日本語単一化文法における並列構造の解析

内野皓介[1]　田辺利文[1]　乙武北斗[1]　吉村賢治[1]

概要：日本語の単一化文法を用いた構文解析における並列構造の処理として，句の比較に基づいた並列構造の推定方法と，並列構造を含む文を単文に分解する方法の提案およびこれを実装した構文解析器の作成について報告する．

キーワード：構文解析，並列構造，単一化文法

Analysis of Coordinate Structure based on Japanese Unification Grammar

KOSUKE UCHINO[1]　TOSHIFUMI TANABE[1]
HOKUTO OTOTAKE[1]　KENJI YOSHIMURA[1]

Abstract: We propose a method of estimation based on phrase comparison and a method of decomposing a sentence containing coordinate structures into single sentences in parsing using a unification grammar of Japanese, and report on the development of a parser that implements these methods.

Keywords: Syntactic Analysis, Coordinate Structure, Unification Grammar

1. はじめに

並列構造は，１文中に同等の機能を持つ複数の単語や句を並べたものである．例えば,『太郎が水を飲む』と『太郎がご飯を食べる』という２つの文を『太郎が水を飲み，ご飯を食べる』という１つの文にまとめる際に使われ，文章の冗長性を無くし，簡潔に記述することに役立っている．一方，並列構造は並列の範囲の曖昧性や並列構造の入れ子を許容する性質があるため，自動翻訳システムや対話システムなどに使用されている構文解析を困難にする要因の一つとなっている．

日本語の構文解析における並列構造の処理に関する先行研究として，京都大学の黒橋らの研究[1][2]がある．この研究における並列構造の推定方法は，まず入力文を形態素に分け，それらを文節(自立語とそれに続く付属語)にまとめ，その文節同士を比較し，独自に定義された類似度の基準により類似度スコアを計算する．例えば,「低水準言語+,」と「高水準言語+と」の類似スコアは品詞の一致と「〜水準言語」の文字列の一致により高いスコアが得られる．このような文節同士の比較をダイナミックプログラミングの手法により，全ての文節に対して検証し，最も類似した２つの文節列を並列構造の要素と推定する．この手法は，様々な並列構造を含む文に対して，高い精度が得られている．しかし,『ユーザがタンクに水，フィルタに新鮮な豆を入れる』のような，並列構造内に連体修飾語(「新鮮な」や「粉末状の」など)を伴う名詞が存在する文の場合，「タンクに水」と「フィルタに新鮮な豆を」の類似度よりも,「水」と「フィルタ」の類似度の方が高くなり，誤った推定をしてしまう．この原因としては，名詞「水」と名詞句「新鮮な豆を」のような句にまとまったもの同士の比較が行えていないことが考えられる．このような「新鮮な」と「豆」が１つの名詞句「新鮮な豆」にまとめることができるという情報は，句構造文法に基づく構文解析の結果得られる解析表から求めることが可能である．

そこで，本研究では，並列構造の推定に文節の比較を用いるのではなく，名詞「水」と名詞句「新鮮な豆」のような句の比較を用いる方法を提案する．これにより，先行研究で対応できていない並列構造内に連体修飾語を伴う名詞が存在する文への対応が可能になる．

また，翻訳や要約，ソフトウェア開発文書の解析などの応用システムで並列構造を含む文の解析結果を利用する場合，単文に分解しておいたほうが扱いやすい．そこで，本研究では文中の並列構造を分解し，並列構造を構成する複数の単文を生成する処理について提案する．

2. 並列構造

並列構造は，その構造から代表的な 5 種類に分類でき，それを構成するための規則から生じる構造上の特性が存在する．この章では，その並列構造の特徴と種類について述べる．なお，本稿では日本語文の形態素解析を通常の学校文法ではなく，音韻論的文法に基づいて行っている[3]．音韻論的文法では，日本語の用言は活用せず，学校文法における用言の活用変化を音韻変化で説明する.「行く」のよう

1 福岡大学
Fukuoka University

に学校文法における 5 段活用の動詞は末尾が子音である子音動詞（ik)，「入れる」のように 1 段活用の動詞は末尾が母音である母音動詞（ire）として分類される．基本的な音韻変化の規則には次の二つがある．

1. 子音脱落
   二つの形態素が連接したときに子音が連続すると文末側の子音が消える．
2. 母音脱落
   二つの形態素が連接したときに母音が連続すると文末側の母音が消える．

例えば，「行く」は子音動詞「行 k」と非過去の接辞「ru」，「入れる」は母音動詞「入 re」と非過去の接辞「ru」に分解される．(r)は子音脱落の規則に従って r が消えることを表している．

