

組合せ範疇文法による ドイツ語文の原因・結果関係の意味表示

高橋直人 一杉裕志 竹内泉

産業技術総合研究所人工知能研究センター

{naoto.takahashi,y-ichisugi,takeuti.i}@aist.go.jp

概要

自然言語において、原因を表す文と結果を表す文の関係は、様々な統語構造で表現可能である。そのような異なる表現の間では、異なった統語構造が意味上の細かな差を生じる事はあっても、一方の文が原因を表し、もう一方の文が結果を表すという基本的な原因・結果関係は共通している。本稿では、ドイツ語で書かれた原因・結果文を組合せ範疇文法に基づいて解析する際に、統語構造が異なっても同一の意味表示が合成できるような語彙項目の設定を提案する。またその際、正書法上必要とされるコンマを積極的に利用して文関係の明示化を試みる。

1 はじめに

人間の知性を他の生物の知性から区別する最大の特徴は、階層構造を持った言語の使用であると言える。

文は語の連続から構成されるが、語の並び方には必然性があり、語順が変われば一般に意味も変わる。また、たとえ表層の語順が同一であっても、内部の統語構造が異なれば文の意味も異なる場合がある。反対に、統語構造の異なる複数の文が（ニュアンスの違いこそあるにせよ）同一の意味を表す場合もある。したがって、自然言語を人間同様に解釈する人工知能を構築するためには、文を単なる語の集合（あるいは語の系列）とみなして統計的に処理するだけでは不十分であり、統語構造に基づいて、論理的に意味を合成する必要がある。

本稿では、統語構造に基づいた論理的な意味合成の具体例として、ドイツ語文における原因・結果関係と、組合せ範疇文法に基づいてそれを解析するための語彙項目設定に関して考察する。

2 組合せ範疇文法

組合せ範疇文法 [1, 2, 3, 4] は語彙化文法の一環であり、古典的範疇文法に関数合成規則を取り入れたものである。組合せ範疇文法の弱生成能力は、文脈自由文法と文脈依存文法の間位置する事が知られており、それゆえ自然言語の文法記述に適していると考えられる。

2.1 統語範疇

通常の句構造文法では、個々の文法範疇に対して固有の非終端記号を割り当てる。たとえば文には S (sentence)、名詞句には NP (noun phrase)、動詞句には VP (verb phrase) 等の記号を割り当てる事が多い。これに対し組合せ範疇文法では、ごく少数の基底範疇と、それらを結ぶ演算子を再帰的に用いて多種多様な統語範疇を表現する。

たとえば英語の動詞句は、

自分自身の左側に主語となる名詞句が隣接した場合、全体として文となるような統語範疇と定義できる。したがって動詞句は、 VP という固有の記号を使わなくとも、 S (文) と NP (名詞句) の組み合わせで $S \setminus NP$ と表現できる。ここで $X \setminus Y$ は、「左側の Y と組み合わせられたときに X となる統語範疇」を意味する。

また英語の他動詞は、

自分自身の右側に目的語となる名詞句が隣接した場合、全体として動詞句 ($S \setminus NP$; 上記参照) となるような統語範疇と定義できる。したがって「他動詞」は $(S \setminus NP) / NP$ と表現できる。ここで X / Y は、「右側の Y と組み合わせられたときに X となる統語範疇」を意味する。

順関数適用規則	$\frac{X/Y:f \quad Y:a}{X:fa} \rightarrow$
逆関数適用規則	$\frac{Y:a \quad X \setminus Y:f}{X:fa} <$
逆関数交差合成規則	$\frac{Y/Z:g \quad X \setminus Y:f}{X/Z:\lambda x.f(gx)} <Bx$

図1 本稿で用いる組合せ規則一覧

2.2 組合せ規則

組合せ範疇文法における「組合せ規則」は、統語範疇と意味表示の両者を同時並列的に合成する物として定義される。このため統語構造と意味表示の間には明確な対応関係が存在する。

組合せ範疇文法で用いられる組合せ規則の数は、通常の句構造文法などにおける文法規則に比べるとはるかに少ない。また、組合せ範疇文法における組合せ規則は、記述対象となる言語には依存せず、原理的にはすべての言語間において共通だとされている¹⁾。言語間に存在する文法上および意味上の差異は、基本的に語彙項目の設定で吸収する事になっている。

