

## Pattern Lattice を使った (ヒトの) 言語知識と処理のモデル化

黒田 航

NICT けいはんな研究所 言語基盤グループ

長谷部 陽一郎

同志社大学/NICT

## 1 はじめに: パターンの理論の必要性

## 1.1 言語の創造性を再考する

N. Chomsky が生成文法の枠組み [1] の中で提唱し、その後、多くの研究者に受け入れられた幾つかのテーゼの一つに、(1) の言語の創造性のテーゼがある:

- (1) a. ヒトは自分が聞いた/読んだことのない新しい表現を聞いた/読んだ時に、それを理解できる。
- b. ヒトは自分が聞いた/読んだことのない新しい表現を作り出せる。

Chomsky は (1) を説明するために生成文法を考案したと主張している。だが、(1) を説明する<sup>1)</sup>のに、本当に生成文法は必要だろうか? 少なくともヒトの記憶に関して別の見方を取り、(2) にあるような仮説を想定する限り、生成文法と同じくらい信憑性のあるモデルを考えることは (理論的には) 難しくない:

- (2) 極端 (に豊か) な記憶の仮説<sup>2)</sup>
  - a. ヒトは思い出せない (=意識の中に呼び出せない) ような記憶を膨大に蓄積している。
  - b. 意識的に思い出せない多くの記憶も、適当な刺激があれば思い出せる。

要するにヒトは経験したことをすべて (暗黙知として) 記憶しているが、そのほとんどが思い出せない状態にある (経験の内容は「覚えるは易し、思い出すは難し」という奇妙な特性をもつ)。

(2) を想定する限り、(1a) の問題は、次のような知識系と処理系を考えることで解決できる見込みがある:

- (3) a. ヒトは自分が聞いたことのない表現  $e$  を聞いた/読んだ時に、それを理解できるのは、 $e$  と同一ではないが、それと部分的に一致する (つまり  $e$  に「似た」) 表現の集合  $e'_1, e'_2, \dots, e'_n$  を (その自覚はないが) 覚えていて、 $e$  の意味を、 $e'_1, e'_2, \dots, e'_n$  の意味を組み合わせで構築している。
- b. ヒトは今までに自分が聞いた/読んだことのない、新しい表現  $e$  を作り出せるのは、 $e$  で言うべき意味を、 $e'_1, e'_2, \dots, e'_n$  の意味  $m'_1, m'_2, \dots, m'_n$  を組み合わせで構成でき、それに見合うように  $e'_1, e'_2, \dots, e'_n$  を統合できるからである。

(3a) は ([3] のような枠組みに限らず)、今の言語処理が明示的、暗示的に置いている想定である。例えば機械翻訳の分野で用例/事例ベースの (機械) 翻訳 [6] や記憶

ベースの (機械) 翻訳 [9] という形で実装されている処理系はいずれもこの想定の下で行われている。だが、(3b) は (3a) ほど一般的には了解されていない。この非対称性は奇妙である。

同じことを別の視点で記述するとこうなる: 現在の言語処理の主流は事例基盤であるが、過去を振替えると、90 年代に文法基盤モデルから事例基盤モデルの移行があったことがわかる。その理由は二つある: 消極面では規則基盤の処理の限界が見え、積極面では大規模コーパスが利用可能になり、統計的手法が分野を席捲した。

だが、言語処理を産出時の処理と受領時の処理に分けた時、うまく行っているのは後者のみである。実際、統計処理と互換性のある事例基盤の言語産出のモデルは、今の時点で明確なモデルとしては存在していない。本発表の狙いは事例集合の Pattern Lattice (PL\*) の理論によってこの空隙を埋めることにある<sup>3)</sup>。

(3b) でも (3a) でも本質的な条件は、効率の良い類似例の検索である。これを可能にするのは何か? — これが PL\* を使って示そうをすることである。

## 1.2 事例と事例集合のパターンのラティス

§1.2.1 で事例の定項の変項化という操作を定義し、その定義の下に §1.2.3 で単一事例  $e$  のパターンラティス  $PL(e)$  を定義する。その定義を、続く §1.2.5 で事例集合を扱えるように一般化する。

