

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4668363号
(P4668363)

(45) 発行日 平成23年4月13日(2011.4.13)

(24) 登録日 平成23年1月21日(2011.1.21)

(51) Int.Cl. F I
G06F 17/27 (2006.01) G06F 17/27 M

請求項の数 24 (全 34 頁)

(21) 出願番号	特願平10-504302	(73) 特許権者	500046438
(86) (22) 出願日	平成9年6月27日(1997.6.27)		マイクロソフト コーポレーション
(65) 公表番号	特表2000-513473 (P2000-513473A)		アメリカ合衆国 ワシントン州 9805
(43) 公表日	平成12年10月10日(2000.10.10)		2-6399 レッドモンド ワン マイ
(86) 国際出願番号	PCT/US1997/011160		クロソフト ウェイ
(87) 国際公開番号	W01998/000793	(74) 代理人	100077481
(87) 国際公開日	平成10年1月8日(1998.1.8)		弁理士 谷 義一
審査請求日	平成16年2月23日(2004.2.23)	(74) 代理人	100088915
審査番号	不服2007-16347 (P2007-16347/J1)		弁理士 阿部 和夫
審査請求日	平成19年6月11日(2007.6.11)	(74) 復代理人	100115624
(31) 優先権主張番号	08/674, 610		弁理士 濱中 淳宏
(32) 優先日	平成8年6月28日(1996.6.28)	(74) 復代理人	100115635
(33) 優先権主張国	米国 (US)		弁理士 窪田 郁大

最終頁に続く

(54) 【発明の名称】 シンタックスツリーから意味的論理形式を計算するための方法とシステム

(57) 【特許請求の範囲】

【請求項1】

自然言語の文に対し論理形式グラフを生成するための、中央演算装置およびメモリを有するコンピュータシステムにおいて実施される方法であって、前記文は、前記文の統語構成体を表すノードを有するシンタックスパーズツリーで表され、前記シンタックスパーズツリーは、前記メモリ中において、リンク構造により接続されるノードを有するデータ構造で表される、そのような方法において、

前記中央演算装置により、前記文中で暗黙に了解される任意の統語構成体に関して、前記メモリ中において、前記シンタックスパーズツリーに統語的役割を追加する段階と、

前記中央演算装置により、前記追加された統語的役割を伴う前記シンタックスパーズツリーに前記シンタックスパーズツリーの前記リンク構造を変更するかまたは前記シンタックスパーズツリーに新規のノードを追加するための意味規則の第1のセットを適用して、前記文の完全な統語的分析結果を表す修整されたシンタックスパーズツリーを形成する段階と、

前記中央演算装置により、前記シンタックスパーズツリーの前記データ構造とは別のデータ構造中に表されている骨格的論理形式グラフを、前記メモリ中において、前記修整されたシンタックスパーズツリーに対して生成する段階と、

前記中央演算装置により、前記メモリ中において、前記生成された骨格的論理形式グラフに対して意味ラベルを追加する段階と、

前記中央演算装置により、意味構成体を追加して前記論理形式グラフを完成させるため、

10

20

意味ラベルを伴う前記論理形式グラフに対し、識別された言語学的な事象に基づいて論理形式グラフを修正する意味規則の第2のセットを適用する段階とを含むことを特徴とする方法。

【請求項2】

前記文が、事前定義された単語の後にくる動詞を省略している場合、統語的役割を追加する前記段階が、省略された動詞に対する統語構成体を追加することを特徴とする請求項1に記載の方法。

【請求項3】

前記事前定義された単語が「to」であることを特徴とする請求項2に記載の方法。

【請求項4】

前記事前定義された単語が「not」であることを特徴とする請求項2に記載の方法。

【請求項5】

前記文が代名詞を失っている場合、統語的役割を追加する前記段階が、失われている代名詞に対する統語構成体を追加することを特徴とする請求項1に記載の方法。

【請求項6】

失われている前記代名詞が、命令文中の「you」であることを特徴とする請求項5に記載の方法。

【請求項7】

前記文が単語「and」または「or」のうちの1つを含む等位構造を含んでいる場合、前記中央演算装置による、統語的役割を追加する前記段階が、統語構成体を追加して前記等位構造を拡大することを特徴とする請求項1に記載の方法。

【請求項8】

前記等位構造が、単語「and」を含んでいることを特徴とする請求項7に記載の方法。

【請求項9】

前記等位構造が、単語「or」を含んでいることを特徴とする請求項7に記載の方法。

【請求項10】

前記中央演算装置による、前記シンタックスパーズツリーに対し意味規則の第1のセットを適用する前記段階が、前記シンタックスパーズツリーにおいて、前記文中において互いに分離しているが互いに隣接して配置された場合言語単位を形成する前記文の異なる部分を識別する段階を含んでいることを特徴とする請求項1に記載の方法。

【請求項11】

前記中央演算装置による、前記シンタックスパーズツリーに対し意味規則の第1のセットを適用する前記段階が、動詞句を前置詞句の目的語を有する動詞に変える段階を含んでいることを特徴とする請求項1に記載の方法。

【請求項12】

前記中央演算装置による、前記シンタックスパーズツリーに対し意味規則の第1のセットを適用する前記段階が、単語「it」を不定詞節で置き換える段階を含んでいることを特徴とする請求項1に記載の方法。

【請求項13】

前記中央演算装置による、前記骨格的論理形式グラフを生成する前記段階が、前記修整されたシンタックスパーズツリーの属性から抽出された属性をラベルとして使って前記骨格的論理形式グラフのノードをラベリングする段階を含むことを特徴とする請求項1に記載の方法。

【請求項14】

前記中央演算装置による、意味ラベルを追加する前記段階が、スピーチの深層部を示す意味ラベルを追加する段階を含むことを特徴とする請求項1に記載の方法。

【請求項15】

前記スピーチの深層部が主語であることを特徴とする請求項14に記載の方法。

【請求項16】

前記スピーチの深層部が目的語であることを特徴とする請求項14に記載の方法。

10

20

30

40

50

【請求項 17】

前記スピーチの深層部が間接目的語であることを特徴とする請求項 14 に記載の方法。

【請求項 18】

前記スピーチの深層部が叙述主格であることを特徴とする請求項 14 に記載の方法。

【請求項 19】

前記スピーチの深層部が補語であることを特徴とする請求項 14 に記載の方法。

【請求項 20】

前記スピーチの深層部が叙述形容詞であることを特徴とする請求項 14 に記載の方法。

【請求項 21】

自然言語で規定された単語の句に対して論理形式グラフを生成するための、中央演算装置およびメモリを有するコンピュータシステムにおいて実施される方法であって、前記自然言語は前記自然言語のシンタックスを規定する文法を有する、そのような方法において、前記中央演算装置により、前記メモリ中において、前記句の初期シンタックスパーstreeを前記自然言語の文法に基づいて生成する段階であって、前記初期シンタックスパーstreeが、リンク構造によって接続され前記句の前記単語の統語構成体を表すノードを含んでいる、そのような初期のシンタックスパーstreeを生成する段階と、

前記中央演算装置により、前記句の中で暗黙に了解される統語構成体に対して統語解析を完了させるため前記初期シンタックスパーstreeのノード群に対し前記シンタックスパーstreeの前記リンク構造を変更するかまたは前記シンタックスパーstreeに新規のノードを追加するための意味規則の第 1 のセットを適用して修整されたシンタックスパーstreeを生成する段階と、

前記中央演算装置により、前記メモリ中において、修整されたシンタックスパーstreeに対して骨格的論理形式グラフを生成する段階であって、前記骨格的論理形式グラフが、複数のリンクによってリンクされた複数のノードを有しかつ前記シンタックスパーstreeのデータ構造から独立したデータ構造中に表されている、そのような骨格的論理形式グラフを生成する段階と、

前記中央演算装置により、前記論理形式グラフを完成させるため、前記骨格的論理形式グラフに対し言語学的な事象を扱う意味規則の第 2 のセットを適用して意味構成体を追加する段階とを含むことを特徴とする方法。

【請求項 22】

前記初期シンタックスパーstreeに対し意味規則の第 1 のセットを適用する前記段階が、句の中で省略されるが人間たる聴手には暗黙に理解される任意の統語構成体に関して、前記メモリ中において、統語的役割を前記シンタックスパーstreeに追加する段階を含んでいることを特徴とする請求項 21 に記載の方法。

【請求項 23】

前記骨格的論理形式グラフに対して意味規則の第 2 のセットを適用する前記段階が、前記メモリ中において、前記生成された骨格的論理形式グラフのノード間の前記リンクに意味ラベルを追加する段階を含んでいることを特徴とする請求項 21 に記載の方法。

【請求項 24】

メモリを有する、自然言語の文に対して論理形式グラフを生成するためのコンピュータシステムであって、前記文が文の統語構成体を表すノードを有するシンタックスパーstreeで表され、前記シンタックスパーstreeが、リンク構造により接続されるノードを有するデータ構造で表される、そのようなコンピュータシステムにおいて、

前記文中で暗黙に了解される任意の統語構成体に関して統語的役割を前記シンタックスパーstreeに追加し、且つ前記追加された統語的役割を伴う前記シンタックスパーstreeの中のノードに対し前記シンタックスパーstreeの前記リンク構造を変更するかまたは前記シンタックスパーstreeに新規のノードを追加することにより当該シンタックスパーstreeが前記文の完全な統語分析結果を表すことになる意味規則の第 1 セットを適用するフェーズ 1 コンポーネントと、

前記メモリ中において、前記修整されたシンタックスパーstreeに対して骨格的論理形

10

20

30

40

50

式グラフを生成するフェーズ2コンポーネントであって、前記骨格的論理形式グラフが前記シンタックスパースツリーの前記データ構造からは独立したデータ構造で表されており、且つノードとリンクを有し、前記ノードは意味構成体に対応しかつ前記リンクは意味構成体間の関係に対応する、そのようなフェーズ2コンポーネントと、

前記メモリ中において、前記生成された骨格的論理形式に意味ラベルを追加し、且つ意味構成体を追加して前記論理形式グラフを完成させるために意味ラベルを伴う前記論理形式グラフに言語学的な事象を扱う意味規則の第2セットを適用するフェーズ3コンポーネントとを備え

前記コンポーネント群は、1または複数の中央演算装置および該中央演算装置上で動作する1または複数の所与のプログラムによって実現されることを特徴とするコンピュータシステム。

10

【発明の詳細な説明】

発明の技術的分野

本発明は自然言語処理（「NLP」）の分野に関しており、より詳しくは、シンタックスツリーから論理形式グラフを生成する方法とシステムに関する。

発明の背景

自然言語を自動的に処理するコンピュータシステムは、種々のサブシステムを用いて、大まかに言って、形態的、統語的、意味的な分析に関する言語学的分野に対応しつつ、入力されたテキスト（以下、入力テキスト）を分析し自然言語を機械的に理解するレベルを達成する。入力テキストをあるレベルまで理解し終わるとコンピュータシステムは、例えば

20

形態分析では入力された単語が識別され、自然言語の話し手である人間が辞書を使って決定できる各単語に対し、情報が提供される。こうした情報は、単語が果たす統語的役割（例えば、名詞又は動詞）と、異なっているが関連性のある単語を生成するために接頭語ないし接尾語を加えることで単語が変化し得る方法とを含んでいる可能性がある。例えば単語fishに加えて、辞書は、fishに関係する又はそこから派生するfishes、

fished、fishing、fisher、fisherman、fishable、fishability、fishbowl、fisherwoman、fishery、fishhook、fishnet、fishy、等の様々な単語を掲げるであろう

30

。統語分析では、各入力文を分析するがその出発点として、入力された単語を形態的に分析することで提供された情報と、入力文が書かれた言語の文法を定義するシンタックス規則のセットとを使用する。以下はシンタックス規則のサンプルである。

文 = 名詞句 + 動詞句

名詞句 = 形容詞 + 名詞

動詞句 = 副詞 + 動詞

統語分析では、シンタックス規則の整ったサブセットを見いだす試みがなされるが、この規則は入力文の単語に適用され、単語のグループを組合わせて句とし、句を組み合わせることで完全な文とするものである。例えばBig dogs fiercely bite.という入力文を考える。上記の単純な3つの規則を使うと、統語分析で、Bigとdogsはそれぞれ形容詞と名詞として識別され、第2規則を適用し名詞句Big dogsが生成される。統語分析で、単語fiercelyとbiteはそれぞれ副詞と動詞として識別され、第3規則を適用し動詞句fiercely biteが生成される。最終的な統語分析で、第2規則を適用し、先に生成された名詞句と動詞句から完全な文を形成することになる。統語分析された結果はパースと呼ばれ、しばしば非輪状下方分岐ツリーとして表され、このツリーは入力単語を表すノード、句読点記号、句、完全な文を表すルートノードを有している。

40

しかし文の中には、複数の異なるパースを持てるものがある。こうした複数パースの古典的文例は、Time flies like an arrow.である。この文の3つ

50

の可能な意味に対応し、少なくとも3つの可能なパースが存在する。第1のパースで、timeは文の主語、fliesは動詞、like an arrowは、動詞fliesを修飾する前置詞句である。しかし、少なくとも予想もしない2つのパースが同様に存在する。第2のパースで、timeはfliesを修飾する形容詞、likeは動詞、an arrowは動詞の目的語である。このパースは、ある種のflies、time fliesが矢を好むないし矢に魅惑されているという意味に対応する。第3のパースで、timeは命令形の動詞、fliesは目的語、like an arrowは動詞timeを修飾する前置詞句である。このパースは、矢が飛ぶ時間を計るように、蠅の飛ぶ時間をたぶんストップウォッチで計る命令に対応する。

統語分析は、シンタックスパースツリーと呼ばれる階層ツリーを少なくとも1つ構成することで達成される。シンタックスパースツリーの各リーフノードは、一般的に入力文の1単語又は句読点記号を表す。シンタックス規則を適用することにより、1つ、2つ、ないし時にはもっと多くの既存ノードへ下から連結される中間レベルのノードを生成することになる。既存ノードは最初、リーフノードのみから成るが、統語分析によりシンタックス規則が適用されるにつれ、既存ノードはリーフノードと中間レベルのノードの両者を含むようになる。1つの完全なシンタックスパースツリーから成るシングルルートノードは完全な文を表す。

意味分析では、入力テキストの意味を述べた論理形式グラフが、シンタックスパースツリーだけで記述できる以上に深い方法で、生成される。論理形式グラフは、言語の話し手である人間により達成されるレベルに類似したレベルで、入力テキストを理解する最初の試みである。

論理形式グラフはノードとリンクを有しているが、先述のシンタックスパースツリーとは異なり、階層的な整理がなされていない。論理形式グラフのリンクには、一对のノード間の関係を示すラベルが付いている。例えば、意味分析では文中のある種名詞が深層主語ないし動詞の深層目的語として識別されるかも知れない。ある動詞の深層主語は行動の行為者であり、ある動詞の深層目的語は動詞で特定された行動の目的語である。ある能動態動詞の深層主語は、文の統語的主語かも知れないし、ある能動態動詞の深層目的語は、当該動詞の統語的目的語であるかも知れない。しかし、受動態動詞の深層主語は、動作主を示すby句で表現されてもよく、受動態動詞の深層目的語は文の統語的主語であってもよい。2つの文例、(1) Dogs bite people. (2) People are bitten by dogs. を取り上げる。最初の文は能動態動詞、二番目の文は受動態動詞を有する。最初の文の統語主語はDogsで、動詞biteの統語目的語はpeopleである。対照的に、二番目の文の統語主語はPeople、動詞句are bittenは、動作主を示すby句by dogsで修飾される。両方の文で、dogsは深層主語であり、peopleは文の動詞ないし動詞句の深層目的語である。文1、2に対して統語分析で生成されたシンタックスパースツリーは上記のように異なってくるが、統語分析で生成された論理形式グラフは同じになり、その理由は2つの文の根底にある意味が同じだからである。

論理形式グラフを生成後さらに意味的处理をすると、より深いレベルの理解を達成するための分析済みのテキストを実世界の概念に関係づける知識データベースに近づくことになる。知識ベースの例は、オンライン百科事典であり、そこからより詳細な定義と文脈情報を特定の単語に関して得ることができる。

以下、3つの - 形態、統語、意味 - NLPサブシステムを、サンプルの入力文The person whom I met was my friend. の処理に関連して述べる。図1は、NLPのサブシステム間の情報の流れを示すブロック線図である。形態サブシステム101は入力テキストを受け取り、単語の識別結果と、各単語が使用され得るスピーチの様々な各部分に対する語義を出力する。統語サブシステム102はこの情報を受け取り、統語規則を適用してシンタックスパースツリーを生成する。意味サブシステム103は、シンタックスパースツリーを受け取り、論理形式グラフを生成する。

図2 - 5は、サンプル入力テキストの入力単語に関して、形態分析中に検索され、電子記

10

20

30

40

50

憶媒体上に記憶された辞書情報を示す。図2は、入力単語 `the 201`、`person 202` に関する辞書記載事項を示す。記載事項 `201` は、キーである `the 203` と属性/値のペアのリストから成る。最初の属性 `Adj 204` は、値として、中括弧 `205`、`206` 内に含まれた記号を有している。この記号は、更に2つの属性/値のペア、(1) `Lemma/the`、(2) `Bits/Sing Plur Wa 6 Det Art B0 Def` から成る。見出し語は、基本的かつ語尾変化しない形式の単語である。故に属性 `Lemma` は、本辞書の記載事項により、基本的かつ語尾変化しない形式の単語として表されることを示している。属性 `Bits` は、ある単語についてのある種の形態的かつ統語的な情報を表す1組の短縮語から成る。この情報は、`the` が(1)単数、(2)複数、(3)語尾変化できない、(4)限定詞、(5)冠詞、(6)通常形容詞、(7)限定的、であることを示している。属性 `204` は単語 `the` が形容詞として働けることを示す。属性 `212` は単語 `the` が副詞として働くことを示す。属性 `Senses 207` では、単語の様々な意味が独立した定義と用例として表され、その一部が中括弧 `208 - 209`、中括弧 `210 - 211` の間の属性/値のペアのリストの中に含まれている。`the` の記載事項に実際に含まれた追加の意味が図2では省かれ、括弧表現 `more sense records 213` によって示される。

