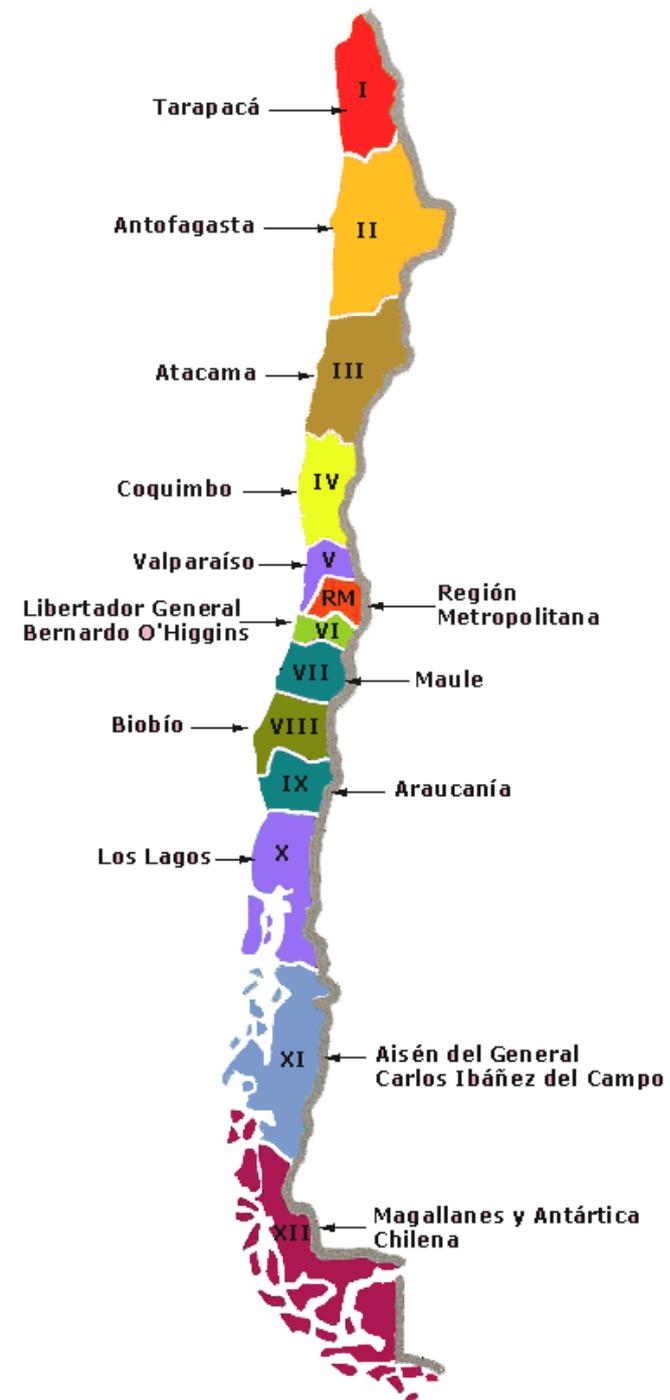


Dungupeyem_alpha_v.01

Analizador morfológico para el mapudungun
basado en transductores de estados finitos

Dungu= palabra, habla | pe= indica proximidad temporal y/o física con los hechos | ye= indica rasgo constante |
m=marca de verbo nominal instrumental, indica instrumento o locación. La traducción sería algo así como
“instrumento que siempre se usa para hacer algo con el lenguaje”, para nosotros: “herramienta del lenguaje”.

- Un poco de historia:



- Breve reseña tipológica del mapudungun

Aglutinación

Composición

Reduplicación

- Fonética y fonología
25 fonemas nativos
6 vocales
19 consonantes
3 glides → semivocales

- Sufijos y slots

100 sufijos

36 slots

raíz - 36 - ... - 1

exclusión: gramatical – semántica

1 – 15: flexivos (fijos)

16 – 27: derivados -[semánticos] (movilidad)

28 – 36: aspecto y valencia (fijos)