組合せ範疇文法で一般に使用される組合せ規則のうち、本稿に登場する規則の一覧を図1に示す。コロンの左側が統語範疇で、コロンの右側がラムダ項による意味表示である。

2.3 語彙項目設定の重要性

本稿の目標は、

統語的には様々に異なった構造を持っているが、人間が読んだ場合には同一の意味を持つと理解されるような複数の文に対して、最終的に同一の意味表示を合成する事

である。組合せ範疇文法は、統語構造と意味表示の間に明確な対応関係を持つので、この目標の達成度を評価するにふさわしいと言える。

文全体の最終的な意味表示は、

- 合成対象となる下位の構成素それぞれの意味表示
 - それら下位の構成素それぞれの統語範疇
 - 下位の構成素間に対して適用される組合せ規則
- という3者から機械的に合成されるわけであるが、

1) 対象言語によっては、使用できる規則の種類に制限が加わる事もある。

これらのうち3番目の組合せ規則は汎言語的に共通であるので、望ましい合成結果、すなわち文全体の望ましい意味表示、を得るためには、合成の材料となる各語彙項目の意味表示と統語範疇を慎重に設定する必要がある。

3 ドイツ語の文法的特徴

3.1 原因・結果表現

本稿で解析対象とする6種類の原因・結果表現を、図2に示す。他の自然言語と同様、ドイツ語においても「Aの結果としてBである」あるいは「Bの原因はAである」のような2文の関係は様々な統語構造で表現される。いずれの文も基本的な意味は同一であるが、それぞれの統語構造を細かく見ると、

- 原因を示す文は、結果を示す文に先行するか、それとも後続するか
- 原因を示す文に標識が付くか、結果を示す文に標識が付くか
- 原因あるいは結果を示す標識は、どのような品詞で表現されるか

などの点でバリエーションがある事が分かる。

3.2 接続詞 weil

従属接続詞 weil は原因を示し、従属複合文中で副文の文頭に立つ。このとき副文内の定動詞は文末に置かれる。また weil によって導入される副文が主文に先行する場合は、副文全体が一つの文肢 [5] を形成するので、後続する主文は定動詞が文頭となる(図2文1, 2)。

3.3 接続詞 denn

並列接続詞 denn は原因を示し、並列複合文中における後続文を導く。この後続文は平叙文なので、定動詞は平叙文内で第2位に位置する(図2文3)。

3.4 副詞 nämlich

接続的副詞 nämlich は原因を示し、並列複合文中で用いられる。通常であれば、文全体を修飾する副詞は文頭に立つ事も可能であるが、この用法としての nämlich が文頭に立つ事はない(図2文4)。

1. 従属複合文（結果文が先行）
Ich trinke Bier, weil ich Durst habe.
I drink beer because I thirst have

2. 従属複合文（原因文が先行）
Weil ich Durst habe, trinke ich Bier.
because I thirst have drink I beer

3. 並列複合文（原因文の文頭に並列接続詞）
Ich trinke Bier, denn ich habe Durst.
I drink beer because I have thirst

4. 並列複合文（原因文の文中に接続的副詞）
Ich trinke Bier, ich habe nämlich Durst.
I drink beer I have because thirst

5. 並列複合文（結果文の文頭に接続的副詞）
Ich habe Durst, deshalb trinke ich Bier.
I have thirst therefore drink I beer

6. 並列複合文（結果文の文中に接続的副詞）
Ich habe Durst, ich trinke deshalb Bier.
I have thirst I drink therefore beer

図2 「喉が渴いたのでビールを飲む」のドイツ語訳6種

3.5 副詞 deshalb

接続的副詞 deshalb は並列複合文中で用いられるが、上記の他の標識と異なり、原因ではなく結果を示す文に付く。また nämlich と違って、文頭に立つ事もできる（図2文5, 6）。

4 ドイツ語用組合せ範疇文法

4.1 基底範疇と統語素性

本稿で使用するドイツ語用基底範疇は名詞句 (NP) と文 (S) の2種類のみであるが、Hockenmaier [6] に準じた統語素性を用いてそれぞれの下位分類を設定している。