自然言語処理の第1ステップで、形態サブシステムは入力テキストの各単語と句読点記号を、別々のトークンとして認識し、各トークンのスピーチの各部に対して属性/値レコードを、辞書情報を使って組立てる。属性とは、特定の属性に対して定義された様々な値の内の1つを有することのできるレコード内のフィールドである。属性/値レコードは更なる処理用に、統語サブシステムへ引き渡され、統語サブシステムが組み立てるシンタックスパーズツリーのリーフノードとして使われる。シンタックスパーズツリーの全ノード及び後続 `NLP` サブシステムで組み立てられる論理形式グラフは、属性/値レコードである。

統語サブシステムは、形態サブシステムから引き渡されたリーフノードに、統語規則を適用し、サンプル入力テキストを表す可能なシンタックスパーズツリーに関してよりレベルの高いノードを組み立てる。完全なシンタックスパーズツリーは、ルートノード、中間レベルノード、リーフノードを含んでいる。ルートノードはサンプル入力テキストに対する統語構成体(平叙文)を表す。中間レベルノードは、中間構成体(動詞、名詞、又は前置詞句)を表している。リーフノードは、属性/値レコードの初期セットを表す。

`NLP` システムの中には、シンタックス規則をトップダウンで適用するものがある。ここに述べる `NLP` システムの統語サブシステムは、リーフノードにシンタックス規則をボトムアップで適用する。即ち、統語サブシステムは、シンタックス規則を1つずつ単一リーフノードに、ペアのリーフノードに、時には大きなグループになったリーフノードに適用しようとする。統語規則が2つのリーフノードを作動対象として要求し、かつ一対のリーフノードの両方もが規則に定められた要件を満たす属性を含んでいる場合、規則が両者に適用されよりレベルの高い統語構成体が作り出される。例えば、単語 `my friend` はそれぞれ形容詞と名詞を表すが、名詞句としてよりレベルの高い統語構成体へと組み合わせることができる。文法規則、「名詞句 = 形容詞 + 名詞」に対応するシンタックス規則により、中間レベルの名詞句ノードが作り出され、`my` と `friend` を表す2つのリーフノードを連結させ、新たに作り出された中間ノードへと至る。新しい各中間レベルノードは作り出されると、既に存在するリーフノードや中間レベルノードと連結されて、シンタックス規則が適用されるノードの全セットの一部となる。シンタックス規則を増大するノードのセットに適用するプロセスは、完全なシンタックスパーズツリーが生成されるまでないしはこれ以上シンタックス規則が適用できなくなるまで続くことになる。完全なシンタックスパーズツリーは、入力文の全単語をリーフノードとして含んでおり、当該文の1つの可能なパースを表している。

統語構文解析の本ボトムアップ方法は、最終の完成したシンタックスパーズツリーには決して含まれないであろう多くの中間レベルノードとサブツリーを作り出す。更には、本構文解析方法は、完全なシンタックスパーズツリーを複数同時に生成することもできる。

10

20

30

40

50

統語サブシステムでは、追加規則が適用できなくなるまで規則を適用し続けることにより、可能な完全なシンタックスパーstree全体に対して徹底的な探索を行うことが可能である。統語サブシステムでは、最も可能性のあるノードを最初に生成するため、種々の自学自習的手法を試みることも可能である。1個ないし数個の完全なシンタックスパーstreeが生成された後には、統語サブシステムでは通常探索を終了することができる。何故なら、入力文を最もよく表すとして選ばれる最も見込みのあるシンタックスパーstreeは恐らく、最初に生成されるシンタックスパーstreeの1つだからである。完全なシンタックスパーstreeが合理的な探索後に生成されない場合、特別な集成規則を適用して生成されるルートノードを使い、最も見込みのあるサブtreeを組み合わせてシングルtreeにすることで、適合する構文解析を達成することができる。

10

図6は、図2-5に示した辞書記載事項に対して、統語サブシステムが作り出した最初のリーフノードを示している。リーフノードには2個の特別なノード601、614が含まれ、それぞれ文の開始と文を終了させるピリオドを表している。602-613の各ノードは、入力単語が文中で表すことのできるスピーチの単一部分を表している。スピーチのこれらの部分は、辞書記載事項中の属性/値のペアとして見いだされる。例えば、リーフノード602、603は単語Theに関するスピーチの2個の可能部分を表しており、図2の属性204、212として見いだされる。

図7-22は、統語サブシステムにより規則を1個づつ適用して組み立てた最終のシンタックスパーstreeを示す。各図は1つの統語規則を適用して、統語構造を表す中間レベルノードを生成する様子を示す。最終シンタックスパーstreeを構成する中間レベルノードを作り出す規則のみを示す。統語サブシステムは、最終のシンタックスパーstreeに含まれずに終わる多くの中間レベルノードを生成する。

20

図7-14では、統語サブシステムにより単純な動詞、名詞、形容詞句を表す中間レベルノードを作り出す1要素統語規則が適用される。図15を皮切りに、統語サブシステムは、動詞、名詞、形容詞句を組み合わせて複数単語の統語構成体にする2要素統語規則を適用開始する。統語サブシステムは適用した場合の成功しそうな可能性に従って規則を発する。統語サブシステムは既存ノードにうまく適用できる規則が見つかるまで、一つ一つ規則を当てはめようとする。例えば図15に見るように、統語サブシステムは、形容詞句と名詞句から名詞句を表すノードを作り出す規則を、首尾良く当てはめている。本規則は、形容詞句と名詞句に必要な特性を定めている。本例の場合、形容詞句は限定詞でなければならない。統語サブシステムがノード1501からノード1503へポイントを逆にたどり、1503に含まれている形態情報にアクセスすることで、ノード1501が限定詞を表していると統語サブシステムは決定する。規則の要求特性を満たす2つのノード1501、1502を置くことで、統語サブシステムは本規則を適用し、2つの単純なフレーズ1501、1502から、名詞句my friendを表す中間レベルノードを作り出す。図22の統語サブシステムは、Begin1という特別なリーフノード2201と、The person whom I met was my friendという動詞句2202と、最後の終了ピリオドを表すリーフノード2203とを組み合わせる3要素規則を適用することにより入力文を表した最終的かつ完全なシンタックスパーstreeを生成し、叙述文を表すノード2204を形成する。

30

40

意味サブシステムは、完全なシンタックスパーstreeから、論理形式グラフを生成する。いくつかのNLPシステムの場合、論理形式グラフはシンタックスパーstreeのノードで構成され、それらのノードに属性と新しい双方向のリンクが加わっている。論理形式グラフはラベル付きで方向性のあるグラフである。論理形式グラフは入力文の意味を表している。各単語に対し形態サブシステムで得られる情報は、シンタックスパーstreeのリーフノードを参照することで、論理形式グラフのノード内から依然として利用可能である。論理形式グラフのリンクの方向とラベルは、論理形式グラフのノードに対する機能的役割を始めとする意味情報を表している。分析中の最中、意味サブシステムは、(1)省略されているが暗黙に了解される単語、(2)動詞句に関して失われているないし不明確な推論と付加詞、(3)前置詞句が言及する目的語、を表すためのリンクとノードを追加

50

する。

図 2 3 は、入力例文に対して意味システムが生成した、完全な論理形式グラフを示す。意味規則を首尾良く当てはめた結果として、意味のあるラベルがリンク 2 3 0 1 - 2 3 0 6 に割り当てられている。2 3 0 7 - 2 3 1 2 の 6 つのノードと、それらの間のリンクが文の意味論的な意味の本質的なコンポーネントを表している。論理形式リンクは一般的に、入力単語におおよそ対応するが、意味論的な意味を伝えるのに必要でないある種の単語、例えば The、whom は論理形式グラフには登場せず、入力動詞、met、was はその不定詞形、meet、be で登場する。ノードはコンピュータシステム中でレコードとして表され、図 2 3 が示さない追加情報を含んでいる。動詞が単数過去形で入力された事実は、動詞 2 3 0 7、2 3 1 0 に対応する論理形式ノード内の追加情報で示される。

シンタックスパースツリーと論理形式グラフの違いは、図 2 3 と図 2 2 の比較から容易に明らかである。図 2 2 に示されたシンタックスパースツリーは、厳密な階層状態に連結された 1 0 個のリーフノードと 1 6 個の中間レベルノードを含んでいるのに対し、図 2 3 に示されたシンタックスパースツリーは、6 個のノードしか含まない。シンタックスパースツリーとは違って、論理形式グラフは階層的に整理されてないことが、ノード 2 3 0 7、2 3 0 8 間で互いに反対方向を向く 2 本のリンクからも明らかである。更に上記から分かるように、ノードはもはや入力単語の正確な形式を表さず、代わりにそれらの意味を表している。

10

後続の自然言語処理のステップは、意味分析後に起こる。そのステップには、知識ベースから得られる追加情報に論理形式グラフを組み合わせる段階と、文のグループを分析する段階と、一般的には、人間が自然言語を処理する文脈環境にほぼ近い豊かな文脈環境を各論理形式グラフの周辺に組み立て試行する段階を含んでいる。

20

論理形式グラフを生成する従来技術には、シンタックスパースツリーに対し、計算上複雑な調整と操作が伴っている。結果的に、NLP システムに新しい意味規則を加えることはますます困難となっている。新しい規則を加えることは、意味システム中に既にプログラム化されている手順論理に対して矛盾する可能性のある新たな手順論理を伴うことになる。更には、シンタックスパースツリーは拡張され、論理形式グラフのノードとして再使用されるので、従来技術の意味サブシステムでは、大量かつ煩雑かつ複雑なデータ構造を産み出すことになる。論理形式グラフのサイズと複雑さがシンタックスパースツリー上に重ね合わされるため、組み合わされたデータ構造をこれ以上使うと、エラーが起きやすく非効率的な事態となる。従って、もっと拡張性がよく管理し易い意味サブシステムを手に入れ、単純な論理形式グラフのデータ構造を作れるようにするのが望ましい。

30

発明の概要

本発明は、NLP システム内で入力文を意味分析するための方法とシステムを対象としている。意味分析サブシステムは、形態サブシステムと統語サブシステムとが生成したシンタックスパースツリーを受け取る。意味分析サブシステムは 2 セットの意味規則を適用し、受け取ったシンタックスパースツリーに対し調整を行う。意味分析サブシステムはその後、第 3 セットの意味規則を適用し、シンタックスパースツリーから骨格的論理形式グラフを作り出す。意味分析サブシステムは最後に、2 セットの追加的な意味規則を骨格的論理形式グラフに適用し、骨格的論理形式グラフのリンクに対し意味論的に意味のあるラベルを提供し、失われているノードに対し追加の論理形式グラフノードを作り出し、冗長な論理形式グラフのノードを統合する。意味分析システムにより生成された最終的な論理形式グラフは、入力文を完全に意味的に分析した結果を表す。

40

【図面の簡単な説明】

図 1 は、NLP システムのサブシステム間の情報の流れを示すブロック線図である。

図 2 - 5 は、入力例文「The person whom I met was my friend」の各単語に関して検索され、電子記憶媒体上に記憶された辞書情報を示す。

図 6 は、入力文を構文解析する際の第 1 ステップとして、統語サブシステムが生成したリーフノードを示す。

50

図 7 2 2 は、統語サブシステムが統語規則を継続して適用して、入力文を構文解析してシンタックスパースツリーを作り出している経緯を示す。

図 2 3 は、入力文の意味を表すため意味サブシステムが生成した論理形式グラフを示す。

図 2 4 は、自然言語処理用の好適なコンピュータシステムのブロック線図を示す。

図 2 5 は、好適な新しい意味分析サブシステムの 3 つのフェーズを示す。

図 2 6 は、新たな意味分析サブシステム (NSS) の流れ線図である。

図 2 7 は、第 1 セットの意味規則を示す。

図 2 8 A は、第 1 セットの意味規則の内の意味規則 P r L F _ _ Y o u の詳細な記述を示す。

図 2 8 B は、第 1 セットの意味規則の内の意味規則 P r L F _ _ Y o u の適用例を示す。

10

図 2 9 は、第 2 セットの意味規則を示す。

図 3 0 A - 3 0 B は、第 2 セットの意味規則の内の意味規則 T r L F _ _ M o v e P r o p の詳細な記述を示す。

図 3 0 C は、第 2 セットの意味規則の内の意味規則 T r L F _ _ M o v e P r o p の適用例を示す。

図 3 1 は、a p p l y _ _ r u l e s の流れ線図を示す。

図 3 2 は、NSS のフェーズ 1 の流れ線図を示す。

図 3 3 は、第 3 セットの意味規則を示す。

図 3 4 A - C は、第 3 セットの意味規則の内の意味規則 S y n T o S e m 1 の詳細な記述を示す。

20

図 3 4 D は、第 3 セットの意味規則の内の意味規則 S y n T o S e m 1 の適用例を示す。

図 3 5 は、NSS のフェーズ 2 の流れ線図を示す。

図 3 6 - 3 8 は、第 4 セットの意味規則を示す。

図 3 9 A は、第 4 セットの意味規則の内の意味規則 L F _ _ D o b j 2 の詳細な記述を示す。

図 3 9 B は、第 4 セットの意味規則の内の意味規則 L F _ _ D o b j 2 の適用例を示す。

図 4 0 は、第 5 セットの意味規則を示す。

図 4 1 A - C は、第 5 セットの意味規則の内の意味規則 P s L F _ _ P r o n A n a p h o r a の詳細な記述を示す。

図 4 1 D は、第 5 セットの意味規則の内の意味規則 P s L F _ _ P r o n A n a p h o r a の適用例を示す。

30

図 4 2 は、NSS のフェーズ 3 の流れ線図を示す。

図 4 3 は、NSS 用のコンピュータシステムのブロック線図である。

図 4 4 - 5 9 は、入力例文用に生成されたシンタックスパースツリーを NSS が処理するにつれ、首尾良く各規則を適用したことを示す。

発明の詳細な説明

本発明はシンタックスパースツリーから論理形式グラフを生成するための新たな意味的方法とシステムを提供する。ある好適実施例では、新しい意味サブシステム (NSS) は意味分析を、(1) シンタックスパースツリーを埋めて、調整する段階、(2) 初期の論理形式グラフを生成する段階、(3) 論理形式グラフのランクに対し意味のあるラベルを生成し、完全な論理形式グラフを組み立てる段階、の 3 つのフェーズに亘って実行する。各フェーズでは、1 組のシンタックスツリーノード又は 1 組の論理形式グラフのいずれかに対して、1 つ又は 2 つのセットの規則が適用される。

40

NSS は、背景の項で先に述べた従来技術による意味サブシステム中で認識された欠落部分に取り組む。NSS の各フェーズは拡張性の効く規則をベースとした単純な方法である。追加の言語学的な事象が認められた場合は、NSS が使用する規則のセットの一つにそれらの事象を扱う規則を容易に含ませることができる。更に NSS の第 2 のフェーズでは、既存のシンタックスパースツリー上に論理形式グラフを重ね合わせるというよりむしろ、全く独立した論理形式グラフが生成される。従って、NSS が生成した論理形式グラフのデータ構造は、従来技術による論理形式グラフのデータ構造に比べ、単純でスペースの

50

効率もよい。

図24はNLPシステム用の好適なコンピュータシステムを示すブロック線図である。コンピュータシステム2401は、中央演算装置、メモリ、記憶装置、入出力装置を有している。NLPサブシステム2406-2409は通常、ディスクのようなコンピュータで読み取り可能な記憶装置からメモリ2404にロードされる。NLPシステムが提供したサービスを使うアプリケーションプログラム2405も通常メモリにロードされる。電子辞書2411はディスク2410のような記憶装置に記憶され、記載事項は、形態サブシステムでの使うため、メモリに読み込まれる。ある実施例の場合、ユーザーは、入力装置2404上に少なくとも1つの自然言語を記入することにより、出力装置2403上に表示されたプロンプトに回答する。自然言語文は、アプリケーションプログラムによって受け取られ、処理され、その後形態サブシステム2406を通じてNLPシステムに引き渡される。形態サブシステムは電子辞書の情報を使って各入力単語を説明するレコードを組み立て、そのレコードを統語サブシステム2407へ引き渡す。統語サブシステムは、入力単語を構文解析し、シンタックスパーズツリーを組み立て、そのシンタックスパーズツリーを意味サブシステム2408へ引き渡す。意味サブシステムは、受け取ったシンタックスパーズツリーから論理形式グラフを生成し、その論理形式グラフを他のNLPサブシステム2409へ引き渡す。アプリケーションプログラムは次に、入力テキストに関してNLPシステムが達成した機械的な理解を利用するために、自然言語サブシステム2409との間で情報を送受信できるようになり、最終的にユーザーに対する回答を出力装置2403上に出力する。

10

20

図25は、好適な新しい意味サブシステムの3つのフェーズを示す。NSSのフェーズ1-3をそれぞれ2502、2504、2506として示す。NSSの各フェーズで入出力されるデータ構造の状態を、ラベル2501、2503、2505、2507として図25に示す。NSSは、統語サブシステムが生成したシンタックスパーズツリー2501を受け取る。NSS2502の第1フェーズは、意味規則を使ってシンタックスパーズツリーを完成させ、完成したシンタックスパーズツリー2503をNSS2504の第2フェーズに引き渡す。NSSの第2フェーズは初期の論理形式グラフ2505を生成し、該論理形式グラフをNSS2506の第3フェーズに引き渡す。NSSの第3フェーズでは、論理形式グラフのリンクへ意味のあるラベルを追加し、入力文の意味的な表示を埋めるためにリンクとノードを新たに追加し、時には冗長なノードを除去するために、意味規則が最初の論理形式グラフに適用される。次に、完全な論理形式グラフ2507は他のNLPサブシステムに引き渡され、論理形式グラフで表される入力文の更なる解釈、疑問への回答、入力文に基づいたデータの準備に使用される。

