） Γ^i_n をイベント論理式の集合，F，F_i をイベント論理式，$F^* = \{F_1, \ldots, F_m\}$（$1 \leq m \leq 3$）とする．そのとき，$\Gamma^i_n$ から Γ^i_{n+1} への導出は以下のように定義される．

(1) $|e_1|_* = |e_2|_*$，$e_1{:}E_1$，$e_1{:}E_2 \in D_{\scriptscriptstyle E}$ かつ $e_1{:}E_1 \approx_* e_2{:}E_2 \not\in \Gamma^i_n$ のとき（ただし，$* \in \{t, l\}$），$\Gamma^i_{n+1} = \Gamma^i_n \cup \{e_1{:}E_1 \approx_* e_2{:}E_2\}$ である．

(2) $|e_1|_t R_i |e_2|_t$，$e_1{:}E_1$，$e_2{:}E_2 \in D_{\scriptscriptstyle E}$ かつ $e_1{:}E_1 R_i e_2{:}E_2 \not\in \Gamma^i_n$ のとき（ただし，$R_i \in \{\prec_t, \lhd_t\}$），$\Gamma^i_{n+1} = \Gamma^i_n \cup \{e_1{:}E_1 R_i\ e_2{:}E_2\}$ である．

(3) $|e_1|_t = |e_2|_t$，$e_1{:}E_1 R_i e_3{:}E_3$（または $e_3{:}E_3 R_i e_1{:}E_1$）$\in \Gamma^i_n$, $e_1{:}E_1$, $e_2{:}E_2$, $e_3{:}E_3$ in Σ_{ev} かつ $e_2{:}E_2 R_i e_3{:}E_3$（または $e_3{:}E_3 R_i e_2{:}E_2$）$\not\in \Gamma^i_n$ のとき（ただし，$R_i \in \{\prec_t, \lhd_t\}$），$\Gamma^i_{n+1} = \Gamma^i_n \cup \{e_2{:}E_2 R_i e_3{:}E_3\}$（または $\{e_3{:}E_3 R_i e_2{:}E_2\}$）である．

(4) $F^* \subseteq \Gamma^i_n$ かつ $F_1 \not\in \Gamma^i_n$ のとき（ただし，F^* と F_1 はタブロー規則の前提と結論），$\Gamma^i_{n+1} = \Gamma^i_n \cup \{F_1\}$ である．

(5) $F^* \subseteq \Gamma^i_n$ かつ $\{F_1, F_2\} \not\subseteq \Gamma^i_n$ のとき（ただし F^* と $F_1; F_2$ はタブロー規則の前提と結論），$\Gamma^i_{n+1} = \Gamma^i_n \cup \{F_1, F_2\}$ である．

(6) $F^* \subseteq \Gamma^i_n$ かつ $\{F_1, F_2\} \cap \Gamma^i_n = \emptyset$ のとき（ただし，F^* と $F_1 \mid F_2$ はタブロー規則の前提と結論），$\Gamma^i_{n+1} = \Gamma^i_n \cup \{F_1\}$ かつ $\Gamma^{i+k}_{n+1} = \Gamma^i_n \cup \{F_2\}$ である（ただし，Γ^{i+k}_{n+1}（$k \geq 1$）は，イベント論理式の新しい集合）．

導出 (1)–(5) は決定的であり，導出 (6) は非決定的である．Γ^i_j が $\{\neg F, F\}$ もしくは $\{et{:}\bot_{\scriptscriptstyle E}(\vec{t})\}$ を部分集合に持つとき，それは矛盾を含んでいるという．

補題 5.1 (i) Γ^i_{n+1} が決定的な導出によって Γ^i_n から導かれたとする．そのとき，Γ^i_n が充足可能ならば，Γ^i_{n+1} も充足可能である．(ii) Γ^i_{n+1}，Γ^{i+k}_{n+1} が非決定的な導出によって Γ^i_n から導かれたとする．そのとき，Γ^i_n が充足可能ならば，Γ^i_{n+1} もしくは Γ^{i+k}_{n+1} が決定可能である．

Proof. 導出 (1)–(3) に対しては，明らかである．導出 (4)–(6) に対して，各タブロー規則の前提が充足可能であるとき，その結論も充足可能であることを証明すればよい．

（引数操作の規則）$\mathcal{I} \models et{:}E(r_1{:}s_1,\ldots,r_m{:}s_m)$ を仮定する．そのとき，$[\![et{:}E]\!]_{\alpha,\Omega} = sb(\vec{d})$ であるので，任意の $\vec{d'} \in [\![et{:}E]\!]_{\alpha,\Omega}$ に対して，$sb(\vec{d'}) \subseteq [\![et{:}E]\!]_{\alpha,\Omega}$ が成り立つ．ゆえに，$\mathcal{I} \models et{:}E(r_1{:}s_1,\ldots,r_{m-1}{:}s_{m-1})$ である．