行く → 行 k + (r)u

入れる → 入 re + ru

また，丁寧体の「行きます」,「見ます」は丁寧の接辞「imas」と非過去の接辞「ru」を使って次のように分解される．ここで(i)は母音脱落の規則に従って i が消えることを表している．

行きます → 行 k + imas + (r)u

見ます → 見(mi) +(i)mas + (r)u

「行った」は子音動詞「行 k」に過去の接辞「ita」が連接したものであるが，このような音便形に対しては特別な規則が働く．

行った → 行(k) + (i:t)ta

ここで，(k)は k が消えることを，(i:t)は i が t に変化することを表している．

2.1 並列構造の特徴

並列構造の特徴として主に以下の 4 つがある．

1. 句を接続する働きを持つ文字列(並列キー)が存在する
   例文では<<と>>で囲まれた文字列が並列キーである．
   ex.『太郎が紅茶に 砂糖 <<と>> ミルク を入 re ru』
2. 並列キーの前と後に同じ形式の文要素(並列要素)が存在する
   例文では{と}で囲まれた文字列が並列要素である．
   ex.『太郎が紅茶に {砂糖} と {ミルク} を入 re ru』
3. 並列要素の単語数は必ずしも同一ではない
   ex.『太郎が{水/を/飲 m} i,{自転車/で/学校/に/行 k} ru』
   例文における / は単語境界である．
4. 文から並列キーと片方の並列要素を取り除いても意味が通る
   ex.『太郎が {水を飲 m} ru』
   ex.『太郎が {自転車で学校に行 k} ru』

2.2 並列構造の種類

並列構造の種類としては基本的な 3 種と応用的な 2 種の計 5 種類がある[4].

基本的な並列構造(3 種)

A) 名詞並列 ：名詞や名詞句の並列
   ex.『太郎が紅茶に {砂糖} と {ミルク} を入 re ru』
B) 述語並列 ：連用修飾句や文法接辞を共有する述語の並列
   ex.『太郎が {お菓子を食 be} ,{コーヒーを飲 m } ru』
   ：連用修飾語を共有しない述語の並列
   ex.『{太郎がお菓子を食 be} , {次郎がコーヒーを飲 m} ru』
C) 部分並列 ：複数の連用修飾句や名詞･名詞句からなり，文法接辞を共有する並列
   ex.『ユーザが {タンクに水}, {フィルタに豆} を入 re ru』

応用的な並列構造(2 種)

D) 入れ子並列 ：並列構造の要素内に並列構造を含む並列
   ex.『太郎が{ (コーヒーに砂糖), (紅茶にミルク)を入 re}, {お菓子を食 be} ru』
E) 連接並列 ：３つ以上の並列構造が連接する並列
   ex.『太郎が{コーヒーに砂糖}, {紅茶にミルク}, {水に氷} を入 re ru』

これらの構造の中で A の名詞並列については，単文の処理を行う句構造解析の中で処理が可能である．また，D の入れ子並列については，出現することがあまりないので，本稿の並列処理では B,C,E を対象とする．

3. 構文解析

本研究の構文解析は，単一化文法を用いたチャート法により構成する．この章では，その単一化文法とチャート法の実装について述べる．

3.1 単一化文法

単一化文法は，素性構造間の単一化によって言語の文法的制約を表す文法である．素性構造は文脈自由文法における非終端記号を単なる記号ではなく，その単語がもつ文法的な属性や意味を素性と素性値の対を要素とする集合で表す．

本研究では，代表的な日本語の単一化文法である日本語句構造文法(JPSG：Japanese Phrase Structure Grammar) [5][6]に基づいて，その原理と素性構造を拡張し，日本語文の解析を形態素解析と構文解析の２段階に区別せずに行う事ができる単一化文法の構築を目指す．JPSG は(1)に示す１つの書き換え規則と素性構造の単一化に関する原理で構成される．

(1) M→ C H

ここで，M,C,H はそれぞれ素性構造で M を親，C,H を子と呼ぶ．特に，H は日本語における右側主要部の規則から

主辞(head)と呼び，左側の子 C と区別する．C と H は次の 3 種類の関係にあるときに結合し，原理に従ってそれぞれの素性構造から M の素性構造が計算される．

1. 補語構造
   C が H の補語になる関係にある
   ex. C:『学校に』と H:『行 k』
   ex. C:『学校』と H:『に』
2. 付加構造
   C が H を付加的に修飾する関係にある
   ex. C:『新鮮な』と H:『豆』
   ex. C:『自転車で』と H: 『行 k』
3. 等位構造
   C と H が述語並列のような並列の関係にある
   ex. C:『お菓子を食 be』と H:『コーヒーを飲 m』