30

NSSの流れ線図を図26に示す。本図はNSSの3つのフェーズ2601、2603、2605が連続して呼び出されていることを示す。以下、NSSの各フェーズにつき詳しく述べる。

NSSフェーズ1 - シンタックスツリーの意味的役割を完成させる

NSSのフェーズ1でNSSは、シンタックスパーズツリーのノードに2組の異なる意味規則を適用し、統語サブシステムから受け取ったシンタックスパーズツリーを変更する。これらの規則により、シンタックスツリーのリンク構造を変えるか、ないしは新たなノードを加えることができる。

40

NSSは統語分析では扱うことのできない種々の可能性のある欠落や省略を解明するために第1セットの意味規則を適用する。第1セットのこれら意味規則を適用すると、入力されたシンタックスパーズツリーに予備的な調整を行うことになる。第1セットの意味規則で取り扱われる言語学的な事象には、単語「to」又は「not」の後で省略されるが人間たる聴手には暗黙に理解される動詞と、命令文中の「you」又は「we」のような失われている代名詞と、単語「and」又は「or」を含む等位構造の拡張と、失われている目的語又は省略された動詞句とが含まれる。図27に、フェーズ1でNSSにより適用される好適な第1セットの意味規則を掲げる。各規則に関して、規則名の後に簡潔な言語学的な事象を示しておく。

50

各意味規則の一般的な形式は、シンタックスパーズツリーのノードないし論理形式グラフのノードに適用される条件のセットと、シンタックスパーズツリー又は論理形式グラフに適用されるアクションのリストである。例えば、NSSは、シンタックスパーズツリーを表すシンタックスレコードのリストに対して第1セットの意味規則内の各規則の条件を適用するが、当該規則の全条件が満たされる各規則に対して、NSSは当該規則に含まれているアクションリストを実行し、結果的にシンタックスパーズツリーに特定の変化が起こる。当然、各意味規則の実際の形式は、シンタックスパーズツリーと論理形式グラフの表示に関する細則に依存するが、これらにシンタックスパーズツリーと論理形式グラフに関しては多くの異なる表示が可能である。以下の図で意味規則を条件付表現を用いて述べるが、この表現では、太活字の単語「If」を先行させ、太活字の単語「Then」を後続させる。意味規則の「If」部分は条件を表し、その条件は、シンタックスパーズツリーノード又は論理形式グラフノードに対して適用され、且つそのノードに適用される当該規則に関し全体として真であると判定されなければならない。また、「Then」表現は、シンタックスパーズツリー又は論理形式グラフ上で実行されるべきアクションのリストを表している。表示されている表現は、当該意味規則に関するコンピュータのソースコード表現と密接に対応している。

10

図28Aは、意味規則の第1セットの内の意味規則PrLF_Youを英語で表現したものである。図28Aで分かるように、「If」表現は、規則が適用されるシンタックスパーズツリーノードの様々な属性の値に関係しており、「Then」表現は、見出し語youに対する代名詞ノードの作り方、当該代名詞ノードに対し親の役をする名詞句ノード、及び作り出されたノードのシンタックスパーズツリーへの取り付け方について規定している。

20

図28Bは、文「Please close the door.」に対して統語サブシステムが生成したシンタックスパーズツリー2801に、意味規則PrLF_Youを適用した例を示す。PrLF_Youを適用した結果、修正されたシンタックスパーズツリー2802となり、2つの新しいノード2803、2804が文のルートノードに接続されている。この意味規則には、命令文で了解済みの「you」を明確にシンタックスパーズツリー中に置く目的がある。

入力シンタックスパーズツリーに対し適用し得る第1セットの意味規則全てが適用済みとなると、NSSは、予備的に調整されたシンタックスパーズツリーのノードに対して、第2セットの意味規則を適用することにより、最も重要な調整を予備的に調整されたシンタックスパーズツリーに行う。この第2セットの意味規則には、遠距離付属事象を識別し解明すること、動詞句を前置詞句付きの動詞に変形すること、ある種のケースで単語「it」を不定詞節に置き換えることという働きをする規則が含まれている。

30

図29に、フェーズ1でNSSにより適用される好適な第2セットの意味規則を掲げる。各規則に関して、規則名の後に簡潔な言語学的な事象を示しておく。図30A-30Bは、意味規則の第2セットの内の意味規則TrLF_MovePropを英語で表現したものである。図30A-30Bから分かるように「If」表現は、規則が適用されるシンタックスパーズツリーノードの様々な属性の値及び様々な関連するシンタックスパーズツリーノードに関係し、「Then」表現は、シンタックスパーズツリーに関するいくぶん複雑な再配置を規定している。

40

図30Cは文「I have no desire to see the man.」に対して統語サブシステムが生成したシンタックスパーズツリー3001に、意味規則TrLF_MovePropを適用した例を示す。PrLF_Youを適用した結果、修正されたシンタックスパーズツリー3002となる。元々のシンタックスパーズツリー中でノード3003によって表されていた不定詞節はその位置が、ノード3004の子供の位置から、修正されたシンタックスパーズツリーのルートDECL1ノードの子供3005の状態へと動かされている。この意味規則は、不定詞節3003のような節を、シンタックスパーズツリー中で低いレベルから高い位置に移動させて、後続でのシンタックスパーズツリーから論理形式グラフへの移行を容易にする目的を有している。

50

本発明の本好適実施例では、意味規則はプログラム言語中のステートメントであり、本ステートメントが実行されると、1つ又は2つ、時にもっと多くの既存のツリーノード又はグラフノードから、新しいツリーノード又はグラフノードが作り出され、新しく作り出されたツリーノード又はグラフノードと既存のツリー又はグラフノードとの間に適当なリンクが作り出される。本好適実施例の場合、意味規則の左側部分は、適用される規則に関して、既存のノード（複数の場合を含む）が有すべき特性を規定している。意味規則の左側部分は、作り出されるべき新しいノードのタイプと、新しいノードの属性の値を規定している。図28と図30に示した規則がこの形式を例示している。

本発明の本好適実施例の場合、各シンタックスパースツリーと各論理形式グラフはノードのリストとして表され、ノード内には、属性値によって表されたノードの間のリンクがある。シンタックスパースツリーに規則を適用することは、連続したノードをノードのリストから選択することと、規則のセットを表す規則リストの各規則を、選択された各ノードに対して、適用しようとするを含んでいる。あるノードがある規則の左側部分に規定された特性を有している場合、当該ノードにその特定の規則を首尾良く適用することができる。時に、規則を首尾良く適用した結果として、新たなノードを作り出してよく、削除するため既存のノードに印付けをしてもよい。

図31にサブルーチン `apply_rules` の流れ線図を示すが、本サブルーチンは、シンタックスパースツリー又は論理形式グラフを表すノードのリストに対して規則セットを適用する。サブルーチン `apply_rules` は、3つのフェーズの各々の最中に、各セットの規則を適用するために、NSSによって呼び出される。ステップ3101で、`apply_rules` は、第1の推論としてのノードのリストと、第2の推論としての規則のリストを受け取る。ステップ3102 - 3107は外部ループを表しており、ループの各々を繰り返し、各規則の入力リストにある入力規則全てを、入力リストから選択された連続するノードに適用しようとする。ステップ3103 - 3106は内部ループを表しており、ループの各々を繰り返して、入力規則のリストから選択されたある入力規則を、ノードの入力リストから選択されたあるノードに適用しようとする。ステップ3102で、`apply_rules` はノードの入力リストから、最初のノードで始まる次のノードを選択する。ステップ3103で、`apply_rules` は規則の入力リストから、最初の規則で始まる次の規則を選択する。ステップ3104で、`apply_rules` は、選択された規則の左側部分に規定された特性を選ばれたノードが有しているかどうかを決定する。規定された特性をノードが有していれば、ステップ3105で、`apply_rules` は選択された規則を、選択されたノードに対して適用する。`apply_rules` が、選択されたノードに対して適用すべき規則がもっとあることをステップ3106で決定した場合、`apply_rules` はステップ3103に戻り、次の規則を選択する。`apply_rules` が、入力規則リストの規則を適用しようとするノードがもっとあるとステップ3107で決定した場合、`apply_rules` はステップ3102に戻り、次のノードを選択する。

NSSの第1フェーズでなされる処理の流れ線図を図32に示す。ステップ3201で変数「パラメーター1」が、シンタックスパースツリーノードのリストとされ、このノードはシンタックスパースツリーにより生成されNSS中に入力されるシンタックスパースツリーから成る。ステップ3202で変数「パラメーター2」が、図27に示されている第1セットの意味規則のリストとされる。ステップ3203でNSSは、サブルーチン `apply_rules` を呼び出し、変数「パラメーター1」と「パラメーター2」を当該サブルーチンに引き渡す。サブルーチン `apply_rules` は、第1セットの意味規則をシンタックスパースツリーに適用し、予備的な調整を達成する。ステップ3204で変数「パラメーター1」が、予備的な調整がなされたシンタックスパースツリーから成るシンタックスパースツリーノードのリストとされる。ステップ3205で変数「パラメーター2」が、図29に示されている第2セットの意味規則のリストとされる。ステップ3206でNSSは、サブルーチン `apply_rules` を呼び出し、変数「パラメーター1」と「パラメーター2」を当該サブルーチンに引き渡す。サブルーチン `apply_`

10

20

30

40

50

r u l e s は、第2セットの意味規則をシンタックスパーズツリーに適用し、予備的な調整を達成する。

N S S フェーズ2 - 初期の論理形式グラフを生成する

N S S のフェーズ2でN S S は、調整済みのシンタックスツリーのノードに第3セットの意味規則を適用する。フェーズ2で規則が首尾良く適用される度に新たな論理形式グラフノードが作り出される。この第3セットの意味規則を適用することにより、N S S は新しい論理形式グラフを作り出す。論理形式グラフのノードは意味論的に意味のある属性と、対応するシンタックスツリーノードへ逆に戻るポイントだけから成る。従来技術による意味サブシステムと異なり、フェーズ2でN S S が生成する論理形式グラフノードは、完全に独立しておりシンタックスパーズツリーノードから識別ができる。N S S はリンクから成る論理形式グラフの骨格を組み立てるが、このリンクは、論理形式グラフのノードを互いに連結し、ノード内に属性として記憶される。

10

図33に、フェーズ2でN S S によって適用される第3セットの意味規則を示す。各規則について、規則名の後に簡潔な言語学的な事象を示しておく。本第3セットの意味規則には、3つの規則しかなく、最初の規則のS y n T o S e m 1 が普通使われる。2番目と3番目の規則は、適当な構文解析が統語サブシステムで生成されずそれ故に調整されたシンタックスパーズツリーが適当なパースノードを含んでいるという特別な状況のみに対して適用される。

図34A - 34Cは、意味規則の第3セットの内の意味規則S y n T o S e m 1 を英語で表現したものである。図34A - 34Cで分かるように、「I f」表現は、規則が適用されるシンタックスパーズツリーノードの様々な属性の値及び様々な関連のシンタックスパーズツリーノードに関係しており、「T h e n」表現は、論理形式グラフノードの作り出し方と、初期の論理形式グラフ内での新しいノードの置き方を規定している。

20

図34Dは、文「T h e b o o k w a s w r i t t e n b y J h o n .」に関して統語サブシステムが生成したシンタックスパーズツリー3401に、意味規則S y n T o S e m 1 を適用した例を示す。S y n T o S e m 1 を適用した結果は、骨格的な論理形式グラフ3402となる。骨格的論理形式グラフは、リンクをラベル付けする一時的な修飾語付きの3つのノードを有する。属性は、ノードが作り出されたシンタックスパーズツリーノードの統語属性に基づいて新たなノードに既に割り当てられている。論理形式グラフ中には、対応するシンタックスパーズツリー中に比べはるかに少ないノードしかなく、その理由は、論理形式グラフが当該文の意味論的意味を表しているからである。原文の単語、t h e、w a s、b y の言語学的な重要性は、論理形式グラフの属性とラベル中に組み込まれており、こうした単語がシンタックスパーズツリー中のリーフノードとしての存在するため生じた複雑なノード階層は、論理形式グラフでは必要でなくなる。

30

図35はN S S のフェーズ2の流れ線図である。ステップ3501で変数「パラメーター1」が、調整済みシンタックスパーズツリーを表すリストのノードとされる。ステップ3502で変数「パラメーター2」が、図33に示されている第3セットの意味規則のリストとされる。ステップ3503でN S S はサブルーチンa p p l y _ r u l e s を呼び出し、第3セットの意味規則を調整済みシンタックスパーズツリーのノードに適用し、これにより、調整済みシンタックスパーズツリーに対応する新しい論理形式グラフを作り出す。

40

N S S フェーズ3 - 論理形式グラフを完成させる

N S S のフェーズ3で、N S S は第4セットの意味規則を骨格的な論理形式グラフに適用し、論理形式グラフのリンクに対して意味論的に意味のあるラベルを追加する。この新たなラベルには、深層主語(D s u b)、深層目的語(D o b j)、深層間接目的語(D i n d)、深層叙述主格(D n o m)、深層補語(D c m p)、深層形容詞(D a d j)が含まれる。図36 - 38に、フェーズ3でN S S によって適用される第4セットの意味規則を示す。各規則について、規則名の後に簡潔な言語学的な事象を示しておく。

図39Aは、第4セットの意味規則の内の意味規則L F _ D o b j 2 を英語で表現したものである。図39Aで分かるように、「I f」表現は、規則が適用されるシンタックスパ

50

ースツリーノードの様々な属性の値に関係しており、「Then」表現は、論理形式グラフノード中のリンクのラベリングを規定している。

図39Bは、文「The book was written by Jhon.」に対してNSSが生成したシンタックスパーズツリー3901に、意味規則LF_Dobj2を適用した例を示す。受動態の節を含む論理形式グラフにLF_Dobj2を適用すると、統語主語が行為の深層目的語として識別される。図39Bに示すように、この識別は、リンク3903を一時的修飾語から深層目的語の関係を示すラベル3904と改称することにより達成される。

フェーズ3の最終ステップとして、NSSは第5セットの意味規則を適用することで、論理形式グラフに対する最終的な調整を行う。このセットの規則は、関係代名詞を先行詞に結びつける、失われている代名詞を見つけ明確に含ませる、数の飛躍を解消する、失われている深層主語を提供する、人称代名詞の冗長例を統合する、意味分析の第1サブステップ中で拡大された等位構造を短縮するという役割を果たす。これらの規則は、代名詞（又は代用形）を取りあげ且つその代名詞が言及する名詞句を識別する問題も扱う。多くの場合、論理形式グラフが提供する情報レベルに関係する正確な名詞句を識別することは可能ではない。その場合は、最も可能性の高い候補のリストを作り出しておき、それ以上の処理は、もっと広い情報が使われているNLPシステム中のより後のステップまで延期する。図40に、フェーズ3でNSSによって適用される第5セットの意味規則を示す。各規則について、規則名の後に簡潔な言語学的な事象を示しておく。

図41A-41Cは、意味規則の第5セットの内の意味規則PsLF_PronAnaphoraを英語で表現したものである。図41A-41Cで分かるように、「If」表現は、規則が適用される論理形式グラフノードの様々な属性の値及び関連の論理形式グラフノードに関係しており、「Then」表現は、代名詞の省略された指示対象を表す論理形式グラフノードの追加について規定している。

図41Dは、文「Mary likes the man who came to dinner, and Joan likes him too.」に対してNSSが生成した論理形式グラフ4101に、意味規則PsLF_PronAnaphoraを適用した例を示している。PsLF_PronAnaphoraを、論理形式グラフの異なる部分中に指示対象付き代名詞ノードを含む論理形式グラフに適用すると、当該代名詞ノードが直接連結される新たなノードを追加させることになる。図41Dでは、PsLF_PronAnaphoraが適用されて新たなノード4103が追加され、ノード「he1」が「man」に言及することが示される。

NSSのフェーズ3でなされる処理の流れ線図を図42に示す。ステップ4201で、変数「パラメーター1」が、NSSのフェーズ2で生成された論理形式グラフから成る論理形式グラフノードのリストとされる。ステップ4202で変数「パラメーター2」が、図36-38に示されている第4セットの意味規則のリストとされる。ステップ4203でNSSは、サブルーチンapply_rulesを呼び出し、「パラメーター1」と「パラメーター2」をサブルーチンに引き渡す。サブルーチンapply_rulesは、第4セットの意味規則を論理形式グラフに適用し、論理形式グラフのリンクに意味論的に意味のあるラベルを追加する。ステップ4204で、変数「パラメーター1」が、ステップ4203で生成され意味のあるラベル付けをされた論理形式グラフから成る論理形式グラフノードのリストとされる。ステップ4205で、「パラメーター2」が、図40に示されている第5セットの意味規則のリストとされる。ステップ4206でNSSは、サブルーチンapply_rulesを呼び出し、「パラメーター1」と「パラメーター2」をサブルーチンに引き渡す。サブルーチンapply_rulesは第5セットの意味規則を論理形式グラフに適用し、最終的な調整を達成する。

図43は、NSSのコンピュータシステムのブロック線図である。コンピュータ4300は、意味規則4308-4308と規則適用エンジン4303の付いているメモリを有する。規則適用エンジンは中央演算装置の制御の下に第5セットの意味規則をシンタックスパーズツリー4301に適用し、対応する論理形式グラフ4302を生成する。シンタッ

10

20

30

40

50

クスパースツリーは形態サブシステムと統語サブシステムで生成するのが好ましいが、両システムをここには図示しない。シンタックスツリーと論理形式グラフは、人間である読者が入力文から得るであろう情報と類似した情報を必要とする後続の仕事を達成するのに使用することができる。例えば文法チェックプログラムは、入力文で述べられたことをより正確かつ簡潔に述べる新たなフレーズを入力文に対して示唆することもできる。又、コンピュータのOSは、入力文が述べる計算上の仕事を実行してもよい。更に別の例として、入力文に含まれている情報を、データベース管理システムで後日の検索のために、分類、記憶することもできる。