（イベント合成の規則 (a)）$\mathcal{I} \models et_1 \circ et_2{:}E_1 \sqcap E_2(\vec{t})$ とする．そのとき，$[\![\vec{t}]\!]_{\alpha,\Omega} \in arg([\![et_1{\circ}et_2{:}E_1{\sqcap}E_2]\!]_{\alpha,\Omega})$（ただし，$[\![et_1 \circ et_2{:}E_1 \sqcap E_2]\!]_{\alpha,\Omega} = I^+(\circ)([\![et_1{:}E_1]\!]_{\alpha,\Omega}, [\![et_2{:}E_2]\!]_{\alpha,\Omega})$）である．定義 4.6 (6) により，$[\![\vec{t}]\!]_{\alpha,\Omega} \in arg([\![et_1{:}E_1]\!]_{\alpha,\Omega}), arg([\![et_2{:}E_2]\!]_{\alpha,\Omega})$ が導かれる．したがって，$\mathcal{I} \models et_1{:}E_1(\vec{t})$ かつ $\mathcal{I} \models et_2{:}E_2(\vec{t})$ となる．

（イベント排他性の規則 (a)）$\mathcal{I} \models et_1{:}E_1(\vec{t})$ と $\mathcal{I} \models et_1{:}E_1 \approx_t et_2{:}E_2$ を仮定する．そのとき，$[\![\vec{t}]\!]_{\alpha,\Omega} \in arg([\![et_1{:}E_1]\!]_{\alpha,\Omega})$ である（ただし，$[\![\vec{t}]\!]_{\alpha,\Omega} = (d_1,\ldots,d_k)$, $tim([\![et_1{:}E_1]\!]_{\alpha,\Omega}) = t$ かつ $loc([\![et_1{:}E_1]\!]_{\alpha,\Omega}) = l$）．よって，$\vec{d} = (d_1,\ldots,d_k,\ldots,d_n,t,l)$ が存在し，$[\![et_1{:}E_1]\!]_{\alpha,\Omega} = sb(\vec{d})$ が成り立つ．定義 4.6 (7) により，$arg\text{-}tim([\![et_2{:}E_2]\!]_{\alpha,\Omega})$ は $(d_1,\ldots,d_k,t)$ を含まない．ゆえに，$\mathcal{I} \not\models et_2{:}E_2(\vec{t})$ である．

他のタブロー規則についても同様に証明される．□

定理 5.1（反駁の健全性） Γ_0^0 をイベント論理式の集合とする．$\Gamma_n^0,\Gamma_n^1,\ldots,\Gamma_n^m$ が Γ_0^0 から導出され，すべての Γ_n^i が矛盾を含んでいるとき，Γ_0^0 は充足不可能である．

Proof. Γ_0^0 から $\Gamma_n^0,\Gamma_n^1,\ldots,\Gamma_n^m$ の導出が存在すると仮定する．もし Γ_n^i が矛盾を含んでいるならば，それはモデルを持たないはずである．補題 5.1 によって，Γ_0^0 は充足不可能である．□

5.2 イベント論理式に対する推論例

前節で導入したタブロー計算を用いて，反駁推論による質問応答の推論例を示す．Γ をイベント論理式の集合，A をイベント論理式とする．Γ における A の恒真性を判定するために，提案したタブロー法を使って $\Gamma \cup \{\neg A\}$ が反駁可能かどうか推論する．次にあげる例において，$\Gamma \cup \{\neg A\}$ が反駁可能ならば質問 A の答えは yes であり，そうでなければ no である．

図 1 は，イベントのソート階層に関する知識とその推論例を示している．図中のシグネチャ1 では，イベントインスタンス e が ⟨ 人 ⟩ の述語として宣言され，イベントソート「凶器で刺す」に属する．そしてイベントのソート階層では，イベントソート「凶器で刺す」,「殴る」は「襲う」のサブソートと宣言される．ここで，事実 1「次郎を凶器で刺す」に対して，質問 1a「誰か人を襲ったか?」の答えは，yes を導く．これは $\{e{:}$凶器で刺す (次郎:男性)$\} \cup \{\neg(\exists X)X{:}$襲う $(y{:}$人$)\}$ にタブロー法を適用した結果，推論過程 1a のように矛盾が導かれ反駁推論が成功するからである．さらに，質問 1b「誰か女性を殴ったか?」については，反駁推論が失敗するので答えは no となる．シグネチャ2 では，イベントインスタンス e_1，e_2 はそれぞれ ⟨ 人, 携帯電話 ⟩ と ⟨ 人, 車 ⟩ の述語として宣言されて，イベントソート「話す」と「運転する」に属する．事実 2 には「太郎は携帯電話で話す」と「太郎は車を運転する」の 2 つのイベントが同じ時間と場所で成り立っている．これに対して，質問 2「太郎は話しながら運転したか?」の答えは，yes となる．この答えは，推論過程 2 によって反駁推論が成功したことによる．

続いて**図 2** は，イベント変数と量化，およびイベント間の時間関係によって表されたイベント規則に関する推論を示している．シグネチャ3 では，イベントインスタンス e_1 は ⟨ 人 ⟩ の述語として宣言され，イベントソート「凶器で刺す」に属している．規則 3 には 3 つのイベント規則が記述されている．その意味は，「人が流血しているならば，それ以前にその人は凶器で刺された」「人が流血しているとき，その間負傷している」「負傷は継続的である」となる．より厳密に説明すると，「人が流血するすべてのイベント X に対して，X 以前の時間に人を凶器で刺すイベント Y が存在する」「人が流血するすべてのイベント X に対して，X を時間的に包含して人が負傷するイベント Y が存在する」「人が負傷するすべてのイベント X に対して，そのすべての時間内で同様に負傷するイベント Y が成り立つ」となる．そこで，事実 3a で「次郎が流血する」が真であるとき，質問 3a「誰か凶器で刺されたか?」の答えは yes となる．これは推論過程 3a によって反駁推論が成功したことによる．さらに，質問 3b「次郎は流血かつ負傷していたか?」の答えは yes である．この反駁推論は推論過程 3b のように成功する．

6. 考察

6.1 イベント・セマンティクスの関連研究との比較

本節では，論理型イベント言語を新たに提案した意義を示すために先行研究との違いを議論する．自然言語の意味記述においては，イベントのインデックス e をとるようなセマンティクスはこれまでいくつか提案されてきた．その端緒は Davidson[6] による．例文

"Brutus stabbed Caesar."