- Forma verbal
raíz + (sufijos derivativos) + 1 + flexivos

- Predicado verbal:

forma finita, persona s3, modo s4

amu- y- m- i

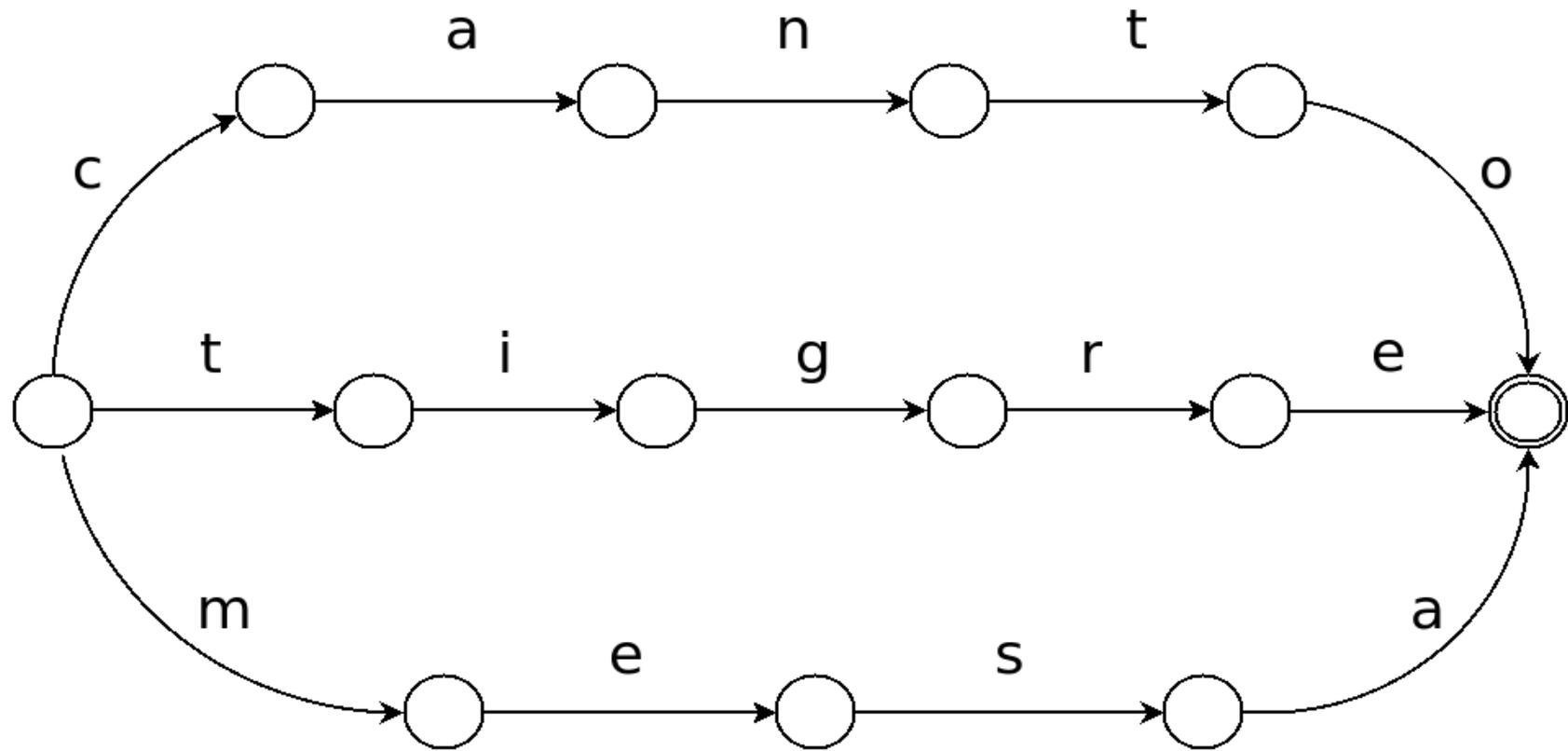
ir- IND⁴- 2³- SG²

tu fuiste

Complejidad: paradigma transitivo

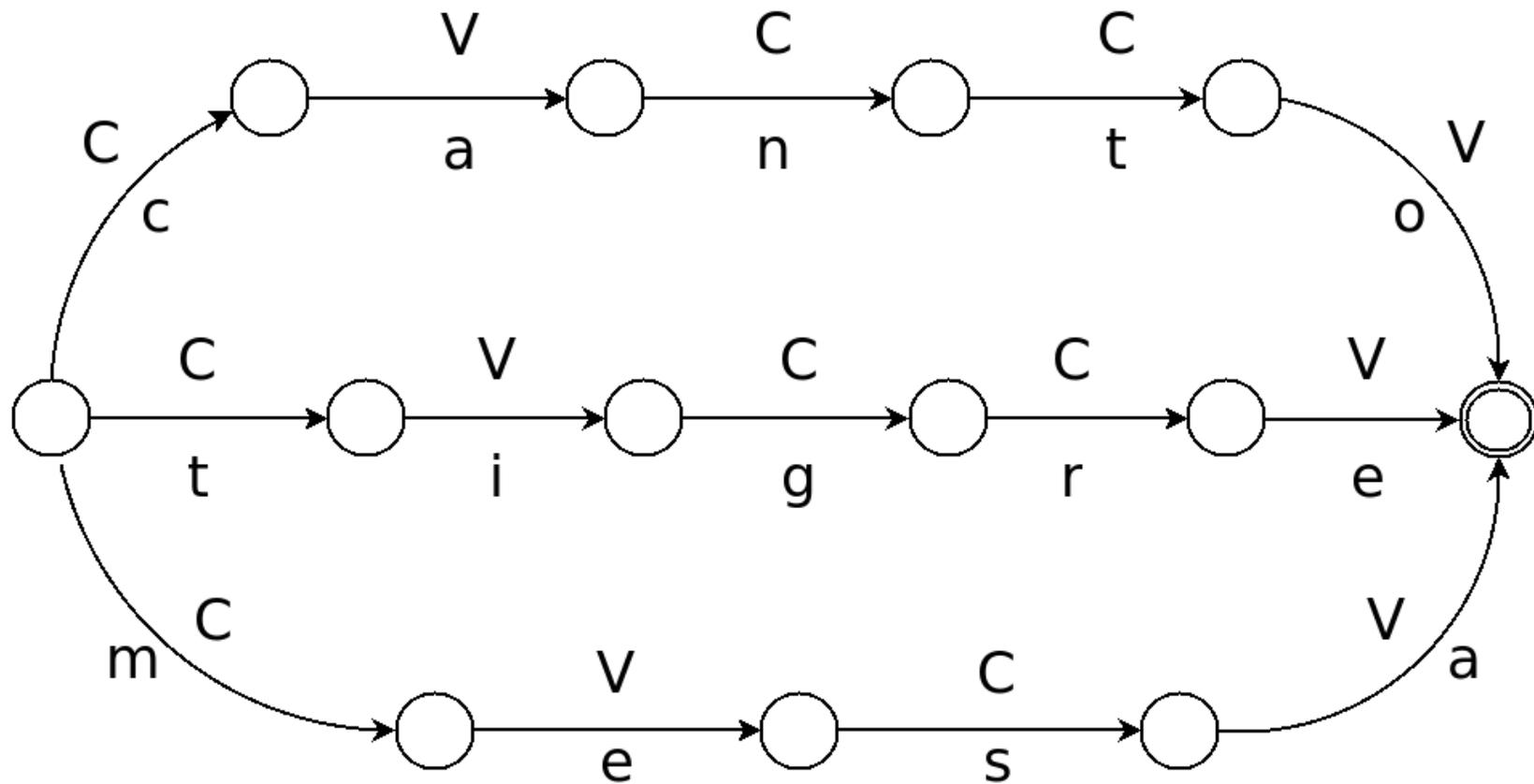
- Morfología de estados finitos
- Transductores de estados finitos

Red de estados finitos



Red que acepta un lenguaje comprendido por tres palabras

- Red de estados finitos de 2 niveles
Transducer



Red de dos niveles o Transductor

- XFST

interfaz → algoritmos

extended regular expressions

reglas de reemplazo

- XFST → Dingupeyem

A Grammar of Mapuche (Smeets)

→ expresiones regulares → XFST

= análisis y generación de predicados verbales del mpudungun

- Morfotáctica
- Método de trabajo
 - 1º descripción AGM
 - duplicación de reglas
 - 2º planilla de cálculo
 - generar restricciones adecuadas

- reglas de variaciones morfofonológicas:
- epéntesis o elisión

[" +VRB""@ns"] : ["@N"{tu}]

```
define RuTrEPENTHn ["@N" -> "" ||
Consonants|SemiVowels _].o. ["@N" -> n ||
Vowels _];
```

- Reemplazo

```
define RuTrCHkiy [{"@KI" -> {ki} || Consonants|  
Semivowels _] .o. [{"@KI" -> y || Vowels _}];
```

Reglas simples de reemplazo: reglas de reescritura

- Restricciones contextuales para sufijos
- Prohibición

```
define RuPrAnotB [ ~$["AffA" ?* "AffB"]];
```

```
define RuPrIDO [ ~$[" +IDO" ?* [" +IMP"|["  
+3""@e"]|[" +3""@ns"]|" +IMP2SG"|" +IMP3SJ"|  
[" +SG""@i"]]]];
```

1. " +IDO" ?* " +IMP"
2. " +IDO" ?* [" +3""@e"]
3. " +IDO" ?* [" +3""@ns"]
4. " +IDO" ?* " +IMP2SG"
5. " +IDO" ?* " +IMP3SJ"
6. " +IDO" ?* [" +SG""@i"]

- Obligatoriedad

```
define RuObCNDIMP [" +CNDNIMP" => ?* ["  
+NEG""@i"] ?* _ ?* [" +IMP1SG"|" +IMP2SG"] ?*];
```

operador de restricción =>

marcador contextual _

1.[?* [" +NEG""@i"] ?* " +CNDNIMP" ?* ["
+IMP1SG"] ?*]