4.1.1 名詞句の統語素性

ドイツ語の平叙文においては、英語同様の「主格名詞句—一定動詞—対格名詞句」という語順に加えて、「対格名詞句—一定動詞—主格名詞句」や「副詞—一定動詞—主格名詞句—対格名詞句」などの語順も可能である。このため、名詞句が文中において果たす主格や対格といった統語的役割を、語順とは独立した形式で示す必要がある。

一般にドイツ語文中における名詞句の格は、冠詞や形容詞語尾などによって表現される事が多い。しかし本稿の範囲内では、主格名詞句としては人称代名詞の ich が、対格名詞句としては無冠詞・無変化の名詞 Durst と Bier が登場するのみである。そのため、名詞句 (NP) に関する統語素性としては、*n*

(nominative, 主格) と *a* (accusative, 対格) の2種類のみを使用する²⁾。

4.1.2 文の統語素性

Hockenmaier[6] では、「ドイツ語平叙文の基底語順は定動詞文頭であり、前域に位置する主語は主題化 (topicalization) によるものである」と解釈されている。またこれによって、主格名詞句、対格名詞句、与格名詞句それぞれの統語範疇を同一に保ったまま、前域要素の入れ替えが可能である事が示されている。

これは興味深い解釈であり、形式主語の es が前域以外では省略可能である理由なども、ここから説明できると思われる。

よって本稿でも [6] 同様、文 (S) の統語素性として、*dcl* (declarative, 定動詞第2位), *vI* (verb initial, 定動詞文頭), *vlast* (verb final, 定動詞文末), *emb* (embedded, 副文) の4種類を使用する。

4.2 語彙項目としてのコンマ

現代ドイツ語の正書法には、複合文内の各文をコンマで区切るという規則がある。本稿ではこの規則を利用して、複合文内の2文を結合するコンマに統語範疇と意味表示を割り当て、原因・結果関係を見通しよく表現する方策を取った。

なお、原因・結果を示す標識語の統語範疇を工夫すれば、コンマを利用しなくても原因・結果関係が表現可能である事は確認済みであるが、これについての詳細は他の機会に譲る。

4.3 標識の位置と意味表示

組合せ範疇文法においては、生成文法で言うところの空範疇は想定されない [1, 7]。また組合せ規則が適用されるのは、実際に音韻表示を持った上で、かつ表層上で直接隣接する統語範疇間のみである。

そのため図2中の文1, 2, 3, 5のように、原因・結果を示す標識 (weil, denn, deshalb など) が文頭に出現する場合には、当該文の意味役割 (原因あるいは結果) を容易に標示する事ができる³⁾が、図2中の文4における nämlich, および文6における deshalb のように、原因・結果を示す標識が文の

2) 文法的には、名詞句の格として他に属格および与格が存在するが、本稿には登場しない。

3) 原因・結果を示す標識が文頭ではなく文末に出現する場合 (日本語の「~だから」など) も、その標識を含む文の意味役割は容易に標示できる。

表 1 語彙項目一覧

音韻表示	統語範疇：意味表示
weil	$S[emb]/S[dcl] : \lambda x.\lambda z.because'(x, z)$
denn	$S[dcl]/S[dcl] : \lambda x.\lambda z.because'(x, z)$
nämlich	$(S\backslash NP)\backslash(S\backslash NP) : \lambda P.\lambda x.\lambda z.because'(Px, z)$
deshalb	$S[dcl]/S[v1] : \lambda x.\lambda z.because'(z, x)$
	$(S\backslash NP)\backslash(S\backslash NP) : \lambda P.\lambda x.\lambda z.because'(z, Px)$
コンマ (,)	$(S[dcl]\backslash S[dcl])/S[dcl] : \lambda P.\lambda x.Px$
	$(S[dcl]\backslash S[dcl])/S[emb] : \lambda P.\lambda x.Px$
	$(S[dcl]\backslash S[emb])/S[v1] : \lambda x.\lambda P.Px$
habe trinke	$(S[dcl]\backslash NP[n])/NP[a] : \lambda y.\lambda x.have'(x, y)$: $\lambda y.\lambda x.drink'(x, y)$
	$(S[v1]/NP[a])/NP[n] : \lambda x.\lambda y.have'(x, y)$: $\lambda x.\lambda y.drink'(x, y)$
	$(S[vlast]\backslash NP[n])\backslash NP[a] : \lambda y.\lambda x.have'(x, y)$: $\lambda y.\lambda x.drink'(x, y)$
ich	$NP[n] : I'$
Bier	$NP[a] : beer'$
Durst	$: thirst'$