に対し，Davidson 流の記述は以下のようになる（例文は文献 17) による）．

$$P\exists e[stabbing(e) \wedge subject(e, Brutus) \wedge object(e, Caesar) \wedge cul(e)]$$

シグネチャ1 e:⟨ 人 ⟩, e:凶器で刺す, 凶器で刺す $\sqsubseteq_E$ 襲う, 殴る $\sqsubseteq_E$ 襲う,
男性 $\sqsubseteq_S$ 人, 女性 $\sqsubseteq_S$ 人

事実 1 $\underline{e\text{:凶器で刺す (次郎:男性)}}_{(0)}$

質問 1a ?-X:襲う (y:人)

答え 1a *yes*, X:襲う $=$ e:凶器で刺す, y:人 $=$ c:男性

質問 1b ?-X:殴る (y:女性)

答え 1b *no*

推論過程 1a

$\Gamma_0^0 = \{\neg(\exists X)(\exists y)X\text{:襲う }(y\text{:人})\} \cup \Gamma_{fact1}$

$\Gamma_1^0 = \Gamma_0^0 \cup \{\neg(\exists y)e\text{:凶器で刺す }(y\text{:人})\}$ (by $\exists X2$)

$\Gamma_2^0 = \Gamma_1^0 \cup \{\underline{\neg e\text{:凶器で刺す (次郎:男性)}}_{(0)}\}$ (by $\exists 2$)

シグネチャ2 e_1:⟨ 人, 携帯電話 ⟩, e_1:話す, e_2:⟨ 人, 車 ⟩, e_2:運転する,
$|e_1| = |e_2|$, 男性 $\sqsubseteq_S$ 人

事実 2 $\underline{e_1\text{:話す (太郎:男性, }c_1\text{:携帯電話)}}_{(0)}$
$\underline{e_2\text{:運転する (太郎:男性, }c_2\text{:車)}}_{(1)}$

質問 2 ?-X:運転する $\sqcap$ 話す (太郎:男性)

答え 2 *yes*, X:運転する $\sqcap$ 話す $=$ $e_1 \circ e_2$:運転する $\sqcap$ 話す

推論過程 2

$\Gamma_0^0 = \{\neg(\exists X)X\text{:運転する} \sqcap \text{話す (太郎:男性)}\} \cup \Gamma_{fact2}$

$\Gamma_1^0 = \Gamma_0^0 \cup \{\neg e_1 \circ e_2\text{:運転する} \sqcap \text{話す (太郎:男性)}\}$ (by $\exists X2$)

$\Gamma_2^0 = \Gamma_1^0 \cup \{e_1\text{:運転する} \approx e_2\text{:話す}\}$ (by Deri.1)

$\Gamma_3^0 = \Gamma_2^0 \cup \{\neg e_1\text{:運転する (太郎:男性)}\}$ (by $c2$)

$\Gamma_4^0 = \Gamma_3^0 \cup \{(\forall x)\neg e_1\text{:運転する (太郎:男性, }x\text{:車)}\}$ (by $a2$)

$\Gamma_5^0 = \Gamma_4^0 \cup \{\underline{\neg e_1\text{:運転する (太郎:男性, }c_1\text{:車)}}_{(0)}\}$ (by $\forall 1$)

$\Gamma_3^1 = \Gamma_2^0 \cup \{\neg e_2\text{:話す (太郎:男性)}\}$ (by $c2$)

$\Gamma_4^1 = \Gamma_3^1 \cup \{(\forall y)\neg e_2\text{:話す (太郎:男性, }y\text{:携帯電話)}\}$ (by $a2$)

$\Gamma_5^1 = \Gamma_4^1 \cup \{\underline{\neg e_2\text{:話す (太郎:男性, }c_2\text{:携帯電話)}}_{(1)}\}$ (by $\forall 1$)

図 1 イベントのソート階層に関する推論
Fig. 1 Reasoning on event sort hierarchy.

ここで，e は固有のイベントのインデックスであり，P は過去（past）を表す様相である．$cul(e)$ は ‘culmination’ を表し，そのイベントが達成されたことを意味する．e に対する制約は形式意味論の発展とともに DRT（Discourse Representation Theory）[9),22)] において用いられるようになる．DRT においては，同例文は以下のような形式で記述される．

$$
\begin{array}{|c|}
\hline
e, x, y, t \\
\hline
Brutus(x) \\
Caesar(y) \\
e : \boxed{x\ stabs\ y} \\
e \subseteq t \\
t < n \\
\hline
\end{array}
$$

ここで，$e \subseteq t$ はイベント e がある時間 t に包含されること，すなわちその時間内に達成（culminate）されることを表し，$t < n$ によりその時間 t は ‘now’ に先行する過去であることを表す．本論文で意図したイベントとその時間との関係記述はこれらの背景研究に依拠する．

