

# 述語項構造シソーラスと EC を用いた 汎用的常識推論エンジンの構築

杉本 智紀 藤田 響平 井坂 諭紀雄 永田 基樹 森 遼太

株式会社 pluszero

{sugimoto.tomoki, fujita.kyohei, isaka.yukitaka, nagata.motoki,  
mori.ryota}@plus-zero.co.jp

## 1 はじめに

我々は日常的に常識を用いた推論を行っている。例えば、以下のような文章を考える。

リビングに新聞がある。

リサが新聞をとってキッチンへ歩いて行った。

新聞がどこにあるかと聞かれれば、誰もがキッチンであると答えるだろう。しかし、このような常識を用いた推論エンジンを汎用的に構築することは未だ実現していない。常識を用いた推論の実現手段の一つとして、以下に示すような1つの事前処理と新しい文章の入力を前提とした3つの処理に分解することが考えられる。

(0) 事前処理として常識を意味表現に変換しておく

(1) 自然言語文に対して意味役割付与を行う

(2) 付与された意味役割を意味表現に変換する

(3) 自然言語文から変換した意味表現と、常識から変換した意味表現を用いて、推論を行う

(1) の中でも、特に日本語を対象にした意味役割付与は、言語学において、項構造から事態間の関係を分析する語彙概念構造が議論されており [1]、竹内らにより語彙概念構造の理論に基づいて述語項構造シソーラスとして体系化されている [2]。述語項構造シソーラスには、常識の一部である動詞の意味構造が網羅的に記述されている。さらに、述語項構造シソーラスの分類体系に基づいて自然言語文を解析し、述語の語義同定及び意味役割付与を行うシステム (ASA) が構築されている [3]。

また、(3) の常識を用いた推論における意味表現と、その推論手法として Event Calculus (EC) [5] が提案されている。EC は解集合プログラミング (ASP) を用いて定式化されている [6]。さらに、ASP のソルバーである *clingo* [7] を用いて、EC による推論を行うシステムとして Event Calculus Answer Set Programming (ecasp) が開発されている [5]。

ASA は (1) に、ecasp は (3) に対応するため、事前処理 (0) と処理 (2) を構築し、組み合わせることで、常識を用いた推論エンジンを構築できると考えられる。

本研究では、ASA による意味役割付与を EC の意味表現へと変換するシステム (ASA2EC) を構築し、述語項構造シソーラスに記述された事態間の関係を EC の公理として大規模に構築した。さらに、構築した EC の公理、ASA2EC、および先行研究である ASA、ecasp を図 1 のように組み合わせることで、常識を用いた推論エンジンを構築した。

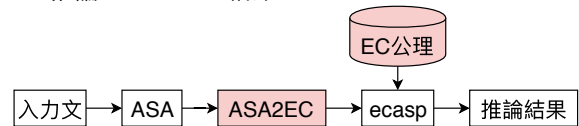


図 1: 常識を用いた推論エンジンの全体図

## 2 関連研究

### 2.1 Event Calculus

EC は event と event による状態 (fluent) の変化を表現し、推論するための意味表現と推論手法である [5]。event と fluent は引数を持つことができ、「私は家に行く。」という event は  $walk("私", "家")$ 、「私は家にいる。」という fluent は  $located("私", "家")$  と表される。

event による fluent の変化は initiates および terminates を用いて記述される。任意の時刻  $T$  において、ある event  $e$  が発生すると、fluent  $f$  が成立する場合、 $initiates(e, f, T)$  と表される。例えば「私が家に行く」ことで「私は家にいる」状態が成立する場合は  $initiates(walk("私", "家"), located("私", "家"), T)$  と表される。同様に、任意の時刻  $T$  においてある event  $e$  が発生すると、fluent  $f$  が成立しなくなる場合、 $terminates(e, f, T)$  と表される。

EC の公理を半自動的に構築した例として、Muellerらは Opem Mind Common Sense [8] というプロジェクトで収集されたデータを元に、event, fluent を 215

