論　文

Technical Papers

文法的不適格文処理のための統合的枠組み

An Integrated Framework for Processing Grammatically Ill-Formed Sentences

今一　修*　松本　裕治*
Osamu Imaichi　Yuji Matsumoto

* 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi 630-01, Japan.

1995年9月20日　受理

Keywords: natural language processing, ill-formedness, robust parsing, integration.

Summary

Interaction with computers using natural language has been a major goal of artificial intelligence. Though many natural language processing systems have been developed, most of them assume that input sentences are grammatically correct. However, when users communicate with the system, they often use grammatically ill-formed sentences, especially in spoken dialogues. For example, the users omit some words, change the word order, or make some careless errors such as agreement errors, misspellings or adding of extra words. To use NLP systems in real applications, we need to construct an NLP system that can handle not only grammatically well-formed inputs but also grammatically *ill-formed* inputs.

This paper describes an integrated method for processing grammatically ill-formed inputs. We use partial parses of the input sentence for recovering from parsing failure. In order to select partial parses appropriate for error recovery, cost and reward are assigned to them. The notion of *cost* and *reward* is introduced in order to select a partial parse appropriate for error recovery. Cost and reward represent the badness and goodness of a partial parse, respectively. The most appropriate partial parse is selected on the basis of cost and reward trade-off. Cost is calculated by *cost-based unification* proposed here.

The system contains three modules. Module $\mathcal{A}$ handles local ill-formedness such as constraint violations. Module $\mathcal{B}$ handles non-local ill-formedness such as word order violations, and Module $\mathcal{C}$ handles non-local ill-formedness such as contextual ellipses. These three modules work in a uniform framework based on the notions of cost and reward.

1. はじめに

自然言語を用いた計算機とのコミュニケーションは，人工知能の分野における主要な研究課題の一つであり，これまでに様々な自然言語処理システムが提案されてきている．それらの多くは文法的に適格な文を対象とするものであり，その処理においては，ある程度の成功を収めている．しかし，現実の文章や話し言葉には様々な非文法的な表現や誤りが含まれているため，実用的なシステムは文法的に適格な文だけではなく，文法的に不適格な文にも適切に対処する必要がある，ということが古くから指摘されている [Weischedel 80]．

問題となる不適格性が構文的なものであり，その種類と数があらかじめ想定できる範囲内であれば，それをシステムの文法に特殊規則として明示的に組み込むことにより，不適格文をあたかも適格文であるかのように処理することができる．しかし，実用的なシステムには，不適格文を処理する能力だけでなく，入力文中の曖昧性を解消する能力も要求されるため，適格文と不適格文を同等に扱うことは曖昧性増大につながり，非現実的である．さらに意味的に不適格な文も扱おうとすれば，極端に制約を緩める必要があり，曖昧性はさらに増える．したがって，入力文が不適格文である可

能性をあらかじめ考慮に入れて文法を緩和しておくのではなく，入力文が適格文として捉えられない場合にのみ何らかの手段を用いて文法を緩和して処理をするべきである．つまり，適格文を対象とした文法を用いて入力文の解析を行ない，それが失敗した段階で，不適格性の同定および修正・回復を行なうのがよい．

Weischedel ら [Weischedel 83] はこのような観点に基づき，**緩和法**と呼ばれる手法を提案している．この手法では，入力文の解析が何らかの不適格性により失敗した場合に，失敗の原因となる制約違反を同定し，その制約を緩和することによって処理を行なっている．しかし，彼らの手法では，複数の失敗の候補の間を比較するメカニズムが与えられていない．

効率良く不適格文を処理するためには，違反している制約の中から失敗の回復に有効なものを選択し，それを緩和することが必要になってくる．本論文では，その選択を適切に行なうために，個々の制約違反に関して，以下の 2 つの指標を考える．

コスト　その制約違反がどの程度のものか．

報酬　その制約違反を緩和したときに，どのような構造が生成されるか．

コストは，その制約を緩和するときの不適格性の許容度を示すもので，その値は小さい方がよい．報酬は，その制約違反を緩和することが，どれくらい失敗の回復に役立っているかを示す尺度であり，その値は大きい方がよい*1．

通常の緩和法では，制約違反によって解析が失敗すると，新たな句はつくられず，その制約を緩和したときに，その句がつくられる．これでは，上記 2 つの指標，特に報酬，を適切に見積ることができない．そこで，制約違反が起こったときでも不適格性を含んだ句をつくり，その際に，コストと報酬を計算して，その句に割り当てる．本稿では，文法記述として HPSG（Head-Driven Phrase Structure Grammar）[Pollard 94] を用いているが，通常の単一化演算では矛盾した情報を扱うことができないので，矛盾した情報を扱えるように拡張することによりコストの計算を行なっている．報酬は不適格性を含んだ句がどのような句かによって決定することができる．もちろん，この不適格性を含んだ句は，通常の解析には用いないので，不適格性を含んだ句をつくることによる通常の解析への影響はない．