入力例文の意味的な処理

例文「The person whom I met was my friend.」のNSSによる一連の処理を以下の説明と図44-59により示す。NSSにより適用される各意味規則を、規則適用した結果を示しながら述べる。

フェーズ1の統語サブシステムからNSSに入力されたシンタックスパースツリーに対し、第1セットの意味規則の予備的な調整規則は、首尾良く適用されてはいない。第2セットの意味規則の1つの主要な調整規則を入力されたシンタックスパースツリーに適用する。図44は、シンタックスパースツリー400の入力された形式を示す。図44のシンタックスパースツリーは図22のシンタックスパースツリーより若干単純になっていることに注目されたい。NSSは、図29が規則1として示す意味規則TrLF_LongDist1をシンタックスパースツリーの関係詞節ノードRELC1、4401に対して首尾良く適用し、調整されたシンタックスパースツリー4402を生成する。意味規則TrLF_LongDist1を適用した結果、直接目的語の属性を名詞句ノード4403中に導き入れ、単語「whom」が句「I met」の直接目的語であることが示されることになる。英語では通常、動詞の直接目的語は動詞の後に続く。構文分解されてシンタックスツリー4400が作り出された文中では、「whom」が「I met」の後に続かないので、「whom」が「I」の直接目的語であるという事実は、統語規則を適用する際に識別されなかった。

第3のセットの意味規則の7つの規則が、NSSのフェーズ2で、首尾よく適用される。図45でNSSは、図33が規則1として示す意味規則SynToSem1を、シンタックスパースツリーの限定された代名詞ノードDET2、4501に適用し、論理形式グラフノード「my」4502を生成する。図46でNSSは、意味規則SynToSem1を、シンタックスパースツリーの名詞句ノードNP4、4601に首尾よく適用して、論理形式グラフノード「friend」4602と、一時的ラベル意味「Tmods」4604の付いたリンク4603を生成する。図47でNSSは、意味規則SynToSem1を、シンタックスパースツリーの名詞句ノードNP3、4701に首尾よく適用し、論理形式グラフノード「I」4702を生成する。図48でNSSは、意味規則SynToSem1を、シンタックスパースツリーの名詞句ノードNP2、4801に首尾よく適用し、論理形式グラフノード「whom」4802を生成する。図49でNSSは、意味規則SynToSem1を、シンタックスパースツリーの関係詞節ノードRELC1、4901に首尾よく適用し、論理形式グラフノード「meet」4902と、一時的意味ラベル「Tmods」4904の付いたリンク4903を生成する。図50でNSSは、意味規則SynToSem1を、シンタックスパースツリーの名詞句ノードNP1、5001に首尾よく適用して、論理形式グラフノード「person」5002と、一時的意味ラベル「Tmods」5004の付いたリンク5003を生成する。図51でNSSは、意味規則SynToSem1をシンタックスパースツリーの叙述分ノードDECL1、5101に首尾よく適用し、論理形式グラフノード「be」5102と、一時的意味ラベル「Tmods」5104の付いたリンク5103を生成する。こうしてNSSのフェーズ2が完了すると、骨格的論理形式グラフが作り出される。

第4のセットの意味規則の6つの規則が、NSSのフェーズ3で首尾よく適用される。図52でNSSは、図36が規則1として示す意味規則LF_Dsub1を、論理形式グラフノード「be」5201に適用し、リンクラベル「Dsub」5201と、一時的意味

10

20

30

40

50

ラベル「Tmods」5204の付いたリンク5203を生成する。図53でNSSは、図37が規則10として示す意味規則LF_Dnomを、論理形式グラフノード「be」5301に適用し、リンクラベル「Dnom」5302を生成する。図54でNSSは、図38が規則21として示す意味規則LF_Propsを、論理形式グラフノード「person」5401に適用し、リンクラベル「Props」5402を生成する。図55でNSSは、図36が規則1として示す意味規則D_Sub1を、論理形式グラフノード「meet」5501に適用し、リンクラベル「DSub」5502を生成する。図56でNSSは、図36が規則3として示す意味規則LF_Dobj1を論理形式グラフノード「meet」5601に適用することで、ノード「meet」をノード「whom」に連結するラベル付きリンク「Dobj」5603を生成する。図57でNSSは、図38が規則22として示す意味規則LF_Opsを、論理形式グラフノード「friend」5701に適用し、リンクラベル「Possby」5702を生成する。

10

第5のセットの意味規則の1つの規則が、NSSのフェーズ3で首尾よく適用される。図58でNSSは、図40が規則1として示す意味規則PsLF_RelProを、図56が5602として示す論理形式グラフノード「whom」に適用し、ラベル付きリンク「Dobj」5801を生成してノード「whom」を除去する。図59でNSSは、図40が規則10として示す意味規則PsLF_UnifyPronsを論理形式グラフに適用し、ノード「I」と「my」を単一のノードに合体させる。これは、NSSが首尾よく適用する最後の規則である。従って図59は、入力文「The person whom I met was my friend .」に対しNSSが生成した最終的かつ完全な論理形式グラフを表す。

20

以上、本発明を好適実施例との関係で述べてきたが、本発明はこれらの実施例に限られるものではない。本発明の精神の下での変更が当業者には明白である。本発明の範囲を以下の請求項により定義する。

【図1】

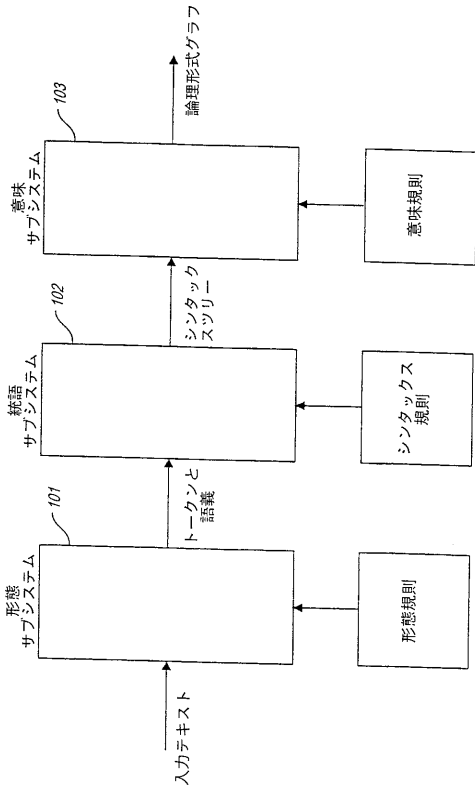


Fig. 1

【図2】

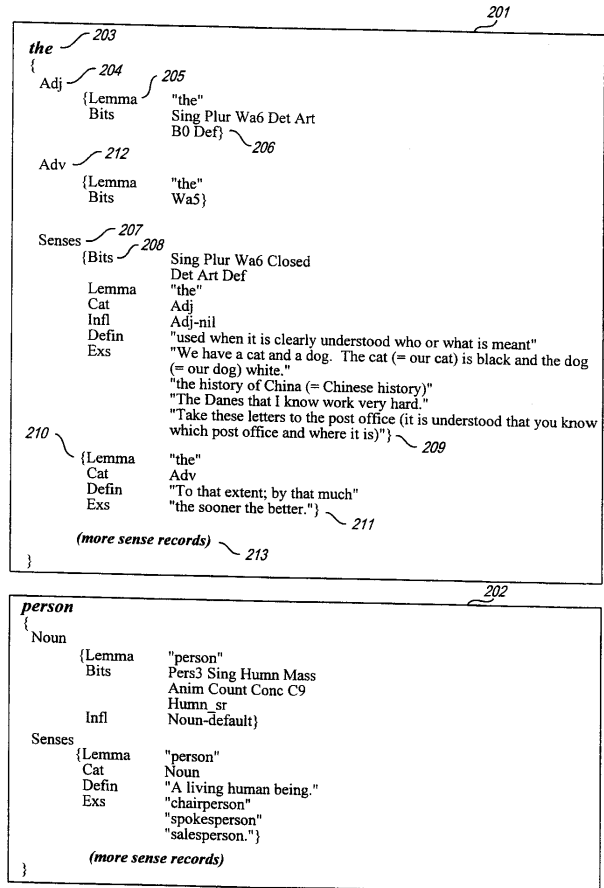


Fig. 2

【 3 】

```

whom
{
  Pron
  {
    {Lemma "who"
     Bits Pers3 Sing Plur Rel Wh
     Humn Obj Anim}
  }
  Senses
  {
    {Lemma "who"
     Bits Pers3 Sing Plur Rel Wh
     Closed Humn Obj Anim
     Cat Pron
     Defin "(the object form of who, used esp. in writing and careful speech)"
     Exs "With whom?"
        "The man with whom he talked."
        "You saw whom?"
        "Whom did they see?"
        "the man (whom) they saw arriving"
        "a man (whom) you may know of?"}
  }
  (more sense records)
}

i
{
  Noun
  {
    {Lemma "i"
     Bits Pers3 Sing TakesAn
     Noun-irreg}
  }
  Pron
  {
    {Lemma "I"
     Bits Sing Nom TakesAn Pers1
     Humn Anim LexCap}
  }
  Senses
  {
    {Lemma "i"
     Cat Noun
     Infl Noun-irreg
     Defin "The ninth letter of the modern English alphabet."}
    {Lemma "I"
     Cat Pron
     Defin "Used to refer to oneself as speaker or writer."}
  }
  (more sense records)
}

met
{
  Verb
  {
    {Lemma "meet"
     Bits Sing Plur Past
     Pastpart
     Infl Verb-meet}
  }
  Senses
  {
    {Lemma "meet"
     Bits Past Pastpart
     Cat Verb}
  }
}

```

Fig. 3

【 4 】

```

was
{
  Verb
  {
    {Lemma "be"
     Bits Pers3 Sing Past Pers1
     Infl Verb-be }
  }
  Senses
  {
    {Lemma "be"
     Bits Past Pastpart
     Cat Verb}
  }
  (more sense records)
}

my
{
  Adj
  {
    {Lemma "I"
     Bits Wa5 Det Poss Pers1 Def
     Gen A0
     Infl Adj-none }
  }
  Ij
  {
    {Lemma "my" }
  }
  Senses
  {
    {Lemma "I"
     Bits Wa5 Closed Det Poss
     Pers1 Def Gen A0
     Cat Adj
     Infl Adj-none
     Defin "belonging to me"
     Exs "my car"
        "my mother"}
    {Cat Ij
     Defin "Used as an exclamation of surprise, pleasure, or dismay"
     Exs "Oh, my! What a tiring day!"}
  }
  (more sense records)
}

```

Fig. 4

【 5 】

```

friend
{
  Noun
  {
    {Lemma "friend"
     Bits Pers3 Sing Humn Anim
     Count Conc Humn_sr N0
     Wrdy
     Infl Noun-default
     Vprp (of to)
     {Bits Humn Count Conc
      Vprp (of) }
     {Bits Humn Count Conc
      Vprp (to) }
  }
  Verb
  {
    {Lemma "friend"
     Bits Inf Plur Pres T1
     Infl Verb-default }
  }
  Senses
  {
    {Lemma "friend"
     Bits Humn Conc
     Cat Noun
     Defin "A person whom one knows, likes, and trusts."}
    {Bits T1
     Lemma "friend"
     Cat Verb
     Infl Verb-default
     Defin "To befriend."}
  }
  (more sense records)
}

```

Fig. 5

【 6 】

```

614 CHAR CHAR1 " "
613 VERB VERB3 "found"
612 NOUN NOUN2 "friend"
611 IJ IJ1 "my"
610 ADJ ADJ3 "my"
609 VERB VERB2 "was"
608 ADJ ADJ2 "not"
607 VERB VERB1 "met"
606 PRON PRON2 "I"
605 PRON PRON1 "whom"
604 NOUN NOUN1 "person"
603 ADV ADV1 "The"
602 ADJ ADJ1 "The"
601 Begin BEGIN

