
自然言語処理パーザの試作

担当：村上 仁一

実験場所：知能棟 2 階 講義室 および演習室

1 実験の目的

我々は、自分の意見や考えを表現し、相手に伝達したり理解するための手段として日本語や英語などの自然言語を用いる。言語の研究は古くからある。古くからはギリシア時代にさかのぼる。この時代、ソフィストが活躍した。その時代において、論理の力で無理やりに相手を打ち負かす詭弁に対抗する能力が要求された。その基本は相対主義である。

計算機 C 担当の実験 III (3 年生後期) では、まず、言語学の基本について学ぶ。そして言語と自然言語の違いについて学ぶ。そして自然言語処理における、基本技術の一つになっている構文解析技法について、その原理を学ぶ。またレポートの書き方について学ぶ。

2 言語学

言語学は、ヒトが使用する言語の構造や意味を科学的に研究する学問である。大きく以下の分野で構成される。

1. 音韻論
音韻体系の研究
2. 形態論
語構造の研究
3. 統語論
文構造の研究
4. 意味論
意味の研究

2.1 言葉の意味

みぎ【右】の意味とは、“東に向いたとき南にあたる方”である。

しかし、“意味”を説明することは、非常に困難である。この意味を説明するための研究は、古くからされてきた。

2.2 “非文”と“正文”

文の意味が理解できる文を“正文”。文の意味が理解できない文を“非文”と言う。

例えば、“私は山に行く”は正文である。“私はウナギだ”は非文である。

2.3 "受理可能"と"受理不可能"

文を入力したとき,"正文"と"非文"に判別できる,仮想機械を想定する.このとき,仮想機械において"受理可能"な文を"正文"とする.また,"受理不可能"は文を"非文"とする.

2.4 文法

"正文"と"非文"に判別できる仮想機械において,使用される規則を"文法"と言う.通常,学校でならう文法とは,異なる.

2.5 パーザ

仮想機械は,"文法"と"単語辞書"を用いて入力文に対して"非文"と"正文"の判別をする.この処理を"パーザ"と言う.図1に例を示す.

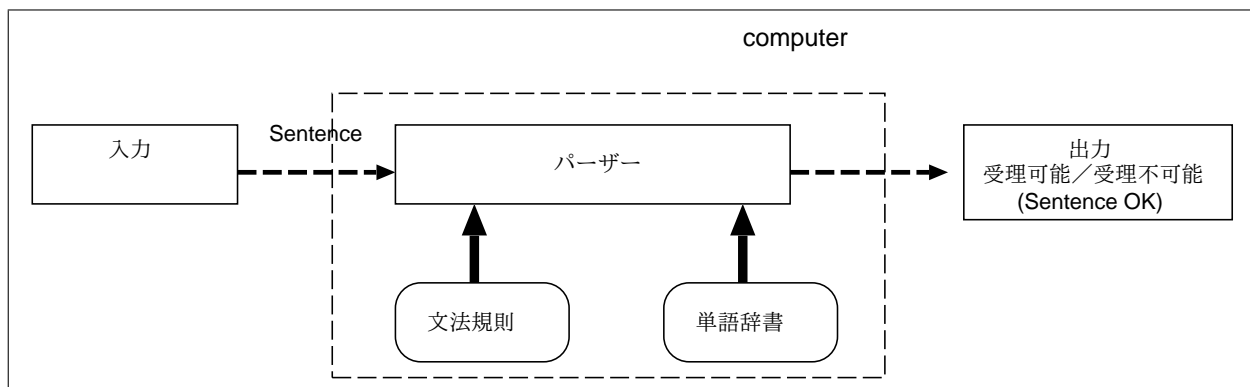


図 1: パーザの動作環境

2.6 形式文法

言語は,無限の文をもつ.したがって言語を規定する文法も,無限の生成能力をもつ.しかし,無限の生成能力にも階層がある.これを階層化するために,通常形式言語が使用される.形式言語 $G = \langle N, T, P, \sigma \rangle$ とする.ここで N は非終端記号の集合, T は終端記号, P は生成規則, σ は初期記号とする.

チョムスキーは形式言語を以下の4つに分類した(図2).

1. 正規文法(ネットワーク文法)

正規文法はチョムスキー階層で最下層にある.そして制約が最も厳しい(一般性がない)書換え規則である.非終端記号を大文字,終端記号を小文字で表すと, $A \rightarrow B, B \rightarrow b$ と書くことが可能である.そして,資源言語では,終端記号が単語に相当し,非終端記号は,品詞に相当する.また,正規言語は有限オートマトンと等価であり,オートマトンで表現すると,理解しやすい.

UNIX で使われる正規表現(regular expression)はこの正規文法をもっている.