通常の解析が入力文の処理に失敗したときに，コストと報酬の指標を用いて失敗の回復に適切な句を選択することによって不適格文処理が行なわれる．通常の緩和法との対応を考えると，不適格性を含んだ句を選択することは，その不適格性を含んだ句をつくるときに違反していた制約を緩和することに相当する．

さらに，語が持っている結合価に関する情報を用いて解析の失敗を回復する手法についても述べる．

本稿で提案する枠組みは，3 つのモジュールから構成される．モジュール $\mathcal{A}$ は，制約違反のような局所的な不適格性を扱い，モジュール $\mathcal{B}$ は，倒置などの非局所的な不適格性を扱う．モジュール $\mathcal{A}, \mathcal{B}$ は，文脈情報を必要としない，一文内で回復可能な不適格性を扱い，一文全体を覆う句を発見することを処理の目的とする．モジュール $\mathcal{C}$ は，これらの出力を受けとり，それに対して文脈情報を用いて適切な解釈を与える．モジュール $\mathcal{C}$ では，省略や断片的発話など，それ単独では意味のある発話として解釈できないものを文脈情報を用いることによって処理を行なう．以上の 3 つのモジュールは，すべてコストと報酬の概念に基づいて協調的に動作するので，単一の枠組み内で統一的に制御することができる．したがって，従来の手法に比べて，柔軟に種々の不適格性を処理することができる．

本稿では，主にモジュール $\mathcal{A}, \mathcal{B}$ について議論する．モジュール $\mathcal{C}$ については現在，研究を進めているところであり，その方向性と考え方を示すに留めておく．

2. 文法的不適格性

2・1 コーパスに出現する不適格文

どのような文法的不適格文が実際のコーパスに出現するかを調べるために，ATR 対話データベース [江原 90] を分析した．分析対象は，国際会議の申し込みに関する参加者と事務局の対話における電話対話文 1000 文（電話データ）とキーボード対話文 1000 文（キーボードデータ）である．分析結果を 表 1 に示す．以下，簡単にそれぞれの言語現象について述べる．

i. 語句の欠落

これは必要な語句が欠落している場合で，助詞の欠落と格要素の欠落に分けることができる．本分析では，電話対話文の 61.9%，キーボード対話文の 51.4%の文で語句の欠落が発生している．文脈から予測できたり，一度話題になった要素は省略されることが多い．また，「私」,「あなた」等のような，話の現場の要素は，主題として言葉で表現されていなくても，すでに話題に出ているとみられるときは省略される．話し言葉では，格関係が文脈から明らかである場合は，格助詞が省略さ

*1　コストと報酬を基準とする考え方は，関連性理論 [Sperber 86] が関連性の程度条件として用いている文脈効果と処理労力から着想を得たものである．本枠組みでは，関連性理論を文脈処理プロセスでの基本メカニズムとして採用している．

表1 コーパスに出現する不適格文の分析結果

不適格な現象	電話データ	キーボードデータ
語句の欠落	619	514
助詞	113	26
格要素	561	495
話者	445	328
聴者	170	185
主題	39	51
制約違反	9	2
余分な語句	644	4
挿入句	10	1
間投詞	635	3
自己修復	256	0
語順誤り（倒置）	5	0
省略	21	69
少なくとも一つ	810	581
計	1000	1000

れることが多い．助詞欠落において，欠落助詞の種類を分析した結果，必須格要素の助詞が欠落する場合がほとんどであることがわかった（電話データで約97%，キーボードデータで約89%）．

ii. 制約違反

日本語には，英語における数と性の一致などの統語的制約が少ないため，統語的制約違反はほとんど現れない．意味的制約違反としては，メタファー，メトニミー，擬人化などの選択規則違反がある．

iii. 余分な語句

話し言葉においては，「あー」，「えーと」などの間投詞，挿入句，自己修復文（言い直し）が頻繁に出現する．本分析でも，これらの現象が話し言葉に特徴的な現象であるという結果が得られている．挿入句は，文の途中に他の文単位が挿入されたものであり，発話の途中で情報の付加や追加をするために用いられる．自己修復文は，例1のように文の途中で発話を一旦中断して語句を言い直すことである．

