程式人

邏輯語為何從 BNF 轉向 PEG ？

lojban

陳鍾誠

ccckmit@gmail.com

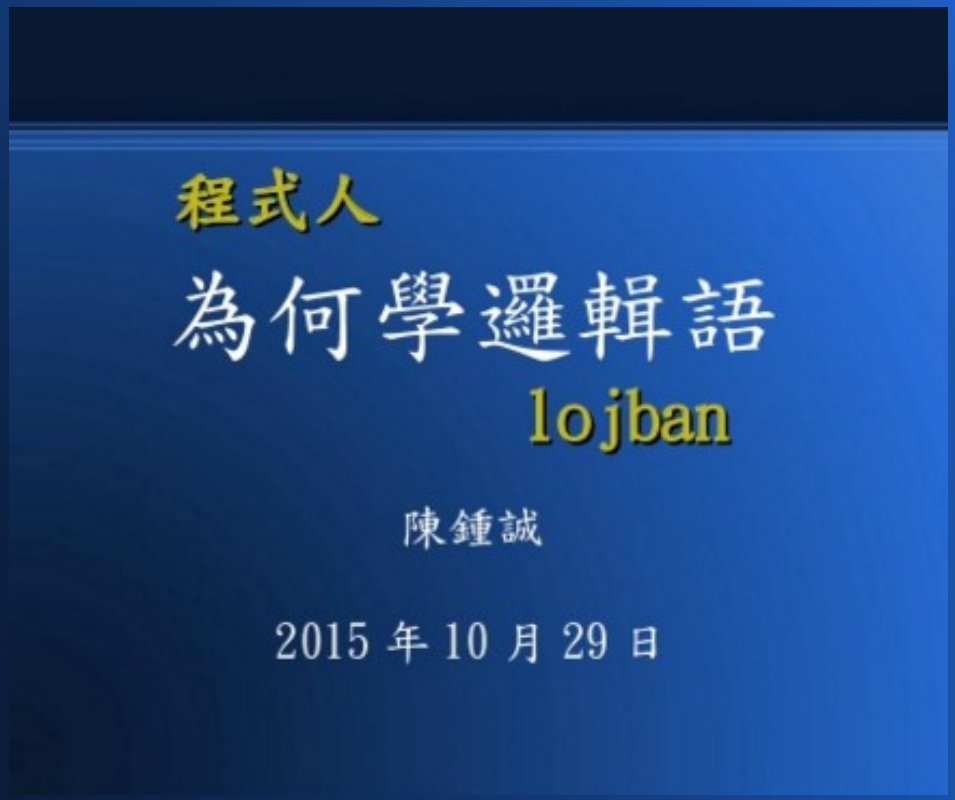
2015 年 11 月 5 日

話說

- 2015 年 10 月的某一天
- 我遇到了邏輯語

詳情請看

- http://www.slideshare.net/ccckmit/lojban



經過了一週的學習之後

我發現

- 目前的邏輯語社群，有一個很嚴重的爭議

這個爭議

- 是有關邏輯語的語法
- 以及邏輯語的 parse 問題

語法有甚麼問題呢？

說來話長

首先、讓我們看一下

- 邏輯語的 BNF

BNF ?

- Backus-Naur Form
- 一種用來描述語法的語法
- 可以說是一種
  - Meta Language

邏輯語的 BNF 位於

https://mw.lojban.org/papri/Lojban_Formal_Grammars

Lojban

Search

Lojban Formal Grammars

3rd Baseline

The 3rd official Baseline of Lojban grammar is available in several formats:

- Extended BNF format (bnf.300)
- YACC format (grammar.300)
- a change-by-change commentary (techfix.300)

The Extended BNF and YACC grammars may be found in HTML format, with indexes for each tag, in Chapter 21 of the Lojban Reference Grammar. As the baseline grammar document, this is the most authoritative source.

If the BNF and YACC versions do not agree, the YACC takes precedence.

Proposed 4th Baseline

.alyn.post. and BPFK are working on a proposal for a 4th Baseline grammar. The 4th Baseline Lojban Machine Grammar will replace the YACC and BNF grammars with a PEG grammar.

你可以看到

- 目前語法為第三版
  - 3rd baseline
- 而且正在提出第四版的草案

問題是

- 第三版的語法，是在 1997 年定義的

```
Lojban Machine Grammar, EBNF Version, 3rd Baseline as of 10 January 1997

This document is explicitly dedicated to the public domain by its author,
the Logical Language Group Inc.  Contact that organization at:
2904 Beau Lane, Fairfax VA 22031 USA 703-385-0273 (intl: +1 703 385 0273)

Explanation of notation:  All rules have the form:

        name<number> = bnf-expression

which means that the grammatical construct "name" is defined by
"bnf-expression".  The number cross-references this grammar with
the rule numbers in the YACC grammar.  The names are the same as those
in the YACC grammar, except that subrules are labeled with A, B, C, ... in the
YACC grammar and with 1, 2, 3, ... in this grammar.  In addition, rule 971 is
"simple_tag" in the YACC grammar but "stag" in this grammar, because of its
frequent appearance.
```

距離現在已經有 18 年了

那麼、為何要定義第四版呢？

當然

- 這代表語法不夠完美
- 而且、是很不完美

為何不完美？

讓我們看一下語法

```
text<0> = [NAI ...] [CMENE ... # | (indicators & free ...)] [joik-jek] text-1

text-1<2> = [(I [jek | joik] [[stag] BO] #) ... | NIhO ... #] [paragraphs]

paragraphs<4> = paragraph [NIhO ... # paragraphs]

paragraph<10> = (statement | fragment) [I # [statement | fragment]] ...

statement<11> = statement-1 | prenex statement

statement-1<12> = statement-2 [I joik-jek [statement-2]] ...

statement-2<13> = statement-3 [I [jek | joik] [stag] BO # [statement-2]]

statement-3<14> = sentence | [tag] TUhE # text-1 /TUhU#/

fragment<20> = ek # | gihek # | quantifier | NA # | terms /VAU#/ | prenex |
        relative-clauses | links | linkargs

prenex<30> = terms ZOhU #

sentence<40> = [terms [CU #]] bridi-tail

subsentence<41> = sentence | prenex subsentence
```