2. 文脈自由文法(context-free grammar)

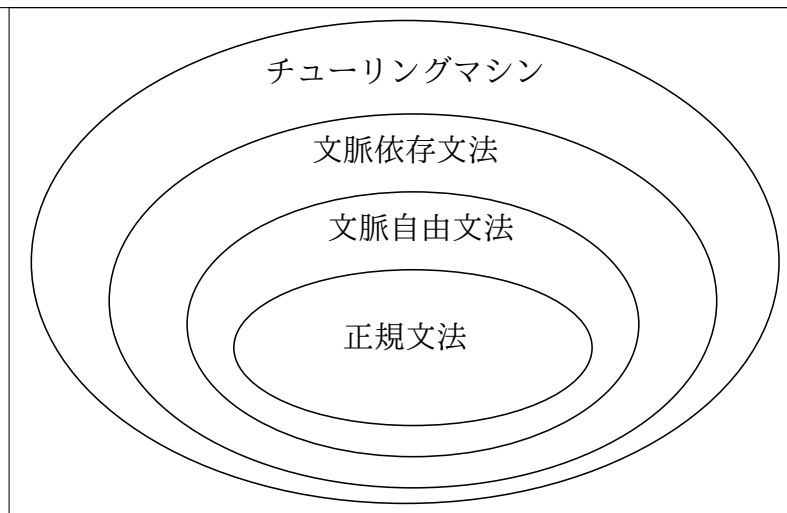


図 2: チョムスキーの言語階層

文脈自由文法 (context-free grammar) の書換え規則は、非終端記号を大文字、終端記号を小文字で表すと、 $A \rightarrow aB$ で、左辺は 1 つの非終端記号でなければならないが、右辺に制約はない。文脈自由言語はプッシュダウンオートマトン (pushdown automaton) と等価である。

文脈自由文法は、すべてのコンピュータ言語にも使われている。例えば Pascal, BASIC, C, Modula-2 や他の言語は文脈自由文法で解析できる。そして、コンピュータ言語のために開発された、わかりやすい構文解析技法は自然言語に適用されている。

3. 文脈依存文法 (context-sensitive grammar)

文脈依存文法 (context-sensitive grammar) の書換え規則は、 $\alpha \rightarrow \beta$ ただし、 $|\alpha| < |\beta|$ と記述できる。ただし、ギリシア文字は、非終端記号、終端記号の総和で、 $|$ は、長さを意味する。文脈依存言語は線形有界オートマトン (linear bounded automaton) と等価である。現在汎用的な文脈依存文法を解析できるアルゴリズムは開発されていない。しかし TAG (tree adjoining grammar) などが文脈自由文法に制限を加える形で開発されている。

4. 無制約文法 (チューリングマシン)

最も一般的な文法は、全く制約のない書換え規則 $\alpha \rightarrow \beta$ をもつ無制約文法 (unrestricted grammar) である。無制約文法による言語は帰納的可算 (recursively enumerable) で、チューリングマシン (Turing machine) に一致する。

3 自然言語処理

3.1 自然言語処理とは

自然言語処理 (Natural-Language Processing, 略して NLP) とは、人間が書いたり、話したりしたりする文章をコンピュータに理解させることである。そのために、文章の中の単語の品詞を決定するための形態素解析、単語間のつながりを調べる構文解析、単語や文の意味を解析する意味解析及び必要な応答文を生成する、などの処理が必要である。

この実験では、この基本技術として、形態素解析や構文解析を学ぶ。なお、これらの解析方法は、本実験では英語に仮定して話を進めるが、C や Basic などの人工言語にも適用できる。

3.2 形態素解析と構文解析

自然言語処理の基本は、形態素解析と構文解析である。形態素解析とは、単語の品詞を決定することであり、構文解析は、単語の係り先を決定することである。形態素解析を行うには、ネットワーク文法で十分であるが、構文解析には、文脈自由文法が必要になる。また、各パーザはそれぞれ異なる方法で文章の構造を解析する。

1. ネットワーク文法（有限状態オートマトン）

形態素解析

2. 文脈自由文法（プッシュダウンオートマトン）

形態素解析 + 構文解析

3.3 ネットワーク文法

3.3.1 概要

ネットワーク文法は、文法規則として、品詞の接続規則が与えられる。また、単語辞書として、単語が属する品詞が与えられる。そして、次にどんな品詞の単語がくるかを、文章の現在の状態（単語の品詞）から予測する。ネットワーク文法の文法規則と単語辞書の例を表 1 に示す。

表 1: ネットワーク文法

文法規則		単語辞書	
start	→ DET (冠詞)		
start	→ NOUN (名詞)		
DET(冠詞)	→ ADJ(形容詞)		
DET(冠詞)	→ NOUN(名詞)		
ADJ(形容詞)	→ NOUN(名詞)		
NOUN(名詞)	→ PREP(前置詞)	the	→ DET(冠詞)
PREP(前置詞)	→ NOUN (名詞)	child	→ NOUN(名詞)
PREP(前置詞)	→ ADJ(形容詞)	runs	→ VERB(動詞)
PREP(前置詞)	→ DET(冠詞)	quickly	→ ADV(副詞)
NOUN(名詞)	→ VERB(動詞)	to	→ PREP(前置詞)
VERB(動詞)	→ ADV(副詞)	the	→ DET(冠詞)
VERB(動詞)	→ DET(冠詞)	large	→ ADJ(形容詞)
ADV(副詞)	→ PREP(前置詞)	house	→ NOUN(名詞)
NOUN(名詞)	→ ADV(副詞)		
ADV(副詞)	→ NOUN(名詞)		
VERB(動詞)	→ end		
NOUN(名詞)	→ end		

