CLL Version 1.1 におけるlujvo作成アルゴリズム

fi'e la .sozysozbot.
(http://twitter.com/sosobotpi)

-1章 注

以下の文章のうち、0章から3章までは2017年5月1日現在 https://goo.gl/fGZvAE にて一般公開している。

0章 概要

ロジバンにおいて、lujvo（複合語）を作る規則は意外とめんどくさいし、聖典CLLの説明は分かりにくい。さらに、英語資料は一応あるもののまとまった日本語資料が見当たらない。
ということで、jvozbaを再発明 (URL: https://goo.gl/hchjGj) した筆者が実装時のわーきゃーを踏まえながら解説記事を書いてみようと思う。
なお、筆者が理解していないので、意味論には触れない。

1章 rafsi

「CLL 4.6に丸投げします」と、言えたらどんなに嬉しいことか。
困ったことに、CLLはrafsiに関して自己矛盾を含んでいるのである。
その話は第6章で触れる[1]として、ここではサラッとrafsiについて解説するのみにしておく。
rafsiとは、複合語(lujvo)を生み出すための形態素で、その形には8種類ある。 Cが子音、CCが「語頭に立ちうる二重子音」を表す。C/Cは「語中にあり得る二重子音」。

3文字:

形	例
Cvv	-bau-
CCV	-jbo-
Cv'v [2]	-ma'o-
CVC	-sel-

CvvとCv'vを合わせてCVVと呼ぶ。なお、vvとしてはai, ei, au, oiの4通りのみが用いられる。

4文字:

形	例
CCVC	-plis-
CVC/C	-karc-

5文字:

形	例
CCVCV	-plise-
CVC/CV	-karce-

なお、5文字rafsiはlujvo末でのみ使うことができる。

結論から言えば、lujvoはこのrafsiをくっつけていけば作れるのだが、そのくっつけるための規則がめんどくさいのである。以下その規則を解説していく。

2章 rafsiをくっつけるPart1

まず大前提として、lujvoはbrivlaなので、lujvo末には母音で終わるrafsiが来なくてはならない。
それが済んだらrafsiをくっつけていけばいいが、諸事情により「ハイフン字」（`y`, `n`, `r`）がrafsi間に挿入されることがある。「諸事情」とは、

- 4文字rafsi
- 禁則回避
- 剥がれ落ち防止

の3パターンである。まずは前2つを解説する。

2-1 4文字rafsi

4文字rafsiの後ろには必ず `y` が必要である。

例：
-plis- + -ladru- → plis**y**ladru

これは、lujvoを一意に分解するのに必要な規則である。lujvoの2文字目には絶対にyが来ないことが知られているので、そのこととこの規則から、先頭を読むだけで {plisyladru} が *[-pli- + syladru] ではなく [-plisy- + ladru] であることを知ることができるのだ。流石LLG、賢い。
ちなみにこの例でyを抜かすと{plisladru}となるが、これは[-pli- + -sla- + -dru-]である。CVC/C型については{karctadji}を考えてみよう。

2-2 禁則回避

そのままくっつけると禁則な子音列が発生する場合、2つのrafsiの間には `y` が要求される。

例：
-fam- + -ma'o- → fam**y**ma'o （同一子音の連続を回避）
-fas- + -bau- → fas**y**bau （無声阻害音＋有声阻害音を回避）
-cen- + -dje- → cen**y**dje （ `ndj` を回避）

この規則はロジバンの音韻構造の制約を直に受けているだけのものであり、それ以上に深い意味は特にない。

3章 rafsiをくっつけるPart2

さて、「lujvoを作る」というのは、そもそもどういうことだろうか。[3]
私の考えでは、「ハンバーガー🍔をつくること」に例えられると思う。[4]
ハンバーガーの内側の具を入れ替えても、それはまだハンバーガーである。[5]
しかし、「一番上に丸くてゴマのついたパン、一番下に平らなパン」という掟を破って層を配置すれば、ハンバーガーではない。

lujvoにおいても同様のことが言える。
前述のとおり、lujvo末のrafsiは母音終わりであることが要求されるが、lujvoの語頭に来るrafsiにも面倒な規則が存在するのだ。

☆**以下の操作は、具を載せ終わって、最後に上の丸いパンを載せるときにのみ、一度だけ、行うこと**☆ （もちろん、具や丸いパンを載せていく過程で2-1と2-2の操作は行うこと）

- case 1: 丸いパンが4文字rafsiまたはCCV rafsiの場合
    - 特別な操作は不要[6]
- case 2: [CVV + CCV]で単語が完成する場合（例： {ma'ovla}, {fu'ivla}）
    - 特別な操作は不要
- case 3: 丸いパンがCVVで、case 2以外のとき
    - CVVの後に `r` を挿入（例： {fa'orma'o}, {fu'irvlarafsi}）