なお，等位構造は並列処理を別途行うためここでは使用しない．

3.2 素性

素性には，表 1 に示すようなものがある．pos や pform, gr は複数個の中の一つを値とする多値素性である．この 3 つの素性は主辞素性であり，head 素性の値である素性構造を構成する要素となる．sem は述語論理式や素性構造など，何らかの形式で表された意味表現を保持する素性である．subcat と adjacent は主辞が必要とする補語の集合を表現している下位範疇化素性と呼ばれる素性である．adjacent は C と H が隣接することを要求する．adjunct の値はその範疇が修飾する主辞の素性構造である．

表 1 素性名と値

素性名	値	備考
pos	v,n,p,…	品詞
pform	ga,wo,ni,…	格助詞
gr	sbj,obj	述語との関係
sem	任意の形式的表現	意味表現
subcat	素性構造の集合	下位範疇化
adjacent	素性構造	隣接を要求
adjunct	素性構造	修飾先の主辞

JPSG の代表的な原理に主辞素性の原理と下位範疇化素性の原理がある．

主辞素性の原理(head 素性)

親 M の主辞素性の値は主辞 H の主辞素性の値と単一化する．

下位範疇化素性の原理(subcat 素性,adjacent 素性)

① 補語素性の場合，親 M の下位範疇化素性の値は，主辞 H の下位範疇化素性の値から子 C(補語)と単一化可能な素性構造を取り除いたものと単一化する．

② 付加構造の場合，親 M の下位範疇化素性の値は，主辞 H の下位範疇化素性の値と単一化する．

③ 等位構造の場合，親 M の下位範疇化素性の値は，子 C と主辞 H の下位範疇化の値と単一化する．

基本的な日本語文の記述に必要な原理としては，その他に以下のようなものがある．

意味素性の原理(sem 素性)

① 補語構造の場合，親 M の意味素性の値は，主辞 H の意味素性の値と単一化する．

② 付加構造の場合，親 M の意味素性の値は，付加語 C の意味素性の値と単一化する．

付加素性の原理(adjunct 素性)

親 M の非局所素性の値は子 C の付加素性の値から主辞 H を取り除いた値と単一化する．

次に，例として名詞「犬」と格助詞「が」，動詞「吠 e(吠える)」，接辞「ita(た)」の素性構造の単一化により，『犬が吠える』という文を形成する過程を示す．まず，名詞「犬」と格助詞「が」の素性構造において単一化を行う．図 1 のように格助詞「が」の adjacent 素性値の要素と名詞「犬」の素性構造が単一化できるので，下位範疇化素性・主辞素性・意味素性の原理に従って後置詞句「犬が」の素性構造が得られる．

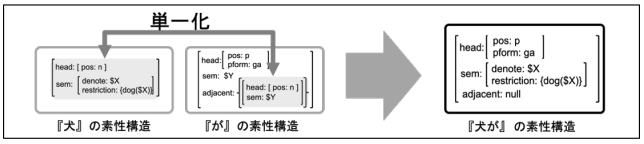

図 1 素性構造「犬」と「が」の単一化

次に，今形成された後置詞句「犬が」と動詞「吠 e(吠える)」の素性構造において単一化を行う．すると，図 2 のように動詞「吠 e」の subcat 素性値の要素と後置詞句「犬が」の素性構造が単一化できるので，下位範疇化素性・主辞素性・意味素性の原理に従って，動詞句「犬が吠 e」の素性構造が得られる．

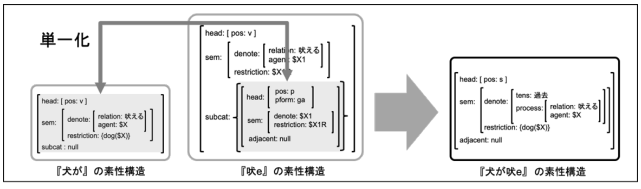

図 2 素性構造「犬が」と「吠 e」の単一化

最後に，動詞句「犬が吠 e」と接尾辞「ita」の素性構造に

おいて単一化を行う．図 3 のように接辞「ita」の adjacent 素性値の要素と動詞句「犬が吠 e」の素性構造が単一化し，下位範疇化素性の原理と主辞素性の原理，意味素性の原理に従って，文「犬が吠えた」の素性構造が得られる．