您可以看到

- 很多項目還分 1, 2, 3, …

```
sentence<40> = [terms [CU #]] bridi-tail

subsentence<41> = sentence | prenex subsentence

bridi-tail<50> = bridi-tail-1
        [gihek [stag] KE # bridi-tail /KEhE#/ tail-terms]

bridi-tail-1<51> = bridi-tail-2 [gihek # bridi-tail-2 tail-terms] ...

bridi-tail-2<52> = bridi-tail-3 [gihek [stag] BO # bridi-tail-2 tail-terms]

bridi-tail-3<53> = selbri tail-terms | gek-sentence

gek-sentence<54> = gek subsentence gik subsentence tail-terms |
        [tag] KE # gek-sentence /KEhE#/ | NA # gek-sentence

tail-terms<71> = [terms] /VAU#/

terms<80> = terms-1 ...

terms-1<81> = terms-2 [PEhE # joik-jek terms-2] ...

terms-2<82> = term [CEhE # term] ...
```

這還不打緊、問題是

- 當 BNF 要轉為 parser 程式時
- 通常要使用像 YACC, Bison 這樣的工具

於是、語法要改寫

- 以符合 YACC, Bison 的寫法

Lojban Formal Grammars

3rd Baseline

The 3rd official Baseline of Lojban grammar is available in several formats:

- Extended BNF format (bnf.300)
- YACC format (grammar.300)
- a change-by-change commentary (techfix.300)

The Extended BNF and YACC grammars may be found in HTML format, with indexes for each tag, in Chapter 21 of the Lojban Reference Grammar. As the baseline grammar document, this is the most authoritative source.

If the BNF and YACC versions do not agree, the YACC takes precedence.

而這個改寫後的版本

- 稱為 grammar.300

```
bridi_tail_50          :  bridi_tail_A_51
                       |  bridi_tail_A_51  GIhEK_KE_814  bridi_tail_50
                                 KEhE_gap_466  tail_terms_71
                       ;

bridi_tail_A_51        :  bridi_tail_B_52
                       |  bridi_tail_A_51  GIhEK_818  bridi_tail_B_52
                          tail_terms_71
                       ;

bridi_tail_B_52        :  bridi_tail_C_53
                       |  bridi_tail_C_53  GIhEK_BO_813  bridi_tail_B_52
                          tail_terms_71
                       ;

bridi_tail_C_53        :  gek_sentence_54
                       |  selbri_130  tail_terms_71
                       ;

gek_sentence_54        :  GEK_807  subsentence_41
                               GIK_816  subsentence_41  tail_terms_71
                       |  tag_491  KE_493  gek_sentence_54  KEhE_gap_466
                       |  NA_445  gek_sentence_54
```

接著有個叫 John Cowan 的人

- 用它創建了一個 parser

Official LLG Parser

Official LLG Parser

The files in this directory are the current distribution of the LLG official Lojban parser, in source form and MS-DOS executable.

Basically, there are two files. /parser.zip contains the MS-DOS executable, whereas /parser.shar.gz contains the source code. To compile the parser from the source code, just say "cc -o parser *.c"; there is no Makefile. The code is in K&R C, but should compile all right using an ANSI compiler. If you have trouble unpacking the ZIP file, use PKZipFix to convert it to MS-DOS ZIP format.

Both versions contain the 'yacc' grammar, the current E-BNF grammar, and a list of technical fixes updating the 2nd baseline grammar. You can process the 'yacc' version with yacc or bison, but do not attempt to use the resulting y.tab.c file in the parser -- the provided grammar.c has been extensively hacked both before and after 'yacc'ing to make it work in the parser.

Much newer version 3.0 resides on John Cowan's site, look for file http://home.ccil.org/~cowan/parser-3.0.00.tar.gz No idea about its official status although.

奇怪的是、這個 parser

- 目前還停留在 DOS 版

而支援 web 的 parser

- 則不是 John Cowan 寫的

Parsers

- http://mw.lojban.org/extensions/ilmentufa/camxes.html - Ilmen's parser (official grammar)
- ilmentufa - Ilmen's parser (both official and experimental grammar)
- la tersmu
- The Official Parser
- (Java) camxes
- www.lojban.org/jboski - A web-based version of of Richard Curnow's jbofihe; acts as a Lojban to English translator. Made by Raphal Poss.
- zantufa, "zabna" parser based on experimental grammars with clear versioning.

Only experimental grammar

- (Haskell) iocixes: Created by la.iocikun. based on zasni gerna with MEX grammar proposal of la xorxes. (zasni gerna peg)
- (JavaScript) ilmentufa: Created by la.ilmen. based on the mhagiwara's camxes.js with plenty of experimental propositions; because the core grammar is camxes.js, it does not include all the debuggings that were done in zasni gerna by la xorxes. (ilmentufa's peg)
- (JavaScript) Another frontend of ilmentufa: Created by la.uilym. based on ilmentufa. It is not necessarily based on the latest version of ilmentufa's peg.