    - ただし、直後に既に `r` がある場合は挿入されるのは `n` （例： {ka'e**n**ri'u}）

- case 4: 丸いパンがCVCのとき
    - 2-2が適用されるなら普段通り。さもなくば3-1へ進む

3-1 tosmabruテスト

先頭にCVCを付けたものがαとβのどちらか片方に該当する場合にのみ、α, βの形ではなく、先頭のCVCの後に `y` をねじ込んだ形が採用される。ちなみに、これをtosmabruテストと呼ぶ。

α [（1個以上のCVC） + CVCCV[7]] かつ、くっつけた時全ての子音結合がCCである
β [（2個以上のCVC） + `y` + （1個以上の任意のrafsi）] かつ、y以前の全ての子音結合がCCである

従って、

- [-cas- -tadji-] → {cas**y**tadji} （全てCC）
- [-kan- -tadji-] → {kantadji} （ `nt` はC/C）
- [-cas- -kan- -tadji-] → {caskantadji} （ `sk` はCCだけど `nt` はC/C）
- [-pas- -paz- -badna-] → {paspazbadna} （ `dn` がC/C）
- [-pas- -pas- -y- -badna-] → {pas**y**pasybadna} （ `dn` はC/Cだが判定にかかるのは `paspasy` の部分まで）

4章 一意性の証明

【長文注意】

以下、上記の方法でlujvoを作れば「(0個以上のcmavo) + lujvo」が「(0個以上のcmavo) + 2個以上のrafsiの列」に一意に分解できることの構成的な証明を行う。なお、以下、C\Cで「語頭に立てない二重子音」を表すとする。

4-1 lujvoの先頭部分に関する考察

まず、3章の全パターンを列挙して調べることで、lujvoの先頭部分としてありうる形が限られることが分かる。（2番目のrafsiが何であれ、必ずCで始まり、その次がyでないことは保障されていることに注意）
便宜上、それぞれの形に番号を振り、先頭のrafsiを強調表示する。

パターン	形
case1	① **CCVC**yC- ② **CVC/C**yC- ③ **CCV**C(y以外の文字)-
case2	④ **CVV**CCV
case3	⑤ **CVV**rC(y以外の文字)- ⑥ **CVV**nr(y以外の文字)-
case4 (2-2が適用されるとき)	⑦ **CVC**yC(y以外の文字)-
case4 (3-1が適用されるとき)	⑧ **CVC**yCVCCVC..CVCCV ⑨ **CVC**yCVC..CVCy-
case4 (それ以外のとき)	⑩ **CVC**/C(y以外の文字)-

これにより、まず、2-1で書いた主張「lujvoの2文字目には絶対にyが来ない」が示され、また「lujvoの3文字目にも絶対にyが来ない」ことも分かる。また、異なる第一rafsiに由来するlujvoの形が重複することがなく、lujvoの形から一意に第一rafsiが判定できることがわかる。
さらに、⑤⑥⑧⑨では第一rafsiの後ろにハイフンが挿入されているが、このハイフンが仮に存在しなかったとしても他種のrafsi列と「は」混同することがない。[8]ゆえに、第一rafsiを取り除いた後の文字列について同様の議論を行うことで[9]、lujvoを構成するrafsi列は一意に決定できる。

4-2 cmavo: 0個 vs. 1個以上

次に、「lujvo」が「(1個以上のcmavo) + lujvo」と一致することはないことを示す。そのために、一致する例が存在するとして矛盾を導く。
一致する例が存在するならば、4-1より、「(1個以上のcmavo) + lujvo」と解釈した時のcmavoの形としてありうるのは1つのCV（②⑦⑧⑨⑩）か1つのCVV（④⑤⑥）である。しかし、②④⑤⑥⑦⑧⑨⑩のいずれの場合でも、「cmavo + lujvo」との解釈は成り立たないことを示す。
②④⑤⑥⑦⑧⑨の場合は単純である。

パターン	理由
②	CVを取り除くと3文字目がyで、これが4-1の結論と矛盾
④	CVVCCVのCVVの内部にはアクセントがあるので、cmavoにはなれない
⑤⑥	CVVを取り除くとlujvoはrCまたはnrで始まることとなるが、これはどちらもCCではない
⑦⑧⑨	CVを取り除くと2文字目がyで、これが4-1の結論と矛盾

さて、問題は⑩である。⑩の先頭からCVを取り除いた形がlujvoとなる可能性を否定しなければならない。⑩は、3-1のtosmabruテストに該当しない場合なので、

1. [（1個以上のCVC） + CVCCV] であるが、くっつけた時一部の子音結合がC\Cである
2. [（2個以上のCVC） + `y` + （1個以上の任意のrafsi）] であるが、y以前の一部の子音結合がC\Cである
3. `y` を含まず、[（1個以上のCVC） + CVCCV]でもない
4. `y` を含んでいるが、[（2個以上のCVC） + `y` + （1個以上の任意のrafsi）] ではない