この文法は、ある状態から他の状態に変わる場合の、正しい遷移の仕方を示す有限状態オートマトンである。たとえば、名詞の次には動詞か前置詞がくる。

図3に表1中の文法規則をオートマトンで示す。

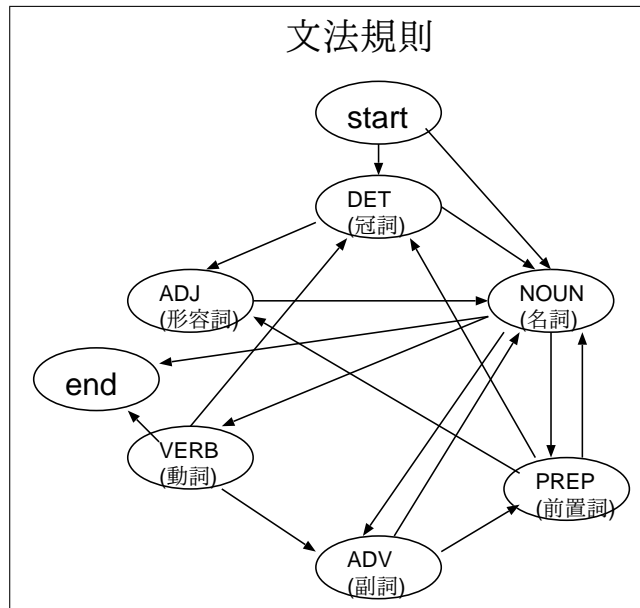


図3: ネットワーク文法の例 (状態遷移図)

3.3.2 プログラムの動作

次の簡単な例題を使って、どのように解析されるか見てみる。

”the child runs quickly to the large house.”

この文章は下の表2のように表1に示される文法に従っている。

表2: 単語と品詞の関係

	the	child	runs	quickly	to	the	large	house	.
start	DET	NOUN	VERB	ADV	PREP	DET	ADJ	NOUN	end
初期状態	冠詞	名詞	動詞	副詞	前置詞	冠詞	形容詞	名詞	最終状態

ここで、図3の状態遷移図 (diagram, ダイアグラム) を見ながら、ネットワーク文法が行っている文章の処理方法を理解してみる。

1. 最初に抽出された単語は the である。the は冠詞であるから、現在の状態を、”DET (冠詞)”にする。状態マシンはここからスタートする。
2. 次の単語 child は名詞であるから、表1から冠詞は名詞と形容詞の遷移が可能である。child は名詞なので、この遷移は成功し、状態は新しい状態 ”NOUN (名詞)”へ移る。

3. 次の単語は動詞 run で、表 1 から名詞からは、動詞と前置詞への遷移が可能である。run は動詞なのでこの遷移は成功し、状態は新しい状態 "VERB (動詞)" へ移る。
4. 次の単語は副詞 quickly であり、状態遷移図を見ると、遷移可能であるから、システムは "ADV "に移る。" ADV "からは "NOUN "と "PREP "の 2 つの可能性しかない。文章中の次の単語は前置詞 to であるから、パーザは "PREP "の状態に向かう。
5. 同様な動作が続いて、最後に形容詞 large が来て "ADJ "への遷移が起こる。状態 "ADJ "は "NOUN "だけにつながっていて、house が品詞分解 (parse) されたときにこのようになる。
6. 最後にパーザはピリオドを呼んで、文章はすべて品詞分解されたことになる。

3.4 文脈自由文法

3.4.1 文法規則

文脈自由文法を理解するためには、ネットワーク文法で使った方法とは全く違う方法で文章の構造を見なければならぬ。文脈自由文法では、文章は各種の項目から成り立っていると考える。項目それぞれはまた別の項目からなっていて、パーザは文章をこれ以上分けられない要素 - 名詞、動詞、形容詞 - などの極小要素にまで分解する。個々の要素がどのように構成されているかを支配している法則は、文法の生成規則 (書き換え規則) と呼ばれている。文脈自由文法のパーザは文章を解析するのに、このルールを使う。

表 3 は文脈自由文法の文法規則と文脈自由文法の単語辞書の例を示している。この規則では右矢印は "生成" を表す。また、この図で NP は "名詞句 (noun phrase) "を、VP は "動詞句 (verb phrase) "を表す。