中央部分に埋め込まれている場合は、その標識を含む文の意味役割の標識がやや難しくなる。本稿では、Steedman[1] および戸次 [3] における英語助動詞 *might* の意味表示を参考に、逆関数交差合成規則を用いてこの問題を解決した。

4.4 語彙項目の設定と意味表示の導出

以上を踏まえ、本稿では表 1 に示す語彙項目設定を提案する。複数の統語範疇が与えられている語彙項目に対しては、状況に応じて適切な統語範疇とそれに対応した意味表示を選択する事とする。その結果、図 3~図 5 に示したように、全 6 文から

$because'(have'(I',thirst'),drink'(I',beer'))$

という共通の意味表示を得る事ができるようになる。(スペースの都合上、6 文のうち 3 文の導出のみを掲載している。)

5 関連研究

Vancoppenolle[8] らは、株式市場領域の文生成を目的としたドイツ語文法を公表した。この文法は、組合せ範疇文法の拡張である「マルチモーダル組合せ範疇文法」を用いて記述されており、OpenCCG⁴⁾ の下で実装されている。OpenCCG は組合せ範疇文法の解析・生成システムであり、様相論理を拡張した Hybrid Logic Dependency Semantics [9] を意味表示として採用している。彼らはドイツ語を部分的に語順自由な言語ととらえ、型変換規則を用いる事でかき混ぜ文や主題化の扱いを工夫している。

4) <https://github.com/OpenCCG/openccg>

Hockenmaier[6] は、既存のドイツ語ツリーバンクである Tiger corpus⁵⁾ を、組合せ範疇文法版のツリーバンクに変換する手順を論じている。またその説明の過程で、ドイツ語用組合せ範疇文法に関する興味深い提案を行っている。ただし、統語範疇に関する記述のみで、意味表示に関しては述べられていない。かき混ぜ文に関しては、各語順に合わせてそれぞれに専用の統語範疇を用意する立場を取っている。

Evang ら [10] は、多言語対応組合せ範疇文法のアノテーションガイドラインと、アノテーション実施のためのツールを公開している。対象とされている言語は、英語、ドイツ語、イタリア語、オランダ語の 4 言語である。また参考のために、4×100 文の平行コーパスと、10K 文の検証済み単言語コーパス、および 82K 文の部分的検証済み単言語コーパスが提供されている。意味表示に関しては述べられていない。

6 おわりに

適切な統語範疇と意味表示を持った語彙項目を設定する事で、6 通りの異なった統語構造を持ったドイツ語の原因・結果文すべてから同一の意味表示を合成する事が可能となった。人間がこの 6 文を読んだ場合も、脳内にはほぼ同一の原因・結果関係が生成されると考えられる。

人間による文理解において興味深いのは、文全体を読み終えてから最適解の探索を開始するのではなく、文頭から順に語を読み進めるに沿って漸進的に解釈を行っていると感じられる点である。

一般の句構造文法では、主語一動詞一目的語という語順の構造を(主語(動詞 目的語))のように解析するので、文頭からの漸進的な解釈のモデル化が難しいが、組合せ範疇文法では型繰り上げ規則と関数合成規則を利用する事で、((主語 動詞)目的語)という解析が可能である⁶⁾。その意味で組合せ範疇文法は漸進的解釈のモデル化に向いている。

また、日本語やドイツ語のように(動詞以外の)語順が比較的自由な言語においては、異なる入力語順から同一の意味的表現が合成される場合がある。これも広い意味では、異なる統語構造から同一の意味表示を求める問題と言える。今後はこれらの点に注目して研究を進めていきたい。