上記の先行研究では，法的推論のように知識ベースを構築してそこから推論結果を出力する実用システムを想定してはいない．そのためイベント記述を分析しても，それに特化した知識表現言語と推論エンジンを実現しているとはいい難い．しかし法的推論システムの研究においては，論理型言語などによる知識表現言語と推論エンジンの設計は最も重要な要素の 1 つである．そこで問題なのは，従来の一階述語論理やそれ

シグネチャ3 e_1:⟨ 人 ⟩, e_1:凶器で刺す

規則 3 $(\forall X)(\forall y)(X$:流血する $(y$:人$) \to (\exists Y)(Y$:凶器で刺す $\prec_t X$:流血する $\wedge\, Y$:凶器で刺す $(y$:人$)))$,
$(\forall X)(\forall y)(X$:流血する $(y$:人$) \to (\exists Y)(X$:流血する $\lhd_t Y$:負傷する $\wedge\, Y$:負傷する $(y$:人$)))$,
$(\forall X)(\forall y)(X$:負傷する $(y$:人$) \to (\forall Y)$ $(Y$:負傷する $\lhd_t X$:負傷する $\to Y$:負傷する $(y$:人$)))$

事実 3 $\underline{e_1\text{:流血する }(c\text{:人})}_{(0)}$

質問 3a ?-Z:凶器で刺す (次郎:人)

答え 3a yes, Z:凶器で刺す $= e_2$:凶器で刺す

質問 3b ?-Z:流血する $\sqcap$ 負傷する (次郎:人)

答え 3b yes, Z:流血する $\sqcap$ 負傷する $= e_1 \circ e_4$:流血する $\sqcap$ 負傷する

推論過程 3a

$\Gamma_0^0 = \{\neg(\exists Z)Z\text{:凶器で刺す (次郎:人)}\} \cup \Gamma_{rule3} \cup \Gamma_{fact3}$

$\Gamma_1^0 = \Gamma_0^0 \cup \{(\forall y)(e_1\text{:流血する }(y\text{:人}) \to (\exists Y)Y\text{:凶器で刺す} \prec_t e_1\text{:流血する} \wedge Y\text{:凶器で刺す }(y\text{:人}))\}$ (by $\forall X1$)

$\Gamma_2^0 = \Gamma_1^0 \cup \{e_1\text{:流血する (次郎:人)} \to (\exists Y)Y\text{:凶器で刺す} \prec_t e_1\text{:流血する} \wedge Y\text{:凶器で刺す (次郎:人)}\}$ (by $\forall 1$)

$\Gamma_3^0 = \Gamma_2^0 \cup \{\underline{\neg e_1\text{:流血する (次郎:人)}}_{(0)}\}$ (by $\to 1$)

$\Gamma_3^1 = \Gamma_2^0 \cup \{(\exists Y)e_2\text{:凶器で刺す} \prec_t Y\text{:流血する} \wedge Y\text{:凶器で刺す (次郎:人)}\}$ (by $\to 1$)

$\Gamma_4^1 = \Gamma_3^1 \cup \{e_2\text{:凶器で刺す} \prec_t e_1\text{:流血する} \wedge e_2\text{:凶器で刺す (次郎:人)}\}$ (by $\exists X1$)

$\Gamma_5^1 = \Gamma_4^1 \cup \{e_2\text{:凶器で刺す} \prec_t e_1\text{:流血する}, \underline{e_2\text{:凶器で刺す (次郎:人)}}_{(1)}\}$ (by $\wedge 1$)

$\Gamma_6^1 = \Gamma_5^1 \cup \{\underline{\neg e_2\text{:負傷する (次郎:人)}}_{(1)}\}$ (by $\exists X2$)

推論過程 3b

$\Gamma_0^0 = \{\neg(\exists Z)Z\text{:流血する} \sqcap \text{負傷する (次郎:人)}\} \cup \Gamma_{rule3} \cup \Gamma_{fact3}$

$\Gamma_1^0 = \Gamma_0^0 \cup \{(\forall y)(e_1\text{:流血する }(y\text{:人}) \to (\exists Y)e_1\text{:流血する} \lhd_t Y\text{:負傷する} \wedge Y\text{:負傷する }(y\text{:人}))\}$ (by $\forall X1$)

$\Gamma_2^0 = \Gamma_1^0 \cup \{e_1\text{:流血する (次郎:人)} \to (\exists Y)e_1\text{:流血する} \lhd_t Y\text{:負傷する} \wedge Y\text{:負傷する (次郎:人)}\}$ (by $\forall 1$)

$\Gamma_3^0 = \Gamma_2^0 \cup \{\underline{\neg e_1\text{:流血する (次郎:人)}}_{(0)}\}$ (by $\to 1$)

$\Gamma_3^1 = \Gamma_2^0 \cup \{(\exists Y)e_1\text{:流血する} \lhd_t Y\text{:負傷する} \wedge Y\text{:負傷する (次郎:人)}\}$ (by $\to 1$)

$\Gamma_4^1 = \Gamma_3^1 \cup \{e_1\text{:流血する} \lhd_t e_3\text{:負傷する} \wedge e_3\text{:負傷する (次郎:人)}\}$ (by $\exists X1$)