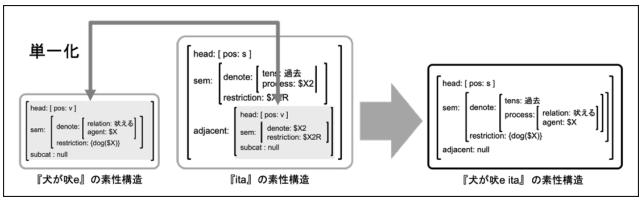

図 3 素性構造「犬が吠 e」と「ita」の単一化

3.3 チャート法

チャート法とは文脈自由文法による構文解析手法の 1 つである．チャートとは，項（term）と呼ばれるデータ構造〈i, j, C → α・β〉の集合である．ここで，i と j は単語の境界位置を表す番号でありこれを節点（vertex）と呼ぶ．C は非終端記号，α と β は終端記号と非終端記号とからなる記号列であり文脈自由文法における書換え規則 C →α β に基づいている．項は構文解析中に得られる部分解析木を意味しており，〈i, j, C → α・β〉は，概略，解析対象の文の単語列 $w_1w_2…w_n$ において，書き換え規則 C →α β における α から単語列 $w_{i+1}…w_j$ が導出されることを意味している．項は C → α・β をラベルとしてもつ弧（edge）として表現でき，チャートをグラフとしてあらわすことができる（図 4）．

項には図 4 に示す〈0, 1, PP → NP・P〉や〈0, 0, S → ・PP VP〉のように不完全な部分解析木に対応するものと〈0, 1, NP → N・〉や〈0, 4, S → PP VP・〉のように完全な部分解析木に対応するものがある．それぞれの項に対応する弧を「不活性弧」,「活性弧」と呼び活性弧は－－➔，不活性弧は➔ で表している．

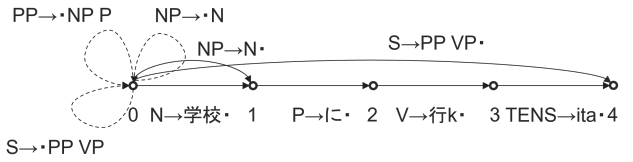

図 4 チャートのグラフ表現（一部）

チャート法による構文解析は，2 つの項を結合することで，より長い単語列の導出が可能であることを意味する項を新たにチャートに追加しながら，単語列全体の導出が可能であることを意味する項がチャート内にできれば構文解析が成功したものとする．例えば，解析対象文に含まれる単語数が n のとき〈0, n, S → α・〉がチャートに含まれていれば構文解析成功となる．チャート法では，ボトムアップ法，トップダウン法のいずれのアプローチでも解析が可能であるが，いずれにおいても 2 つの項を結合する，いわゆる基本規則が必要である．またボトムアップ法，トップダウン法ではそれぞれが必要とする規則があり，以下，基本規則を含めそれぞれの規則について簡単に説明する．

基本規則

チャートに項〈i, j, A → α・B β〉と項〈j, k, B → γ・〉が存在するならば，項〈i, k, A → α B・β〉を作成してチャートに追加する．α，β，γ は空列のこともある．

チャート法ではこの基本規則とボトムアップの規則かトップダウンの規則の適用を繰り返す．

ボトムアップの規則

チャートに項〈i, j, B → β・〉を追加するときには，A → B α の形式を持すべての書き換え規則について，項〈i, i, A → ・B α〉を作成してチャートに追加する．ここで α は空列のこともある．

トップダウンの規則

チャートに項〈i, j, A → α・B β〉を追加するときには，B → γ の形式をもつすべての書き換え規則について，項〈j, j, B → ・γ〉を作成してチャートに追加する．

チャートを用いたボトムアップの構文解析アルゴリズム

本研究では項のデータ構造を
〈i, j, FS, P, Clist, Hlist〉
とした．ここで，i と j は単語の境界位置につけた番号，P は句の品詞，FS は句の素性構造である．Clist は FS における subcat 素性値または adjacent 素性値のどちらかの要素の pos 素性値のリストである．Clist が空(φ)でないとき FS は書き換え規則（P→CH）の H になる可能性があり，Clist の要素は C になり得る素性構造の pos 素性値を示している．また，Hlist は FS における adjunct 素性値の要素の pos 素性値のリストである．Hlist が空(φ)でないとき FS は書き換え規則（P→CH）の C になる可能性があり，Hlist の要素は H になり得る素性構造の pos 素性値を示している．Clist と Hlist の値は必要なときに素性構造 FS から取り出すことができるが，構文解析の効率を上げるために，項を作成する時点で取り出しておく．解析で用いる日本語単一化文法の書き換え規則（P→CH）はチョムスキー標準形になっており，標準のチャート法において C → α・β の中で「・」を用いて表している情報は Clist と Hlist で表現している．

また，アルゴリズムにはチャートを用いたボトムアップの構文解析アルゴリズムを利用した．今回のアルゴリズムにおいて，agenda と chart は項を格納するためのリストであり，n は入力文の単語数であり，文末の節点の番号にな

