

自然言語処理システム開発環境 LangLAB

東京工業大学 情報工学科 田中研究室

1. 目的

自然言語処理システムを効率良く開発するためのよりよい環境を提供すること

2. 構成

- ・ 左外置変形に関する記述が容易な文法形式 XGS を BUP-XG 節に変換するトランスレータ
- ・ DCG 形式で記述された辞書を TRIE 構造の辞書に変換するトランスレータ
- ・ 文法の誤りの検出、修正を助けるデバッガ (開発中)

3. 特徴

- ・ 左外置変形を考慮した文法形式 (XGS) が扱える
- ・ DCKR による意味処理の記述ができる
- ・ 時間的・空間的効率化と熟語処理のために TRIE 構造の辞書を導入

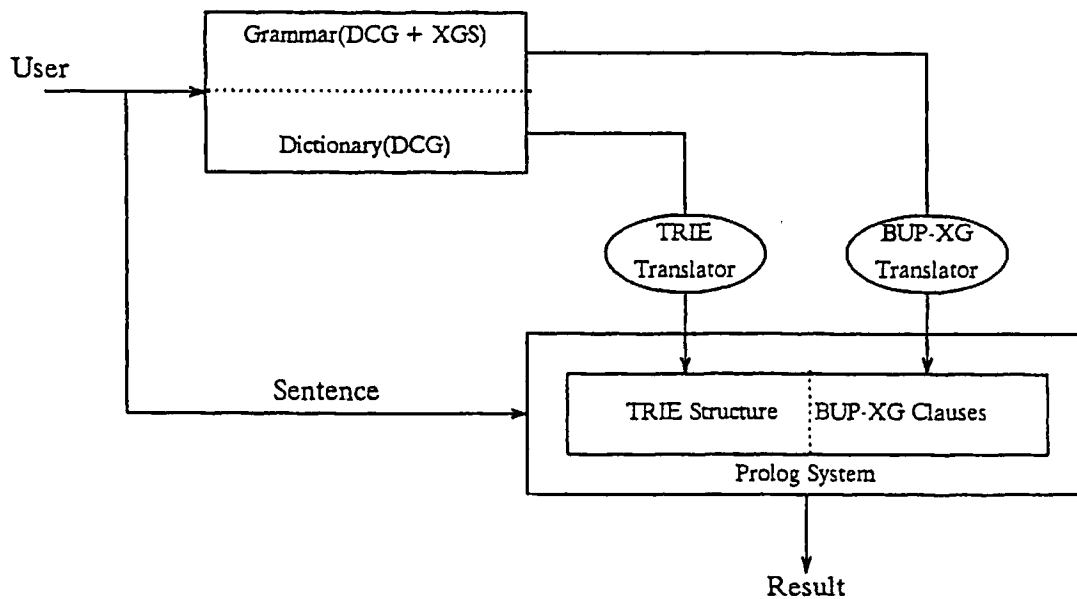


図1 LangLAB の構成

4. 応用例1 英文構文解析システム

4.1. 特徴

- ・ 文法規則数は約200
- ・ XGSの特徴を積極的に利用している
- ・ 統一的な構文情報の検査プログラムで構文木を減らしている

4.2. 辞書記述例

```
adj([[cmp]' + '],additional) --> [additional].
adv([[cmp]' - '],alone) --> [alone].
art([[cnt]' + '],[sgl]' + '],[spl][a]],a) --> [a].
n([[cnt]' - '],acid) --> [acid].
np(B,[]) --> [not,only],np(A,_),[but,also],np(B,_).
v([[pat,5,9,13,15,16,17,18,19],[p,for,of]],allow) --> [allow].
```

4.3. 文法記述例

```
ncomp_t([],Ncomp_t_S) -->
  srel(_,Srel_S).

np(Np_A,Np_S) ==>
  detp(Detp_A,Detp_S),
  nomhd(Nomhd_A,Nomhd_S), { unify(Detp_A,Nomhd_A,Np_A) },
  ( [] ;
    ( ncomp(_,Ncomp_S) ; ncomp_t(_,Ncomp_t_S)../np(Np_A,Np1_S) )
  ).

srel(Sd_A,Srel_S) -->
  < relpro(_,Relpro_S) >,
  sd(Sd_A,Sd_S).
```