雖然、這很正常

- 寫 DOS 程式不會寫 WEB 的人
- 隨便找都很多

但是

- 新版的 parser
  - 有些不支援舊版的標準語法(3.0)
  - 只支援 4.0 的實驗性語法

還好

- ilmen 寫的 ilmentufa

  有支援舊版的正式官方語法

讓我們來用用看

mw.lojban.org/extensions/ilmentufa/camxes.html

Type any Lojban text in the following textarea. The result will be parsed as you type:

```
mi klama vi vu
```

Output mode: Prettified [Grammar file] [Switch to experimental grammar]

```
([mi CU] [klama {<vi KU> <vu KU>} VAU])
```

看起來好像不錯

- 但是如果我們切到 raw output 模式

您會看到這個畫面

mw.lojban.org/extensions/ilmentufa/camxes.html

Type any Lojban text in the following textarea. The result will be parsed as you type:

```
mi klama vi vu
```

Output mode: Raw output [Grammar file] [Switch to experimental grammar]


```
[
  "text",
  [
    "text_1",
    [
      "paragraphs",
      [
        "paragraph",
        [
          "statement",
          [
            "statement_1",
            [
              "statement_2",
              [
                "statement_3",
                [
                  "sentence",
                  [
                    [
                      "terms",
                      [
                        "terms_1",
                        [
```

再看深一點

mw.lojban.org/extensions/ilmentufa/camxes.html

Type any Lojban text in the following textarea. The result will be parsed as you type:

```
mi klama vi vu
```

Output mode: Raw output [Grammar file] [Switch to experimental grammar]

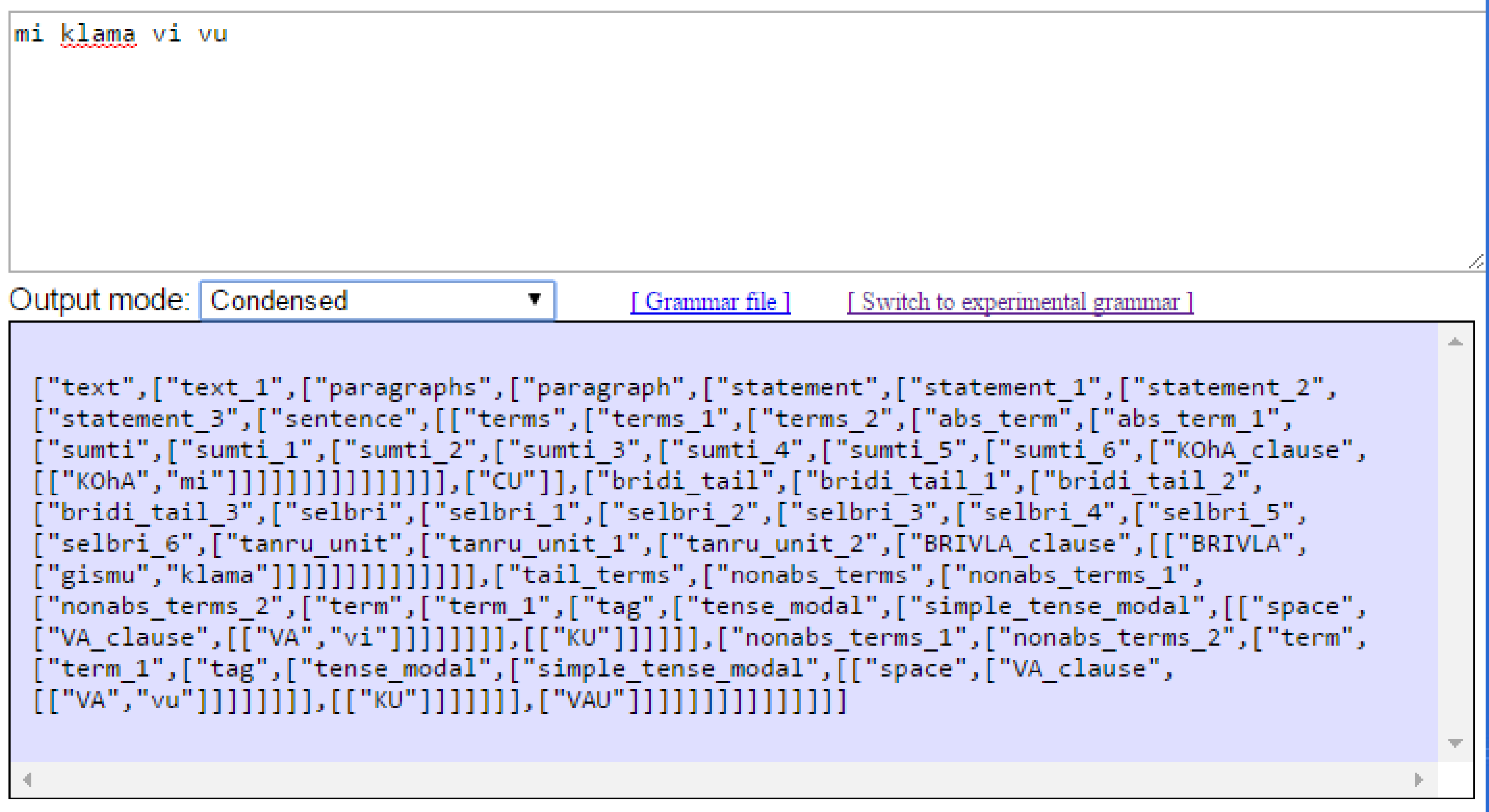

```
[
  "abs_term_1",
  [
    "sumti",
    [
      "sumti_1",
      [
        "sumti_2",
        [
          "sumti_3",
          [
            "sumti_4",
            [
              "sumti_5",
              [
                "sumti_6",
                [
                  "KOhA_clause",
                  [
                    [
                      "KOhA",
                      "mi"
                    ]
                  ]
                ]
              ]
```