る．

step.1　chart と agenda を空（φ）にする．
step.2　0 ≦ i ＜ n である整数 i について節点 i と i+1 の間にある単語 w のすべての範疇 A に対して，項〈i, i+1, w の FS, P, Clist, Hlist〉を agenda の末尾に追加する．
step.3　agenda = φ ならば step.7 に行く．
step.4　agenda の先頭の項を取り出す．この項を　E =〈i, j, fs, P, Clist, Hlist〉とする．
step.5　E と同じ項が chart に存在しない場合
　step5-1.　E を chart に追加する．
　step5-2.　chart に E と隣接していて E の fs と単一化可能な fs を持っている項 F〈j, k, FS, P, Clist, Hlist〉が存在するならば項 U〈i, k, FS, P, Clist, Hlist〉を作成して agenda の末尾に追加する(FS, P, Clist,Hlist はEとFを単一化したものを使用).
step.6　step.3 に行く．
step.7　chart に項〈0, n, FS, s, φ, φ〉が存在するならば終了する．
step.8　chart 内で i, k 間の長さが最も長く Clist が φ でない項 X を探す．
step.9　項〈i, i, FS, P, Clist, Hlist〉を作成して agenda の末尾に追加する（i は項 X の i と同値，FS は空所を示す単語 ε のものを使用）．
step.10　step3.に行く

このアルゴリズムにおいて, step.8 と step.9 は省略に対処するために設けたゼロ代名詞（ε）の処理である．単文の解析では，まず省略がないものとして解析を行い，解析が失敗したときに省略の処理に入る．本稿では，ゼロ代名詞の処理の詳細については割愛する．

4. 提案手法

3 章のアルゴリズムは並列構造を含まない単文の構文解析を行うアルゴリズムである．このアルゴリズムで解析に失敗した場合，文中に並列構造があるものとみなして並列構造の推定と単文への分解を行う．

4 章では並列構造の推定と単文への分解を行う手法について述べる．並列構造の推定は，失敗した構文解析の結果から得られる句情報の比較によって行う．また，並列構造の推定結果を用いて並列構造を含む文から並列構造を構成する単文を生成する(並列構造の分解).

4.1 並列構造の推定

概要

並列構造の特性により，並列構造には並列キーとなる文字列が存在し，並列構造の範囲を決定することができる．その範囲において類似性のある並列要素の組を句の比較により求め，並列構造を推定する．

定義

1. 形態素列が構成可能な部分木の根のことを句と呼び, その形態素列の句から得られる構文情報 (素性構造) のことを句の情報と呼ぶ. 形態素列が構成可能な部分木は複数現れることもある．
2. 句の類似性の有無を判定する際に使用する句の情報は, その句の品詞情報 (pos) とその句が必要とする補語 (subcat, adjacent) の数，助詞の場合，その助詞の型の情報 (pform) を持つ．

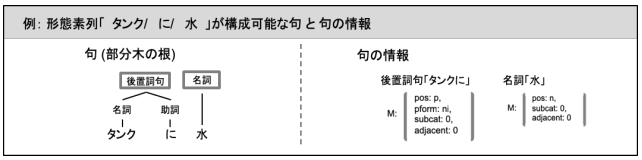

図 5　句の情報

並列キーの位置の特定

並列キーは「と」や「,」などの語や記号であることがわかっており個数も限定されるため，並列キーは網羅的に登録できる．形態素解析後の単語列中において並列キーが存在する場合にその位置を検出する．

検討すべき並列要素の組み合わせ

並列構造を構成する個々の並列要素は，並列キーの前方と後方にそれぞれ一つずつ存在することがわかっている．並列キーが解析対象文の i 番目の単語(w_i)であるとする．ここで $\mathbf{W}_{i,j}$ を，w_i から w_j までの単語列，n を単語数であるとすると，並列キーの前方に位置する並列要素候補として $\mathbf{W}_{i-1,i-1}$,$\mathbf{W}_{i-2,i-1}$, … ,$\mathbf{W}_{1,i-1}$ の i-1 通り, 同様に並列キーの後方での候補は $\mathbf{W}_{i+1,i+1}$, $\mathbf{W}_{i+1,i+2}$, … , $\mathbf{W}_{i+1,n}$ の n-i 通りとなる．例文「ユーザがタンクに水，フィルタに豆を入 re ru」では図 6 のように，並列キーの前方の並列要素は[ユーザがタンクに水]〜[水]の 5 通り, 並列キーの後方の並列要素は[フィルタに豆を入 re ru]〜[フィルタ]の 6 通りである．