例1 会議のこと，あのー，登録のことで伺いたいのですが．

iv. 語順の誤り

語順誤りの典型例は倒置である．日本語では，動詞の下位範疇化要素の語順は比較的自由であるが，動詞とその下位範疇化要素の語順は自由ではなく，下位範疇化要素は動詞の前に出現しなければならない．例えば，「太郎が学校へ行く」や「学校へ太郎が行く」は文法的であるが，「行く，学校へ太郎が」や「太郎が行く，学校へ」などは非文法的である．本分析によれば，倒置の0.5%しか出現せず，頻出する言語現象ではないことがわかる．

v. 省　略

ここで言う省略は，狭義の省略であり，必須格欠落のようなゼロ代名詞化は含まない．つまり，談話中での，それ自体では不完全で断片的な文法単位（主に名詞句）の使用を指している．日本語では，文脈から明らかな場合，主題（名詞句）だけを残して，述語部分を省略することができる．この場合，文の形を保つために，判定詞を付けるのが普通である．

例2 甲: 誰が来たの？
乙: 太郎です．

例2の対話において，乙は，甲の質問に対して，「太郎が来た」という完全な文で答えずに，答えの焦点となる要素「太郎」に判定詞「です」を付けた「太郎です」という断片文で答えることができる．

vi. 縮　約

話し言葉では，単語や語句の縮約が起こることがよくある．縮約の仕方は，各単語によって異なるが，よく使われるものとしては，次のような例がある．

- ～のだ → んだ

例3 ちょっと，急ぎますんで失礼します．

- ～ければ → きゃ
- ～れば → りゃ

例4 やめたきゃ，やめりゃいいじゃないか．

このような縮約は，対話文特有の語彙として書き言葉用の辞書に追加登録することによって処理することが可能である．したがって本稿では，縮約を不適格であるとは考えない．

2・2 不適格文の分類

前節の分析をもとに，不適格文を以下のように分類する．

タイプ1 制約違反
タイプ2 構造違反
タイプ3 文脈情報を要する不完全で断片的な句

タイプ1は文法が課している制約条件（統語的制約，意味的制約）の違反である．本稿では，助詞欠落もタイプ1に属するものと考える．日本語文法によっては，「名詞+助詞（後置詞）」を後置詞の投射（後置詞句）とし，後置詞を主辞と考えるものがあるが，本稿では，「名詞+助詞」を名詞の投射とし，名詞を主辞として捉える．助詞は（表層）格マーカーと考え，動詞が下位範疇化する名詞句は，助詞で格マーキングされていなければならないとする．したがって，助詞が欠落した名詞句は，この制約条件に違反していることになる．

日本語では制約自体が少ないため，このタイプに属する言語現象の数は少ないが，英語では，種々の統語的制約があるため，このタイプに属する言語現象の数は多く，これまで行なわれてきた研究のほとんどがこ

のタイプの不適格性を対象としている．

タイプ2には倒置などの語順誤りが分類される．例5では，名詞句「その本を」が倒置されている．

例5 私は読みました，その本を．

このタイプの不適格性は文法規則で許されていない統語構造を持っている．つまり，動詞句「私は読みました」と名詞句「その本を」をこの順で結合する文法規則はなく，文法規則に課せられた制約条件を緩和してもこの不適格性を扱うことはできない．本枠組みでは，このような不適格性に対して，語の結合価情報を用いて処理を行なう．

タイプ3には省略や格要素の欠落が分類される．この不適格性を処理するためには文脈情報が必要となる．例2で，乙の「太郎です」という答えは，甲の質問「誰が来たの？」という文脈がなければ解釈することができない．省略や格要素の欠落を不適格文と捉えることに関しては，様々な見解があるが，本研究では，このような省略は文脈情報がなければ適切に解釈できないという点で不適格であると考えている．したがって，厳密に言えば，文法的不適格文ではなく文脈的不適格文とでも言うべきものである．

タイプ1とタイプ2の不適格文は，文脈情報を使わずに処理することができ，一方，タイプ3の不適格文は，文脈情報を用いて処理しなければならない．また，タイプ1の不適格性は単一化文法 [Shieber 86] の枠組みでは単一化の失敗として捉えることができる．タイプ2とタイプ3の不適格性を処理するためには語の結合価情報や文脈情報を必要とする．

本稿では，以上3つのタイプを対象とし，自己修復，間投詞などの余分な語句は扱わない．

3. コスト付き単一化

単一化文法の枠組みでは，文法に課せられた制約条件の充足/違反を単一化演算の成功/失敗として捉えている．古典的な単一化では，単一化する素性構造間に矛盾する情報が検出されると単一化演算に失敗するため，矛盾する情報を扱うことができない．本稿では，不適格性を含んだ構造（つまり，矛盾する情報を含んだ構造）をつくるために単一化演算を拡張する．この拡張した単一化を**コスト付き単一化**と呼ぶ．