表 3: 文脈自由文法

文法規則			単語辞書	
SENTENCE(文)	→ NP(名詞句) + VP(動詞句)			
NP(名詞句)	→ DET(冠詞) + NOUN(名詞句)	the	→	DET(冠詞)
NP(名詞句)	→ DET(冠詞) + NP(名詞句)	child	→	NOUN(名詞)
NP(名詞句)	→ PREP(前置詞) + NP(名詞句)	runs	→	VERB(動詞)
NP(名詞句)	→ ADJ(形容詞) + NOUN(名詞)	quickly	→	ADV(副詞)
VP(動詞句)	→ VERB(動詞) + NP(名詞句)	to	→	PREP(前置詞)
VP(動詞句)	→ VPa(動詞句) + NP(名詞句)	the	→	DET(冠詞)
VP(動詞句)	→ VERB(動詞) + ADV(副詞)	large	→	ADJ(形容詞)
VP(動詞句)	→ VERB(動詞)	house	→	NOUN(名詞)
VPa(動詞句)	→ VERB(動詞) + ADV(副詞)			
VPa(動詞句)	→ VERB(動詞) + NP(名詞句)			

名詞句は前置詞句の中では再帰的になる。動詞句は名詞句を呼び出すので間接的には再帰的である。

これらのルールをどのように文章に適用するかを見るために次の図 4 を使って考えてみる。

文脈自由文法は一種の木 (tree) を形成する。この木はパーザが文章を解釈する方法を表すので、パーザ木と呼ばれる。文脈自由文法は実際に構成している句から単語へと構成されているので、個々の単語だけではなく句全体を抽出しやすい。個々の単語のように句も構文解析でき、個々の句がどこから来たのかも知ることができる。

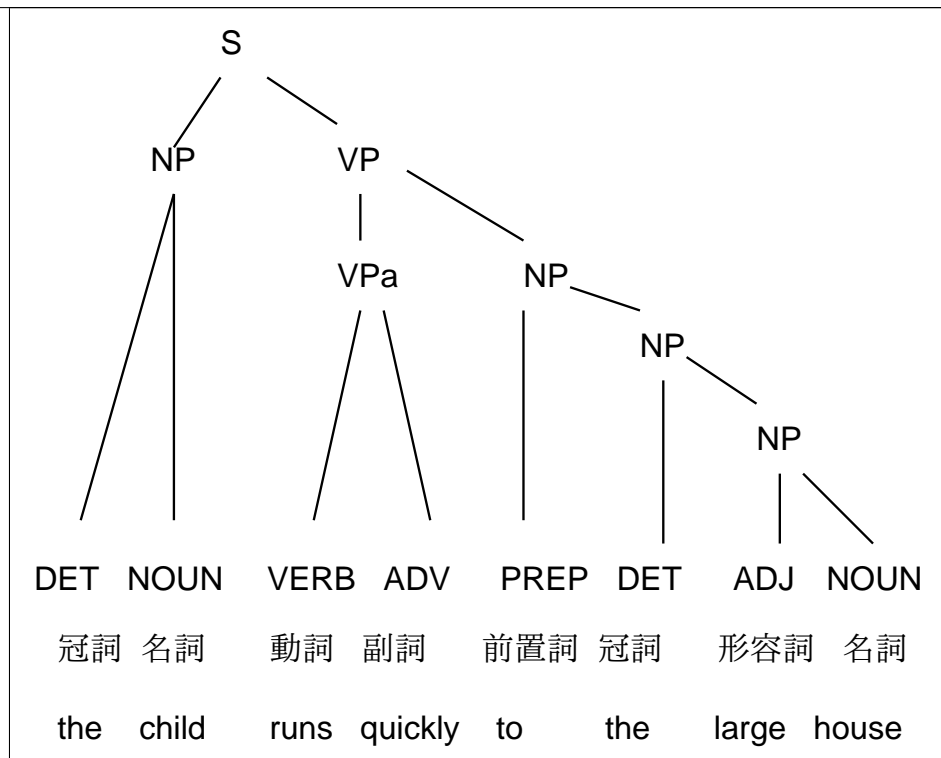


図 4: パーザ木の例

なお，図 4 に示されるような構文解析木は，コンピュータでは表 4 に示されるような括弧を利用した式 (S 式) で表現される．

表 4: S 式の例

<p>(S (NP (DET "the") (NOUN "child")) (VP (VPa (VERB "runs")(ADV "quickly")) NP (PREP "to") (NP (DET "the")(NP (ADJ "large") (NOUN "house"))))))</p>
--

3.4.2 文脈自由文法のパーサの問題

文脈自由文法のパーサには，単純な Top-Down 法，CYK 法，Early 法，LR 法などが知られている．単純な Top-Down パーサは，もっともプログラミングが容易だが，計算効率は悪く，また記述できない生成規則が存在する (左再帰規則)．また，計算量は，文の長さを N とすると， $O(N^N)$ になる．一方 CYK 法は，効率的で計算量は $O(N^3)$ である．Early 法は，もっとも問題がないが，プログラムの作成は困難である．LR 法は，メモリを必要とするが，通常もっとも高速に解析できる．本実験では，Top-Down 法および CYK 法を利用した文脈自由文法の実験を行う．