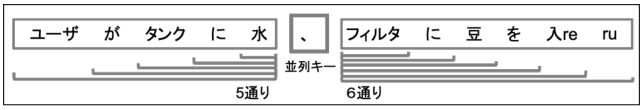

図 6 検討すべき並列要素

類似判定

並列構造を構成する個々の並列要素は，共に同じ働きを持つ句で構成されており，個々の並列要素の間には類似性があると考える．そこで，検討すべき並列要素のそれぞれの組み合わせにおいて，類似判定を行い，類似性があれば並列構造であると判定し，類似性がなければ並列構造では

ないと判定する．ここでの類似性の有無の判定は，構文解析から得られる句の情報の比較により行う．構文解析から得られる情報を簡易的に示すと図 7 のような構文木となる．

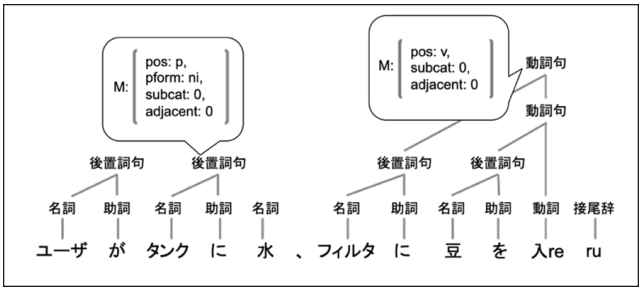

図 7 構文解析によって得られる構文木

ここで，組み合わせにおける前方の要素候補(単語列)を $\mathbf{W}_a$，後方の要素候補を $\mathbf{W}_b$ とする．並列要素の判定は次の通りとする．

1. 構文解析中の部分木より，$\mathbf{W}_a$ を導出する句の素性構造 FSa を抽出する．句が複数の場合には FSa は素性構造のリストとなる．$\mathbf{W}_b$ に関しても同様に行い FSb とする．
2. 1.で得られた FSa と FSb に対し，対応する位置での素性構造がすべて単一化可能であればこの組み合わせを並列構造とみなし，$\mathbf{W}_a$ と $\mathbf{W}_b$ を並列要素とする．

例えば，前方の並列要素が「タンクに水」で，後方の並列要素が「フィルタに豆を」の組み合わせについて類似判定を行う (図 8 参照)．まず，前方の並列要素「タンクに水」は構文解析の情報から，後置詞句「タンクに」と名詞「水」の２つの句の情報が得られる．次に，後方の並列要素「フィルタに豆を」は構文解析の情報から，後置詞句「フィルタに」と後置詞句「豆を」の２つの句の情報が得られる．そして，前方と後方の並列要素の句情報が単一化できるか検証する．この場合，後置詞句「タンクに」と後置詞句「フィルタに」は単一化が可能であるが，名詞「水」と後置詞句「豆を」は単一化できない．よって「タンクに水」と「フィルタに豆を」の組み合わせに類似性はなく，並列構造ではないと判定される．

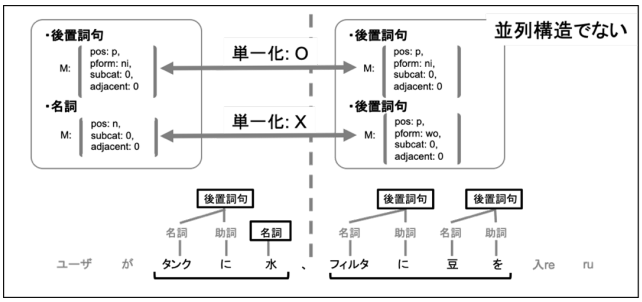

図 8 並列構造の推定_例 1

次に，前方の並列要素が「タンクに水」で，後方の並列要素が「フィルタに豆」の組み合わせについて類似判定を行う(図 9 参照)．まず，前方の並列要素「タンクに水」は構文解析の情報から，後置詞句「タンクに」と名詞「水」の２つの句の情報が得られる．次に，後方の並列要素「フィルタに豆」は構文解析の情報から，後置詞句「フィルタに」と名詞「豆」の２つの句の情報が得られる．そして，前方と後方の並列要素の句の情報が単一化できるかを検証する．この場合，後置詞句「タンクに」と後置詞句「フィルタに」は単一化可能であり，名詞「水」と名詞「豆」も単一化可能である．よって，全ての句情報において単一化可能であったため，類似性ありと判定し，「タンクに水」と「フィルタに豆」の組み合わせは並列構造であると推定する．