發現甚麼問題了嗎？

- 舊版的剖析樹很深、很深

如果改成 condensed 模式

- 會看到這個畫面



mi klama vi vu

Output mode: Condensed

```
["text",["text_1",["paragraphs",["paragraph",["statement",["statement_1",["statement_2",
["statement_3",["sentence",[["terms",["terms_1",["terms_2",["abs_term",["abs_term_1",
["sumti",["sumti_1",["sumti_2",["sumti_3",["sumti_4",["sumti_5",["sumti_6",["KOhA_clause",
[["KOhA","mi"]]]]]]]]]]]]]]],["CU"]],["bridi_tail",["bridi_tail_1",["bridi_tail_2",
["bridi_tail_3",["selbri",["selbri_1",["selbri_2",["selbri_3",["selbri_4",["selbri_5",
["selbri_6",["tanru_unit",["tanru_unit_1",["tanru_unit_2",["BRIVLA_clause",[["BRIVLA",
["gismu","klama"]]]]]]]]]]]]]],["tail_terms",["nonabs_terms",["nonabs_terms_1",
["nonabs_terms_2",["term",["term_1",["tag",["tense_modal",["simple_tense_modal",[["space",
["VA_clause",[["VA","vi"]]]]]]]],[["KU"]]]]]],["nonabs_terms_1",["nonabs_terms_2",["term",
["term_1",["tag",["tense_modal",["simple_tense_modal",[["space",["VA_clause",
[["VA","vu"]]]]]]]],[["KU"]]]]]]],["VAU"]]]]]]]]]]]]]]]
```

很難看懂！

這還不打緊

- 假如我們將《北風與太陽》
  http://mw.lojban.org/extensions/ilmentufa/o/
  這篇故事的邏輯語，放進去
  剖析

會得到這個結果

mw.lojban.org/extensions/ilmentufa/camxes.html

Type any Lojban text in the following textarea. The result will be parsed as you type:

```
ni'o la berti brife jo'u la solri pu troci lo ka cuxna lo poi me vo'a vau traji be lo ka vlipa i
ca'o bo lo pa litru noi dasni lo glare kosta cu klama
i le re mei pu simxu lo ka tugni fi lo nu lo traji be lo ka clira fa lo nu ce'u snada lo ka
gasnu lo nu le litru co'u dasni le kosta cu traji lo ka vlipa
i ba bo la berti brife co'a traji cupra lo brife i ku'i lo nu by by zenba lo ka cupra lo xo kau
brife cu rinka lo nu le litru zu'e ri zenba lo ka se tagji le kosta i ba bo la berti brife co'u
troci i ba bo la solri co'a glare dirce i ba zi bo le litru co'u dasni le kosta
i se ki'u bo la berti brife cu bilga lo ka tugni fi lo nu la solri cu traji lo ka vlipa
```

Output mode: Condensed [Grammar file] [Switch to experimental grammar]

```
SyntaxError: Expected [,] but "v" found.
```

還好、只要我們將換行拿掉

- 變成這樣

mw.lojban.org/extensions/ilmentufa/camxes.html

Type any Lojban text in the following textarea. The result will be parsed as you type:

```
ni'o la berti brife jo'u la solri pu troci lo ka cuxna lo poi me vo'a vau traji be lo ka vlipa i ca'o bo lo
pa litru noi dasni lo glare kosta cu klama i le re mei pu simxu lo ka tugni fi lo nu lo traji be lo ka clira
fa lo nu ce'u snada lo ka gasnu lo nu le litru co'u dasni le kosta cu traji lo ka vlipa i ba bo la berti
brife co'a traji cupra lo brife i ku'i lo nu by by zenba lo ka cupra lo xo kau brife cu rinka lo nu le litru
zu'e ri zenba lo ka se tagji le kosta i ba bo la berti brife co'u troci i ba bo la solri co'a glare dirce i
ba zi bo le litru co'u dasni le kosta i se ki'u bo la berti brife cu bilga lo ka tugni fi lo nu la solri cu
traji lo ka vlipa
```

就可以正常 parse 了

ni'o la berti brife jo'u la solri pu troci lo ka cuxna lo poi me vo'a vau traji be lo ka vlipa i ca'o bo lo pa litru noi dasni lo glare kosta cu klama i le re mei pu simxu lo ka tugni fi lo nu lo traji be lo ka clira fa lo nu ce'u snada lo ka gasnu lo nu le litru co'u dasni le kosta cu traji lo ka vlipa i ba bo la berti brife co'a traji cupra lo brife i ku'i lo nu by by zenba lo ka cupra lo xo kau brife cu rinka lo nu le litru zu'e ri zenba lo ka se tagji le kosta i ba bo la berti brife co'u troci i ba bo la solri co'a glare dirce i ba zi bo le litru co'u dasni le kosta i se ki'u bo la berti brife cu bilga lo ka tugni fi lo nu la solri cu traji lo ka vlipa