5) <https://www.ims.uni-stuttgart.de/en/research/resources/corpora/tiger/>

6) 常に漸進的解析が可能になるわけではない [11]。

謝辞

本研究は JSPS 科研費 JP22K12188 の助成を受けたものです。

参考文献

- [1] Mark Steedman. **The Syntactic Process**. The MIT Press, 2000.
- [2] Mark Steedman. **Surface Structure and Interpretation**. The MIT Press, 1996.
- [3] 戸次大介. 日本語文法の形式理論. くろしお出版, 2010.
- [4] 峯島宏次. 論理と文法. 数学セミナー, Vol. 59, No. 3, pp. 30–35, 2020.
- [5] 在間進. [改訂版] 詳解ドイツ語文法. 大修館書店, 2006.
- [6] Julia Hockenmaier. Creating a CCGbank and a wide-coverage CCG lexicon for German. In **Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics**, pp. 505–512, Sydney, Australia, July 2006. Association for Computational Linguistics.
- [7] M. Steedman. Grammar, interpretation, and processing from the lexicon. In W. Marslen-Wilson, editor, **Lexical representation and process**, pp. 463–504. The MIT Press, Cambridge, MA, 1989.
- [8] Jean Vancoppenolle, Eric Tabbert, Gerlof Bouma, and Manfred Stede. A german grammar for generation in openccg. In **H. Hedeland, T. Schmidt, K. Wörner (eds.): Multilingual Resources and Multilingual Applications. Proc. of the Conference of the German Society for Computational Linguistics and Language Technology (GSCL), Hamburg, 2011. Working Papers in Multilingualism, Series B**, No. 96, pp. 145–150, 2011.
- [9] Jason Baldrige and Geert-Jan Kruijff. Coupling CCG and hybrid logic dependency semantics. In **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 319–326, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.
- [10] Kilian Evang, Lasha Abzianidze, and Johan Bos. CCG-web: a new annotation tool and a first quadrilingual CCG treebank. In **Proceedings of the 13th Linguistic Annotation Workshop**, pp. 37–42, Florence, Italy, August 2019. Association for Computational Linguistics.
- [11] Vera Demberg. Incremental derivations in CCG. In **Proceedings of the 11th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+11)**, pp. 198–206, Paris, France, September 2012.