$\Gamma_5^1 = \Gamma_4^1 \cup \{e_1\text{:流血する} \lhd_t e_3\text{:負傷する}, \underline{e_3\text{:負傷する (次郎:人)}}_{(1)}\}$ (by $\wedge 1$)

$\Gamma_6^1 = \Gamma_5^1 \cup \{\neg e_1 \circ e_4\text{:流血する} \sqcap \text{負傷する (次郎:人)}\}$ where $|e_1| = |e_4|$ (by $\exists X2$)

$\Gamma_7^1 = \Gamma_6^1 \cup \{\underline{\neg e_1\text{:流血する (次郎:人)}}_{(0)}\}$ (by $c2$)

$\Gamma_7^2 = \Gamma_6^1 \cup \{\underline{\neg e_4\text{:負傷する (次郎:人)}}_{(2)}\}$ (by $c2$)

$\Gamma_8^2 = \Gamma_7^2 \cup \{(\forall y)(e_3\text{:負傷する }(y\text{:人}) \to (\forall Y)\ (Y\text{:負傷する} \lhd_t e_3\text{:負傷する} \to Y\text{:負傷する }(y\text{:人})))\}$ (by $\forall X1$)

$\Gamma_9^2 = \Gamma_8^2 \cup \{e_3\text{:負傷する (次郎:人)} \to (\forall Y)\ (Y\text{:負傷する} \lhd_t e_3\text{:負傷する} \to Y\text{:負傷する (次郎:人)})\}$ (by $\forall 1$)

$\Gamma_{10}^2 = \Gamma_9^2 \cup \{\underline{\neg e_3\text{:負傷する (次郎:人)}}_{(1)}\}$ (by $\to 1$)

$\Gamma_{10}^3 = \Gamma_9^2 \cup \{(\forall Y)(Y\text{:負傷する} \lhd_t e_3\text{:負傷する} \to Y\text{:負傷する (次郎:人)})\}$ (by $\to 1$)

$\Gamma_{11}^3 = \Gamma_{10}^3 \cup \{e_4\text{:負傷する} \lhd_t e_3\text{:負傷する} \to e_4\text{:負傷する (次郎:人)}\}$ (by $\forall X1$)

$\Gamma_{12}^3 = \Gamma_{11}^3 \cup \{\underline{\neg e_4\text{:負傷する} \lhd_t e_3\text{:負傷する}}_{(3)}\}$ (by $\to 1$)

$\Gamma_{13}^3 = \Gamma_{12}^3 \cup \{\underline{e_4\text{:負傷する} \lhd_t e_3\text{:負傷する}}_{(3)}\}$ (by Deri.3)

$\Gamma_{12}^4 = \Gamma_{11}^3 \cup \{\underline{e_4\text{:負傷する (次郎:人)}}_{(2)}\}$ (by $\to 1$)

図 2 イベント規則に関する推論
Fig. 2 Reasoning on event rules.

に基づいた論理プログラミング言語では述語表現は基本的にプロパティを表すので，イベント記述が頻繁に出現する法律推論の知識表現言語としては不十分という点である．特に法律知識の法令文や判例を表すには動的なイベント表現やイベントの型表現が必要で，さらにそれらのイベントを述語や変数としても扱う試みは New HELIC-II をはじめとする研究でも行われてきた．

本研究はイベント記述のために H 項の動詞的な概念を導入した New HELIC-II の研究に続き，イベント記述に特化した論理型言語を提案している．実際 New HELIC-II に不足している要素には以下があげられる．

(1) イベントの解釈に含まれる時間と場所
(2) イベントの合成・排他性
(3) (1) と (2) およびイベントのインスタンスと型表現を統合した意味論と推論メカニズム

イベントのインスタンスと型表現は，New HELIC-II では変数項と定数項として H 項にもすでに導入されている．本研究ではさらにイベントがインスタンス化されるときにイベントが起こった時間情報と場所情報を解釈に含めている．その解釈により複数のイベントインスタンスの合成やイベントソート間の排他性を整合的に扱うことができる．

本研究の最も重要な意義は，(1) と (2) の拡張とイベントのインスタンスと型表現をあらかじめ論理型言語に組み込み，**イベントの多様な表現を統合した意味論と推論メカニズム**を提案していることにある．論理型言語の拡張を議論するとき，**表現力**と**推論力**の違いを考えなければならない．本言語では，高階論理や一般量化子のような表現力の向上を提案したのではなく，それまでにないイベントのシンタックスを導入して推論力を向上させている．たとえば，イベントソート間の排他性を New HELIC-II の H 項で表すならば，以下の 3 つのルールを追加すればアドホックには対応できるかもしれない．

$r1 :: \bot(\text{事象 }1 = A/\top, \text{事象 }2 = B/\top)$
$\quad \leftarrow A/\text{証言する}\ (\text{主体} = X/\text{人間},$
$\qquad\qquad \text{時間} = Y/\top, \text{場所} = Z1/\top),$
$\quad\ B/\text{黙秘する}\ (\text{主体} = X/\text{人間},$
$\qquad\qquad \text{時間} = Y/\top, \text{場所} = Z2/\top)$
$r2 :: -A/\text{証言する}\ (\text{主体} = X/\text{人間},$
$\qquad\qquad \text{時間} = Y/\top, \text{場所} = Z1/\top)$
$\quad \leftarrow A/\text{黙秘する}\ (\text{主体} = X/\text{人間},$
$\qquad\qquad \text{時間} = Y/\top, \text{場所} = Z/\top)$
$r3 :: -A/\text{黙秘する}\ (\text{主体} = X/\text{人間},$
$\qquad\qquad \text{時間} = Y/\top, \text{場所} = Z1/\top)$
$\quad \leftarrow A/\text{証言する}\ (\text{主体} = X/\text{人間},$
$\qquad\qquad \text{時間} = Y/\top, \text{場所} = Z/\top)$