Output mode: Condensed [Grammar file] [Switch to experimental grammar]


```
["text",["text_1",[["NIhO_clause",[["NIhO","ni'o"]]]],["paragraphs",["paragraph",["statement",
["statement_1",["statement_2",["statement_3",["sentence",[["terms",["terms_1",["terms_2",["abs_term",
["abs_term_1",["sumti",["sumti_1",["sumti_2",["sumti_3",["sumti_4",["sumti_5",["sumti_6",["LA_clause",
[["LA","la"]]],["sumti_tail",["sumti_tail_1",["selbri",["selbri_1",["selbri_2",["selbri_3",["selbri_4",
["selbri_5",["selbri_6",["tanru_unit",["tanru_unit_1",["tanru_unit_2",["BRIVLA_clause",[["BRIVLA",
["gismu","berti"]]]]]]]]]],["selbri_4",["selbri_5",["selbri_6",["tanru_unit",["tanru_unit_1",
["tanru_unit_2",["BRIVLA_clause",[["BRIVLA",["gismu","brife"]]]]]]]]]]]]]]]],["KU"]]]]],[[["joik_ek",
["joik_ek_1",["joik",["JOI_clause",[["JOI","jo'u"]]]]]],["sumti_3",["sumti_4",["sumti_5",["sumti_6",
["LA_clause",[["LA","la"]]],["sumti_tail",["sumti_tail_1",["selbri",["selbri_1",["selbri_2",["selbri_3",
["selbri_4",["selbri_5",["selbri_6",["tanru_unit",["tanru_unit_1",["tanru_unit_2",["BRIVLA_clause",
[["BRIVLA",["gismu","solri"]]]]]]]]]]]]]]]],["KU"]]]]]]]]]]]]]]]],["CU"]],["bridi_tail",["bridi_tail_1",
["bridi_tail_2",["bridi_tail_3",["selbri",["tag",["tense_modal",["simple_tense_modal",[["time",
[["time_offset",["PU_clause",[["PU","pu"]]]]]]]]]],["selbri_1",["selbri_2",["selbri_3",["selbri_4",
["selbri_5",["selbri_6",["tanru_unit",["tanru_unit_1",["tanru_unit_2",["BRIVLA_clause",[["BRIVLA",
["gismu","troci"]]]]]]]]]]]]]]]],["tail_terms",["nonabs_terms",["nonabs_terms_1",["nonabs_terms_2",["term",
["term_1",["sumti",["sumti_1",["sumti_2",["sumti_3",["sumti_4",["sumti_5",["sumti_6",["LE_clause",
[["LE","lo"]]],["sumti_tail",["sumti_tail_1",["selbri",["selbri_1",["selbri_2",["selbri_3",["selbri_4",
["selbri_5",["selbri_6",["tanru_unit",["tanru_unit_1",["tanru_unit_2",["NU_clause",[["NU","ka"]]],
["subsentence",["sentence",["bridi_tail",["bridi_tail_1",["bridi_tail_2",["bridi_tail_3",["selbri",
["selbri_1",["selbri_2",["selbri_3",["selbri_4",["selbri_5",["selbri_6",["tanru_unit",["tanru_unit_1",
["tanru_unit_2",["BRIVLA_clause",[["BRIVLA",["gismu","cuxna"]]]]]]]]]]]]]]]],["tail_terms",["nonabs_terms",
["nonabs_terms_1",["nonabs_terms_2",["term",["term_1",["sumti",["sumti_1",["sumti_2",["sumti_3",["sumti_4",
["sumti_5",["sumti_6",["LE_clause",[["LE","lo"]]],["sumti_tail",["relative_clauses",["relative_clause",
["relative_clause_1",["NOI_clause",[["NOI","poi"]]],["subsentence",["sentence",["bridi_tail",
```

如果用 prettified 模式

結果會好看很多！

Type any Lojban text in the following textarea. The result will be parsed as you type:

```
ni'o la berti brife jo'u la solri pu troci lo ka cuxna lo poi me vo'a vau traji be lo ka vlipa i ca'o bo lo
pa litru noi dasni lo glare kosta cu klama i le re mei pu simxu lo ka tugni fi lo nu lo traji be lo ka clira
fa lo nu ce'u snada lo ka gasnu lo nu le litru co'u dasni le kosta cu traji lo ka vlipa i ba bo la berti
brife co'a traji cupra lo brife i ku'i lo nu by by zenba lo ka cupra lo xo kau brife cu rinka lo nu le litru
zu'e ri zenba lo ka se tagji le kosta i ba bo la berti brife co'u troci i ba bo la solri co'a glare dirce i
ba zi bo le litru co'u dasni le kosta i se ki'u bo la berti brife cu bilga lo ka tugni fi lo nu la solri cu
traji lo ka vlipa
```

Output mode: Prettified [Grammar file] [Switch to experimental grammar]

```
(ni'o [{<(¹[la {berti brife} KU] [jo'u {la solri KU}]¹) CU> <pu troci> <lo (¹ka [cuxna {lo <(²poi [{me vo'a
MEhU} vau] KUhO²) (²traji [be {lo <ka (³vlipa VAU³) KEI> KU} BEhO]²)> KU} VAU] KEI¹) KU> VAU} {i ca'o bo
<lo (¹[pa BOI] litru [noi {dasni <lo (²glare kosta²) KU> VAU} KUhO]¹) KU> cu} {klama VAU}] [i {<(¹le [re
mei] KU¹) CU> <pu simxu> <lo (¹ka [tugni {fi <lo (²nu [{<lo (³traji [be {lo <ka (⁴clira [fa {lo <nu (⁵[ce'u
CU] [snada {lo <ka (⁶gasnu [lo {nu <(⁷[le litru KU] CU⁷) (⁷co'u dasni⁷) (⁷le kosta KU⁷) VAU> KEI} KU] VAU⁶)
KEI> KU} VAU]⁵) KEI> KU}] VAU⁴) KEI> KU} BEhO]³) KU> cu} {traji <lo (³ka [vlipa VAU] KEI³) KU> VAU}] KEI²)
KU>} VAU] KEI¹) KU> VAU} {i ba bo <la (¹berti brife¹) KU> CU} {co'a <traji cupra>} {lo brife KU} VAU] [i
ku'i] [{<lo (¹nu [{<(²by by²) BOI> CU} {zenba <lo (²ka [cupra {lo <(³[xo kau] BOI³) brife> KU} VAU] KEI²)
KU> VAU}] KEI¹) KU> cu} {rinka <lo (¹nu [{<(²le litru KU²) (²zu'e ri²)> CU} {zenba <lo (²ka [{se tagji} {le
kosta KU} VAU] KEI²) KU> VAU}] KEI¹) KU> VAU}] [i ba bo {<la (¹berti brife¹) KU> CU} {co'u troci} VAU] [i
ba bo {<la solri KU> CU} {co'a <glare dirce>} VAU] [i {ba zi} bo {<le litru KU> CU} {co'u dasni} {le kosta
KU} VAU] [i {se ki'u} bo {la <berti brife> KU} cu] [bilga {lo <ka (¹tugni [fi {lo <nu (²[{la solri KU} cu]
[traji {lo <ka (³vlipa VAU³) KEI> KU} VAU]²) KEI> KU}] VAU¹) KEI> KU} VAU])
```