## 1.2.1 事例の定項の変項化の定義

事例  $e$  を適切に  $T(e) = [t_1, t_2, \dots, t_n]$  に分割するモデルが与えられているとする (この理論が満足すべき条件については §2.1.2 で後述する)。この仮定の下で、 $T$  の適当な定項  $t_i$  を変項  $X$  で置換する操作を  $t_i$  の  $X$  による変項化と定義し、この操作の産物を  $e$  から派生したパターン (patterns derived from  $e$ ) と定義する。例えば  $e =$  「彼は歌って踊った」の分割は (適当な分割のモデル  $M$  の下では)  $T(M, e) = [\text{彼}, \text{は}, \text{歌っ}, \text{て}, \text{踊っ}, \text{た}]$  であり、5 番目の部分「踊っ」を  $X$  で変項化したものは「彼は歌って  $X$  た」というパターンである。変項  $X$  のタイプに制約を設けるかどうかは独立の問題とする。

## 1.2.2 言語情報の内部表現に関する想定

知覚された言語情報は内部表現 (mental representations) に変換され、処理・維持される必要がある。内部表現は、言語学や言語処理では言語表現の解析 (parses) と同一視されるのが通例であるが、私たちはモデル化の出発点として、解析が (再利用の可能性を最大にするために) 次の特徴をもつべきである<sup>4)</sup> i) 最小限の事前知識で盲目的に/機械的に実行可能 (intelligence-free);

<sup>1)</sup> 本稿では追求しないが、(1a) と (1b) は同列に扱うことはできない。実際、これらの間には驚くべき非対称性がある。ヒトが (1a) の意味で創造的なのは妥当な想定かも知れないが、ヒトの産出は実際にはかなり保守的であり、ヒトが (1b) の意味で創造的であるかどうかは疑問が残る。

<sup>2)</sup> 詳細は [12] を参照されたい。

<sup>3)</sup> もう一つの動機は第一著者が協力者と一緒に進めている複層意味フレーム分析 (MSFA) を使った意味タグづけ [14] で、状況の記述と優先して対応づけるべき超語彙パターンをなるべく効率良く見つけたいという希望である。これは非線型表現のデータベース化 [11] の効率化とも関係する。

<sup>4)</sup> 句構造はこれらの条件 (特に (i) の条件) を満足しない。

ii) 再分析 (=解析のやり直し) 不要 (reanalysis-free) . 私たちは §1.2.1 で定義したパターンがこれらの条件を満たす内部表現であると考えられる .

### 1.2.3 単一事例 $e$ のパターンラティス $PL(e)$

§1.2.1 の変項化の定義の下で, 事例  $e$  のパターンラティス  $PL(e)$  を次のように定義する:

- (4) a.  $e$  の分割の結果を  $T(e) = [t_1, t_2, \dots, t_k]$  とする .  $T(e)$  の要素を再帰的に単一の変項で変項化し, 得られたパターンのべき集合を  $P(e)$  とする .
- b. 次の is-a 関係の下での  $P(e)$  の半順序集合を  $e$  のパターンラティス  $PL(e)$  と定義する .
- c.  $p_i, p_j \in P(e)$  であるパターン  $p_i$  の  $n$  番目の要素  $p_i[n]$  とパターン  $p_j$  の  $n$  番目の要素  $p_j[n]$  との関係で, i)  $p_i[n] = p_j[n]$  であるか, ii)  $p_j[n]$  が変項ならば,  $[p_i \text{ is-a } p_j]$  である .

$PL(e)$  の頂点 (top) は  $k$  個の変項のみからなるパターンで,  $PL(e)$  の底 (bottom) は  $k$  個の定項のみからなる事例  $= e$  である .

豊かな事例記憶の想定の下では, パターンは事例集合へのインデックスになっていけばよい . 別の言い方をするとパターンはスキーマとして事例とは独立に自律的な内容をもっている必要はない (その内容は常に事例集合の値の期待値として与えられる) .

### 1.2.4 パターンのランクと $PL(e)$ の部分集合への分割

パターン  $p$  に含まれる定項の数を  $p$  のランクと定義する . 例えば  $p = [\text{彼, は, } X, \text{ て, } X, \text{ た}]$  (i.e.,  $[p_i, \text{は}, V_1, \text{て}, V_2, \text{た}]$  is-a  $p$ ) のランクは 4 である .

ランクは定項の数で  $PL(e)$  を部分集合に分割する . 一般に  $e$  が  $k$  個の部分に分割される時,  $PL(e)$  の Rank 0 は  $k$  個の変項のみからなるパターン  $XX \dots X (= \text{top})$  のみをもつ集合, Rank  $k$  は  $\{e (= \text{bottom})\}$  である .