しかしこの場合，3 つの欠点が生じる．第 1 には，ユーザがすべてのイベントソートに関してルールを入力しなければならない．したがって，知識ベースに情報が追加されるたびに，新しいイベント表現すべてについて上記のルールを書き足す必要がある．第 2 は，ユーザごとにイベントに関する知識表現と推論メカニズムを考えなければならない．対して本言語では，タブロー法にイベントの推論規則が追加されているので汎用的に推論を実現してくれる．第 3 は，第 2 にも関連してイベント表現の意味論が定義されていないことである．論理型言語の大きな利点の 1 つに厳密な意味論の定義により推論の整合性を保証することがある．本研究ではイベント自体が持つ複雑さゆえにそれを特徴付ける複雑なシンタックスを採用しているが，そのおかげでシンタックスに従えば意味論の下で安全な推論が保証される．

法的推論のような高度な知識表現には，推論力の拡張が重要なことは一階述語論理と法的推論システムの知識表現言語との違いからも明らかである．たとえば，New HELIC-II の H 項で洗練されたシンタックスでも，一階述語論理の表現力を駆使すれば reification などにより同じ知識ベースが記述可能なので，数学的には言語の表現力が向上したわけではない．しかし H 項が有用で優れているのは，複雑な知識に適したシンタックスとそれを処理する推論メカニズムの拡張にある．したがって，本言語はイベントの知識表現と推論を目指して次の流れで拡張された結果といえる．まず論理プログラミングに型表現の拡張を施して LOGIN [1] とフレーム論理[13] が得られ，さらに（イベント）述語の型表現により New HELIC-II が得られる．そのうえイベント表現が精緻化されると同時に推論力が拡張されて論理型イベント言語が得られる．

＜イベント表現の拡張＞

論理プログラミング
↓
LOGIN とフレーム論理
↓
New HELIC-II
↓
論理型イベント言語

ただしこの拡張の流れはイベント記述に限った話で，New HELIC-II には本研究にはない多くの機能が提案されている．

6.2 法的推論への有用性

前節に続いて法律知識の表現と推論を実現する基盤として本言語の有用性を述べる．本論文でのイベントセマンティクスは法的推論を契機に設計・開発したために法的推論を主たる目的とした研究内容になっているが，もちろん汎用の意味記述装置である．しかしその設計・開発過程より法的推論に顕著にその効果が現れると判断される．具体的には本研究では，次に示すような法律知識の表現を容易にして，その結果，事例を常識知識，法令文と法律の解釈，および判例ルールに適用してある種の法的判断を推論することを目標にしている．たとえば，前章の推論例で「e_1:話す (太郎:男性, c_1:携帯電話)」と「e_2:運転する (太郎:男性, c_2:車)」の事実を用いたが，それらに以下の常識知識，法令文と法律の解釈，および判例ルールが追加された場合を考える．

＜常識知識＞

運転する $\sqcap$ 話す $\sqsubseteq$ 不注意運転する
運転する $\sqcap$ よそ見する $\sqsubseteq$ 不注意運転する
不注意運転する $\sqsubseteq$ 過失行為をする
医療ミスする $\sqsubseteq$ 過失行為をする

＜法令文や法律の解釈＞

$(\forall X)(\forall Y)(\forall x)(\forall y)(\forall z)$
(X:運転する (x:人, y:車) $\wedge$ Y:話す (x:人, z:携帯電話) $\wedge$ (Y:話す $\approx$ X:運転する) $\rightarrow$ 道路交通法違反 (x:人))

＜判例ルール＞

$(\forall X)(\forall x)$(X:不注意運転する (x:人) $\rightarrow$ $(\exists Y)$ $(\exists y)$(Y:ひく(x:人, y:歩行者)) $\rightarrow$ 重罪 (x:人))

常識知識では，イベントソート「運転する $\sqcap$ 話す」（話しながら運転する）と「運転する $\sqcap$ よそ見する」（よそ見して運転する）が「不注意運転する」の下位ソートとして宣言されている．さらに「不注意運転する」と「医療ミスする」が「過失行為をする」の下位ソートとなる．これらに加えて法令文や法律の解釈として,「車を運転しているとき，携帯電話で話しをするのは道路交通法違反である」ことを示す．また，判例ルールでは,「不注意運転して人をひいたとき，重罪になる」ことを示す．これらの知識により，推論システムを使えば「太郎が道路交通法違反である」ことが導かれるし，もし「太郎が歩行者をひいた」ならば判例ルールから「太郎は重罪である」ことが導ける．