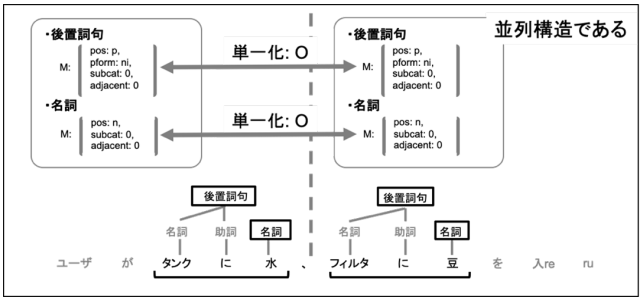

図 9 並列構造の推定_例 2

4.2 並列構造を含む文の単文化

概要

並列構造の特性の 1 つとして，元の文から並列キーと片方の並列要素を取り除いても意味が通ることがわかっている．この特性と並列構造の推定結果に基づき，文中の並列構造内の並列要素を 1 つずつ抽出・加工することで，元の文を 2 つの単文に言い換えることを考える．

定義

1. 文中の並列キーを K，並列構造における並列要素を，文前方から A，B とする．
2. 生成された単文のリストを result とする．

単文の生成

1. 文 X 中の並列キーK と並列要素 B の形態素を全て取り除いたものを文 X’として作成する.
2. 文 Y 中の並列キーK と並列要素 A の形態素を全て取り除いたものを文 Y’として作成する.

例えば,「ユーザがタンクに水，フィルタに豆を入 re ru」のような，並列キーK が「,」で， 並列要素 A が「タンクに水」，並列要素 B が「フィルタに豆」の文における文 X と文 Y から, 図 10 のような文 X’と文 Y’が作成される.

文X:「ユーザがタンクに水、~~フィルタに豆~~を入re ru」	→	文X':「ユーザがタンクに水を入re ru」
文Y:「ユーザが~~タンクに水、~~フィルタに豆を入re ru」	→	文Y':「ユーザがフィルタに豆を入re ru」

図 10 単文化の例

単文リスト result への単文の追加

並列構造を分解して得られた文 X に対して構文解析を行い成功した場合に文 X を result リストの末尾に追加する. ただし，文 X 中に並列キーが存在する，つまり単文中に並列構造が存在すると推測される場合，および result リストに既に同じものが存在する場合には文 X は result リストには追加しない.

並列構造を３つ以上含む文の取り扱い

並列構造を３つ以上含む文を解析する場合，並列構造の除去が完了していない単文が生成されるため，その文に対して並列構造の除去を行う必要がある. その方法としては，生成された単文に対して再度並列構造の推定と並列構造の分解を行う.

例文「太郎がコーヒーにミルク，紅茶に砂糖，水に氷を入 re ru」のような２つの並列キーを含む文の場合， １つ目の並列キー「,」について並列構造の推定と並列構造の分解を行うと図 11 のような結果が得られる.

解析対象：「太郎がコーヒーにミルク、紅茶に砂糖、水に氷を入re ru」
並列構造の推定：「コーヒーにミルク」、「紅茶に砂糖」
並列構造の分解：文X 「太郎がコーヒーにミルク、水に氷を入re ru 」 文Y 「太郎が紅茶に砂糖、水に氷を入re ru 」

図 11 並列構造の分解の例_途中経過

文 X，Y には並列構造が含まれているため再度文中の並列キー「,」に対して，並列構造の推定と並列構造の分解を行う．並列構造の推定により，文 X の並列構造は「コーヒーにミルク」と「水に氷」であると推定される．文 Y の並列構造は「紅茶に砂糖」と「水に氷」であると推定される. そして，並列構造の分解を行う．それぞれ文 X と Y から図 12 のような結果が得られる.

解析対象：「太郎がコーヒーにミルク、水に氷を入re ru」
並列構造の推定：「コーヒーにミルク」、「水に氷」
並列構造の分解：文X'「太郎がコーヒーにミルクを入re ru」 文Y'「太郎が水に氷を入re ru」
解析対象：「太郎が紅茶に砂糖、水に氷を入re ru」
並列構造の推定：「紅茶に砂糖」、「水に氷」
並列構造の分解：文X''「太郎が紅茶に砂糖を入re ru」 文Y''「太郎が水に氷を入re ru」

図 12 並列構造の分解の結果

5. 実験

本研究では，単一化文法を用いた構文解析器 UGP(Unification Grammar Parser)を作成し，本稿で提案した手法である並列構造の推定と並列構造の分解による並列構造の処理を実装した．この章では，句の比較に基づいた並列構造の推定が有効であるか，並列構造を含む文から並列構造を構成する単文を生成できるか，３種類の並列構造について例文を用意し，実験を行った結果を示す.

1. 述語並列

主語を共有する述語並列を含む 1-1 文と主語を共有しない述語並列を共有する 1-2 文について実験を行った(図 13). その結果より，1-1 文と 1-2 文において，並列構造の推定と並列構造の処理が期待通り動作することが確認できた.