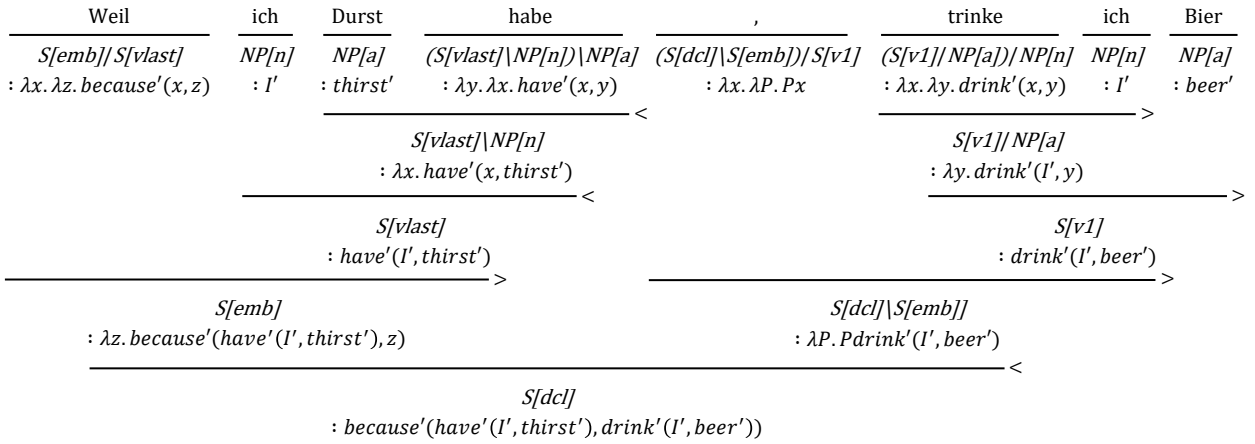


図3 “Weil ich Durst habe, trinke ich Bier.”の導出

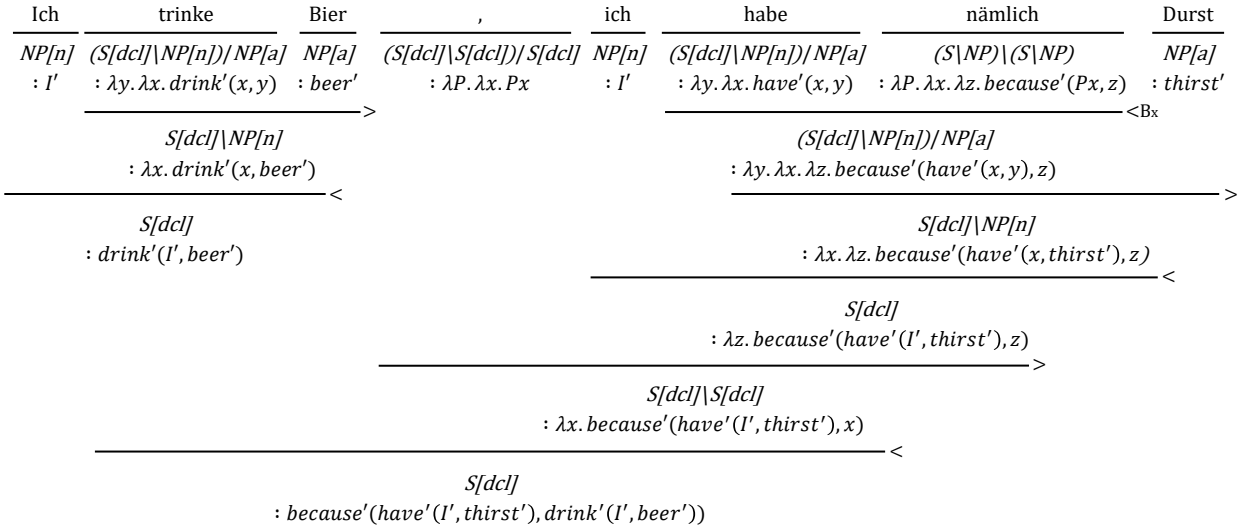


図4 “Ich trinke Bier, ich habe nämlich Durst.”の導出

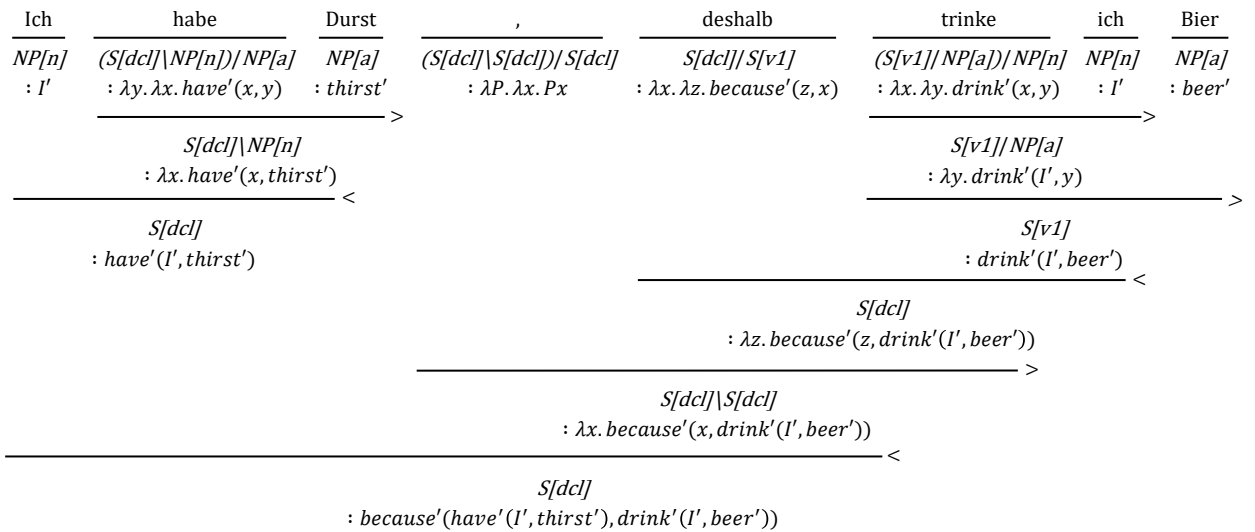


図5 “Ich habe Durst, deshalb trinke ich Bier.”の導出