という4通りの場合があり得る。

まず1.の場合、先頭からCVを取り除くとCCVCCV..C\CV..CVCCVといった形になる。左から1つずつrafsiを削っていくとき、4-1.よりその削り方は一意に決定できる（CCVという形で必ず削られる）が、すると途中でC\CV-という形に遭遇してしまう。当然これはrafsiとして許されないので、こう分解することは許容されない。2.も同様であり、CCVCCV..C\CV..CCVCy-という形を左から削っていくとC\CV-に遭遇してしまう。

3.の場合、元々の分解方法では途中で「CVCCVでも4文字rafsiでもCVCでもないrafsi」に遭遇することとなる。そのrafsiの左には1つ以上のCVC rafsiがあるので、lujvoの先頭からCVを削った場合

- ⑪ **CVC**/CVV- [CVVが後続する時]
- ⑫ **CVC**/CCV- [CCV, CCVC, CCVCVが後続する時]
- ⑬ **CVC**/CVC\CV [CVC\CVが後続する時]

から先頭のCVが（直接的に、またはCCVCCV..CCVの最後2文字として吸収されて）失われた場合にrafsi列として成立しないことを示せばよい。 まず、上での3文字目と4文字目の2つのCがC\Cである場合、先頭のCVを失うとrafsi列と見ることはできない。次に、CCVV-, CCCV-で始まるrafsi列はないので⑪⑫は除外でき、CCVC\CVという形はCCVを削るしかなくC\CVが露出し不適。以上より3.も否定できる。

最後に4.の場合。2.の場合を除いていることから、[CVC + (CVCが0個以上) + (CVC以外が1個以上) + (CVCが0個以上) + `y` + （1個以上の任意のrafsi）] として表せる。つまり、「CVC+CVV」「CVC+CCV」「CVC+CCVC」「CVC+CVC/C」のいずれかの組み合わせが列中に存在するが、最初の3つは先頭のCVが直接的または間接的に削られた際CCVV-またはCCCV-というrafsi列としてあり得ないものとなり、「CVC+CVC/C」は2-1より「CVC+CVC/C+ `y` 」であり、先頭のCVが消えるとCCV + C/Cy-というあり得ない形となる。

以上より、「lujvo」が「(1個以上のcmavo) + lujvo」と一致することはないことが示された。

4-3 最後に

最後に、2種類の「(0個以上のcmavo) + lujvo」が一致することはないことを示す。「(m個以上のcmavo) + lujvo1」からなるAと、「(n個以上のcmavo) + lujvo2」Bから導き出される文字列が一致したと仮定して矛盾を導く。
まず、m=nの場合を考える。この場合、明らかにcmavo列は一致していて、ゆえにlujvo1とlujvo2も一致する。

次にm>nのとき、A,Bの先頭から共にn個のcmavoを削ると「((m-n)個のcmavo) + lujvo1」と「lujvo2」が一致することになるが、これは4-2に反する。AとBを入れ替えればm<nの場合も示せる。

以上より、一意性が示された。

5章 おまけ1: lujvoの得点の話

lujvoには「得点」が割り振られている。同じrafsi列から作られたlujvoの中でも、得点が最も低いものが「好ましい」（原文: preferable）とされている。
得点の計算式自体はCLL4.12に載っているが、厳密に1の位まで得点を計算する必要があることはほとんどなく、以下の目安を覚えておけば十分であろう。

1. 短いlujvoが勝ち（ただし、アポストロフィは0.5字相当）
2. 引き分けならハイフン字（ `y` , `n` , `r` ）が少ない方が勝ち
3. それでも引き分けならCVC < CCV < Cvv

6章 おまけ2: CLLの自己矛盾

公式の説明CLL4.6は自己矛盾を含む。というのも、

> Each gismu always has at least two rafsi forms; one is the gismu itself (used only at the end of a lujvo), and one is the gismu without its final vowel (used only at the beginning or middle of a lujvo).

という記述ををbroda, brode, brodi, brodo, broduに当てはめれば、-brod-はこの5種のgismuを表すrafsiになるはずだが、そうするとその直前にある

> each such rafsi represents only one gismu.

に矛盾するからだ。
ちなみに、CLL4.15を見ると、

> The following gismu were not made by the gismu creation algorithm.

という文章の後にbrodVが紹介されているので、brodVが公式見解上gismuであることは確実である。
詳しくは、https://goo.gl/zM7M9u を参照のこと。

脚注

1: 話題が逸れるので分離

2: アポストロフィは文字数に入れないので-ma'o-も3文字rafsi
3: なんかポエム始まったぞ
4: 駄目だこいつ…早くなんとかしないと…
5: 「肉が入ってないものは厳密にはハンバーガーとは呼べない」などの指摘は受け付けません。ご了承ください。
6: 念のため書いておくと、丸いパンが4文字rafsiだったら当然規則2-1より `y` が入る
7: CVC/CVとは言っていない。ゆえに{mabru}にはマッチするが{mapku}にはマッチしない
8: 後述する通り、万が一ハイフンがない場合cmavo + lujvoと 「は」混同することに注意
9: ただし、lujvo末には5文字rafsiが登場しうるという点に留意する必要はある