コスト付き単一化は以下の性質を持つ．

(1) 単一化する素性構造間に矛盾した情報が含まれていない場合は，古典的な単一化と等価である．

(2) 単一化する素性構造間に矛盾した情報が含まれている場合，

(a) コスト付き単一化は常に成功する．

(b) 結果となる素性構造にコストが付与される．

(c) 結果となる素性構造は矛盾した情報を保持する．

矛盾した情報は，**矛盾集合**によって保持される．ある素性に対して，単一化する素性構造間で値が異なっている場合に，これらの値を要素として持つ矛盾集合をつくり，この矛盾集合を結果となる素性構造におけるその素性の値とする．例えば，素性構造AのNUMBER素性の値が **singular**，素性構造BのNUMBER素性の値が **plural** である場合，コスト付き単一化の結果として得られる素性構造のNUMBER素性の値は $\top\{\mathbf{singular}, \mathbf{plural}\}$ となる．$\top$ は矛盾を示す記号である．

結果として得られる素性構造には，それに含まれる矛盾の度合に応じて**コスト**が付与される．素性構造のコスト C は以下の式で計算される．

$$C = \sum_{f \in \mathcal{F}} w_f \cdot e_f \tag{1}$$

ここで，$\mathcal{F}$ は素性の集合，f はその要素，w_f は素性 f に対する重み，e_f は素性 f が持っている余分な要素の数である．矛盾集合を値として持っていない素性の場合，e_f の値は0である．矛盾集合を値として持っている素性の場合，矛盾集合の要素数から1を引いたものが e_f の値となる．矛盾を含まない素性構造では，すべての素性に対して e_f の値が0であるのでコスト値は0となる．矛盾の度合の尺度としてコストを用いるためには，各素性に対する重み w_f を適切に設定する必要がある．矛盾を生じやすい素性と矛盾を生じにくい素性に対して適切な重みを，コーパス等の分析によって決定していく必要があるが，現在の実装では，すべての素性に対して重み w_f を1にしてコスト計算を行なっている．

コスト付き単一化の例を一つ示しておく．素性構造(2)と(3)のコスト付き単一化を行なうと，結果は素性構造(4)となり，コストの計算式(1)を用いると，この素性構造のコスト値は3であることがわかる．

$$\begin{bmatrix} \text{NUMBER} : \mathbf{plural} \\ \text{PERSON} : \mathbf{3rd} \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} \text{NUMBER} : \mathbf{singular} \\ \text{PERSON} : \top\{\mathbf{1st}, \mathbf{2nd}\} \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} \text{NUMBER} : \top\{\mathbf{singular}, \mathbf{plural}\} \\ \text{PERSON} : \top\{\mathbf{1st}, \mathbf{2nd}, \mathbf{3rd}\} \end{bmatrix} \tag{4}$$

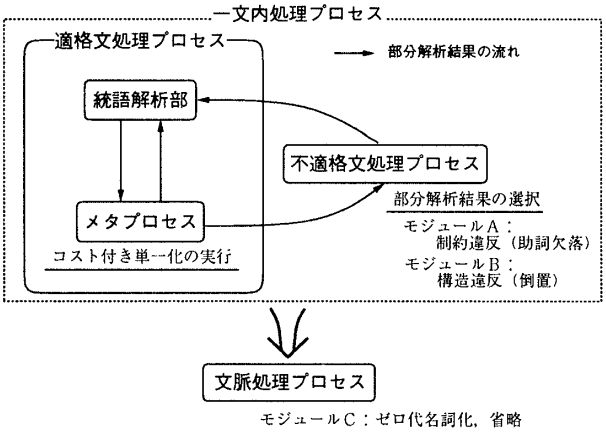


図1　提案する枠組みの全体構成

4. 不適格文回復のため統合的手法

4・1 全体構成

本稿で提案する枠組みは，一文内処理プロセスと文脈処理プロセスに分けられる（図1）．一文内処理プロセスは，適格文処理プロセスと不適格文処理プロセスから構成される．不適格文処理プロセスは，適格文処理プロセスが入力文の処理に失敗した場合にのみ起動される．以下，各プロセスについて順に説明する．

4・2 適格文処理プロセス

適格文処理プロセスは統語解析部とメタプロセスから構成されている．統語解析部では，上昇型チャート法[Kay 80]に基づいて入力文の解析が行なわれるが，ここでは，文法規則に課せられた制約条件の評価，つまり，コスト付き単一化は行なわれない．統語解析部は，新しくつくられる句とその句がつくられるために満足すべき制約条件の組をメタプロセスに送る．メタプロセスでは，コスト付き単一化を実行して矛盾の有無を調べ，その有無に応じて新しくつくられる句のフィルタリングを行なう．

例えば，以下の文法規則を考えてみる．

動詞句 → 名詞句，動詞，{ 制約条件 }.