```

Fig. 6

【 図 7 】

規則：形容詞を形容詞句へ
ADJ1 → AJP1

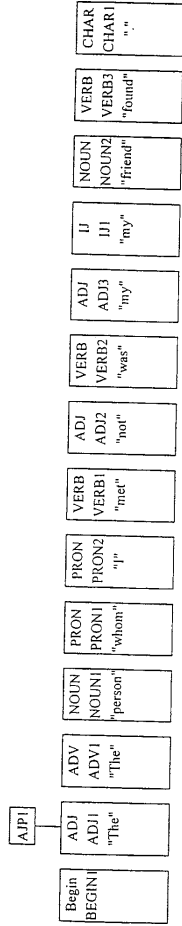


Fig. 7

規則：名詞を名詞句へ
NOUN1 → NP1

【 図 8 】

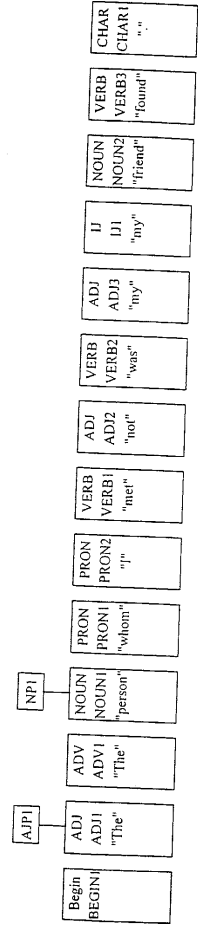


Fig. 8

【 図 9 】

規則：代名詞を名詞句へ
PRON1 → NP2

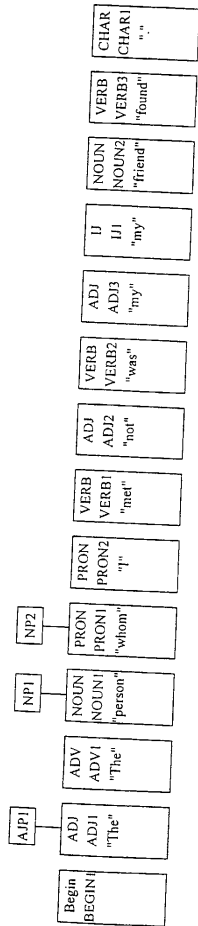


Fig. 9

規則：代名詞を名詞句へ
PRON2 → NP3

【 図 10 】

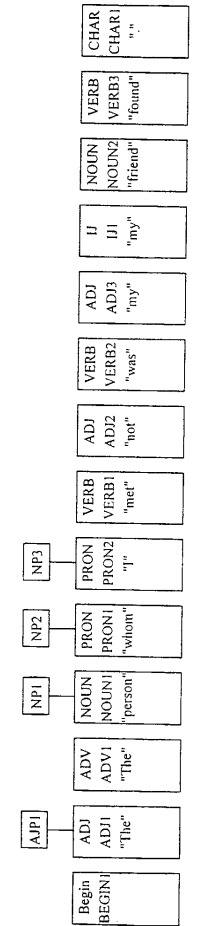


Fig. 10

【 1 1 】

規則：動詞を動詞句へ
VERB1 → VP1

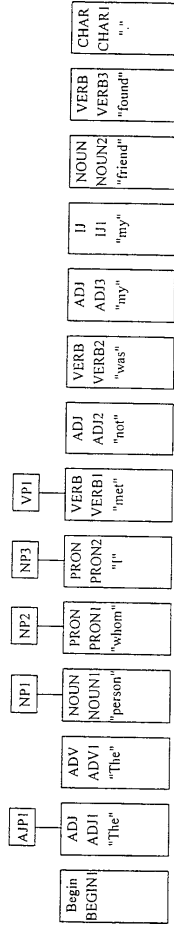


Fig. 11

【 1 2 】

規則：動詞を動詞句へ
VERB2 → VP2

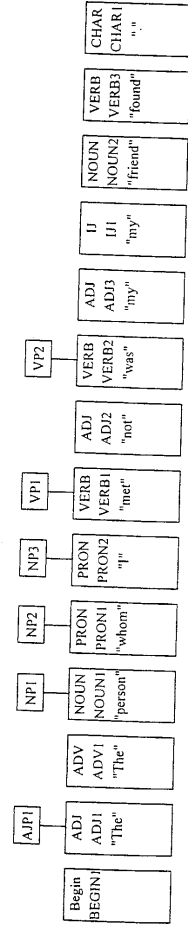


Fig. 12

【 1 3 】

規則：形容詞を形容詞句へ
ADJ3 → AJP2

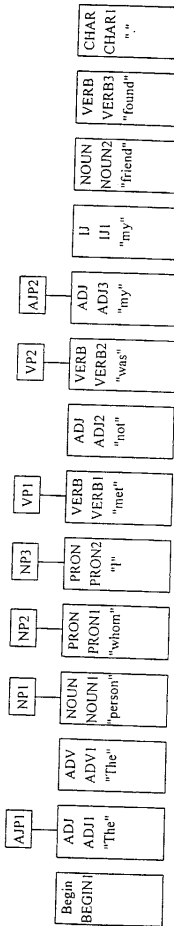


Fig. 13

【 1 4 】

規則：名詞を名詞句へ

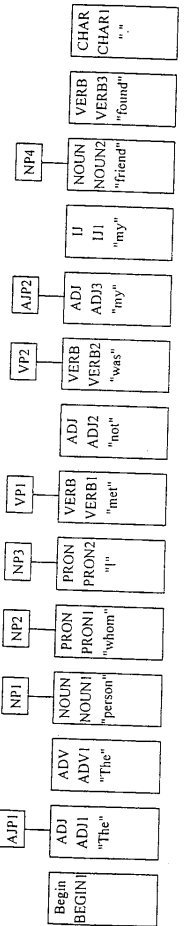


Fig. 14

【 図 15 】

規則：歴定詞付きの名詞句へ
AJP2, NP4 → NP5

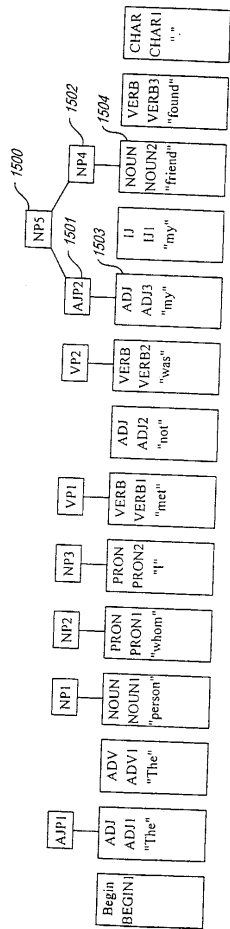


Fig. 15

規則：他動詞の目的語としての名詞句を有する動詞句へ
VP2, NP5 → VP3

【 図 16 】

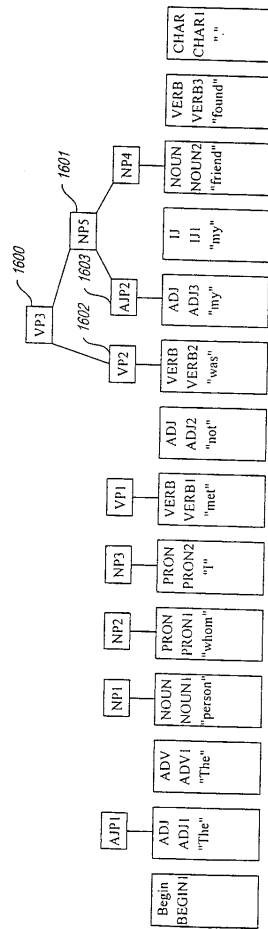


Fig. 16

【 図 17 】

規則：主語としての名詞句を有する動詞句へ
NP3, VP1 → VP4

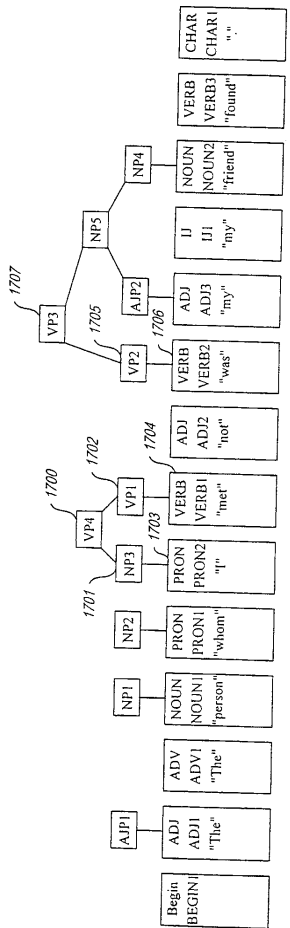


Fig. 17

規則：主語化
NP2, VP4 → VP6

【 図 18 】

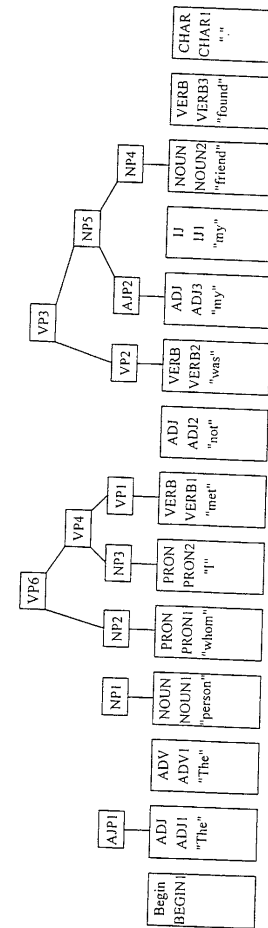


Fig. 18

【 図 19 】

規則：関係詞節付の名詞句
NP1, VP6 → NP9

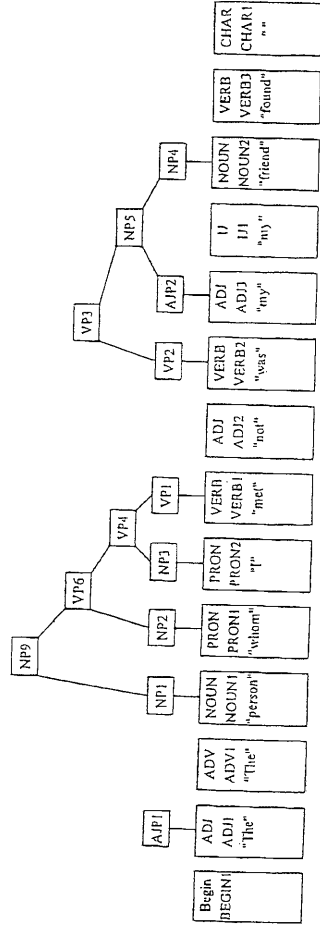


Fig. 19

規則：明確な数量詞を有する名詞句へ
AJP1, NP9 → NP11

【 図 20 】

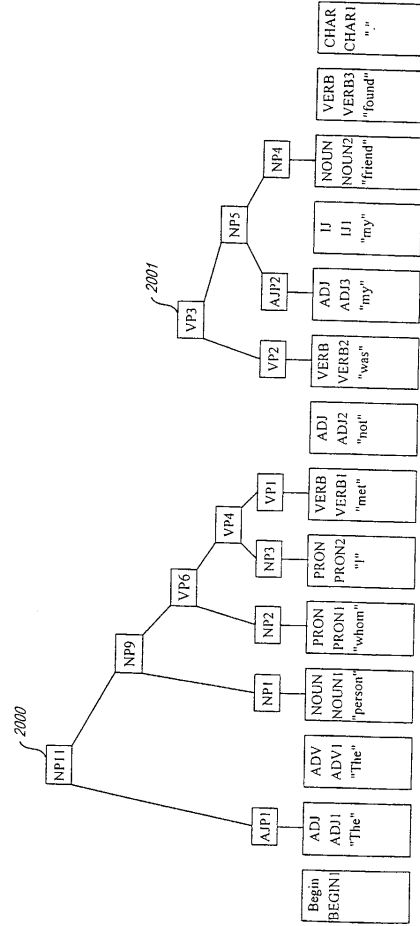


Fig. 20

【 図 21 】

規則：名詞句主語付きの動詞句へ
NP11, VP3 → VP9

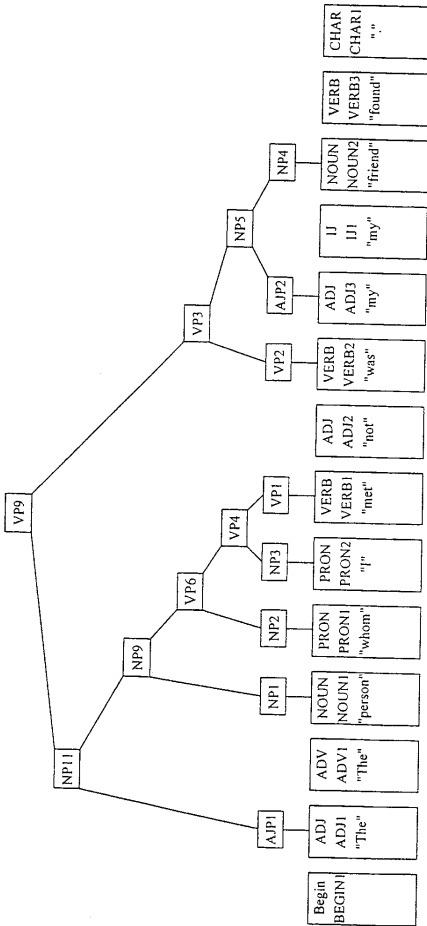


Fig. 21

規則：Beginから"."に至るまでの平綴文
BEGIN1, VP9, CHAR1 → DECI

【 図 22 】

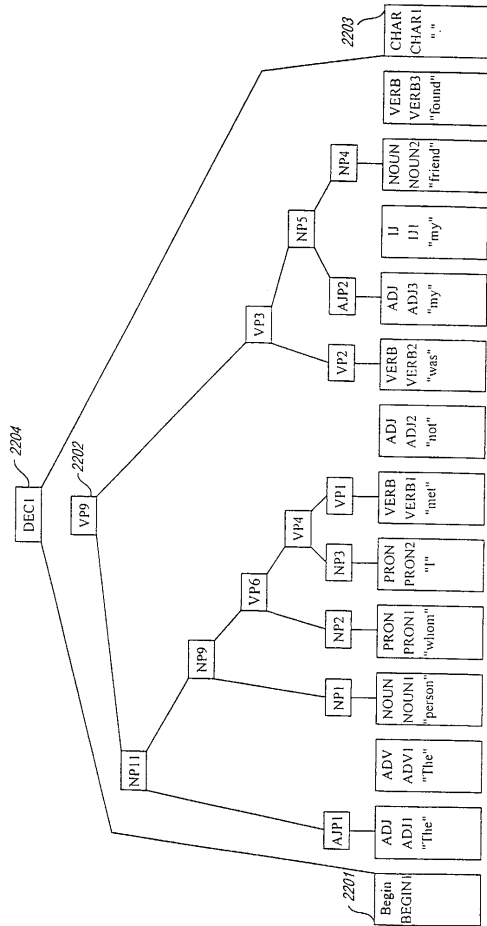


Fig. 22

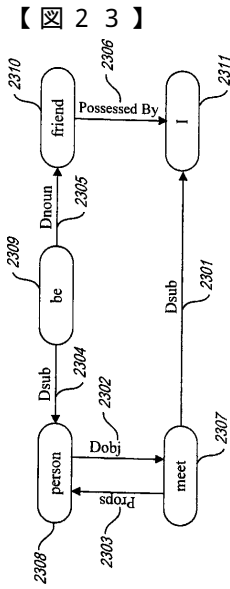


Fig. 23

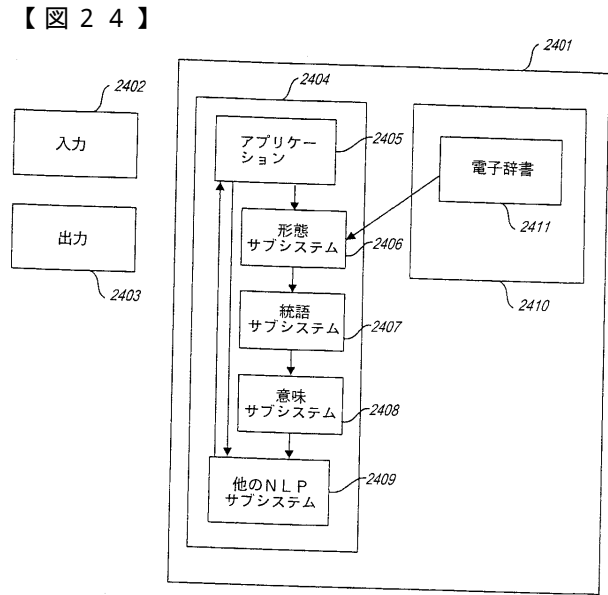


Fig. 24

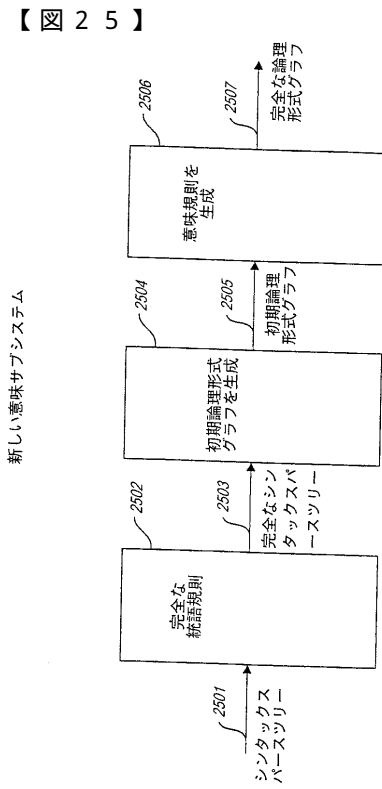


Fig. 25

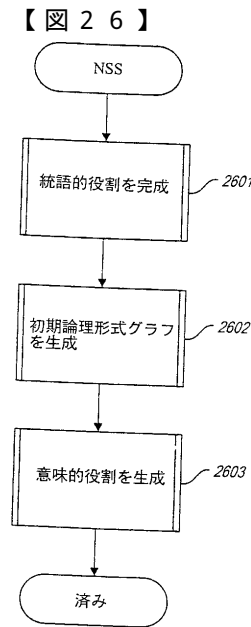


Fig. 26

【 27 】

- PrLF_NPQuantOf: for NPs like "a number of books," makes "books" the head and "a number of" the modifier
- PrLF_PPQuantOf: same but for PPs, like "with a number of books"
- PrLF_notAnaphora: prepares to fill VP anaphora like "John thought he would go but Jim thought not _____"
- PrLF_soAnaphora: prepares to fill VP anaphora like "Mary wondered if it was true but Jane knew so _____"
- PrLF_toAnaphora: prepares to fill VP anaphora like "Chris wanted to go but Pat didn't want to _____"
- PrLF_You: supplies the understood "you" in commands like "(You) please close the door"
- PrLF_HowAbout: supplies the understood "you" in constructions like "How about (you) closing the door?"
- PrLF_We: supplies the understood "we/us" in constructions like "Let's (us) go to the movies"
- PrLF_I: supplies the understood "I" in, for example, "(I) thank you" or "(I) Have not yet received your letter"
- PrLF_SubjectMods: connects "we" and "all" in, e.g., "We are all reading the book"; connects "he" and "hungry" in, e.g., "He arrived hungry"
- PrLF_RightShift: connects "the man" and "who was my friend" in, e.g., "The man arrived who was my friend"
- PrLF_InflCPP: prepares for correct interpretation in constructions like "a person on whom to rely"
- PrLF_QuantifierEllipsis: having to do with the resolution of pronoun references
- PrLF_PossessivePronHead: having to do with the resolution of pronoun references
- PrLF_PossibleCorefsOfProns: having to do with the resolution of pronoun references
- PrLF_VPAnaphora: identifies and fills missing arguments in all cases of VP anaphora, e.g., "Sarah likes basketball and I do too"
- PrLF_DistCoords: distributes elements across coordinated structures, like "They washed _____ and dried the dishes"

Fig. 27

【 28 B 】

パースツリーが表わす文: "Please close the door."
統語サブシステムが生成したシタックスパースツリー

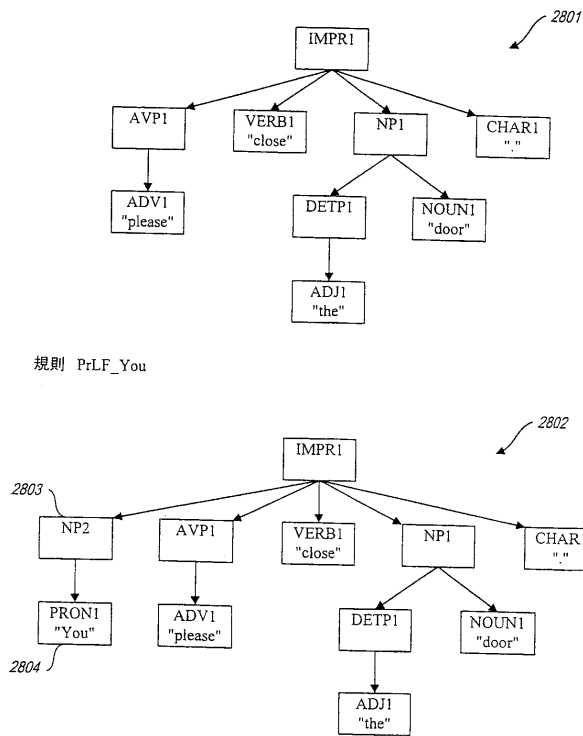


Fig. 28B

【 28 A 】

PrLF_You

If the Syntax Record

has the attribute "Infinitive" and does not have the attribute "Subject" or has the attribute "Verb Phrase Invert" and does not have any of the attributes "Object2," "Yes/No/Question," or "Old Subordinate Clause" and does not meet the "There Subject Test" and does not have the "Coordinate Constructions" attribute and does not have any premodifiers with the node type "Auxiliary Phrase" or the attribute "Modal Verb" and does not have any premodifiers with the lemma "let" or the node type "Adverbial Phrase," and does not have the node type "Abbreviated Clause," "Auxiliary Phrase," "Complement Clause," "Infinitive Clause," "Noun Relative," "Past Participle Clause," or "Relative Clause" and does not have a parent with the node type "Past Participle Clause" and if the head of the parent has node type "Conjunction," then the parent does not have a "Subject" attribute and does not have the node type "Auxiliary Phrase," "Complement Clause," "Infinitive," "Noun Relative," or "Relative Clause" and if there is an Auxiliary Attribute on its Head then for all its Premodifiers their Lemma must not be "neither" nor "so," and if it has a Do Modifier, then it must have an Infinitive attribute and either there must not be a Modal on the First Verb Attribute, or the Lemma of its First Verb must be either "dare" or "need," and if it has a Perfective attribute, then its Lemma must be do, and if it has a Verb Phrase Invert attribute, then either there must not be an L9 attribute or there must not be a Comma attribute and for all of its Premodifiers their node type must not be equal to "Prepositional Phrase" and for all of its Premodifiers their node type must either not be "Adverbial Phrase" or there must be a Comma attribute or the node type of their Head must be an Interjection, and has neither "ect" nor "ect." as its Lemma, and if its Lemma is "suffice," then the Lemma of its Object1 cannot be "it," and if its Lemma is "thank," then the Lemma of its Object1 cannot be "you,"

Then

create a pronoun record for the lemma "you"; make the Subject attribute of the syntax record be a copy of the pronoun record and set the Segtype to be "NP," set the node type to be Segtype, and set the head attribute to be the pronoun record; and set the premodifiers of the syntax record to be the value of the subject attribute plus all of the original premodifiers and set the Undersubject attribute flag.