3.4.3 Top Down 法の動作

Top Down 法の動作は， $S \rightarrow NP + VP$ を初期値として，規則を適用させる．そして，解析が不可能であった場合，バックトラックをおこなって解析を行う．入力文が全て解析できたとき，パーザーは解析を止める．

表 3 におけるパーサーの動作の 1 例を表 9 に示す。表 9 中において \rightarrow は、生成規則が受理されたことを、 \nrightarrow は、生成規則が受理されなかったことを意味する。また、生成規則の添字は、その生成規則の親で統一している。注意してトレースしてもらいたい。

3.4.4 Top Down 法の問題点 (左再帰規則)

テキスト中のパーサは、単純な Top-Down 法のパーサである。このパーサは、プログラム量が少なくすむ反面、文法に制限がある。以下の表 5 のような左再帰の文法は、無限ループにはいるため、記述できない。

表 5: Top-Down パーサでは記述してはいけない文法例 (左再帰規則)

VP(動詞句) \rightarrow VP(動詞句) + NP(名詞句)

3.4.5 CYK 法の動作

CYK 法は、文脈自由文法を解析する 1 つの手法 (パーサ) である。CYK 法は、ボトムアップの left-to-right 型の探索アルゴリズムである。

Top-Down 法と比較すると、大幅に計算量が削減できる。計算量は入力文の長さを n とすると $O(n^3)$ である。また、左再帰規則の制限がない。

表 3 におけるパーサーの動作の 1 例を図 5 に示す。図 5 中において、左の数字は i を、上の数字は j を、意味する。そして、右の数字は計算順序を示している。

まず CYK 法では、 $A_{i,i}, i = 1, n$ において入力単語がはいる。次に、単語にたいする品詞に書き換え規則が動く。計算順序は、下三角行列の下から順に上がっていく形になる。具体的には以下のように計算される。

$A_{1,1}, A_{2,2}, A_{3,3}, A_{4,4}, A_{5,5}, A_{6,6}, A_{7,7}, A_{8,8},$
 $A_{1,2}, A_{2,3}, A_{3,4}, A_{4,5}, A_{5,6}, A_{6,7}, A_{7,8},$
 $A_{1,3}, A_{2,4}, A_{3,5}, A_{4,6}, A_{5,7}, A_{6,8},$
 $A_{1,4}, A_{2,5}, A_{3,6}, A_{4,7}, A_{5,8},$
 $A_{1,5}, A_{2,6}, A_{3,7}, A_{4,8},$
 $A_{1,6}, A_{2,7}, A_{3,8},$
 $A_{1,7}, A_{2,8},$
 $A_{1,8}$

そして、 $A_{i,j}$ には $A_{i,k} A_{k+1,j}$ の書き換え規則が記憶される。例えば、

$$\begin{aligned}
 A_{3,7} &= A_{3,3} A_{4,7} \\
 &= A_{3,4} A_{5,7} \\
 &= A_{3,5} A_{6,7} \\
 &= A_{3,6} A_{7,7}
 \end{aligned}$$

から書き換え規則が適応される。

なお、書き換え規則の後の数字は、どの書き換え規則かを示す通し番号である。この図 5 では $A_{1,3}$ には $A_{2,1}$ の NP_9 と $A_{3,3}$ の VP_3 の書き換え規則 S が入る。

4 実験の内容

4.1 ネットワーク文法の実験

4.1.1 手による解析

1. ネットワーク文法の実験 1

文法規則には表 1 を，単語辞書は表 6 を使用して，“I saw a girl with a telescope” を手で形態素解析しなさい．このとき表 8 に相当するものを書きなさい．

表 6: ネットワーク文法の実験 1 の単語辞書

I	→	NOUN(名詞)
saw	→	VERB(動詞)
a	→	DET(冠詞)
girl	→	NOUN(名詞)
with	→	PREP(前置詞)
telescope	→	NOUN(名詞)

4.1.2 プログラムによる解析

1. ネットワーク文法の実験 2

表 1 において定義されている文法を用いて，“the child runs quickly to the large house.” が，形態素解析されるプログラムを作成しなさい．そのときの出力として，表 2 に示されるような品詞を出力しなさい．

2. ネットワーク文法の実験 3

文法規則には表 1 を，単語辞書は表 6 を使用して，“I saw a girl with a telescope” をプログラムで形態素解析しなさい．そのときの出力として，表 2 に示されるような品詞を出力しなさい．

4.2 文脈自由文法

4.2.1 手による解析

1. 文脈自由文法の実験 1

文法規則には表 7 を，単語辞書は表 6 を使用して，“I saw a girl with a telescope” を手で top down 法で構文解析しなさい．このとき表 9 に相当するものと図 4 に相当するものを書きなさい．また S 式も書きなさい．