統語解析部に名詞句と動詞の並びが与えられると，この文法規則を適用することによって動詞句がつくられるが，ここでは制約条件（例えば，名詞句は格マーキングされていなければならない，など）は評価されない．メタプロセスは，この動詞句と制約条件の組を受けとり，コスト付き単一化を実行することによって制約条件を評価する．新たなコストが検出された場合，この動詞句は以後の処理には使用されない．新たなコストが検出されなかった場合，この動詞句は統語解析部に送られ処理が継続される．

また，メタプロセスは解析が失敗した場合の不適格文処理のために，解析途中で生成されたすべての部分解析結果を不適格文処理プロセスに送る．

4・3 一文内での回復処理

〔1〕 部分解析結果の選択

入力文の解析に失敗すると，不適格文処理プロセスは，メタプロセスによって送られてきた部分解析結果の中から失敗を回復するために適切なものを選択する．この選択された部分解析結果が，統語解析部ですでに使われているものかどうかで，モジュール $\mathcal{A}, \mathcal{B}$ の起動が制御される．

失敗回復に適切な部分解析結果を選択するために，以下の基準を導入する．

- **部分解析結果の選択基準**
  - ○ 報酬最大・コスト最小の部分解析結果を選択.

報酬は以下の基準で決定される．

- **報酬の基準 A**
  - ○ 動詞句
  - ○ 名詞句
  - ○ その他の句
- **報酬の基準 B**
  - ○ 入力文の最も広い範囲を覆う句
  - ○ 最も右側にある句（日本語の場合）

報酬の基準は，報酬の順序付けを表している．基準Aは，動詞句，名詞句，その他の句の順で報酬が高いことを表している．基準Aだけで報酬の順序付けが決定できない場合，基準Bによってさらに順序付けがなされる．つまり，入力文中のより広い範囲を覆う句が報酬が高く，さらに，入力文中のより右側に位置する句の方が報酬が高い．

〔2〕 モジュール $\mathcal{A}$

選択基準に従って選ばれた部分解析結果が統語解析部でまだ用いられていない場合，モジュール $\mathcal{A}$ が起動され，タイプ1の不適格性が処理される．モジュール $\mathcal{A}$ が行なう処理は，この選択された部分解析結果を統語解析部に渡すことである．つまり，この部分解析結果に含まれる制約違反を緩和することになる．

例6のような助詞が欠落している文を考える．

例6 太郎ご飯食べる．

この文は，名詞「太郎」の後に助詞「が」，名詞「ご飯」の後に助詞「を」が欠落しているために解析に失敗するが，コスト付き単一化によりコスト付き部分解析結果は生成されている．図2において，点線で示されている弧がコスト付きの部分解析結果である．上側の部

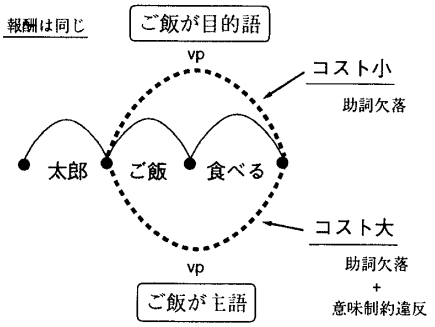


図2 モジュール $\mathcal{A}$ の例

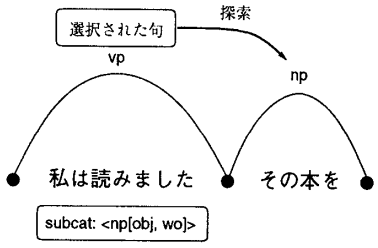


図3 モジュール $\mathcal{B}$ の例

分解析結果は，名詞「ご飯」が動詞「食べる」の目的語となる句であり，これに含まれる不適格性は，助詞欠落である．下側の部分解析結果は，名詞「ご飯」が動詞「食べる」の主語となる句であり，これに含まれる不適格性は，助詞欠落と意味的制約違反である．意味的制約違反は,「ご飯」は「食べる」の主語としては不適切であることに起因する．部分解析結果の選択基準にしたがって，報酬最大・コスト最小の部分解析結果を選択する．この例の部分解析結果は同じ報酬を持っているが，コストを比較すると，上側の部分解析結果の方がコスト値が小さいので，こちらが選択される．モジュール $\mathcal{A}$ は，この部分解析結果を統語解析部に送り，それを受け取った統語解析部は処理を再開する．