Fig. 28A

【 29 】

- TrLF_LongDist1: locates NPs that are removed from their semantic heads and reattaches them, e.g., "Who did John say that Mary likes_(who)?"
- TrLF_LongDist2: performs the same kind of long-distance attachment for AJs, INFLCS, PPs, PRPRTCLS, PTPRTCLS, SUBCLS
- TrLF_PhrasalVerb: defines semantic objects of certain verbs when they appear hidden inside PPs: "his hat" is really the semantic object of "took off" in "He took off his hat"
- TrLF_ControlwNP: e.g., in "Chris told Pat what to eat," "Pat" is really the subject of "eat" and "what" is its object
- TrLF_ControlwAJP: e.g., in "I find this difficult to believe," "this" is really the object of "believe"
- TrLF_ForInflcl: used in "for-to" constructions, e.g., in "For Mary to talk to John is easy," "Mary" is really the subject of "talk"
- TrLF_ForInflclCoords: used in "for-to" constructions that have coordinated PPs
- TrLF_MoveProp: given our strategy for attachment, it is sometimes necessary to move clauses from a lower to a higher level so that the proper argument structure can be assigned
- TrLF_ControlatVP: e.g., in "Farmers grow food by using salt water," "farmers" is really the subject of "use salt water"
- TrLF_PropsAsArgs: some clauses (propositions) can be arguments, e.g., in "Has he to answer the letter?" the object of "has" is "to answer the letter"
- TrLF_Extraposition: e.g., in "It makes me happy to meet you," the real subject of "makes" is "to meet you" -- "it" is an empty word and must drop out
- TrLF_FillCoords: fills in missing arguments in coordinated structures
- TrLF_RedefineSubject: e.g., in "What is John's address?" we interpret "John's address" as the logical subject even though it is not in canonical subject position

Fig. 29

【 図 3 0 A 】

TrLF MoveProp

If the syntax Record has either a node type of Abbreviated Clause, Infinitive Clause, Present Participle Clause, Past Participle Clause or if it has a Gerund attribute and an Object of a Prepositional Phrase and if it has Premodifiers, then the node type of all Premodifiers must be either Auxiliary Phrase, Adverbial Phrase, or Prepositional Phrase, and the node type of the Head attribute of the Parent is not "verb" and this syntax record is the last of the post modifiers of its parent and this syntax record is not in the coordinates attribute of its parent and among the ancestors of the parent there is a record whose node type of the Head is "Verb" but none of those ancestors can have a Coordinates attribute (this record will later be referred to as "same ancestor") and there should be no For To Prepositional Phrase attribute on the parent, and if the node type equals Infinitive Clause, then there must be either no WH attribute on PP obj of the parent or the syntax record is not equal to the Nominal Relative of the parent, and if the node type is either Present Participle or Past Participle, then its Parent does not have an Object of a Prepositional Phrase, and if the node type is a Present Participle Clause, then there must be an 'ING' Complement on the same ancestor and if the node type is a Past Participle Clause, then there must be a V8 (code from Longman's dictionary) attribute on the same ancestor and if there is an X1 attribute on the syntax record then there must not be an Object 1 and there is no B3 attribute on its parent, and this syntax record must follow the head of the same ancestor or there is a passive attribute on the same ancestor and if the Lemma of the Parent is 'certain' then the node type of the parent must not be an Adjective Phrase and if the Lemma of the Preposition is either "as" or "of," then there must be a To Noun attribute of its Parent and if the Lemma of the same ancestor is either "be" or "become" then either the node type of the Parent must be an Adjective Phrase or there must be a WH attribute on the Parent or there must be both a To Noun attribute on parent and no There Subject Test on the same ancestor or the Lemma of the Parent must be one of the following: "delight," "horror," "joy," "pleasure," "riot," "shame," "surprise," "terror."

Fig. 30A

【 図 3 0 B 】

TrLF MoveProp

Then the syntax record whose attributes will be changed is the same ancestor syntax record (see above); if the Parent of the syntax record has the Subject attribute and the Parent of syntax record also has the Object attribute, then delete the object attribute from the ancestor; if the Parent of the syntax record has the Subject attribute and the Parent of the Syntax Record does not also have the Object 1 attribute, then set the Subject attribute of same ancestor to be the syntax record; if the same ancestor has the DI (Longman code) attribute and there is an Object Complement attribute and no Indirect Object attribute and there is a To Infinitive on the syntax record and the Parent of syntax record is the Object and there is no WH attribute on the Parent of Syntax Record and either there is an Animate attribute on Parent of syntax record or there is a Case attribute on Parent of Syntax Record and the Lemma of the Parent of the syntax record is not "it" or there is a Human attribute on the Parent of Syntax Record or there is a Proper Name attribute on Parent of syntax record, then make the Indirect Object Attribute on same ancestor equal to that of the Parent of syntax record; if there is a To Infinitive attribute on the syntax record and no Passive attribute on same ancestor, then make the Predicate Complement attribute equal to the syntax record; if the Parent of syntax record is in the Propositions attribute of same ancestor, then take that Propositions list and replace the Parent of the syntax record with the syntax record itself in the propositions list; delete the Infinitive attribute of the Parent of the syntax record; delete the Alternatives attribute on the syntax record; reattach the syntax record to the same ancestor.

Fig. 30B

【 図 3 0 C 】

パースツリーが表わす文: "I have no desire to see the movie."

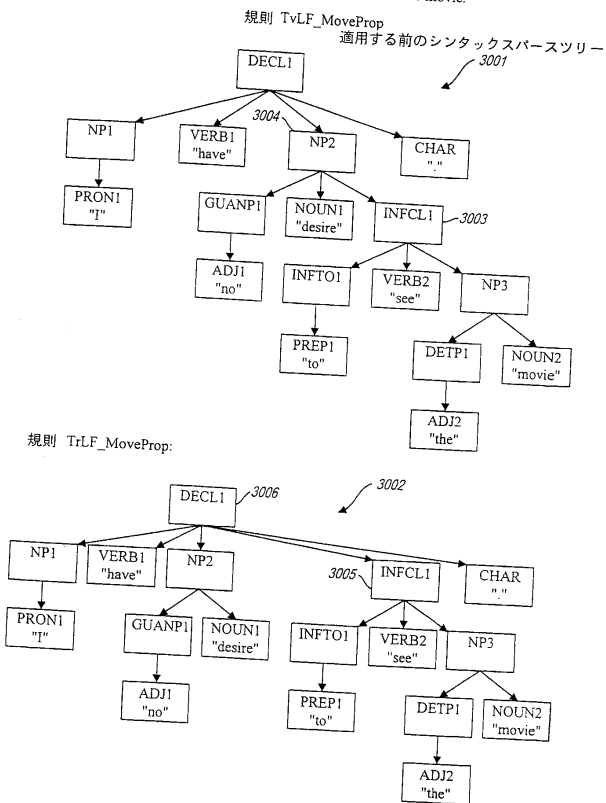


Fig. 30C

【 図 3 1 】

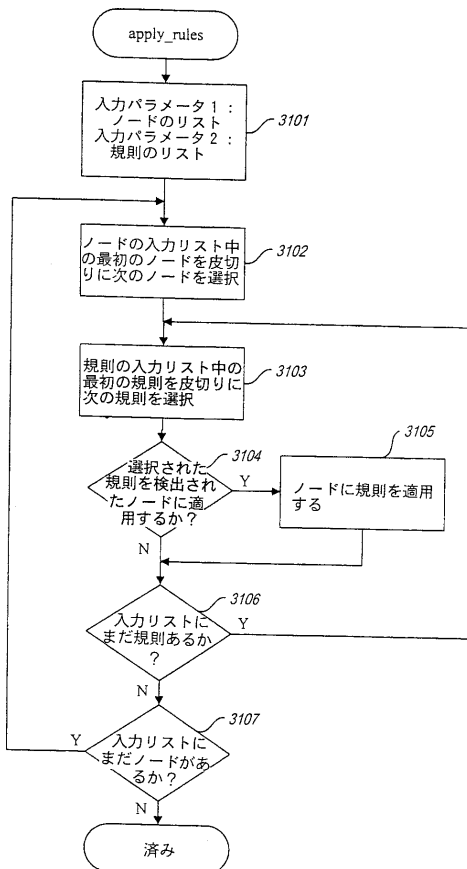


Fig. 31

【 図 3 2 】

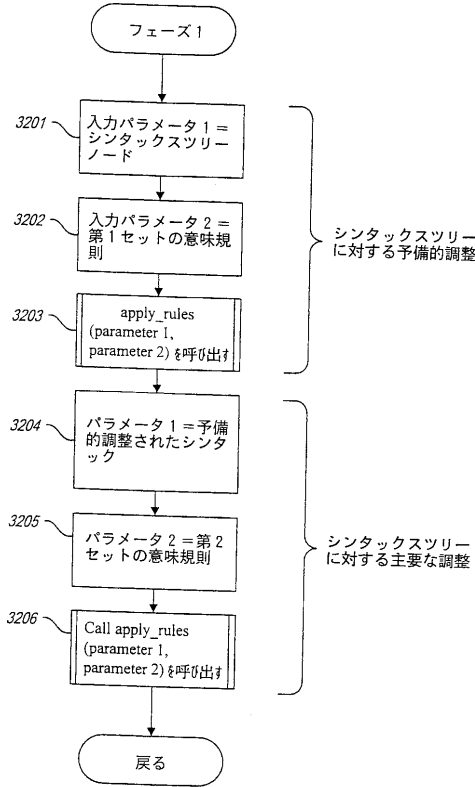


Fig. 32

【 図 3 3 】

1. SynToSem1: creates semantic nodes and a basic semantic graph in es
2. SynToSem2: creates the top-level semantic node and graph for fitted parses
3. SynToSem3: creates semantic nodes for a special subclass of elements in fitted parses

Fig. 33

【 図 3 4 A 】

Rule SynToSem1

If

the Syntax Record

- has a Head and
- there is no Subordinate Conjunction and
- there is no Correlative and
- there is no "It subject" and
- there is no "There subject" and
- there is no Ancestor of the Head for which it is true that that node is the Emphatic of its Parent and is not a fraction and the head node is not a verb and
- if the segment is the Relative Pronoun of its Parent,
 - then there must not be a Nominal Relative on the Object of its Parent and for all of its Parents last records there must not be a VPDone attribute and if the lemma equals "that"
 - then there must not be an Extra Position on the Parent of the Parent and the node type is not "Auxiliary Phrase," "To Infinitive," "Determiner Phrase," or "Tag" or
 - there is a Possessive attribute or
 - there is an EVR attribute or
 - the Lemma equals "other" or
 - there are Coordinates and for all of those coordinates there is either a Possessive attribute or an EVR attribute or the lemma is "other" and
- if the node type is "Adverb Phrase"
 - then if the node type of Parent equals Prepositional Phrase
 - then the segment must not be the first of the Premodifiers of its Parent and
 - either the Lemma must not be equal to 'well' or there must not be any Degree attribute or there must not be any Weak Obligation on the Parent and
- If the node type of the Head is a Conjunction or a Preposition,
 - then the segment node must not be a Conjunction of the Parent and the segment node must not be a Preposition of the Parent and
- If the node type is a Conjunctive Phrase
 - then there must not be any Coordinates of the Parent or there must not be a Coordinate Conjunction attribute and
- If the node type is a Quantifier Phrase,
 - then the Lemma of the Head must not be "no" and
- If the word could have been an Interjection
 - then the node type must not be an Adverb Phrase or
 - there must be Premodifiers or
 - there must be no comma or
 - the segment must be the Post Adverbial of the Parent or
 - the number of Post Modifiers must be greater than one and
- If there is an Intensifier attribute
 - then either the node type of Head of Parent is a "verb" or
 - the node type of Parent equals "fitted" or
 - there is an Adverbial Phrase attribute or
 - there is a WH marker and a Nominal Relative on the Parent and
- If there is a Preposition attribute,
 - then there must be an Object of the Prepositional Phrase or
 - there is a Particle attribute on the Parent or
 - the word also could have been an Adverb and

Fig. 34A

【 図 3 4 B 】

Rule SynToSem1

If the Lemma is "also," "so," or "too,"

- then there must not be a VPDone attribute on the Parent and

If the Lemma is "as" or "than"

- then there must not be a Comparative on the Parent and

If the Lemma equals "for"

- then there must not be a "for to" Preposition on the Parent and

If the Lemma equals "it"

- then if there is a Topic Clause on the Parent
- then the segment must be equal to the Subject of the Parent or
- the segment must be equal to the Object of the Parent and

If the Lemma equals "it"

- then the segment must not be in the Premodifiers of the Parent or
- If there is an Extra Position on the Predicate Adjective of the Parent
- then there must not be a Right Shift attribute on the Parent and
- if there is a WH Question attribute on the Parent
- then there is no "To Infinitive" attribute on the Predicate Complement of the Parent and it's
- not the case that for any of the Post Modifiers of the Parent that there is a "For io" prepositional phrase on the first of the Premodifiers and

If the Lemma equals "let"

- then the node type is not equal to "Adverb Phrase" and

If the Lemma equals "not"

- then there must be a Coordinate Conjunction on the Parent and

If the Lemma equals "there"

- then there must not be any Skipover attribute and
- either there must not be any "Yes No" question on the parent or
- there must not be a Copulative on the Parent or
- there must be a T1 attribute on the Parent or
- the first token integer must be greater than the first token integer of the Subject of the Parent and

If the Lemma is "whether" or "whether or not"

- then the node type of the Nominative Relative must not be an Infinitive Clause" and

the Lemma must not be "etc.," "etc.," "the," "hm," "mm," "uh," or "um"

Then

(If syntax node was kept, then create a corresponding semantic node.)

If the node type of the syntax node is a Noun Phrase and

- there are Bases on the syntax node and
- there is a Subject or an Object on the syntax node,

then make the Predicate equal to the Lemma of the first Basis of the syntax node

Else if there is a Proper Noun attribute on the syntax node and

- if there is a dictionary entry for that word,

then make the Predicate equal to that dictionary entry

Else set the Predicate equal to the Lemma of the syntax node

If the word could have been a Verb and has a Present Participle attribute and

- if for any of the Premodifiers of the syntax node there is a Possessive or
- if the Lemma of the Preposition of the first of the Postmodifiers of the syntax node is "by," "for," "of," or "to"

Fig. 34B

【 図 3 4 C 】

Rule SynToSem1

then make the Predicate equal to the Lemma of the Verb entry of the Part of Speech Record

Copy the appropriate fields from syntax node to the semantic node.

Go through each of the Premodifiers of syntax record and examine each Premodifier

For each record of Premodifiers of the syntax record

if there is a semantic node on the record and

if the semantic node of the record is not in the temporary modifiers attributes of this semantic record and there is no Skipover attribute on the record and the record is not equal to the Preposition of the Parent of the record and the record is either not in the Coordinates of syntax record or there is a Coordinate of the Prepositional Phrase on syntax record, or Coordinate Subordinate Clauses

then add the Semantic node of the record to the Temporary Modifiers attribute on this semantic record

For each record of the Postmodifiers of the syntax record

if there is a semantic node on record and

if the semantic node of record is not in the Temporary Modifiers attributes of this semantic record and there is no Skipover attribute on record and record is either not in the Coordinates of syntax record or there is a Coordinate of the Prepositional Phrase on syntax record or Coordinate Subordinate Clauses

then add the Semantic node of the record to the Temporary Modifiers attribute on this semantic record

If there are Coordinates of the syntax record and no Coordinates of the Prepositional Phrase on that syntax record and no Coordinate Subordinate Clauses

then

for each of the Coordinates of syntax record

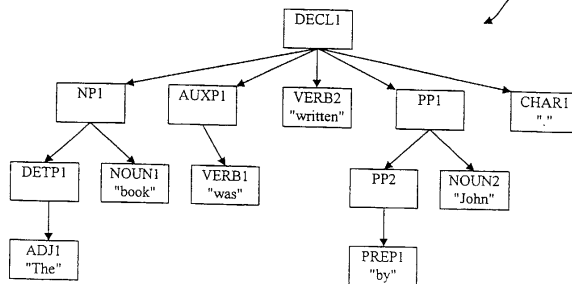
if there is a Semantic node on record,

then add that Semantic node to Coordinates attribute on this new Semantic record.

【 図 3 4 D 】

シンタックスパースツリーが表わす文: "The book was written by John."

規則 SynToSem1 を適用する前のシンタックスツリー



規則 SynToSem1:

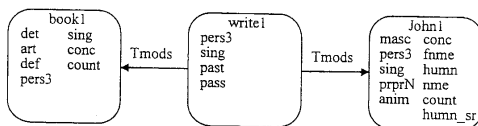


Fig. 34D

Fig. 34C

【 図 3 5 】

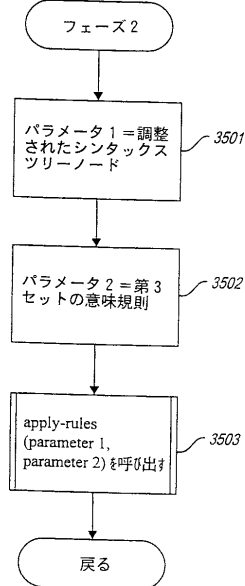


Fig. 35

【 図 3 6 】

1. LF_Dsub1: creates the Dsub (deep subject) label for subjects of clauses in the active voice
2. LF_Dsub2: for passive-voice clauses, if there is a "by"-PP, identifies this PP as the Dsub of the action
3. LF_Dobj1: creates the Dobj (deep object) label for, e.g., direct objects of clauses in the active voice
4. LF_Dobj2: for passive clauses, identifies the syntactic subject as the deep object of the action
5. LF_Dobj3: for clauses like "The door opened," identifies "the door" as the logical object of the action
6. LF_Dobj4: for constructions like "the nomination of the candidate," identifies "the candidate" as the logical object of an action of nominating
7. LF_Dind1: creates the Dind (deep indirect object) label for, e.g., "Mary" in "John gave Mary the book"
8. LF_Dind2: identifies the deep indirect object ("Mary") in paraphrases like "John gave the book to Mary"

Fig. 36

【 37 】

- 9. LF_Dind3: chooses the right deep indirect object in trickier constructions like "The book was given her"; "She was given the book"
- 10. LF_Dnom: creates the Dnom (deep nominative) label for predicate nominative, e.g., "our friends" in "They are our friends"
- 11. LF_Dcmp1: identifies the complement ("president"; "italic") in, e.g., "elect Tom president"; "make the word italic"
- 12. LF_Dcmp2: identifies the complement in trickier constructions, e.g., in "He gave Tom a place to call his own," "his own" is the Dcmp of "call"
- 13. LF_Dadj: creates the Dadj label for predicate adjectives, e.g., "blue" in "The sky is blue"
- 14. LF_CausBy: creates a causative relation where appropriate, e.g., "why" in "Why did you say that?"
- 15. LF_LocAt: creates a locative relation where appropriate, e.g., "where" in "Where did you find that?"
- 16. LF_TimeAt: creates a temporal relation where appropriate, e.g., "what day" in "What day did you read that?"
- 17. LF_Manr: creates a manner relation where appropriate, e.g., "how" in "How did you do that?"