2. 文脈自由文法の実験 2

文法規則には表 7 を，単語辞書は表 6 を使用して，“I saw a girl with a telescope” を手で CYK 法で構文解析しなさい．このとき図 4 に相当するものと図 5 に相当するものを書きなさい．

表 7: 文脈自由文法の実験 3 の文法規則

SENTENCE(文)	→	NP(名詞句)	+	VP(動詞句)
NP(名詞句)	→	NOUN(名詞)		
NP(名詞句)	→	DET(冠詞)	+	NOUN(名詞)
VP(動詞句)	→	VERB(動詞)		
VP(動詞句)	→	VERB(動詞)	+	NP(名詞句)
VP(動詞句)	→	VERB(動詞)	+	SS
SS	→	NP(名詞句)	+	PP(前置詞句)
PP(前置詞句)	→	PREP(前置詞)	+	NP(名詞句)

4.2.2 プログラムによる解析

1. 文脈自由文法の実験 3

表 3 において定義されている文法を用いて, “the child runs quickly to the large house.” が, 構文解析されるプログラムを作成しなさい. このとき CYK 法を利用すること, また図 5 に示されるような表と, 表 4 で示されるような S 式を出力しなさい.

2. 文脈自由文法の実験 3

文法規則には表 7 を, 単語辞書は表 6 を使用して, “I saw a girl with a telescope” をプログラムで構文解析しなさい. このとき CYK 法を利用すること, また図 5 に示されるような表と, 表 4 で示されるような S 式を出力しなさい.

4.3 日程

実験は以下の日程で行う.

1. 1 週目

集合場所: 講義室

ネットワーク文法の講義

ネットワーク文法の実験 1

ネットワーク文法のプログラムの仕様書書き

2. 2 週目

集合場所: 講義室

ネットワーク文法のプログラミング

ネットワーク文法の実験 2

ネットワーク文法の実験 3

3. 3 週目

集合場所: 講義室

文脈自由文法の講義

文脈自由文法の実験 1

文脈自由文法の実験 2

文脈自由文法のプログラムの仕様書書き

4. 4,5 週目

集合場所：演習室

文脈自由文法のプログラミング

文脈自由文法の実験 3

5. 6,7 週目

集合場所：講義室

レポート書き

4.4 仕様書

仕様書は、他人がプログラムや仕様書のメンテナンスができるように記述する文章である。仕様書は、外部仕様書と内部仕様書に別れる。それぞれ以下のことを考慮して書くこと。

1. 外部仕様書

外部仕様書とは、プログラム全体の動作を記述したものである。

外部使用書には、入力データや、使用するデータベース、および出力データの概要を記述すること。また、プログラムの各モジュールを記述すること。

2. 内部仕様書

内部仕様書は、各モジュールの詳細を記述したものである。

内部仕様書には、各モジュールの役割と、各モジュールごとの、入力データや、使用するデータベース、および出力データの詳細を記述すること。詳細はコンピュータ入力のための詳細な format も含む。

4.5 プログラム

プログラムは、内部仕様書を書いたのち、以下のことを考慮して書くこと。

1. プログラムと辞書の分離

文法規則と単語辞書は file として作成すること。パーサのプログラムは、文法規則と単語辞書の file を読み込んで動作させること。

2. 分割コンパイルと makefile

プログラムは、分割して、サブルーチンを多く利用すること。分割コンパイルが可能なように makefile を使うこと。

3. ローカル変数とグローバル変数

ローカル変数とグローバル変数を明確に意識して作成すること。

4. コメント

できるだけ、プログラム中にコメントを書くこと。

4.6 レポート

1. 目的

レポートは、自分が行った実験を、他人に理解してもらう報告書である。そのため、自分のメモではなく、他人に読んでもらうことを前提に記述すること。そのために、わかりやすく書くこと。

2. 章構成

必ず章に分割して書くこと。章構成は、最低限の章を含むこと。

- (a) 始めに
- (b) 言語学の概要
- (c) 実験の目的
- (d) 実験の結果
- (e) まとめ
- (f) 考察
- (g) 参考文献

3. 言語学の概要

この章において、以下の言葉の定義を書くこと。また、それぞれの関係を述べること。

- (a) "言葉の意味 "
- (b) "正文 "
- (c) "非文 "
- (d) "受理可能 "
- (e) "受理不可能 "
- (f) "文法 "

4. はじめ 本文 まとめ

本文の目的、実験内容、結果は "はじめに " "本文 " "まとめ " で3回繰り返すこと。

5. 参考書

日本語の文章を書く上で参考になる本

日本語の作文技術 本田勝一 朝日文庫 ISBN4-02-260808-0 540 円

実戦・日本語の作文技術 本田勝一 朝日文庫

6. 考察

レポートの考察には、以下の内容を含むこと

- (a) "I saw a girl with a telescope" には大きくわけて2つの意味(解釈)がある。このときの構文解析結果と、日本語の訳を示せ。また、複数の解析結果を出力できるように文脈自由文法のパーザを作成するには、どのようにしたらよいか考察せよ。
ヒント telescope の使用目的。