表 1: 述語項構造シソーラスと EC の公理の対応関係

見出し語	大分類1	小分類1	小分類2	深層格	Frame	構築された公理
長い	状態変化なし	長/短	長い	対象	[1]の長さや時間が長い状態 である	$\text{fluent}(\text{long}(\text{target}))$
開く	状態変化あり	形状(開閉)	開ける	対象	([動作主]の働きかけで) [1]が開いた状態 になる	$\text{event}(\text{open}(\text{target}))$ $\text{fluent}(\text{opened}(\text{target}))$ $\text{initiates}(\text{open}(\text{target}), \text{opened}(\text{target}), \text{time})$

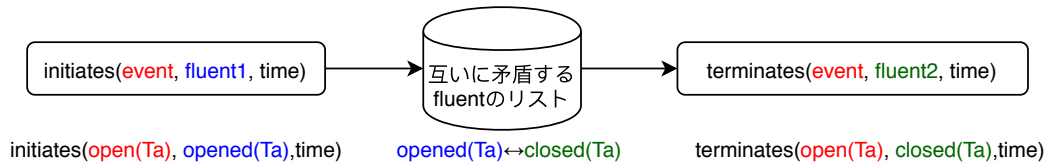


図 2: terminates の構築方法

個ずつ, initiates, terminates を 41 個ずつ作成したことが挙げられる [5].

## 2.2 述語項構造シソーラス

述語項構造シソーラスとは, 動詞, 形容詞, 形容動詞に関して語義概念, 意味役割, 概念フレームを記述した語彙資源である [2]. 語義概念とは語の意味のことである. 例えば, 動詞「動く」の語義概念は「状態変化あり-位置変化-位置変化(物理)-位置そのものの変化」となっている. 多義語は意味ごとに異なる語義概念が記述されている. 意味役割は述語と項の関係を表すものであり, 例えば, 動詞「動く」の意味役割は対象となっている. 概念フレーム(以下 Frame)は, 述語が示す事象が発生した後どのような状態になるか, または示す状態がどのような状態であるかについて記述したものである. 例えば, 動詞「動く」の概念フレームは「([動作主]の働きかけで) [1]の位置が変化する」となっており, 形容詞「長い」の概念フレームは「[1]の長さや時間が長い状態である」となっている.

## 2.3 ASA

日本語意味役割付与システム(ASA)とは自然言語文を入力とし, 入力文に対して述語項構造解析を行って述語の語義を同定し, 係り受け関係にある項に意味役割を付与するシステムである [3]. ASA の出力は全て述語項構造シソーラスに準じている.

## 3 提案手法

本章では, 述語項構造シソーラスを用いた EC の公理の大規模構築手法, ASA の出力を EC 表現に変換する方法を述べる.

### 3.1 EC の公理の大規模構築

表 1 に述語項構造シソーラスを用いた EC の公理の大規模構築の概要を示す. まず, 大分類 1 が「状態変化なし」である語から fluent を構築する. 次に, 「状

態変化あり」の語に対して, 小分類 1 と小分類 2 から event を, 付与されている Frame から fluent を構築する. その後, 「状態変化あり」の語から構築した event と fluent を組み合わせて initiates を構築する. 最後に, 構築した initiates と, 独自に定義した互いに矛盾する fluent を組み合わせて, terminates を構築する.

#### 3.1.1 event, fluent の構築

まず, 「状態変化なし」となっている語から fluent を構築する. fluent は以下のように記述され, fluent の名称と引数が必要である.

$$\text{fluent}(\text{opened}(Ta)) :- \text{target}(Ta).$$

fluent の名前は小分類 1 と小分類 2 を結合して定義し, 引数は深層格を用いて定義する.

次に, 「状態変化あり」の語に対して, 小分類 1 と小分類 2 から event を, 付与されている Frame に記述された状態から fluent を構築する. event は以下のように記述され, event の名称と引数が必要である. fluent と同様に, event の名前は小分類 1 と小分類 2 を結合して定義し, 引数は深層格を用いて定義する.

$$\text{event}(\text{open}(Ta)) :- \text{target}(Ta).$$

fluent については, その名称を Frame から定義する. 例えば, 表 1 の「開く」の Frame に対応する fluent の名称は「開いた状態」とする. 引数は深層格ではなく, 同じ見出し語から構築された event の引数を元に, 以下のように定義する.

(1) event の引数に「動作主」と「対象」の両方が入っている場合は event の引数から「動作主」を除いたものを fluent の引数として用いる.