Fig. 37

【 38 】

- 18. LF_Ptcl: creates a Ptcl node to refer to particles in phrasal verb constructions
- 19. LF_PrpCnjs: creates temporary relations for PPs and subordinate clauses by naming these relations with the word that is the preposition or conjunction
- 20. LF_PrpCoord: handles cases of coordinated PPs or subordinate clauses
- 21. LF_Props: lists remaining clausal adjuncts for any given node
- 22. LF_Ops: identifies logical operators in noun phrases, e.g., "all" in "all my children"
- 23. LF_Nadj: lists remaining adjectives that premodify nouns
- 24. LF_Mods: lists remaining non-clausal modifiers for any given node

Fig. 38

【 39 A 】

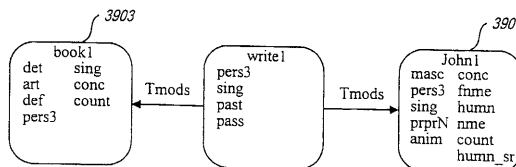
Rule LF_Dobj2

If the Semantic Record
 doesn't already have a Deep Object,
 and has a Passive attribute,
 and has a Subject on its syntactic record (SynNode), and this Subject (which is a syntactic record) has a SemNode attribute (i.e., it has a corresponding semantic record)
 and there are no Coordinates
 and if there is a Predicate Complement attribute on its syntactic record, then the node type is not "COMPCL" (i.e., it is not a complement clause, as in: "some people were convinced that he had written a book")
 and if the SynNode record has either a D5, D6, ObjC, or Psych feature²
 then either the Object of the SynNode is not a noun phrase,
 or the SynNode has an X1³ feature (as in: He was named Arles")
 or the Object of the SynNode has an Animate feature
 or there is a Case feature on the Object of the SynNode and its Lemma is not "it"

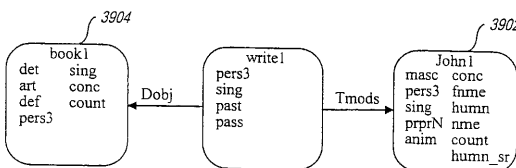
Then,
 give the Semantic record a Dobj attribute with, as its value, the semantic record corresponding to the Subject on the syntactic record
 and, remove what is now the value of Dobj attribute from the list of Tmods

【 39 B 】

論理形式で表現した文: "The book was written by John."
 規則 LF_Dobj2 を適用する前の論理形式



規則 LF_Dobj2:



² D5, and D6 are features from Longman's Dictionary of Contemporary English; ObjC is verb subcategory for verbs which show object control (e.g., I want Harry to wash the car) and Psych is a verb subcategory for verbs like "scare" "excite".

³ X1 is a feature from Longman's Dictionary of Contemporary English.

Fig. 39A

Fig. 39B

【 4 0 】

1. PsLF_RelPro: identifies proper referents for relative pronouns, e.g., "who" refers to "the man" in "the man who came to dinner"
2. PsLF_ReciprocalAnaphora: handles reciprocal pronouns like "each other" and "one another"
3. PsLF_ReflexiveAnaphora: handles reflexive pronouns like "myself, yourself, him/herself," etc.
4. PsLF_PronAnaphora: identifies possible NP referents for most pronouns
5. PsLF_PronoAnaphora: handles special cases of pronouns which can agree with just about any NP
6. PsLF_NumberEllipsis: handles reference for number words, e.g., "A bird in the hand is worth two (birds) in the bush"
7. PsLF_FillInHead: adds "DUMMY" as a head word in special cases of unclear referents
8. PsLF_NumberCritique: takes note of pronouns that disagree in number with their referents
9. PsLF_FillSub: fills in "x" as a placeholder for the deep subject in cases where that is missing, e.g., in passives like "The door was opened"
10. PsLF_UnifyProns: if two pronoun nodes refer to the same referent, this rule unifies them
11. PsLF_UnifyCopies1: unifies some nodes that should be identical
12. PsLF_UnifyCopies2: unifies other nodes that should be identical
13. PsLF_RaiseModality: deletes some verbs when they serve only an aspectual purpose, e.g., in "We used to go there," "used to" is deleted from the graph
14. PsLF_RaisePcs: makes fitted parses easier to read

Fig. 40

【 4 1 B 】

Rule PsLF_PronAnaphora

- and if there is an ancestor of the possible referent that has a Coords attribute (i.e., has coordinate constituents) (but before there is an ancestor with a Subject attribute) then this ancestor is the same as the ancestor of this record that has a Coords attribute (but before there is an ancestor with a Subject attribute)
- then if this record is a possessive (e.g., "his" in "John saw his son") add the possible referent to the list of possible referents (the value of the Refs attribute) if:
- the possible referent is a genitive
 - and node type of the head of the possible referent is not a Noun
 - and the possible referent precedes this record (i.e., the semantic record being processed in this rule)
- or if:
- the possible referent is not the first of this record's Parents
 - and the first of the Parents of the possible referent is not the first of this record's Parents
- and if the possible referent follows this record and if any of the possible referent's ancestors have Coordinate constituents, then there should be no ancestor of this record for which the Parent has Coordinate constituents and for which the Parent is the same as the ancestor of the possible referent that has Coordinate constituents (but before there is an ancestor whose node type is "NP")
- or else if the node type of Parent of this record's syntactic record is "TAG" (i.e., if the pronoun is in a tag question)
- add the possible referent to the list of possible referents (the value of the Refs attribute) if:
- the possible referent is the Subject of the Parent of the Parent of this record (e.g., "they" refers to "someone" in: "Someone painted in here, didn't they?")
- or else:
- if
- this record is a prepositional phrase
 - and this record precedes the Subject of this record's Parent
 - and the possible referent is the Subject of this record's Parent
- then add the possible referent to the list of possible referents (the value of the Refs attribute);
- else if
- this record is not possessive
 - and this record precedes the possible referent
 - and node type of the head of the possible referent is "NOUN" and is not a Dummy noun (i.e., one that cannot be a possible referent)
 - and if this record is not one of possible referent's ancestors
 - and if it is not the case that there is an ancestor of this record that has Coordinate constituents and the Lemma of that ancestor is "but" and that ancestor is also an ancestor of this record which has Coordinate constituents
- then add the possible referent to the list of possible referents (the value of the Refs attribute)

Fig. 41B

【 4 1 A 】

Rule PsLF_PronAnaphora

- If the Semantic Record**
- has a Pers3 attribute, i.e., it is not either first (e.g., I or we) or second person (e.g., you)
 - and the node type of the head of its syntactic record is either "PRON" (pronoun) or the node type of the head of its syntactic record is "ADJ" (adjective) and it has a possessive attribute and is not Reflexive
 - and none of the premodifiers of the Parent of its syntactic node has the Lemma "own" and the Pred of this semantic record is not "each other" or "one another"
 - and does not have NonRef attribute (NonRef is an attribute set on words that cannot have a reference, such as true numbers, as in: One plus one is two.
 - and does not have a Negation attribute
 - and if it has an Indefinite attribute, then there must also be a Definite attribute
 - and is not a Wh- word (it does not have a WH attribute)
 - and is not a Relative
 - and is not a Distal (Distl) or a Proxal (Proxl) determiner (e.g., "this" "that")
- Then**
- add a FindRef attribute to the semantic record
 - for each of the records in the list of possible referents;¹
 - if
 - the possible referent has a corresponding semantic record
 - and the possible referent is not the same as this record (i.e., the antecedent of a noun phrase can not be the noun phrase itself)
 - and if the head of both the possible referent and of this record's SynNode are pronouns (i.e., have the node type "PRON" as their head), then the possible referent must precede this record (no forward reference to a pronoun; an example of forwards (cataphoric) reference is: with his hat on, the teacher left the room, where "his" refers forward to "teacher"
 - and if the possible referent is the ancestor of the syntactic record of this record, then that ancestor must have a Prp attribute (i.e., must have a postmodifying Prepositional phrase), and its preposition must be either "in", "to", "for", or "by"
 - and there is no Time or Space feature on the possible referent
 - and this record and the possible referent agree in number
 - and this record and the possible referent agree in gender
 - and if the Lemma of the SynNode is "they" and the possible referent can be a Mass noun (i.e., the possible referent has a Mass feature), then the possible referent must also be a Count noun (i.e., it must also have a Count feature).
 - and if the Lemma of the SynNode is "they" and the possible referent has a Sing feature (can be Singular), and the possible referent does not have a Plur feature (i.e., it cannot be Plural), then the possible referent is either a Count noun, or the possible referent is a Coordinated noun phrase, or it has a Universal feature, or the possible referent is indefinite and has no possessive, or the possible referent has a Proxal feature,

¹ this list is created in a PrLf rule, so, after syntactic processing but before most logical form processing (it is a list of syntactic records). This is a list of all the words in the sentence which can be referred to, i.e., most of the nouns and pronouns in the sentence

Fig. 41A

【 4 1 C 】

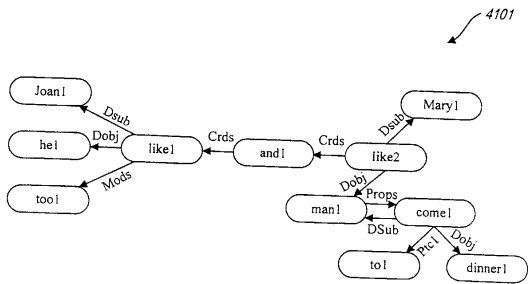
Rule PsLF_PronAnaphora

- else if
- the possible referent is a Prepositional Phrase
 - and the Parent of the possible referent is not the Parent of this record's syntactic record
 - and if the Parent of the possible referent is an Adjective Phrase, then the Parent of the possible referent precedes this record
- then add the possible referent to the list of possible referents (the value of the Refs attribute)
- else if
- there is no ancestor of the possible referent for which the Lemma is "be" (but before there is an ancestor with a Subject) that is the same as the ancestor of this record for which the Lemma is "be" (but before there is an ancestor with a Subject)
 - and none of the Parents on the semantic record of the possible referent is the same as the possible referent
 - and if this record precedes the possible referent, then the Head of the possible referent is not either a Noun or an Adjective
- then add the possible referent to the list of possible referents (the value of the Refs attribute)
- if the possible referent was added to the list of possible referents (the value of the Refs attribute) then add of RefOf attribute to the possible referent and add this record to that list (provide cross pointers: this record gets a Ref attribute pointing to possible referents, and the possible referents each get a RefOf attribute, pointing back to this record.

Fig. 41C

【図41D】

論理形式で表現した文: "Mary likes the man who came to dinner, and Joan likes him too."
規則 PsLF_PronAnaphora を適用する前の論理形式



規則 PsLF_PronAnaphora:

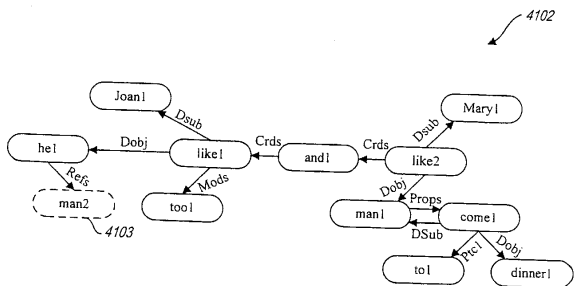


Fig. 41D

【図42】

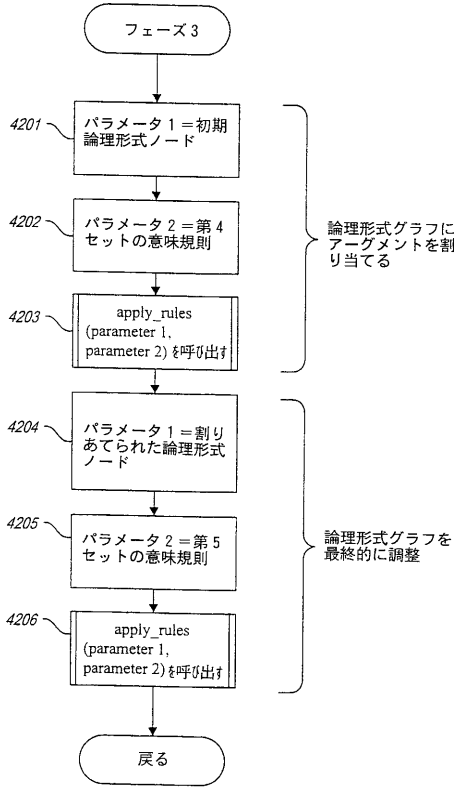


Fig. 42

【図43】

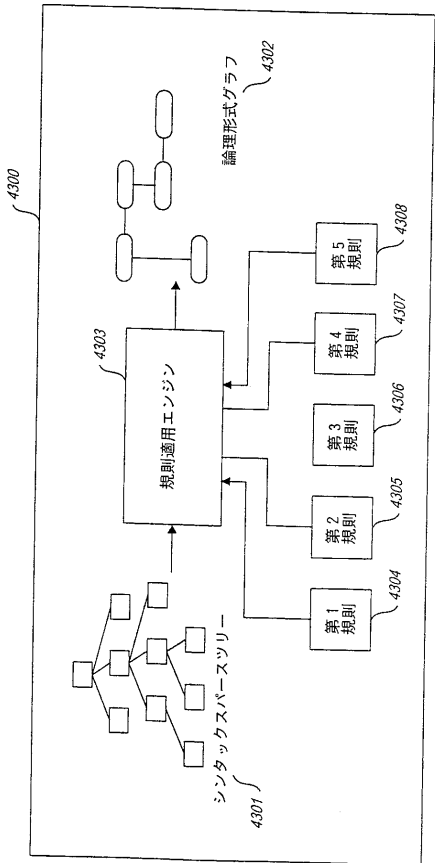


Fig. 43

【図44】

規則: TrLF_LongDist1でRELCL1("whom I met")を修正する

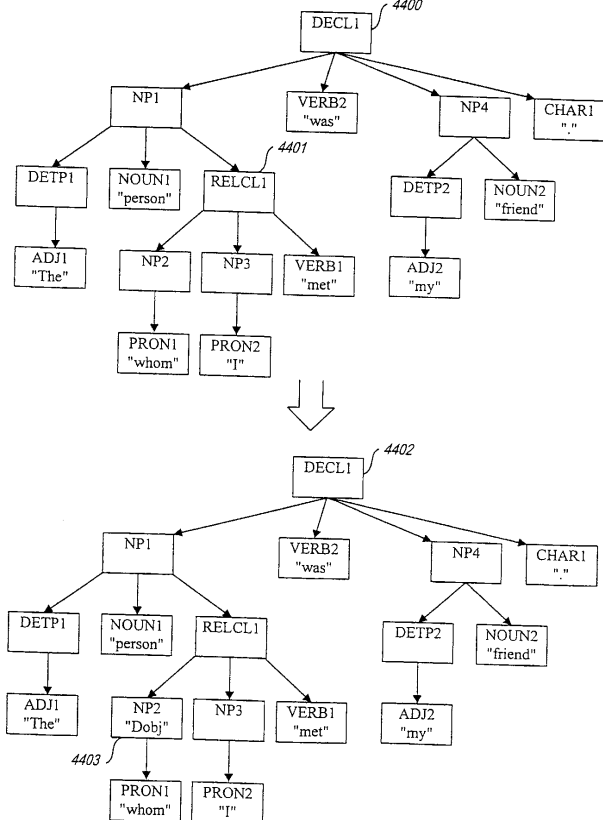
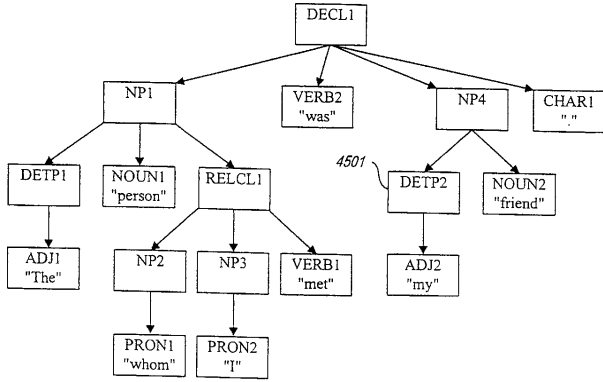


Fig. 44

【 図 4 5 】



規則：SynToSem1でDETP2から論理形式グラフノードを ("my") 作り出す

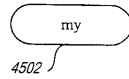
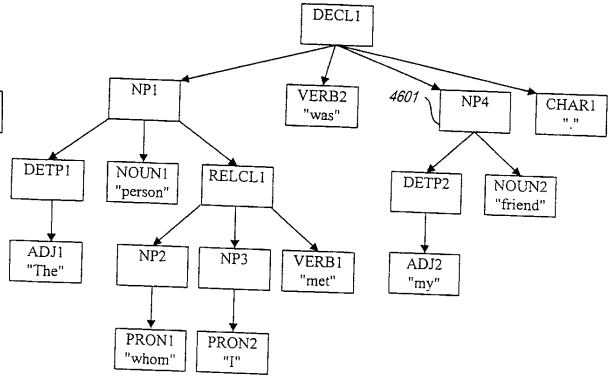