2.[?* [" +NEG""@i"] ?* " +CNDNIMP" ?* ["
+IMP2SG"] ?*]

- Obligatoriedad condicionada
transitivización de intransitivos

Slot ad-hoc [" +VI""@0"] : 0

 [" +VI""@e"] : 0

```
define VERBFORM [VROOT (INTR) (TRFAC)  
(PASS1A2A) (EIDO) (DS)];
```

- Reglas contextuales

```
define RULE01 [["@VI" => ?* _ ?* " +VI" ?*] .o. ~$  
["@VT" ?* " +VI"]];
```

```
define RULE02 ["@VT" => ?* _ ?* [["@ +PASS"|" +1A"|" +2A"]|[" +EDO"|" +IDO"]]]];
```

```
define RULE03 [ ~$ [["@ +VI""@0"] ?* [["@ +PASS"|" +1A"|" +2A"|" +EDO"|" +IDO"]]]];
```

```
define RULE04 [ ~$ [["@ +PASS"|" +1A"|" +2A"] ?* [["@ +EDO"|" +IDO"]]]];
```

```
define RULE05 [["@ +DS3A"|" +DS12A"] => ?* " +IDO" ?* _ ?*];
```

- Análisis y generación

xfst[1]: apply up *afka*

acabar +FAC → acabar +VI@e +FAC

acabar +FAC → acabar +VI@0 +FAC

xfst[1]: apply down acabar +VI@0 +FAC

afka

xfst[1]: apply down acabar +VI@e +TR +IDO
+DS3A@e

Aftuemew

- *Semi-obligatoriedad*

- Temas verbales (stem)
simples → raíz v + sufijos
no verbales: raíz nv + VRB³⁶
- Slot 36 – Verbalizadores {VRB}
-∅- sustantivos, adjetivos, numerales,
adverbios

che: persona; *che-∅*: convertirse en persona,
recobrar la sobriedad

-nge-, sustantivos, adjetivos, numerales,
chum- (cómo)

rume küme wentru- nge- y- Ø

mucho bueno hombre-VERB³⁶-IND⁴- 3³

él es muy buen hombre

-tu-, sustantivos

kitra: pipa; *kitra-tu-*: fumar en pipa

-(n)tu-, adjetivos

küme: bueno; *küme-ntu-*: gustar de, pensar que (algo) es bueno

-l-, sustantivos, adverbios, numerales, tunte (cuanto)

kiñe: uno; *kiñe-l-*: dar un (algo)

-ye-, sustantivos

Lalla- ye- w- y- u

suegra/yerno-VERB36-REF31-IND4-1NS3-DL2
somos *llalla* el uno del otro

- Reduplicación
- Slot 36, Tematizadores verbales {SFR}
 - tu- → *la-la-tu-*: desmayarse; *la-* morir
 - nge- → *kinging-kinging-nge*: quejarse como un perro; *kinging*: gimotear
 - ye- → *pi-pi-ye-*: repetir incesantemente(tr.); *pi*: decir
 - ∅- → *tril-tril-∅-*: tiritar de frío o fiebre

- Compile-replace

```
define PREREDUP [0 .x. "^[" %{}];
```

```
define POSREDUP [" +RED" .x. %} %^ 2 "^)];
```

```
define 2ONROOT PREREDUP ONROOT  
POSREDUP;
```

```
define 2VROOT PREREDUP VROOT POSREDUP;
```

```
define RuRRON [[["@ON" ?* " +RED"] => _ ["  
+SFR"" @0"] ?*] .o. ~$["@ON" ?* " +RED"] ?* "  
+VBR"]];
```

```
define RuRRV [[["@VI"|"@VT"] ?* " +RED"] => _  
[" +SFR"] ?*] .o. ~$[[["@VI"|"@VT"] ?* " +RED"] ?  
* " +VBR"]];
```

- Análisis y generación de raíces reduplicadas

xfst[1]: apply up *ngayngay*

OTC: rebuznar +RED +SFR@0

xfst[1]: apply up *allküallküye*

oír +RED +SFR@i

xfst[1]: apply down OTC: rebuznar +RED
+SFR@0

ngayngay

xfst[1]: apply down oír +RED +SFR@i
allküallküye

Versión Alfa

En las fases del desarrollo del software se utiliza el término **alfa** informalmente para referirse a una fase donde un producto todavía es inestable, aguarda a que se eliminen los errores o a la puesta en práctica completa de toda su funcionalidad, pero satisface la mayoría de los requisitos.