では上記による法律知識の表現と推論において，本言語の優位性は何だろうか？ その答えとして，言語の拡張がイベント表現を簡潔にしており，それにもかかわらず推論規則の追加がイベント表現の推論力を高めていることがいえる．特に顕著なのは，**イベントが時間と場所を含めて解釈**され，加えて**イベントの合成を扱える能力**である．形式オントロジの研究分野でも議論されているように，イベントは人や物など可算物のインスタンスにはない性質を持ち，2 のイベントインスタンスを合わせればそれもまた 1 つのイベントインスタンスになる．したがって，常識知識で「運転する $\sqcap$ 話す $\sqsubseteq$ 不注意運転する」と宣言されているとき，イベントソート「話す」と「運転する」のそれぞれのインスタンス e_1，e_2 が合成されて 1 つのインスタンス $e_1 \circ e_2$ が生成され，それが上位イベントソート「不注意運転する」の 1 インスタンスとなる．判例ルールではこのイベントソートを用いているが，そのとき本言語では「不注意運転する」のイベントインスタンスが合成されたものかどうかは気にする必要がない．また,「不注意運転する」と同様にイベントソート「医療ミスする」の上位ソートに「過失行為をする」があるが「医療ミスする」のインスタンスは合成でないかもしれないので,「過失行為をする」は合成・非合成の両方のインスタンスを持つ可能性がある．このようなイベント合成などがあらかじめ意味論や推論に組み込まれているので，ユーザが意識しないで直観的にイベントソートの関係のみで階層が構築できるのは大きな利点である．

既存の言語では New HELIC-II やフレーム論理であっても，時間と場所やそれらに基づいて 2 つのイベントインスタンスが合成され 1 つのインスタンスを生成することは想定されていない．そのため事実，常識知識，法令文と法律の解釈，および判例ルールのすべての知識ベースで時間と場所を引数に加えて明示するとともに，複数のイベントインスタンスがどのように合成されるかを考えて擬似的に引数，変数や制約関係を駆使して記述しなければならない☆．これは言語のユーザに過負担であるし，適切に推論がもたされるかにも不安が残る．上記の常識知識を書くときにルール表現を用いて引数で明示的に時間・場所とイベント合成を考慮するのでは，イベントソートの概念階層で簡潔に常識知識を表す本来の利便性を失う．さらに判例ルールの記述では，常識知識に明示された複雑な引数

☆ フレーム論理は引数と変数だけでなく，オブジェクト指向のインスタンスと型を拡張した優れた表現を備える．しかし本問題は，時間や場所の情報を含み，静的なオブジェクトとは明らかに異質で動的なイベントのインスタンスと型である．フレーム論理にはイベントに特化したインスタンスと型はなく，イベントの性質をサポートするには冗長な表現を補う必要がある．たとえば，時間や場所，イベント識別子を格納する引数，合成・排他性にともなう時間，場所やイベントの制約関係がある．

構造を調べて「不注意運転する」のインスタンスに合成がありうるかを調べるのは不便であるばかりか，常識知識を共有知識ベースとする際にも障害となる．

6.3 実装に関して

将来的に本推論システムを実装するとき，いくらかの問題が残る．本言語では，合成イベントの包含関係が束ではなく半順序関係なので，推論の計算量が指数的に増加する可能性がある．その解決策にはタブロー法を改良して，論理プログラミングで用いられる導出原理を設計すれば，推論が効率的になる．その際，イベント階層が束でなければ，文献 5) で紹介されているソート階層の修正方法によって，下限を人工的に追加しイベントソート間の単一化が実現できる．また，合成イベントの表現は一種の 2 項関数と見なせば従来のソート項の単一化が適用できるだろう．

さらに本論文では，実際に法的知識ベースを構築して推論を実現する際に，閉世界仮説（CWA）を導入すべきか言及していない．時間と場所の情報によって各イベントが正確に記述できるようになって，ある時空間でのイベントの発生と未発生を明示できる．しかし，もし「発生していないイベントの否定情報をすべて記述しなければならない」ようなフレーム問題を回避するには，閉世界仮説が有用である．そのため実装では，本推論システムを少し改良して知識ベース内に現れるすべてのイベントインスタンス $e{:}E(\cdots)$ からその否定言明 $\neg e{:}E(\cdots)$ を質問応答システムに適用する．その結果 *no* が返ってきたら，$\neg e{:}E(\cdots)$ を知識ベースに追加すればデフォルト規則と同じ働きになりフレーム問題が解消される．

7. ま と め

本論文では法的推論を想定して，刑法における判例記述に不可欠なイベント表現とその推論に特化した論理型言語を提案した．具体的には，判例記述で用いられるイベントの多様な論理的性質をサポートするために，イベントの量化，ソート，合成や排他性の表現を導入してソート付きのイベント言語を形式化した．従来のアプローチとは異なって，イベントがインスタンス，ソート，述語や変数といった様々な形態で利用されることを想定した知識表現を提案している．その表現は，順序ソート論理のソート階層のようにイベントのソート階層を導入し，二階述語論理の述語定数や述語変数のようにイベントインスタンスとイベント変数をとらえている．さらに，ソートモデルを変形してイベント言明の意味論を定義する試みを行った．また法的推論システムに組み込む推論メカニズムへの応用を見すえて，イベント論理式のタブロー法による推論体系を新たに設計した．イベント合成の推論では，現実の述語引数が重要なものから並べられることを前提に，引数の並びが短い方に単一化するメカニズムを用いている．これは，引数の役割を明示する煩わしさを避けるとともに表現の簡潔さを保つためである．