Fig. 45

【 図 4 6 】



規則：SynToSem1でNP4から論理形式グラフノード"friend"を作り出す

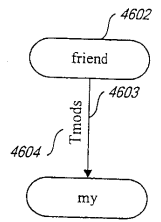
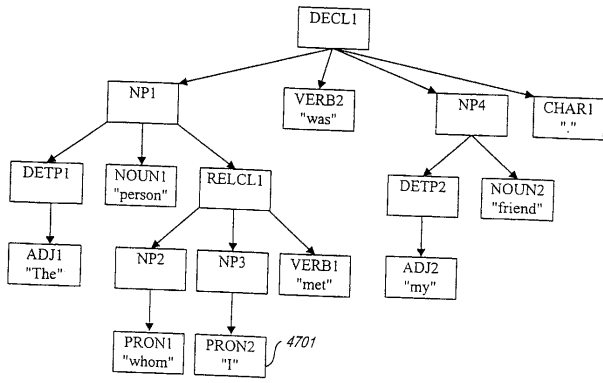


Fig. 46

【 図 4 7 】



規則：SynToSem1でNP3から論理形式グラフノード"I"を作り出す

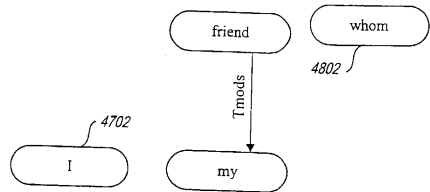
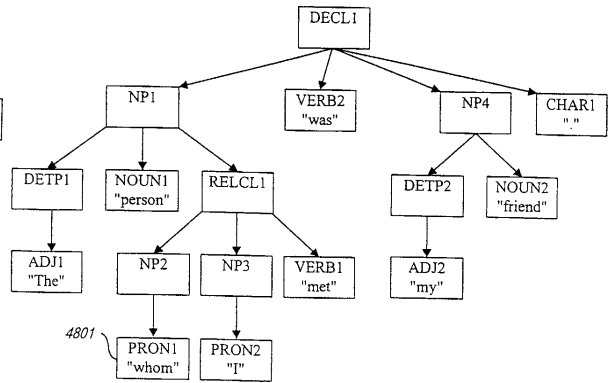


Fig. 47

【 図 4 8 】



規則：SynToSem1でNP2から論理形式グラフノード"whom"を作り出す

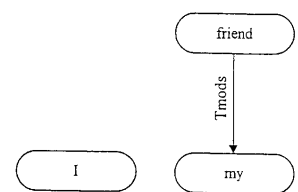
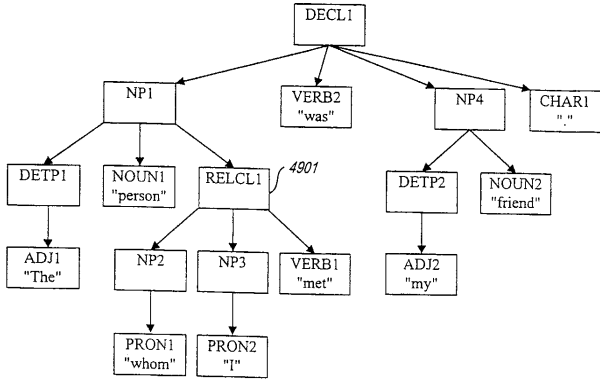


Fig. 48

【 図 4 9 】



規則：SynToSem1でRELCL1から論理形式ノード“meet”を作り出す (“whom I met”)

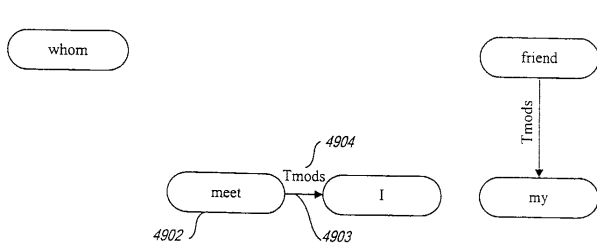
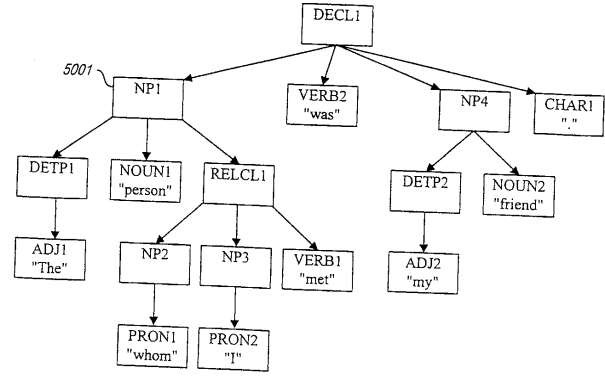


Fig. 49

【 図 5 0 】



規則：SynToSem1でNP1から論理形式ノード“person”を作り出す (“The...met”)

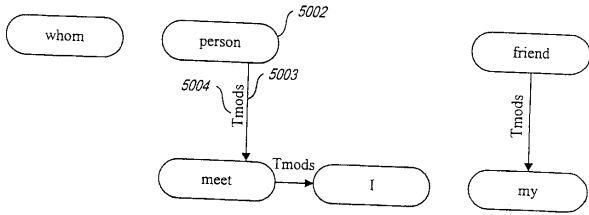
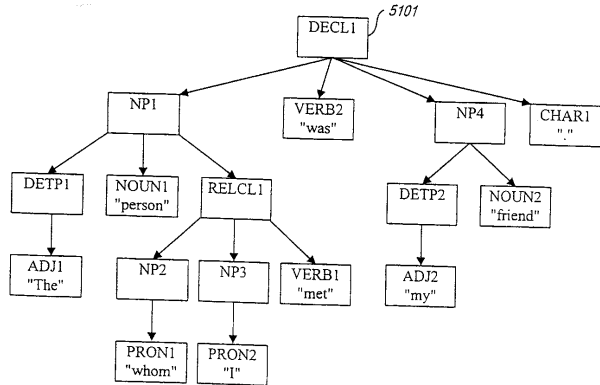


Fig. 50

【 図 5 1 】



規則：SynToSem1でDECL1から論理形式グラフノード“be”を作り出す

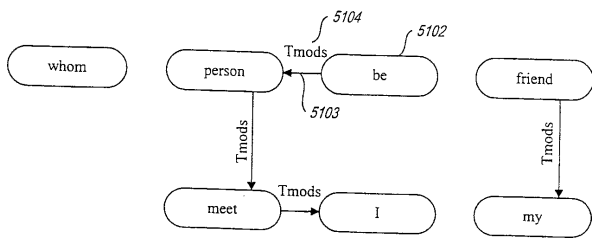
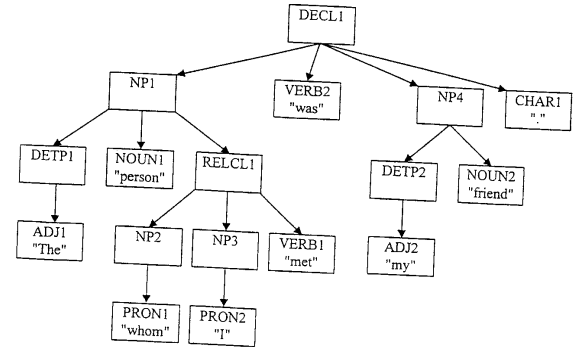


Fig. 51

【 図 5 2 】



規則：ノードbeにLE.Dsub1を適用してリンクにラベルを貼りもう1つのリンクを作り出す

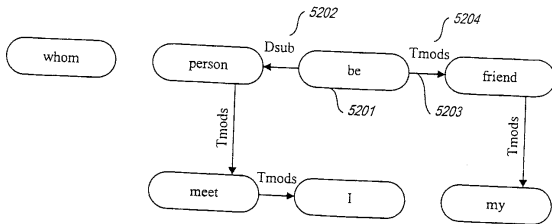
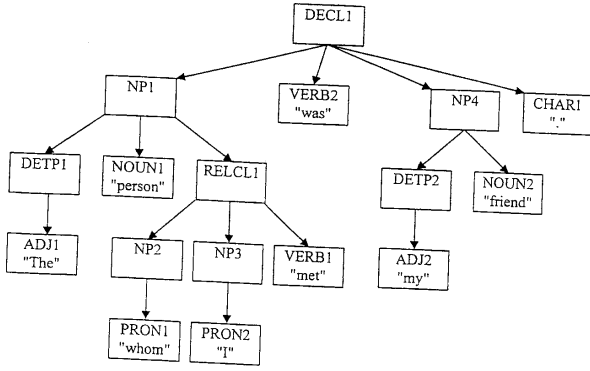


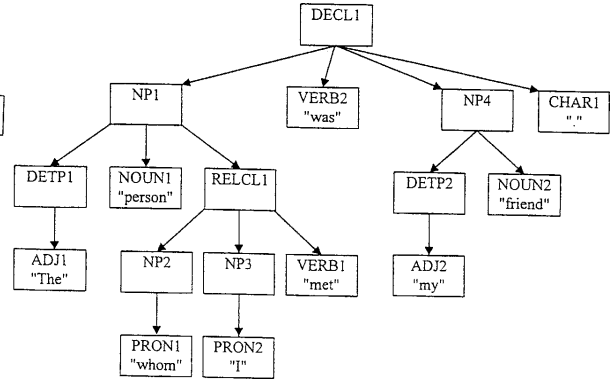
Fig. 52

【 図 5 3 】



規則：ノードbeにLF_Dnomを適用してリンクにラベルを貼る

【 図 5 4 】



規則：ノードpersonにLE_Propsを適用してリンクにラベルを貼る

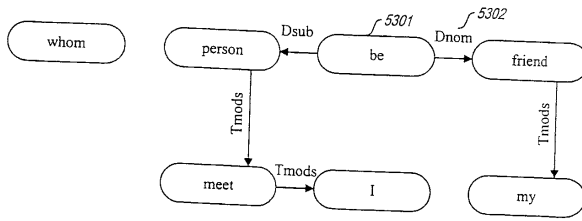


Fig. 53

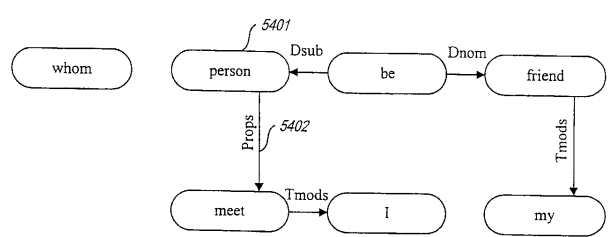
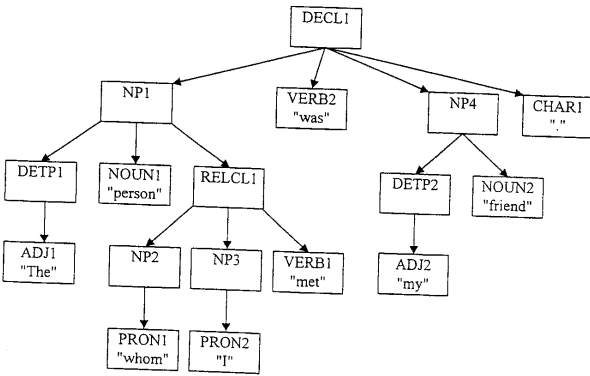


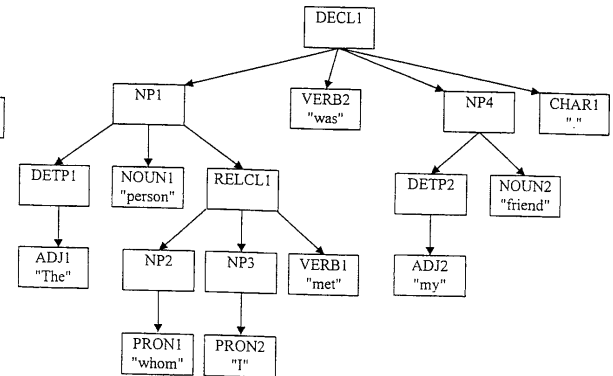
Fig. 54

【 図 5 5 】



規則：ノードmeetにLF_Dsub1を適用しリンクにラベルを貼る

【 図 5 6 】



規則：ノードmeetにLF_Dobj1を適用してリンクを追加しリンクにラベルを貼る

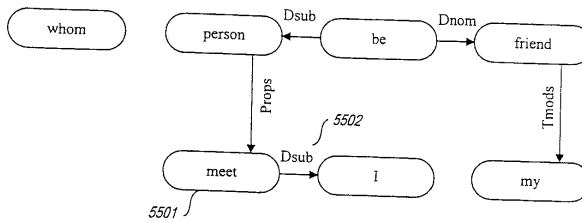


Fig. 55

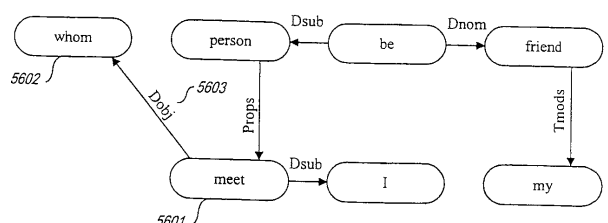
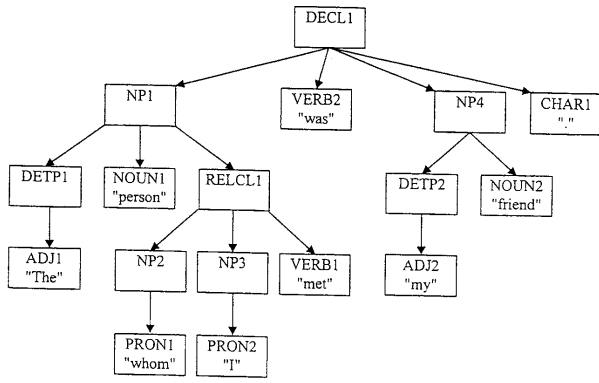


Fig. 56

【 図 5 7 】



規則：ノードfriendにLF_Opsを適用してリンクにラベルを貼る

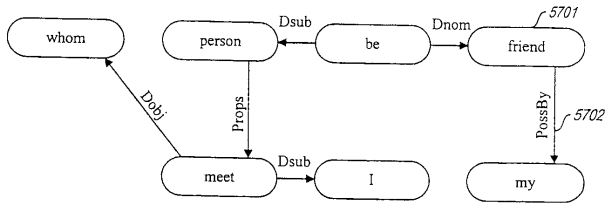
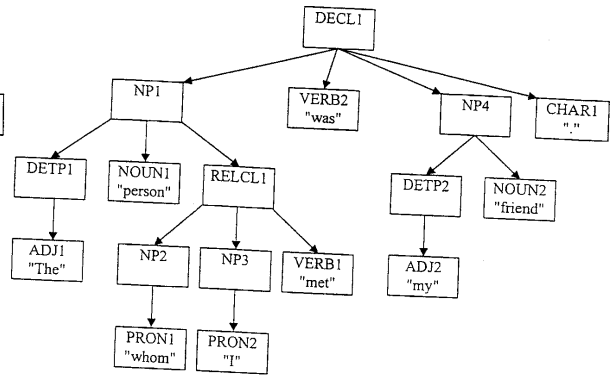


Fig. 57

【 図 5 8 】



規則：ノードwhomにPsLF_ReIProを適用してノードを除きリンクを追加する

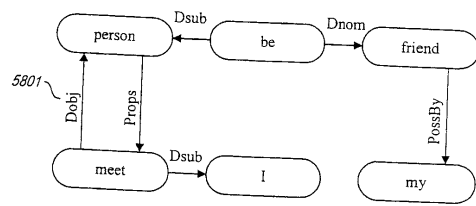
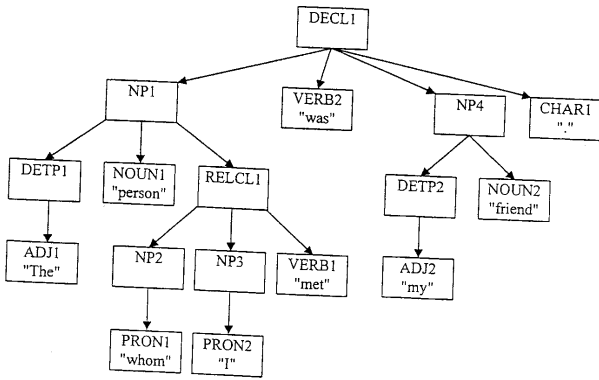


Fig. 58

【 図 5 9 】



規則：PsLF_UnifyProns で2つのノードI,myを1つのノードに統合する

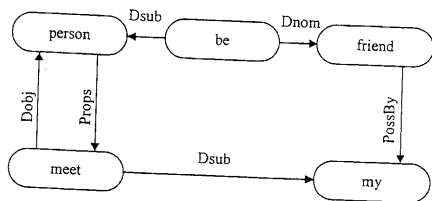


Fig. 59

フロントページの続き

(72)発明者 ハイドーン ジョージ
アメリカ合衆国 ワシントン州 98008 ベルビュー ノースイースト ワンハンドレッド
アンドシックスティフィフス プレイス 3211

(72)発明者 ジャンセン カレン
アメリカ合衆国 ワシントン州 98008 ベルビュー ノースイースト ワンハンドレッド
アンドシックスティフィフス プレイス 3211

合議体

審判長 田口 英雄

審判官 長島 孝志

審判官 久保 正典

(56)参考文献 特開平3 - 83167 (JP, A)
特開平5 - 197712 (JP, A)
特開平2 - 291076 (JP, A)
特開平2 - 187857 (JP, A)
特開平4 - 47364 (JP, A)