解析対象：1-1文「太郎がお菓子を食be、コーヒーを飲m ru」
並列構造の推定：「お菓子を食be」,「コーヒーを飲m」
出力： ********** 解析成功 ['太郎', 'が', 'お菓子', 'を', '食be', 'ru'] ['太郎', 'が', 'コーヒー', 'を', '飲m', 'ru'] **********
解析対象：1-2文「太郎がお菓子を食be、次郎がコーヒーを飲m ru」
並列構造の推定：「太郎がお菓子を食be」,「次郎がコーヒーを飲m」
出力： ********** 解析成功 ['太郎', 'が', 'お菓子', 'を', '食be', 'ru'] ['次郎', 'が', 'コーヒー', 'を', '飲m', 'ru'] **********

図 13 実験_述語並列を含む文

2. 部分並列

並列構造内に連体修飾語を伴う名詞が存在しない 2-1 文と並列構造内に連体修飾語(形容動詞)が存在する 2-2 文について実験を行った(図 14). その結果より，2-1 文と 2-2 文において，並列構造の推定と並列構造の処理が期待通り動作することが確認できた.

解析対象：2-1文「太郎がタンクに水、フィルタに豆を入re ru」
並列構造の推定：「タンクに水」,「フィルタに豆」
出力： ********** 解析成功 ['太郎', 'が', 'タンク', 'に', '水', 'を', '入re', 'ru'] ['太郎', 'が', 'フィルタ', 'に', '豆', 'を', '入re', 'ru'] **********
解析対象：2-2文「太郎がタンクに水、フィルタに新鮮な豆を入re ru」
並列構造の推定：「タンクに水」,「フィルタに新鮮な豆」
出力： ********** 解析成功 ['太郎', 'が', 'タンク', 'に', '水', 'を', '入re', 'ru'] ['太郎', 'が', 'フィルタ', 'に', '新鮮な', '豆', 'を', '入re', 'ru'] **********

図 14 実験_部分並列を含む文

3. 連接並列

名詞並列が３つ並んだ連接並列を含む 3-1 文について実験を行った(図 15). その結果より，並列構造の推定と並列構造の処理が期待通り動作することが確認できた.

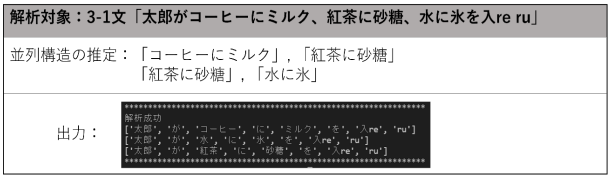

図 15 実験_連接並列を含む文

6. おわりに

本論文では，日本語句構造文法(JPSG)に基づく単一化文法を用いた構文解析における並列構造の処理として，句の比較に基づいた並列構造の推定方法と，並列構造を含む文を単文に分解する方法を提案した．そして，本稿で提案する並列構造の処理を実装した単一化文法を用いた構文解析器 UGP(Unification Grammar Parser)を作成し，3 種類の並列構造に対して例文による実験を行った．その結果，並列構造の各種類において適切な出力が得られることが確認できた．このことから，構文解析の結果得られる句の情報(pos：品詞情報, subcat, adjacent：下位範疇化素性の数, pform：助詞の型)を並列構造の推定に用いることは有効であることがわかった.

本研究で作成した構文解析器は，素性構造辞書の登録単語数が少ないこともあり，実際に出現する大量の文を用いた実験を行なっていない．また，本研究では入れ子並列は文中に出現する頻度が低いと考えたため実験の対象とはしなかった．よって，今後の課題としては，素性構造辞書の増強を図り，一般的な文を用いた定量的な実験を行うなど検証を重ね，その結果を基に句の比較を行う際に必要な情報を決定すること及び入れ子並列構造の推定と分解の機能を実装することが考えられる.

謝辞

本研究は科研費（No.24500052）の助成を部分的に受けている.

参考文献

[1] 黒橋禎夫, 長尾真, “長い日本語文における並列構造の推定,” 情報処理学会論文誌, 第 33 巻, 8 号, pp. 1022－1031, 1992.
[2] 黒橋禎夫, 長尾眞, “並列構造の検出に基づく長い日本語文の構文解析,” 自然言語処理, 第 1 巻, 1 号, pp. 35-57, 1994.
[3] 吉村賢治, 自然言語処理の基礎[新訂版], サイエンス社, 2012.
[4] 首藤公昭, 吉村賢治, 津田健蔵, “日本語技術文における並列構造,” 情報処理学会論文誌, 第 27 巻, 2 号, pp. 183-190, 1986.
[5] 郡司隆男, 自然言語, 日本評論社, 1994.
[6] 吉村賢治, “日本語単一化文法による形態素解析と構文解析の融合,” 福岡大学工学集報, pp. 15-21, 2017.