假如果我們切到新版 4.0 的實驗性語法上

- 會看到這個結果

mw.lojban.org/extensions/ilmentufa/camxes-exp.html

Type any Lojban text in the following textarea. The result will be parsed as you type:

```
ni'o la berti brife jo'u la solri pu troci lo ka cuxna lo poi me vo'a vau traji be lo ka vlipa i
ca'o bo lo pa litru noi dasni lo glare kosta cu klama i le re mei pu simxu lo ka tugni fi lo nu lo
traji be lo ka clira fa lo nu ce'u snada lo ka gasnu lo nu le litru co'u dasni le kosta cu traji
lo ka vlipa i ba bo la berti brife co'a traji cupra lo brife i ku'i lo nu by by zenba lo ka cupra
lo xo kau brife cu rinka lo nu le litru zu'e ri zenba lo ka se tagji le kosta i ba bo la berti
brife co'u troci i ba bo la solri co'a glare dirce i ba zi bo le litru co'u dasni le kosta i se
ki'u bo la berti brife cu bilga lo ka tugni fi lo nu la solri cu traji lo ka vlipa
```

Output mode: Prettified [Grammar file] [Switch to standard grammar]

```
(ni'o [{<(¹la [berti brife] KU¹) (¹jo'u [la solri KU]¹)> <CU (¹pu troci¹) (¹lo [ka {CU <cuxna
(²lo [{poi <CU (³me vo'a MEhU³) vau> KUhO} {traji <be (³lo [ka {CU <vlipa VAU>} KEI] KU³)
BEhO>}] KU²) VAU>} KEI] KU¹) VAU>} {i ca'o bo <lo (¹[pa BOI] litru [noi {CU <dasni (²lo [glare
kosta] KU²) VAU>} KUhO]¹) KU> <cu (¹klama VAU¹)>}] [i {<le (¹[re BOI] mei¹) KU> <CU (¹pu
simxu¹) (¹lo [ka {CU <tugni (²fi [lo {nu <(³lo [traji {be <lo (⁴ka [CU {clira <fa (⁵lo [nu
{ce'u <CU (⁶snada [lo {ka <CU (⁷gasnu [lo {nu <(⁸le litru KU⁸) (⁸CU [co'u dasni] [le kosta KU]
VAU⁸)> KEI} KU] VAU⁷)> KEI} KU] VAU⁶)>} KEI] KU⁵)> VAU}] KEI⁴) KU> BEhO}] KU³) (³cu [traji {lo
<ka (⁴CU [vlipa VAU]⁴) KEI> KU} VAU]³)> KEI} KU]²) VAU>} KEI] KU¹) VAU>} {i ba bo <la (¹berti
brife¹) KU> <CU (¹co'a [traji cupra]¹) (¹lo brife KU¹) VAU>}] [i ku'i] [{lo <nu (¹[{by by} BOI]
[CU {zenba <lo (²ka [CU {cupra <lo (³[{xo kau} BOI] brife³) KU> VAU}] KEI²) KU> VAU}]¹) KEI>
KU} {cu <rinka (¹lo [nu {<(²le litru KU²) (²zu'e ri²)> <CU (²zenba [lo {ka <CU (³se tagji³)
(³le kosta KU³) VAU> KEI} KU] VAU²)>} KEI] KU¹) VAU>}] [i ba bo {la <berti brife> KU} {CU <co'u
troci> VAU}] [i ba bo {la solri KU} {CU <co'a (¹glare dirce¹)> VAU}] [i {ba zi} bo {le litru
KU} {CU <co'u dasni> <le kosta KU> VAU}] [i {se ki'u} bo {la <berti brife> KU} {cu <bilga (¹lo
[ka {CU <tugni (²fi [lo {nu <(³la solri KU³) (³cu [traji {lo <ka (⁴CU [vlipa VAU]⁴) KEI> KU}
VAU]³)> KEI} KU]²) VAU>} KEI] KU¹) VAU>}])
```

但不管是甚麼模式

- 速度都很慢
- 北風與太陽短短一兩百字
- 得花上五秒鐘才能剖析完

對於常用編譯器的程式人而言

- 這很難接受！

因為

- 幾千行甚至萬行的程式碼
- 通常編譯也不會花這麼久

那麼為何？

- ilmentufa 要這麼久呢？

讓我們繼續追根究柢

- 肉搜出 ilmentufa 的原始碼

發現果然在 github 裏面

GitHub, Inc. [US] https://github.com/Ilmen-vodhr/ilmentufa

This repository
Search

Pull requests Issues Gist

Ilmen-vodhr / ilmentufa

Watch 6 Star 7 Fork 11

Syntactical and semantical parser of Lojban text

438 commits 1 branch 0 releases 9 contributors

Branch: master ilmentufa / +

Ilmen-vodhr Merge pull request #125 from mezohe/master ··· Latest commit 1584bf2 14 days ago

glosser	Trying to fix glosser/glosser.htm	2 years ago
ircbot	"gimyinda" and "ni'o ni'o pa mo'o ni'o" fixes	4 months ago
node_modules	Delete node-debug	7 months ago
LICENSE	Updated LICENSE with author names	a year ago
README.md	Update README.md	7 months ago
camxes-builder.js	Parser builders: Fix - the code snippet appended to the pa...	a year ago
camxes-exp-builder.js	Parser builders: Fix - the code snippet appended to the pa...	a year ago
camxes-exp-changelo...	update camxes-exp-changelog	5 months ago
camxes-exp.html	fixing mistypes in camxes html files. Sockets are coming to...	8 months ago
camxes-exp.js	dragau lo me ma'oi fu'a moi	14 days ago
camxes-exp.js.peg	dragau lo me ma'oi fu'a moi	14 days ago
camxes-mh.js	various	8 months ago
camxes-mh.js.peg	various	8 months ago
camxes-pamoi.peg	original camxes from	8 months ago
camxes-std-changelo...	change "text" from _node to _node_nonempty to fix parse ...	11 months ago