参考文献

- 田中穂積他,"自然言語処理のためのソフトウェアシステム LangLAB", in Proc. *The Logic Programming Conference '86*,pp.5-12,1986.
- Matsumoto Y. et.al.,"BUP-A Bottom-up Parser Embedded in Prolog", *New Generation Computing*, Vol.1, No.2, pp.145-158,1983.
- 今野聡,田中穂積,"左外置を考慮したボトムアップ構文解析システム", コンピュータソフトウェア,岩波書店,pp.19-29,1986.
- Pereira,F. and Warren,D.,"Definite Clause Grammar for Language Analysis - A Survey of the Formalism and a Comparison with Augmented Transition Networks",*Journal of Artificial Intelligence*,Vol.13,pp.231-278,1980.

```

|-bun
  |-bun
    |-kouchishiku
    | |-meishiku
    | | |-daimeishi -- 私
    | |-joshi
    | |-kakujoshi -- が
    |-bun
      |-kouchishiku
      | |-meishiku
      | | |-daimeishi -- 彼
      | |-joshi
      | |-kakujoshi -- に
      |-bun
        |-kouchishiku
        | |-meishiku
        | | |-meishi -- ドア
        | |-joshi
        | |-kakujoshi -- を
        |-bun
          |-jutsugo
          | |-doushi1
          | | |-doushi
          | | |-gokan -- 開け
          |-sieki
          | |-gokan -- させ
          |-kako
          | |-gokan -- た

```

私が彼にドアを開けさせた。

Total Processing Time = 6017 sec.

```

number of tree was : 1.
number of wf_goal was : 31.
number of wf_dict was : 16.
number of fail_goal was : 41.

```

he#19::

door#21::

det:the

the

```
[vp18~make#18,(tense:past,agent:i#16,goal:(vp6ab~open#20),individual_of(make)),
 [past],[i#16,individual_of(i)],
 [vp6ab~open#20,(agent:he#19,object:door#21,individual_of(open)),
 [he#19,individual_of(he)],
 [door#21,(det:the,individual_of(door)),[the]]]]
```

(akeru#20)@sieki#18::

goal:akeru#20

tense:past

agent:watashi#16

akeru#20::

agent:kare#19

object:doa#21

kare#19::

doa#21::

past

watashi#16::

```
[(akeru#20)@sieki#18,
 (goal:akeru#20,tense:past,agent:watashi#16,individual_of((akeru#20)@sieki)),
 [akeru#20,(agent:kare#19,object:doa#21,individual_of(akeru)),
 [kare#19,individual_of(kare)],[doa#21,individual_of(doa)]],
 [past],[watashi#16,individual_of(watashi)]]
```

5. 応用例2 機械翻訳システム

5.1. 特徴

- ・ トランスファー方式である
- ・ 文法規則は変換により解析と生成の両方に使用できる
- ・ 意味構造変換規則の記述が簡単である

5.2. 解析例

Input sentences
i made him open the door.

No. 1

```
| - sentence
  | - sdec
    | - subj
      | - np
        | - pron -- i
      | - vp
        | - v -- made
        | - obj
          | - np
            | - pron -- him
          | - infinitive1
            | - vp
              | - v -- open
              | - obj
                | - np
                  | - ddet
                    | - the -- the
                  | - nomhd
                    | - n -- door
```

```
vp18~make#18::
  tense:past
  agent:i#16
  goal:(vp6ab~open#20)
```

past

i#16::

```
vp6ab~open#20::
  agent:he#19
  object:door#21
```

4.5. 解析速度

Interpreted Code of C-Prolog			
No.	Time [msec]		Tree
	First	Total	
1	1149	1249	1
2	1833	2166	1
3	2166	4649	3
4	4799	5666	1
5	4683	5899	3
6	9416	11283	2
7	14483	21400	6
8	12000	12133	2
9	11350	21400	7

例文

1. I open the window.
2. Diagram is an augmented grammar.
3. The structural relations are holding among constituents.
4. It is not tied to a particular domain of applications.
5. Diagram analyzes all of the basic kinds of phrases and sentences.
6. This paper presents an explanatory overview of a large and complex grammar that is used in a sentence.
7. The annotations provide important information for other parts of the system that interpret the expression in the context of a dialogue.
8. For every expression it analyzes, diagram provides an annotated description of the structural relations holding among its constituents.
9. Procedures can also assign scores to an analysis, rating some applications of a rule as probable or as unlikely.

4.4. 解析例

Input sentences

i open the door which he made.

No. 1 time : 2100.01 msec

```
| - sentence
  | - sdec
    | - sd
      | - np
        | | - ppron -- i
          | - vp
            | - tensed_v
              | - v_comp
                | - v -- open
                  | - np
                    | - detp
                      | | - det
                        | | - art -- the
                          | - nomhd
                            | | - n -- door
                              | - ncomp_t
                                | - srel
                                  | - relpro -- which
                                    | - sd
                                      | - np
                                        | | - ppron -- he
                                          | - vp
                                            | - tensed_v
                                              | - v_comp
                                                | - v -- made
                                                  | - np -- t
```

> total time : 3116.7 msec

> tree : 1

> wf_goal : 23

> wf_dict : 9

> fail_goal : 27
=====