(2) (1) 以外は event の引数を fluent の引数とする.

(1) は, 「(動作主) が (対象) を (着点) に置く」というような場合に対応する. event の引数から (動作主) を除くことで, fluent の引数として (対象) と (着点)



図 3: ASA2EC の構造

が用いられ、(対象)が(着点)にいる状態を表す fluent を定義することができる。(2)は、「(対象)が開く」というような場合に対応する。event の引数である(対象)が fluent の引数として用いられ、(対象)が開いた状態を表す fluent を定義することができる。

### 3.1.2 initiates の構築

initiates は以下のように記述され、event と fluent の組が必要である。前述の方法で「状態変化あり」の語から構築した event と fluent の組を用いて構築する。

$$\text{initiates}(\text{open}(Ta), \text{opened}(Ta), T) \\ \text{:- target}(Ta), \text{time}(T).$$

### 3.1.3 terminates の構築

terminates の構築手法の概要を図 2 に示す。terminates は「ある event と、その影響で成立しなくなる fluent との関係」を表したものであるが、直接的に述語項構造ソーラスに記述されているわけではない。本研究では、互いに矛盾する fluent の組を定義し、一方を成立させる event が発生したとき、もう一方の矛盾する fluent は成立しなくなるべきであるという仮定をもとに、terminates を構築した。互いに矛盾する fluent の組は、表 2 の 5 パターンの関係性をもとに定義した。構築した terminates の具体例を以下に示す。

$$\text{terminates}(\text{open}(Ta), \text{closed}(Ta), T) \\ \text{:- target}(Ta), \text{time}(T).$$

表 2: terminates の構築に必要な fluent の組

矛盾の種類	fluent1	fluent2
否定	existence(Target)	not_existence(Target)
対義語	opened(Target)	closed(Target)
位置の違い	located(Agent,Place1)	located(Agent,Place2)
所有者の違い	have(Agent1,Target)	have(Agent2,Target)
存在と消滅	existence(target)	disappear(Target)

## 3.2 ASA の出力の EC 表現への変換

常識を用いた推論エンジンを構築するためには、ASA による意味役割付与を EC の意味表現である event, fluent, object に変換する必要がある。ASA2EC の概要を図 3 に示す。

event や fluent への変換には、公理の構築において得られた、述語項構造ソーラスの語義概念から event と fluent への変換表を用いる。変換表によって、ASA の出力から得られる語義概念が event や fluent に変換される。event や fluent に必要な引数は、ASA の出力の深層格から取得する。語義概念が「状態変化あり」の場合は happens(event,time) に、「状態変化あり」の場合は holdsAt(fluent,time) に変換される。

ASA の出力における語義概念及び意味役割の情報が全て述語項構造ソーラスに準じているために EC 表現への変換が可能となっている。

## 4 結果

提案手法により構築した EC の公理、ASA2EC 表現への変換、および ASA, ecasp を図 1 のように組み合わせ、常識を用いた推論エンジンを構築した。推論システムの汎用性と、自然言語文入力時のシステムの推論結果を以下に示す。

### 4.1 推論システムの汎用性

本研究により、event を 2263 個、fluent を 3427 個、initiates を 2057 個、terminates を 39110 個獲得した。述語項構造ソーラスに記述された全ての事態間の意味関係を、公理として推論システムに組み込むことができている。述語項構造ソーラスは基本語意味データベース Lexeed の動詞、形容詞、形容動詞をほとんど網羅しているため、基本語意味データベース Lexeed の動詞、形容詞、形容動詞を用いた文を EC 表現に変換することが可能となっている。