Code
Issues 29
Pull requests 0
Wiki
Pulse
Graphs

HTTPS clone URL
https://githu

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop
Download ZIP

仔細看

GitHub, Inc. [US] https://github.com/Ilmen-vodhr/ilmentufa

ilmentufa

'la ilmentufa' is a syntactical and not yet semantical parser for the Lojban language.

Read more about it at http://lojban.org/papri/La_Bangu:_ilmentufa

=== Requirements ===

For generating a PEGJS grammar engine from its PEG grammar file, as well as for running the IRC bot interfaces, you need to have Node.js installed on your machine.

For generating PEGJS engine, you may need to get the Node.js module 'pegjs'. For running the IRC bots, you may need to get the Node.js module 'irc'.

However, as the necessary node_modules are already included in this project, I think you'll probably not have to download any of the aforementioned modules. ;)

=== Building a PEGJS engine ===

After having entered the ilmentufa directory, run the following command:

$ node [builder-filename]

For example, "node camxes-builder" for building the standard grammar engine or "node camxes-exp-builder" for experimental grammar.

Now, the grammar engine should have been created/updated, and be ready for use. :)

=== Running the IRC bots ===

Nothing easier; after having entered the ilmentufa directory, run the command "node ircbot/camxes-bot" or "node ircbot/cipra-bot" (the latter is for the experimental grammar). The list of the channels joined by the bot can be found and edited within the bot script.

您會發現

- ilmentufa 專案中有很多以 peg 為附檔名的檔案
- 並且採用了 peg.js 這個專案來將這些 peg 語法轉換為程式

這就是 PEG. js 的官網

pegjs.org

PEG.js

Parser Generator for JavaScript

Home | Online Version | Documentation | Development

PEG.js is a simple parser generator for JavaScript that produces fast parsers with excellent error reporting. You can use it to process complex data or computer languages and build transformers, interpreters, compilers and other tools easily.

Features

- Simple and expressive grammar syntax
- Integrates both lexical and syntactical analysis
- Parsers have excellent error reporting out of the box
- Based on parsing expression grammar formalism — more powerful than traditional LL(k) and LR(k) parsers

Try PEG.js online

— or —

```
npm install pegjs
```

— or —

```
bower install pegjs
```

— or —

Download browser version

問題是

- PEG. js 真的比較慢嗎？

讓我們 google 一下

發現以下網頁

blog.gmane.org/gmane.comp.parsers.peg.general/month=20110801

peg@lists.csail.mit.edu

Discussion of parsing expression grammars and packrat parsing

Dustin Voss | 1 Aug 01:04 2011 headers

GROUP

gmane.comp.parsers.peg.general.

ADVERTISEMENT

PROJECT WEB PAGE

Discussion of parsing expression grammars and packrat parsing

SEARCH ARCHIVE

Search

Re: Res: Known LPEG ports/implementations

My parser (written in Dylan) supports selective memorization on a rule-by-
"default." That said, it parses fastest with memorization enabled on a few
most. It parses much slower if I disable memorization for all rules, and i
enable memorization for all rules.

On Jul 31, 2011, at 3:03 PM, Volker Birk wrote:

> On Sun, Jul 31, 2011 at 02:42:53PM -0700, Sérgio Medeiros wrote:
>> Actually, it seems that PEG libraries that implement
>> the Packrat algorithm (that guarantees linear parsing
>> time) turn it off by default, because the memorization

其中實測了 PEG. js 的速度

Francisco Tolmasky | 1 Aug 01:17 2011 **headers**

Gmail **Re: Res: Known LPEG ports/implementations**

I have actually been running speed tests for a while against the following:

- PEG.js: seems to be the most popular current JS PEG
- Jison: most popular JS parser (used by CoffeeScript), but not PEG
- Narcisses: a hand rolled JS parser in JS

I have been so far comparing the parse times of jquery.js (since this is a pretty language features). My results on my Macbook are:

language.js: 7173 ms

peg.js: 94644 ms

Narcissus: 865 ms

jison: 6372 ms

The speed test is available here: **https://github.com/tolmasky/ParserSpeed** and can

您可以看到 PEG.js 真的很慢

- 比 jison 慢 10 倍以上
- 比 Narcissus 慢一百倍以上

為何 Narcissus 可以這麼快？

- 因為 Narcissus 是手工打造的 javascript compiler
- 而不是用 compiler compiler 將 BNF/PEG 轉為 compiler 的

但是 jison 和 PEG.js 很像

- 只是 jison 採用 BNF 語法
- 而 PEG.js 採用 PEG 語法

雖然 PEG 語法比 BNF 稍強

- 但是這代價未免也太大了。

我想，邏輯語的新版 parser 開發者

- 應該是為了簡化剖析過程的難度，決定採用了 PEG

而且這些 parser

- 幾乎都衍生自 camxes 這個核心版本
- 因此都採用 PEG 也就不令人意外了。

但是衍生的效應就是

- Parse 速度非常非常慢！

那麼，採用 PEG 有甚麼好處呢？

我繼續追根究柢

- 結果發現這篇文章
  - Issues With The Lojban Formal Grammar
  - http://users.digitalkingdom.org/~rlpowell/hobbies/lojban/grammar/

Issues With The Lojban Formal Grammar
這篇文章最後一段提到下列問題

Old Approach

For a while I was trying to adjust the BNF grammar (after conversion to ABNF) to do The Right Thing with respect to elidable terminators, because that's the hard part. I have since come to the conclusion that elidable terminators can merely be made optional if longest-match disambiguation is used, but that puts us in the realm of specifying the parser again, which is what I was trying to avoid.