### 1.2.5 事例集合のパターンラティス $PL^*$

$PL(e)$  は一つの事例  $e$  のパターンラティスである .  $PL(e)$  は  $e$  と分割数が同じ事例と互換性があるが, 異なる分割数の  $PL$  とは互換性がない . この点は次の仕方で異なる長さのパターンを統合することで解決できる:

- (5) 変項の再帰的単純化:<sup>5)</sup> 任意の連続した  $l$  個の変項列  $X$  と連続した  $l-1$  個の変項列  $X'$  について,  $[X' \text{ is-a } X]$  が成立する .

事例集合  $E = \{e_1, e_2, \dots, e_n\}$  のパターンラティス  $PL(E)$  を参照の便宜のため,  $PL^*$  で表わす .

### 1.3 $PL^*$ 上の統語処理と意味処理

ランクが  $k$  の事例  $e$  は, ランクが  $k-1$  の (超) 語彙的パターン  $p_1, p_2, \dots, p_k$  の重ね合わせ (=素性の論理和) である . 従って,  $e$  は  $p_1, p_2, \dots, p_k$  から非排他的に意味的, 音韻的資源を継承する . これは  $e$  の意味処理の際に, ランクが低い超語彙的パターンの方がランクの相対的に高い (超) 語彙的パターンよりも事例に「近く」, その分だけ影響が強いことを意味している (§2.2.2 で例を示す構文効果の原因はこれだと考えられる) .

なお, 本稿では  $PL^*$  上での意味処理の詳細に立ち入る余裕はない . 興味がある方は [13] を参照されたい .

### 1.4 $PL^*$ の試験的実装: Pattern Lattice Builder

§1 の  $PL^*$  の定義に基づいて  $PL^*$  の処理システム Pattern Lattice Builder (PLB) を試作的に実装し, [http://](http://www.kotonoba.net/rubyfca/pattern)

[www.kotonoba.net/rubyfca/pattern](http://www.kotonoba.net/rubyfca/pattern) で公開した . i) 一行に一事例 (要素分割はスペース挿入で指定) で  $N$  行までの入力を受けつけ,  $PL^*$  を可視化する (上記の Web サーバー上での  $N$  の上限は 30 個だが, ローカルインストールでは自由に変更できる) . その際, ii) 指定した事例に寄与する is-a リンクを色づけする; iii) 同一ランク内でパターンのもつ事例数の  $z$  スコアを求め, それを色温度に変換した (これにより, 生産的=データの説明力の高いパターンをそうでないパターンから区別できる)<sup>6)</sup> の機能をもたせた . 図 1 に {彼, は, 歌っ, た; 彼, は, 踊っ, た; 彼, は, 歌っ, て, 踊っ, た; 彼, は, 踊っ, て, 歌っ, た} を与えた時の PLB の出力例を示す . Rank 3 では [彼 は た] の生産性をもっとも高く, それに続くのが [彼 歌った] と [彼 踊った] であることがわかる .

## 2 議論

$PL$  には実装に抛らない不利点と利点とがある . これらについておのおの論じる .

### 2.1 $PL^*$ 基盤の記述の不利点

#### 2.1.1 分割数の増大と組み合わせ爆発

Pattern Lattice を使った最大の難点は, おそらく (A) 組み合わせ爆発に起因する記憶と処理の非効率性, 並びに (B) 規模の拡大可能性 (scalability) の問題である .

$PL(e)$  のノード数は,  $e$  が  $n$  個の分割をもつ場合,  $2^n$  である .  $PL^*$  の複雑性  $C$  は,  $e$  ごとの分割の数  $k$ , 分割の異なり数  $l$  で決まるが,  $l$  より  $k$  に依存する度合いが強い . 分割数が大きくなると組み合わせ爆発が起こる .

事例の分割数  $k$  に計算論的な上限があるという事実は, 言語処理の観点から見る限り難点でしかないが, 言語の認知科学の観点から見ると, 逆に重要な含意をもつ .

#### 2.1.2 処理範囲の最適化

今のところ十分な根拠を示すことはできないが, 組み合わせ爆発に関連して一つ, 興味深いと思われる点がある:

- (6) 特に  $e$  の分割数と  $PL(e)$  の複雑性の対応には, (分割数が 7 を境にして) (相転移に似た) 質的变化がある .