-
- (b) “Time flies like an arrow” と “Fruit flies like an apple” の解釈について、以下の項目について考えること。
- i. 翻訳の違い
 - ii. 構文解析の違い
 - iii. 文法規則と単語の登録方法
 - iv. コンピュータでどのようにすれば、曖昧性が解消されるか
- (c) “Time flies like an arrow” をコンピュータで解釈した場合、どれくらいの解釈の数があるが考えてみよ。
- (d) 感想
実験の感想を述べよ。
- (e) 付録
以下のものを付録として提出すること。
- i. hand free の結果
 - ii. Shell
 - iii. makefile
 - iv. ソースコード
 - v. 実行結果、解析途中の結果（正常終了、異常終了と共に）

7. 提出期限

印刷したものを 1507 の TA に週の水曜日の 13 時までに提出すること。また同時に電子化されたものはすべてメールで “murakami@eecs.tottori-u.ac.jp” まで送ること。subject は “実験 III レポート 学籍番号 名前 ” とすること。

— 1 —

4.7 実験のメモ

1. tex

レポートは、tex で記述すること。

2. 図

図をできるだけ、書くこと OpenOffice を利用し、epsf 形式で保存して Tex に書き込むこと。

3. editor

editor は emacs を利用すること。

4. コメント

プログラムには、日本語のコメントを書くこと。

5. 必要な本

以下のコマンドが必要である。必要ならば UNIX の本を持ってくること。

make gcc gdb emacs リダイレクト (<, >) パイプ (|)

以下の関数が必要である。必要ならば C のプログラミングの本を持ってくること。

fopen fgets fscanf printf strcpy strcmp strcat

参考文献

- [1] J. ホップクロフト, J. ウルマン著「オートマトン言語処理計算論」(サイエンス社)
- [2] 長尾真著「言語工学」(昭晃堂)
- [3] 野村浩郷著「自然言語処理の基本技術」電子情報通信学会(コロナ社)
- [4] 田中穂積著「自然言語解析の基礎」(産業図書)
- [5] H. シュルト著「Cで学ぶAI」
- [6] 長尾真「自然言語処理」(岩波書店) 1996

表 8: NetWork 文法の動作

start										
		the								
start	DET									
		the	child							
start	DET	NOUN								
		the	child	runs						
start	DET	NOUN	VERB							
		the	child	runs	quickly					
start	DET	NOUN	VERB	ADV						
		the	child	runs	quickly	to				
start	DET	NOUN	VERB	ADV	PREP					
		the	child	runs	quickly	to	the			
start	DET	NOUN	VERB	ADV	PREP	DET				
		the	child	runs	quickly	to	the	large		
start	DET	NOUN	VERB	ADV	PREP	DET	ADJ			
		the	child	runs	quickly	to	the	large	house	
start	DET	NOUN	VERB	ADV	PREP	DET	ADJ	NOUN		
		the	child	runs	quickly	to	the	large	house	.
start	DET	NOUN	VERB	ADV	PREP	DET	ADJ	NOUN	end	

表 9: Top Down パーザの動作

1	S_1	\rightarrow	$NP_1 + VP_1$	
2	NP_1	\rightarrow	$DET_2 + NOUN_2$	1 の NP を処理する .
3	DET_2	\rightarrow	the	
4	$NOUN_2$	\rightarrow	child	1 の NP が受理されたので VP を処理する .
5	VP_1	\rightarrow	$VERB_5 + NP_5$	
6	$VERB_5$	\rightarrow	runs	
7	NP_5	\rightarrow	$DET_7 + NOUN_7$	
8	DET_7	\rightarrow	quickly	
9	NP_5	\rightarrow	$DET_7 + NP_7$	
10	DET_7	\rightarrow	quickly	
11	NP_5	\rightarrow	$PREP_{11} + NP_{11}$	
12	$PREP_{11}$	\rightarrow	quickly	
13	NP_5	\rightarrow	$ADJ_{13} + NOUN_{13}$	
14	ADJ_{13}	\rightarrow	quickly	
15	VP_1	\rightarrow	$VP_{a15} + NP_{15}$	
16	VP_{a15}	\rightarrow	$VERB_{16} + ADV_{16}$	
17	$VERB_{16}$	\rightarrow	runs	
18	ADV_{16}	\rightarrow	quickly	
19	NP_{15}	\rightarrow	$DET_{19} + NOUN_{19}$	
20	DET_{19}	\rightarrow	to	
21	NP_{15}	\rightarrow	$DET_{21} + NP_{21}$	
22	DET_{19}	\rightarrow	to	
23	NP_{15}	\rightarrow	$PREP_{23} + NP_{23}$	
24	$PREP_{23}$	\rightarrow	to	24 が受理
25	NP_{23}	\rightarrow	$DET_{25} + NOUN_{25}$	
26	DET_{25}	\rightarrow	the	
27	$NOUN_{25}$	\rightarrow	large	26 が受理されず .
28	NP_{23}	\rightarrow	$DET_{28} + NP_{28}$	
29	DET_{28}	\rightarrow	the	25 が受理
30	NP_{28}	\rightarrow	$DET_{30} + NOUN_{30}$	
31	DET_{30}	\rightarrow	large	
32	NP_{28}	\rightarrow	$DET_{32} + NP_{32}$	
33	DET_{32}	\rightarrow	large	
34	NP_{32}	\rightarrow	$PREP_{34} + NP_{34}$	
35	$PREP_{34}$	\rightarrow	large	
36	NP_{32}	\rightarrow	$ADJ_{36} + NOUN_{36}$	
37	ADJ_{36}	\rightarrow	large	
38	$NOUN_{36}$	\rightarrow	house	