I still think that one could probably get BNF to do The Right Thing with respect to elidable terminators, but it would be Very, *VERY* hard. I would be surprised if it could be done without expanding the grammar by a factor of 20 or so. No, that's not an exaggeration or a joke.

Update 20 Sep 2005: I am no longer so sure that it's possible, but I don't have any good reason; just a change in my gut feeling. I also think a 20 times size increase is probably conservative.

To give you a sense of what I mean, consider fixing 'kei'. This requires having the grammar descending from a NU clause to eat all brivla it sees until the next kei. Because BNF is inherently ambiguous, forcing this requires that every place where two brivla could occur next to each other be re-written`to only form two separate selbri when there is a kei between them, but only inside a NU clause. If this is possible in BNF/CFGs, and I'm not totally certain it is, it requires nearly doubling the size of the grammar because you have to have everything under 'subsentence' copied into a "[foo]_during_NU" form, or whatever.

When you're done with that, try another big elidable terminator, like 'ku'. This will require the same thing, but the ku additions to the grammar and the nu additions to the grammar must work nested, in either order. That's two more complete sets, not including the 'ku' or 'kei' sets. You now have a grammar on the order of four times the original size, and you've fixed only two elidable terminators.

Good luck; let me know when you're done.

這個問題

- 是關於 elidable terminators 的
- 它會造成 longest-match disambiguation
- 如果要解決會造成語法大膨脹

  it would be Very, VERY hard. I would be surprised if it could be done without expanding the grammar by a factor of 20 or so. No, that's not an exaggeration or a joke.

那麼

- 到底 elidable terminators 是甚麼東東呢？

繼續追下去、我們查到這個

https://mw.lojban.org/papri/elidable_terminator

Lojban Page Discussion Search Guest

Learning Ask question

Questions Web chat

News Contacts

Upload file
Special pages
Print as PDF

What links here
cmavo
currently proposed experimental cmavo
description 110
grammar
ku
. . .

elidable terminator

From the Book, chapter 19, section 17: The elidable terminators are

cmavo	selmaho	meaning
be'o	BE	sumti attached to a tanru unit
boi	PA/BY	number or lerfu string
do'u	COI/DOI	vocative phrases
fe'u	FIhO	ad-hoc modal tags
ge'u	GOI	relative phrases
kei	NU	abstraction bridi
ke'e	KE	groups of various kinds
ku	LE/LA	description sumti
ku'e	PEhO	forethought mekso
ku'o	NOI	relative clauses
li'u	LU	quotations
lo'o	LI	number sumti
lu'u	LAhE/NAhE+BO	sumti qualifiers

問題是

- 這些邏輯語的字我全都不認識

只好查字典

查第一個 be’o

1 definition found

From Lojban to English :

```
       Word: be'o [jbovlaste]
       Type: cmavo
  Gloss Word: end linked sumti
    selma'o: BEhO
  Definition: elidable terminator: end linked sumti in specified description.
```

再查第二個 boi

1 definition found

From Lojban to English :

```
       Word: boi [jbovlaste]
       Type: cmavo
  Gloss Word: end number or lerfu
     selma'o: BOI
  Definition: elidable terminator: terminate numeral or letteral string.
```

再查第三個 do’u

1 definition found

From Lojban to English :

```
        Word: do'u [jbovlaste]
        Type: cmavo
  Gloss Word: end vocative
     selma'o: DOhU
  Definition: elidable terminator: end vocative (often elidable).
```

再查第四個 fe’u

1 definition found

From Lojban to English :

```
        Word: fe'u [jbovlaste]
        Type: cmavo
  Gloss Word: end modal selbri
     selma'o: FEhU
  Definition: elidable terminator: end nonce conversion of selbri to modal;
       usually elidable.
```

再查第五個 ge’u

1 definition found

From Lojban to English :

```
       Word: ge'u [jbovlaste]
       Type: cmavo
  Gloss Word: end relative phrase
    selma'o: GEhU
  Definition: elidable terminator: end GOI relative phrases; usually elidable
      in non-complex phrases.
```

再查第六個 ge’u

From Lojban to English :

```
       Word: kei [jbovlaste]
       Type: cmavo
  Gloss Word: end abstraction
      rafsi: kez
    selma'o: KEI
  Definition: elidable terminator: end abstraction bridi (often elidable).
```

再查第七個 ke’e

From Lojban to English :

```
       Word: ke'e [jbovlaste]
       Type: cmavo
  Gloss Word: end grouping
      rafsi: kep ke'e
    selma'o: KEhE
  Definition: elidable terminator: end of tanru left grouping override
      (usually elidable).
```

應該查得夠多了！

- 我想得回去學學邏輯語

但是、我猜測

- 這些詞代表某個子句或項目的結尾
- 而且子句或項目可能很長
- 因此造成難以處理的情況

這也正是

- 為何要從 BNF 切換到 PEG 的原因
- 但是切換到 PEG 的代價實在太大
- 速度慢上十倍！

我想

- 這會是個值得研究的問題

但是要研究之前

- 先讓我再多學一些邏輯語
- 否則只能瞎子摸象了！

再見了！

Good bye

Co’o ro do