なお，報酬最大・コスト最小の部分解析結果が複数存在する場合（例えば，意味情報などを用いても助詞欠落の格が決定できない場合）は，それらはすべて選択され，曖昧性が生じるが，後の処理で解消可能であれば解消される．例えば,「太郎が花子誉める」という文において，動詞句「花子誉める」内では花子の格を決定することができない．しかし，名詞句「太郎が」とこの動詞句を結びつける際に，花子が動詞「誉める」の目的格であることが決定される．

〔3〕 モジュール $\mathcal{B}$

選択された部分解析結果が，統語解析部ですでに用いられており，その句が未完成な句である場合，モジュール $\mathcal{B}$ が起動され，タイプ2の不適格性を扱う．モジュール $\mathcal{B}$ のアルゴリズムは以下の通りである．

(1) 選択された句の結合価情報（下位範疇化情報，修飾情報）を参照することにより，この句が完成するために必要な句が何であるかを同定する．

(2) その候補となる句の探索を行なう．

(3) 発見された句が完成した句であるか未完成の句であるかに応じて以下の処理を行なう．

- 完成した句のとき
  - 選択された句がその結合要素に課している制約条件を評価する．
  - 選択された句と発見された句を結びつけた結果を統語解析部に送る．
- 未完成の句のとき
  - この句を次の選択された句とし，このアルゴリズムを再帰的に呼び出す．

例えば，例7の倒置文を考えてみる（図3参照）．

例7 私は読みました，その本を．

この文では動詞句「私は読みました」と名詞句「その本を」が倒置されている．この2つの句をこの順序で結合させる文法規則は存在しないため，この2つの句を覆うコスト付きの句は生成されず，この不適格文はモジュール $\mathcal{A}$ では扱うことができない．まず，句の選択基準によって，動詞句「私は読みました」が選択される．この句は，すでに統語解析部で用いられている句であるので，モジュール $\mathcal{B}$ が起動される．モジュール $\mathcal{B}$ は，この動詞句「私は読みました」の結合価情報，つまり，下位範疇化情報を参照し，この動詞句が完成した句となるために必要な句が，助詞「を」で格マーキングされた名詞句であることを同定する．次に，この名詞句を探索し，名詞句「その本を」を発見する．動詞句「私は読みました」が，その下位範疇化要素である名詞句に課している制約条件を評価し，動詞句「私は読みました」と名詞句「その本を」を結合させて動詞句「私は読みました，その本を」をつくり，これを統語解析部に送る．

この例において，倒置されている名詞句が「その本」のように助詞が欠落している場合，動詞句「私は読みました」がその下位範疇化要素である名詞句に課している制約条件の評価の際に，矛盾が発見されるため，新たにつくられる動詞句「私は読みました，その本」はコストを持つことになる．したがって，統語解析部にこの動詞句が送られても利用されないので，さらに不適格文処理が続けられる．その過程で，この動詞句が選択され，モジュール $\mathcal{A}$ によってこの句が統語解析部に送られることにより，処理が完結する．

4・4 文脈情報を用いた処理

一文内処理プロセスの目的は，一文全体を覆う句を発見することである．モジュール C では，その結果を受けとり，それに対して文脈情報を用いて適切な解釈を与える．モジュール C は，タイプ3の不適格性を扱い，意味処理を中心に行なう．

例えば，例8のような発話対を考えてみる．

例8 甲: 何を読んでいるの？
乙: 小説．

乙の「小説」という発話に対し，一文内処理プロセスは，名詞「小説」を結果としてモジュール C に送る．このような断片的な発話に対して省略されている要素（この例では，動詞「読む」）を補うことにより解釈するのではなく，この断片的な発話が文脈上で適切なものと解釈できればよいと考える．この例では，名詞「小説」が，甲の質問「何を読んでいるの？」の適切な答えであればよいとする．名詞「小説」は,「何を読んでいるの？」の意味表現において具体化されていない目的格を占める要素として意味的に適切であるので，この場合「小説」という名詞の断片は，この文脈で適切に解釈できる．

このモジュール C に関しては，関連性理論が提唱している関連性の原則による処理の実現に向けて，研究を行なっている．

5. 従来の手法との比較

緩和法は，Weischedelら [Weischedel 83] によって提案されたものである．彼らの手法には，

(1) ATNを用いたバックトラック処理に基づいているため，誤り箇所と解析の失敗の位置にずれを生じることがある．
(2) 失敗の原因毎に処理の仕方を考える必要がある．
(3) 複数の失敗の候補の間を比較するメカニズムが与えられていない．