分割数  $k$  は多ければ良いというわけではなく, 課題に応じて最適値が決まるようである . 経験的には, 単文の項構造を記述するのに十分な被覆率を確保するには 7 個 (= [3, 1, 3]) の分割が必要で, 頻度は低い少し複雑な場合を取り入れるためには 9 個 (= [4, 1, 4]) の分割が必要になるという感じである . 具体的に言うと,  $PL(e)$  を構成する全パターンのうち, 意味をもつという直観が容易に得られるパターンの比率  $r$  を考えると, 文節数が 7 個を超えると,  $r$  が急に低くなるように思える<sup>7)</sup> . これは依存関係の候補を見つけるための窓が, 標的の左 (=過去) に -3 個 (か -4 個), 右 (=未来) に +3 個 (か +4 個) ぐらいのスパンにあるということである . この観察は今のところ主観的なものにすぎないが, 将来的には 7 個を境に本当にそのような変化があるのかを検証したい .

#### 2.1.3 分割の最適化と処理の複層化

分割数/計算の複雑性の自然な上限が存在するならば, それは言語単位の時間的幅/処理の深さに応じて, 最適な分割が, 異なったレベルに幾つか併存することが理論的に必然化すると考えられる . 例えば (M) 語の内部

<sup>6)</sup> 使用したのは <http://www.graphviz.org/doc/info/colors.html> の rdbu9 color scheme (区間の幅は 0.5z) .

<sup>7)</sup> この数に不思議な数 7 [8] との関係を読み取るのは, 必ずしも牽強附会とは言えないだろう .

<sup>5)</sup> PLB の実装では  $l$  個の変項を一つの変項に置換するという簡略形で実装している .

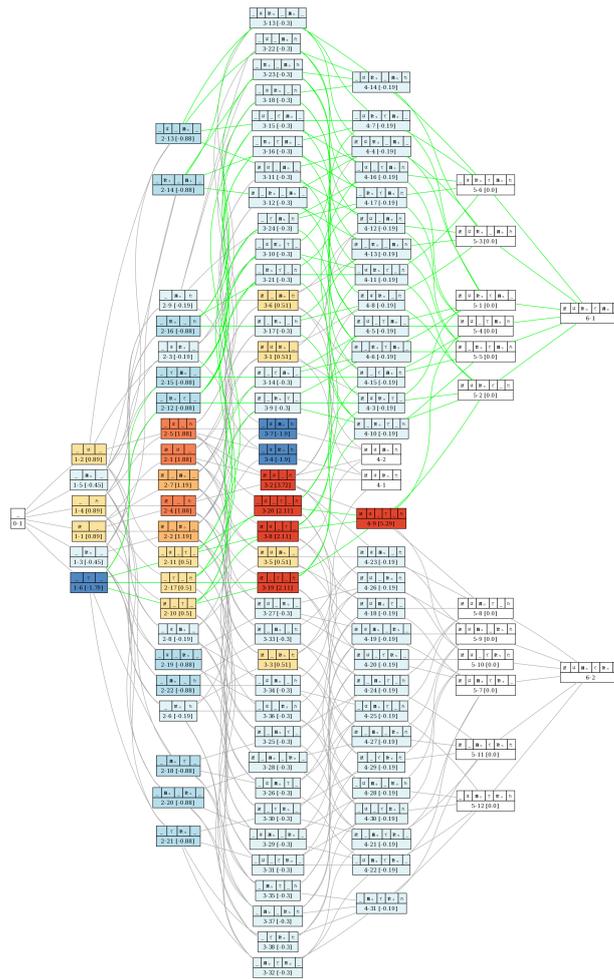


図1 Rank 0, Rank 1, ..., Rank 6 のパターンの個数は、おのその 1, 6, 22, 38, 31, 12, 2 (合計 112) である

構造=形態論での分割の最適化, (S) 文の内部構造=狭義の統語論での分割の最適化, (D) 話の内部構造=広義の統語論での分割の最適化は別のものであり, かつ, おののが言語の処理レベルに対応していると考えられる. 分割単位の恣意性は, PL\*の理論の弱点というより, 処理の複層化された分割の複数の最適化を保証する利点であると考えべき可能性が残される<sup>8)</sup>.