今後の課題として，イベント言明に付随して日付や時刻に対する知識表現を扱う必要性が考えられる．そうしたイベントの時間関係に関するより詳細な記述や推論を可能にするためには，イベント言明に暗黙に含まれる時間情報と，具体的に明示された時間情報との相互関係を考慮する必要がある．たとえば，イベント論理と時間論理を融合することで時間情報との相互関係を推論へ反映できると考えられる．

謝辞 本研究の一部は，科学研究費補助金・若手研究（B）（No.17700164）の支援を受けて実施された．

参 考 文 献

1) Aït-Kaci, H. and Nasr, R.: LOGIN: A Logic Programming Language with Built-In Inheritance, *Journal of Logic Programming*, Vol.3, No.3, pp.185–215 (1986).
2) Allen, J.F. and Ferguson, G.: Actions and Events in Interval Temporal Logic, *Journal of Logic and Computation*, Vol.4, No.5, pp.531–579 (1994).
3) Barwise, J.: Constraints, Channels, and the Flow of Information, *Situation Theory and Its Applications*, Peters, S., Katagiri, Y., Israel, D. and Aczel, P. (Eds.), Vol.3, CSLI, Stanford University (1993).
4) Barwise, J., Gabbay, D. and Hartonas, C.: On the Logic of Information Flow, *Logic, Language, and Computation, vol.1*, Seligman, J. and Westerstahl, D. (Eds.), CSLI, Stanford University (1996).
5) Cohn, A.G.: Taxonomic reasoning with many sorted logics, *Artificial Intelligence Review*, Vol.3, pp.89–128 (1989).
6) Davidson, D.: The logical form of action sentences, *Essays on actions and events*, pp.105–148, Clarendon Press, Oxford (1980).
7) Galton, A. and Augusto, J.C.: Two Approaches to Event Definition, *Proc. DEXA 2002*, LNCS 2453, pp.547–556 (2002).
8) Gatziu, S. and Dittrich, K.R.: Events in an Active Object-Oriented Database System, *Proc. 1st International Workshop on Rules in Database Systems*, Workshops in Computing, pp.23–29, Springer (1994).
9) Kamp, H. and Reyle, U.: *From Discourse to*

Logic, Kluwer Academic (1993).
10) Kaneiwa, K. and Mizoguchi, R.: Ontological Knowledge Base Reasoning with Sort-Hierarchy and Rigidity, *Proc. 9th International Conference on the Principles of Knowledge Representation and Reasoning* (*KR2004*), pp.278–288 (2004).
11) Kaneiwa, K. and Tojo, S.: Event, Property, and Hierarchy in Order-Sorted logic, *Proc. 1999 Int. Conf. on Logic Programming*, pp.94–108, The MIT Press (1999).
12) 兼岩　憲，東条　敏：イベントとプロパティの区別を導入した型階層論理，コンピュータソフトウェア，Vol.17, No.2, pp.10–24 (2000).
13) Kifer, M., Lausen, G. and Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages, *J. ACM*, Vol.42, No.4, pp.741–843 (1995).
14) Landman, F.: *Structures for Semantics*, Kluwer (1991).
15) Nitta, K., et al.: Knowledge Representation of New HELIC-II, *Workshop on Legal Application of Logic Programming, ICLP '94* (1994).
16) Nitta, K., et al.: New HELIC-II: A software tool for legal reasoning, *Proc. 5th Int. Conft. on Artificial Intelligence and Law*, ACM Press (1995).
17) Parsons, T.: *Events in the Semantics of English*, MIT press (1990).
18) Sadri, F. and Kowalski, R.A.: Variants of the Event Calculus, *Proc. 12th International Conference on Logic Programming*, Sterling, L. (Ed.), pp.67–82, MIT Press (1995).
19) Schmidt-Schauss, M.: *Computational Aspects of an Order-Sorted Logic with Term Declarations*, Springer-Verlag (1989).
20) Socher-Ambrosius, R. and Johann, P.: *Deduction Systems*, Springer-Verlag (1996).
21) Sowa, J.F.: *Knowledge Representation: logical, philosophical and computational foundations*, Brooks/Cole (2000).
22) 東条　敏：言語・知識・信念の論理，オーム社 (2006).
23) van Benthem, J.: *The logic of time: a model-theoretic investigation into the varieties of temporal ontology and temporal discourse*, Synthese Library, Vol.156, Reidel (1983).

(平成 18 年 11 月 21 日受付)
(平成 19 年 9 月 3 日採録)

兼岩　　憲（正会員）

1993〜1996 年富士通（株）．2001 年北陸先端科学技術大学院大学情報科学研究科博士後期課程修了．同年国立情報学研究所情報学基礎研究系助手．2006 年 4 月より，（独）情報通信研究機構研究員，国立情報学研究所客員准教授．情報科学博士．論理プログラミング，順序ソート論理，知識表現，形式オントロジ，UML ダイアグラム矛盾検証の研究に従事．2005 年人工知能学会論文賞．ソフトウェア科学会，電子情報通信学会，人工知能学会，ALP，ASL 各会員．

東条　　敏（正会員）

1981 年東京大学工学部計数工学科卒業，1983 年東京大学大学院工学系研究科修了．1983〜1995 年三菱総合研究所．1995 年北陸先端科学技術大学院大学情報科学研究科助教授，2000 年同教授．博士（工学）．自然言語の形式意味論および人工知能の論理の研究に従事．人工知能学会，ソフトウェア科学会，言語処理学会，認知科学会，Folli 各会員．