という欠点があった．(1) は，解析手法としてチャート法などのようなバックトラックを伴わない手法を用いることにより回避できる．(2) は，不適格性を個々の制約が違反したものとして扱うのではなく，単一化文法のように単一化の失敗としてメタレベルで捉えることによって，あらかじめ想定していなかった失敗についても扱うことができる．(3) に関しては，本稿で提案したコストと報酬を基準とした手法を用いることにより複数の失敗の候補の間の比較を行なうことができる．

緩和法を応用したものとして，段階的に文法規則を緩和する手法がある [Douglas 92]．この手法では，文法をいくつかのレベルに分け，入力文の解析に失敗した場合に文法のレベルを段階的に緩めて処理を行なっている．しかし，どのように文法を適切にレベル分けするのか，という問題点がある．我々の手法は，コスト付き単一化における矛盾度（コスト）によって連続的に不適格性のレベルを設定することができる．

緩和法とは異なるアプローチをとっている研究もいくつか行なわれている．

Jensen[Jensen 83] は，解析に失敗した場合，それまでに得られた構造の中から一番もっともらしい構造の列を選び，それを後の処理（意味処理）の入力とするfitted parse と呼ばれる手法を提案している．この方法は，解析システムを引き続く処理の前処理と位置付け，何らかのもっともらしい結果を返すようにしている．文献 [Hobbs 92, McDonald 92] においても 部分的に解析できた構造の中から意味のある情報を抽出する部分解析と呼ばれる手法が提案されている．

しかし，これらの手法では，失敗した箇所から先の解析は断片を集める以外に方法がなく，解析の初期の段階で失敗が発生した場合，構成素間の関係を正確に捉えることができない．これに対して，我々の手法では，不適格文処理によって失敗箇所の回復をした後は，通常の処理によって処理が引き継がれるために，構成素間の関係を適切に捉えることができる．

文献 [Kato 94, Mellish 89] では，チャート法を拡張することによって，語の欠落や未知語，語の誤りなど統語的な不適格性を扱う手法が提案されている．基本的な手法としては，まず，上昇型チャート法により入力文の解析を行ない，解析に失敗した場合に下降型チャート法を用いて，失敗の原因を同定するものである．

Wilksが提案した優先意味論 [Fass 83] の枠組みでは，制約条件を優先条件と捉え直すことにより，意味処理を中心とした手法が用いられている．優先条件は，必ず満足されなければならない条件ではなく，あくまでも優先条件であるので，違反が起こった場合は単にその条件は適用されないだけである．この手法では，単一の枠組みの中で意味的な制約が曖昧性解消に用いられたり，必要な場合は，それらを緩和したりしている．このように，いくつかの解釈の中から最も確からしい解釈を優先させることは，不適格文処理においても有効である．しかし，この手法は独立した統語解析部を持たず，意味中心の手法にどのように統語的な制約を組み込むかは明らかにされていない．

これら2つの手法は，統語情報，意味情報をそれぞれ単独に用いて不適格文を処理している．しかし，よ

り柔軟に不適格文の処理を行なうためには，統語情報，意味情報を統合して利用することが必要である．本稿では，HPSGを用いることにより，統語情報，意味情報，さらに，文脈情報を統合的に利用することができ，より柔軟な処理が可能である．

6. おわりに

本稿では，統合的に不適格文処理を行なう枠組みについて提案した．本枠組みは3つのモジュールからなっている．モジュール $\mathcal{A}$ は，従来研究で対象とされてきた制約違反全般を扱うことができ，モジュール $\mathcal{B}$ では，制約違反に起因しない現象を語の情報を利用することにより扱うことができる．モジュール $\mathcal{C}$ は，文脈情報を用いることにより，一文内だけでは処理できない現象を扱う．これら3つのモジュールがすべて，コストと報酬の概念をもとに単一の枠組みに統合されており，全体として柔軟な不適格文処理が実現されている．コストにより部分解析結果に含まれる矛盾の度合が表され，報酬により部分解析結果が失敗の回復に役立つ度合が表されている．また，個々のモジュールは記述のレベルでは明示的に分離されているため，容易に拡張を行なうことができる．

本稿では，一文内で回復可能な不適格性に対する処理を中心に述べた．モジュール $\mathcal{A}, \mathcal{B}$ に関しては，計算機上でプロトタイプシステムを実装して，その有効性を確認した．大規模なシステムを構築するためには，3章で述べたように，コストの見積りを適切に行なう必要があり，コーパス等から適切なコストの見積りを得ることを考えている．また，日本語では，一文内だけでは処理できない，文脈情報を必要とする不適格性が数多く存在する．本枠組みでは，モジュール $\mathcal{C}$ がこれらの不適格性に対処する．今後は，関連性理論やその実装 [Hirasawa 95, Poznański 92] を参考にしながら，モジュール $\mathcal{C}$ の定式化および実装を行なっていく予定である．

◇参考文献◇

[Douglas 92] Douglas, S. and Dale, R.: Towards Robust PATR, *Proc. of COLING-92*, pp.468–474 (1992).