#### 2.1.4 段階を踏んだパターンの獲得

組合わせ爆発は機械上の言語処理で問題になるばかりでなく, 言語獲得においても深刻な問題となる. だが, 別の見方をすると, 言語獲得の際に子供がどうやって組合わせ爆発を問題を回避しているかという形で, 言語獲得の謎を解明する契機になる可能性もある.

語彙的パターン (=R1 のパターン) とランクの低い超語彙的パターン (e.g., R2, R3) は, 分割数の影響を強く受けず, それらは分割数が大きい場合でも流用可能であることに注意されたい. これから, 子供はランクの低い

語彙的, 超語彙的パターンを先に獲得し, それをランクの高い, 複雑な事例に流用するという戦略を取っている可能性が考えられる. これは規模の拡大可能性を保証する「最初は少なく (starting small)」 [4, 2] の原理に従っていると考えられる.

以上の理由から, PL\*では分割単位の認定の問題を,  $e$  の分割数  $k$  の決定の問題から意図的に独立させない.

#### 2.2 PL\*基盤の記述の利点

PL の利点は (以上の不利点と引換えにはあるが) 少なくとも (7) に示した, 互いに関連しあった利点をもつ:

- (7) a. ヒトの言語処理の記述と説明において, 言語学的理論 (aka 先入観) の干渉を最小限にできる (少なくとも句構造は不要であり, 極端なことを言うと品詞ラベルも不要).
- b. 構文効果 [5, 15, 10] に代表される超語彙的パターン/非線型表現 [11] の意味貢献を非アドホックに, 体系的に記述可能

##### 2.2.1 言語処理での「文法」の役割の最小化

PL\*基盤の処理システムでは, 「文法」の役割は極小化されている. 極論すると, PL\*基盤の処理システムは「辞書」だけで動いていると言っても良い (明らかに句構

<sup>8)</sup> 日本語に関して言うと, M, S, D レベルでの助詞の機能分化がありそうだ: 「〜と」「〜て」「〜で」「〜た」「〜だ」は述語間の共起関係 (D レベル) を, 「〜が」「〜を」などは述語内の要素間の共起関係 (S レベル) を, 「〜な」「〜の」(と「〜に」) は, 句内の要素間の共起関係 (M レベル) をエンコードしているようだ.

造はない) . そればかりか, 品詞ラベルすら無用化されている(少なくとも変項の実現値は意味的に制約されるので, 品詞の上での制約は(あって困らないが) 必要不可欠ではない) . しかし, 記述に必要な般化は十分に起こっており, 効果的な選択制限の記述すら可能である .

実際, この特徴の派生的な効果として, 池原ら [11] が進めてきたパターン翻訳で非線型パターンを自動的に発見することが可能である<sup>9)</sup> .

### 2.2.2 構文効果の説明の実例

李 [15] は (8) の用法で二格名詞句を認可するのは「消え(る)」の語彙的な意味ではないと論じている:

- (8) a. 患者が診察室に消えた  
b. テールランプが(暗)闇に消えた

構文上の意味は [N1 が N2 に V] というパターンに帰着できるわけではない . (9) は移動の意味はもたない:

- (9) a. 彼が知人に会った  
b. 子供が親に似ている (のは当然だ)

李の結論は, 構文上の意味の担い手は抽象的なパターン [N1 が N2 に V] ではなく, [[Human] が [Location ∨ Space] に V] のような, N1, N2 の意味クラスに言及するもう少し具体的なパターンだというものである .

意味クラスは (i) 事例集合を通じて (分布類似度の高いクラスとして) 獲得される, (ii) 意味クラスは (有限集合に限って言うと) 値の集合で表現できるという二点を考えると, PL は明示的に [N1 が N2 に V] のような「格パターン」の N の意味クラスに言及してはならないが, それを表わすのと同じタイプの一般化を表現できる . そればかりでなく, PL ベースの記述では, パターン間の階層的關係を明示的, かつ体系的に記述できるという利点がある . 実際, 次のことが PL\* の定義から予測され, 事実は予測の通りだと思われる:

- (10) パターンを構成する変項は, ランクが高いほど (e.g.,  $R=1, 2$ ) 潜在的意味クラスとの対応が弱く, ランクが低いほど意味クラスとの対応が良い .  
(11) 構文「効果」は (10) の想定の下で作用する超語彙的パターンの変項の補完の産物である<sup>10)</sup> .

