The JavaScript Spanish-English Grammar

Francesco Zamblera

August 10, 2010

Abstract

This article describes the grammar used by the Spanish-English NSM translator.

A short introduction (section 1) summarizes some basic computational issues, explaining the format of the grammar and its function in the computational process. Then sections 2 and 3 describe, respectively, the English and the Spanish morphological components of the grammar; while in section 4 the Spanish-English component is addressed.

1 Introduction

This article describes the javascript source code of the grammar used by the Spanish-English and English-Spanish NSM translator.

The translation process is based on pattern matching: a pattern in the source language is recognized and changed immediately into the corresponding pattern in the source language. Before pattern matching, however, the input text is analyzed morphologically, and an abstract representation of the morphemes is sometimes introduced; so I cannot say that there is absolutely no intermediate metalanguage involved.

Each component of the grammar consists of a list of javascript rules, and each rule is a pair or triple of strings.

The grammar thus written is compiled into a cascade of regular expressions, which are matched to the input text to produce the result. The rules are applied from first to last, in the order in which they are written, in a unique pass, without backtracking.

The compilation can be in both directions: the source grammar of the Spanish-English grammar can be compiled into both a Spanish-English and an English-Spanish tranaslator.

Almost all NSM constructions are recognized, although some bugs remain, which will be noticed as the discussion goes on. One of the main drawbacks of the approach is that a textual level is still lacking.

I have tried to implement all the specifications for Spanish *Metalengua semántica natural* described by Catherine Travis ([Tra02]). Spanish NSM is

based on Colombian Spanish, so the unmarked form for the second person pronoun is *usted*, and the unmarked future tense is the periphrastic construction *ir* a + infinitive.

To understand the following discussion, just a few hints about javascript data structures are needed:

- A *string* is a sequence of characters, delimited by double quotes. If the string needs to span several lines, a backslash is placed before the carriage return:

  ```
  "I am a string"
  "I am a string too \
  and I span two lines"
  ```

- A *list* is a sequence of items, comma-separated. Each item can be a string, a number, another list, and other. Here is a three-number list:

  ```
  [1, 2, 3]
  ```

  and this is a list of four strings:

  ```
  ["string one", "string two", "string three", "string four"]
  ```

  and here is a mixed list with five items (three nombers, a string and another list):

  ```
  [1, 2, 3, "ciao", ["a", "b"]]
  ```

- You will often see *comments* in source code. A comment begins with two slashes (//) and goes up to the end of the line.

  Each rules is a pair or triple of strings, where *pair* and *triple* are simply two- and three-element lists:

  ```
  ["string_1", "string_2"]            // a two-member rule
  ["string_1", "string_2, "string_3"] // a three-member rule
  ["string_1",
   "string_2,
    "string_3"]                 // another three-member rule
  ```

- A datum (like a list, a string, or a number) can be memorized in a *veriable*. The following javascript instruction declares a variable whose name is `pippo` and whose value is the string `hello`:

  ```
  var pippo = "hello"
  ```

while the following instruction, (actually the first of the English morphological component), declares the variable `eng_morph`, which contains the rule list:

3a ⟨*English-morphology* 3a⟩≡
```
var eng_morph =  [
⟨English morphological rules 4a⟩
];
```

To see how a whole component look like, here is a "grammar" which replaces every occurence of a lower-case vowel in the input string with the corresponding upper-case vowel. The grammar is memorized in the variable `toy_grammar`:

```
var toy_grammar = [
  ["a", "A"],
  ["e", "E"],
  ["i", "I"],
  ["o", "O"],
  ["u", "U"]
]
```

The actual grammar is obviously more complicated. It has three components with the same structure as the above toy grammar:

- the English morphological component;
- the Spanish morphological component;
- the Spanish-English grammar

The English morphological component is memorized in a different file, so that it can be loaded into other grammars; the Spanish morphological component and the Spanish-English grammar are both in the file `spaCO_eng.js`. I named the file after the Ethnologue codes for English and Spanish, writing `spaCO` instead of simply `spa` as a reminder that we are dealing here with Colombian Spanish. The structure of the file is:

3b ⟨*spaCO-eng.js* 3b⟩≡
```
⟨Spanish morphology 15c⟩
⟨Spanish English grammar 25a⟩
```

2 The English Morphology Component

When the program is translating from English to Spanish, this component performs a preliminary morphological analysis of the English words received in input. This analysis allows for later translation rules to be expressed more straightforwardly.

In the inverse direction (Spanish to English), the English morphological component works in reverse: it synthesizes, for example the word "did" from the string `PRET do`, where `PRET do` is the output of a previuous translation process which replaced, for example, "hizo" into `PRET hace` and then into `PRET do`.

2.1 Simple rules

The simplest form of a rule is:

$$[string_1, string_2] \tag{1}$$

where $string_1$ and $string_2$ are javascript strings, delimited by double quotes.