[江原 90] 江原暉将，井ノ上直己，幸山秀雄，長谷川敏郎，庄山富美，森元逞：ATR対話データベースの内容，Technical Report TR-I-0186, ATR自動翻訳電話研究所 (1990).

[Fass 83] Fass, D. and Wilks, Y.: Preference Semantics, Ill-Formedness, and Metaphor, *Computational Linguistics*, Vol.9, No.3-4, pp.178–187 (1983).

[Hirasawa 95] Hirasawa, J.: Contextual Interpretation of Utterances in Relevance Theory, Master's Thesis, NAIST-IS-MT351092, Nara Institute of Science and Technology (1995).

[Hobbs 92] Hobbs, J.R., Appelt, D.E., Bear, J. and Tyson, M.: Robust Processing Real-World Natural-Language Texts, *Proc. of 3rd Applied Natural Language Processing of ACL*, pp.186–192 (1992).

[Jensen 83] Jensen, K., Heidorn, G.E., Miller, L.A. and Ravin, Y.: Parse Fitting and Prose Fixing: Getting a Hold on Ill-formedness, *Computational Linguistics*, Vol.9, No.3-4, pp.147–160 (1983).

[Kato 94] Kato, Y.: Yet Another Chart-Based Technique for Parsing Ill-Formed Input, *Proc. of 4th Applied Natural Language Processing of ACL*, pp.107–112 (1994).

[Kay 80] Kay, M.: Algorithm Schemata and Data Structure in Syntactic Processing, Technical Report CSLI-80-12, Xerox PARC (1980).

[McDonald 92] McDonald, D.D.: An Efficient Chart-based Algorithm for Partial-Parsing of Unrestricted Texts, *Proc. of 3rd Applied Natural Language Processing of ACL*, pp.193–200 (1992).

[Mellish 89] Mellish, C.S.: Some Chart-based Techniques for Parsing Ill-Formed Input, *Proc. of 27th Annual Meeting of ACL*, pp.102–109 (1989).

[Pollard 94] Pollard, C. and Sag, I.A.: *Head-Driven Phrase Structure Grammar*, The University of Chicago Press (1994).

[Poznański 92] Poznański, V.: *A Relevance-Based Utterance Processing System*, PhD Thesis, University of Cambridge (1992).

[Shieber 86] Shieber, S.M.: *An Introduction to Unification-Based Approaches to Grammar*, No.4 in CSLI Lecture Notes, CSLI (1986).

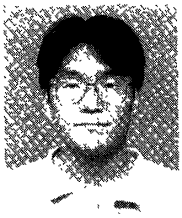
[Sperber 86] Sperber, D. and Wilson, D.: *Relevance*, Blackwell (1986).

[Weischedel 80] Weischedel, R.M. and Black, J.E.: Responding Intelligently to Unparsable Inputs, *Computational Linguistics*, Vol.6, No.2, pp.97–109 (1980).

[Weischedel 83] Weischedel, R.M. and Sondheimer, N.K.: Meta-rules as a Basis for Processing Ill-Formed Input, *Computational Linguistics*, Vol.9, No.3-4, pp.161–177 (1983).

〔担当編集委員：石崎　俊，査読者：安原　宏〕

著者紹介



今一　修(学生会員)

1969年生まれ．1993年京都大学工学部電気工学第二学科卒業．1995年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．現在，同博士後期課程在学中．自然言語処理の研究に従事．言語処理学会会員．〈osamu-im@is.aist-nara.ac.jp〉

松本　裕治(正会員)

1955年生まれ．1977年京都大学工学部情報工学科卒業．1979年同大学大学院工学研究科修士課程情報工学専攻修了．同年電子技術総合研究所入所．1984～85年英国インペリアルカレッジ客員研究員．1985～87年（財）新世代コンピュータ技術開発機構に出向．京都大学工学部助教授を経て，1993年奈良先端科学技術大学院大学情報科学研究科教授，現在に至る．専門は自然言語処理．情報処理学会，日本ソフトウェア科学会，言語処理学会，日本認知科学会，AAAI，ACL，ACM各会員．〈matsu@is.aist-nara.ac.jp〉