## 3 終わりに

### 3.1 課題と将来への展望

PLB にデータベースをもたせ, 超語彙的パターンのデータベース化を行いたい . これにより十分な被覆率をもった超語彙的パターン/非線型表現=構文のデータベースが得られる可能性が現実的なものとなる .

### 3.2 言語の創造性は「豊かな事例記憶」の随伴事象

本発表で私たちは (3) の記憶ベースの言語知識のモデル化として事例集合の Pattern Lattice (PL\*) を提案した<sup>11)</sup> . PL は事例基盤の言語処理で有用なデータ構造に基礎を与えるだけでなく, 理論言語学で用法基盤アプ

ローチ (Usage-based Approach) と呼ばれる枠組みにも理論的基礎を提供すると考えられる .

最後に用法基盤/事例基盤モデルは次の重要な含意をもつことを指摘して本論文を終えることにしたい: (3) で特徴づけた記憶ベースの言語知識と処理のモデル化が正しいならば, i) 言語の創造性は豊かな記憶の産物の随伴事象である; ii) 表層形に関するスキーマの知識 (e.g., コロケーション) が深層にあると想定される概念構造と同じ位か, あるいはそれよりも重要である .

## 参考文献

- [1] N. Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA, 1965.  
[2] C. M. Conway, M. R. Ellefson, and M. H. Christiansen. When less is less and when less is more: Starting small with staged input. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society*, pp. 270–275. Mahwah, NJ: Lawrence Erlbaum, 2003.  
[3] W. Daelemans and A. van den Bosch. *Memory-Based Language Processing*. Cambridge University Press, 2005.  
[4] J. L. Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, Vol. 48, No. 1, pp. 71–99, 1993.  
[5] A. D. Goldberg. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press, 1995.  
[6] J. Hutchins. Example-based machine translation: A review and commentary. *Machine Translation*, Vol. 19, pp. 197–211, 2005.  
[7] K. Kuroda. *Foundations of PATTERN MATCHING ANALYSIS*. PhD thesis, Kyoto University, Japan, 2000.  
[8] George A. Miller. The magical number seven, plus or minus two. *The Psychological Review*, Vol. 63, No. 2, pp. 81–97, 1956.  
[9] S. Sato and M. Nagao. Toward memory-based translation. In *Proceedings of COLING-90, Helsinki, Finland*, pp. 247–252, 1990.  
[10] 中本敬子, 李在鎬, 黒田航. 日本語の語順選好は動詞に還元できない文レベルの意味と相関する: 心理実験に基づく日本語の構文研究への提案. *認知科学*, Vol. 13, pp. 334–352, 2006. 「文理解」特集号.  
[11] 池原悟, 徳久雅人, 村上仁一, 佐佐木昌, 池田尚志, 宮崎正弘. 非線形な重文複文の表現に対する文型パターン辞書の開発. *情報処理学会研究報告*, Vol. NL-170, No. 25, pp. 157–164, 2005.  
[12] 黒田航. 徹底した方法基盤主義の下での文法獲得: 「極端に豊かな事例記憶」の仮説で描く新しい筋書き. *月刊言語*, Vol. 36, No. 11, pp. 24–34, 2007.  
[13] 黒田航. なぜ超語彙的パターンよりも重要なのか? <http://clsl.hi.h.kyoto-u.ac.jp/~kkuroda/papers/constructions-as-bias-sources.pdf>, 2008.  
[14] 黒田航, 李在鎬, 渋谷良方, 井原均. 複層意味フレーム分析 (の簡略版) を使った意味役割タグづけの現状. *言語処理学会 14 回大会発表論文集*, 2008.  
[15] 李在鎬. 他動詞のゆらぎ現象に対する「構文」的アプローチ. *言語科学論集*, Vol. 7, pp. 1–20, 2001.

<sup>9)</sup> Pattern Lattice の理論化の動機の一つはこれであった .

<sup>10)</sup> 例えば, 図 1 の 5-2 [彼は歌って . た] の空所に [踊っ] を補完するのは, 日本人を母語にする話者には特に難しいことではない . このような種の補完が暗黙に起っていることで構文効果が生じると説明すれば, 具体的な語に言及しない抽象的構文 (e.g., [N1 が N2 に V]) が移動の意味をもっている/エンコードしているという (過剰般化に繋がる) 想定はしなくて済む .

<sup>11)</sup> 第一著者が開発した Pattern Matching Analysis (PMA) [7] は, 記憶ベースの記述モデルの具現化の一つとして構想された .