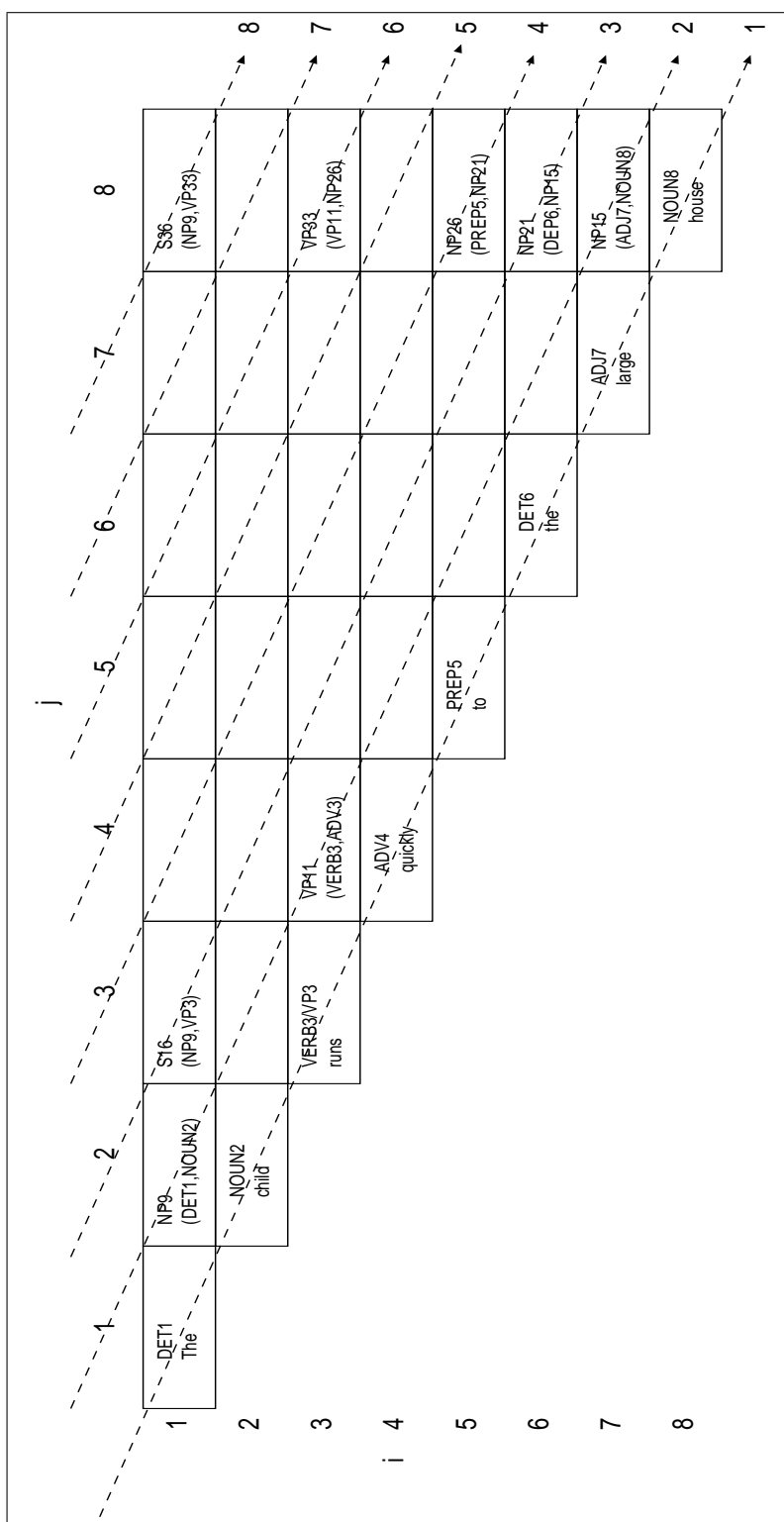


図 5: CKY パーザの動作