### 4.2 推論結果

本研究では表 3 の 10 文を入力して推論結果を確認した。「私は本を買う。」を入力とした時の詳細な

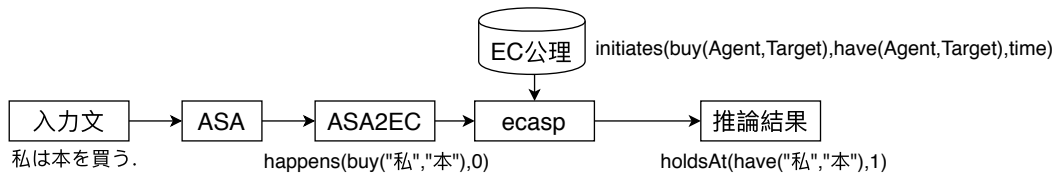


図 4: 推論結果の具体例

表 3: 自然言語文での推論エンジンの検証結果

入力文	EC表現	EC公理	推論結果
私は家にいる 私は学校に行く	located("私","家") walk("私","学校")	initiates( walk("私","学校"), located("私","学校"), 1) terminates( walk("私","学校"), located("私","家"), 1)	-located("私","家") +located("私","学校")
私は本を持っている 私は本を友達に貸す	have("私","本") lend("私","本","友達")	initiates( lend("私","本","友達"), have("友達","本"), 1) terminates(lend("私","本","友達"), have("私","本"), 1)	-have("私","本") +have("友達","本")
ドアが閉じている ドアが開く	closed("ドア") open("ドア")	initiates( open("ドア"), opened("ドア"), 1) terminates( open("ドア"), closed("ドア"), 1)	-closed("ドア") +opened("ドア")
父が眠っている 父が起床する	asleep("父") wake("父")	initiates( wake("父"), awake("父"), 1) terminates( wake("父"), asleep("父"), 1)	-asleep("父") +awake("父")
本が存在する 本が消滅する	existence("本") disappear("本")	initiates( disappear("本"), disappeared("本"), 1) terminates( disappear("本"), existence("本"), 1)	-existence("本") +disappeared("本")

推論結果を図 4 に示す。「私は本を買う。」という文が、ASA と ASA2EC により  $happens(buy("私","本"),0)$  という EC 表現に変換され、構築した公理  $initiates(buy(Agent,Target),have(Agent,Target),T)$  と ecasp を用いて、 $holdsAt(have("私","本"),1)$  が推論されたことを表している。これは、「主体が客体を買うと、主体が客体を所有する状態になる」という常識を用いており、「私が本を買う」ことで、「私が本を持っている状態になった」ことを意味する。このように、構築した推論システムを用いることで、常識を用いた推論が可能となっている。

## 5 おわりに

本研究では、ASA の出力を EC の意味表現へと変換するシステム (ASA2EC) を構築し、述語項構造ソーラスに記述された事態間の関係を EC の公理として大規模に構築した。構築した EC の公理、ASA2EC、ASA、ecasp を組み合わせることで、常識を用いた推論エンジンを構築した。文章入力時の出力結果から推論エンジンが正しく推論できることを確認した。

## 参考文献

- [1] 影山太郎. 動詞意味論: 言語と認知の接点. くろしお出版, 1996.
- [2] 竹内孔一, 石原靖弘, 竹内奈央. 述語項構造のソーラス分類と意味役割の設計について. 人工知能学会全国大会, p. 2D4OS03a1, 2013.
- [3] 竹内孔一, 土山傑, 守屋将人, 森安祐樹. 類似した動作や状況を検索するための意味役割及び動詞

語義付与システムの構築. 電子情報通信学会技術研究報告. NLC, 言語理解とコミュニケーション, 109(390), pp. 1-6, 2010.

- [4] 笠原要, 佐藤浩史, 田中貴秋, 藤田早苗, 金杉友子, 天野成昭. 「基本語意味データベース: Lexeed」の構築. 情報処理学会研究報告自然言語処理 (NL), 2004.1(2003-NL-159), pp. 75-82, 2004.
- [5] Mueller, Erik T. Commonsense reasoning: an event calculus based approach. Morgan Kaufmann, 2014.
- [6] Kim, Tae-Won, Lee, Joohyung, and Palla, Ravi. Circumscriptive event calculus as answer set programming. In Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009), pp. 823-829, 2009.
- [7] Gebser, Martin, Kaminski, Roland, Kaufmann, Benjamin, and Schaub, Torsten. Answer set solving in practice. San Rafael, CA: Morgan & Claypool, 2013.
- [8] Singh, Push, Lin, Thomas, Mueller, Erik T., Lim, Grace, Perkins, Travell, and Zhu, Wan L. Open Mind Common Sense: Knowledge acquisition from the general public. OTM Confederated International Conferences "On the Move to Meaningful Internet Systems". Springer, Berlin, Heidelberg, pp. 1223-1237, 2002.