The English morphological component begins with some rules, not so different from those of the toy grammar above, which allow the system to recognize contracted forms in input, and to replace them with the long forms:

4a ⟨*English morphological rules* 4a⟩≡ (3a) 4b▷

```
["cannot", "can not"],
["don't", "do not"],
["doesn't", "does not"],
["didn't", "did not"],
```

When the system is translating from English to Spanish, the first string is matched against the input, and, if a match is found, the second string is substituted for the first one. So, "cannot" is changed into "can not", "don't" into "do not", etc.

When the system is translating from Spanish to English, the direction is reversed: $string_2$ is the match, and $string_1$ the replacement.

After such "orthographical" rules, true morphological analysis can begins:

4b ⟨*English morphological rules* 4a⟩+≡ (3a) ◁4a 5▷

```
["does", "3 do"],
["#did#", "#PRET do#"],
["#has#", "#3 have#"],
```

the form "does" is recognized as the present, third person singular (here written as 3) of the stem "do", and the form "did" as the preterite form of the same stem.

A word can be preceded or followed by the character # which means "word boundary". That is, the system must recognize "did" as the PRET form of "do" only if "did" is not a part of a larger word.

It is safe to insert such boundaries when we are dealing with short words, in order for the system not to confuse them with parts of longer words.

2.2 Ordering of rules

In the following rules, the existential predicate "there is" is recognized as a different item from the form "is", and marked as such:

```
["there is not", "NO EXIST"],
["there is", "EXIST"],
["#is#", "#3 be#"],
```

When the system is translating from English into Spanish, the match "there is not" is tried *before* the match "there is". If the order were reversed, the system would no more be able to recognize "there is not", because "there is" would have already been substituted by "EXIST". So, the longest match is to be placed first.

However, this won't work in reverse mode, when the system is translating from Spanish to English: in that case, rules are attempted from last to first, so the system will first match the string "NO EXIST" against the rule "EXIST" → "there is". As the match will be found, the input string "NO EXIST" will be replaced by "NO there is", and the context for the application of the rule

```
["there is not", "NO EXIST"],
```

will be destroyed. (This was in fact an actual bug in the grammar, and the sentence "no hay personas así" were translated as "NO there are persons like this"). I corrected the bug by reformulating the rules like this:

5 ⟨*English morphological rules* 4a⟩+≡ (3a) ◁4b 6▷

```
// EXISTENCE
   ["there is", "EXIST"],
   ["EXIST not", "NO EXIST"],
```

In the same way, other forms of the existence predicate can be recognized:

6 ⟨*English morphological rules* 4a⟩+≡ (3a) ◁5 7▷

```
["there are", "p EXIST"],
["p EXIST not", "NO p EXIST"],

["there was", "#PRET EXIST#"],
["PRET EXIST not", "NO PRET EXIST"],

["there were", "p PRET EXIST"],
["p PRET EXIST not", "NO p PRET EXIST"],
["there AUX be not", "NO AUX EXIST", "AUX=can,will,could,would"],
["there AUX be", "AUX EXIST", "AUX=can,will,could,would"],
```

Let's see how the system analyzes and synthesizes the string "there are not".

In translating from English to Spanish, the rules are interpreted as:

$$there \quad are \rightarrow p \quad EXIST$$
$$p \quad EXIST \quad not \quad \rightarrow NO \quad p \quad EXIST$$

and "there are not" first becomes "p EXIST not" (rule 1) and then "NO p EXIST".

When the system is teranslating from Spanish to English, the rules are interpreted as follows:

$$NO \quad p \quad EXIST \rightarrow p \quad EXIST \quad not$$
$$p \quad EXIST \rightarrow there \quad are$$

if the input string is "NO p EXIST", the first rule will change it into "p EXIST not". The second rule will recognize "p EXIST" and change it into "there is", leaving "not" as it stands. The whole will thus become "there is not".

In the following rules, I will use a little trick: as the most frequent form of negation is the one with the auxiliary "do", I will use it as the unmarked form. Therefore, instead of a "do-insertion" rule, I write a "do-deletion" rule with the verb "to be":

7 ⟨*English morphological rules* 4a⟩+≡ (3a) ◁6 8▷

```
// Forms of TO BE

  ["#you were not#", "#you do not PRET be#"],
  ["#were not#", "#do not p PRET be#"],
  ["#were#", "#p PRET be#"],

  ["#I am not#", "#I do not be#"],
  ["#I am#", "#I be#"],

  ["#I was not#", "#I do not PRET be#"],
  ["#I was#", "#I PRET be#"],
  ["#was#", "#3 PRET be#"],

  ["#are#", "#p be#"],
  ["#is#", "#3 be#"],
```

although these forms look awful, we are just performing a sort of grammatical analysis:

```
["#you were not#", "#you do not PRET be#"],
```

means simply: analyze the string "you were not" into $you + NEGATION + PRET_{tense} + be_{stem}$.

2.3 Variables

While the verb "to be" has peculiar inflected forms, other verbal primes have a more patterned behaviour. Let's analyze the "s-form" of a verb into $3 + stem$. Using simple rules, we would have to write something like this:

```
["touches", "3 touch"],
["happens", "3 happen"],
["moves",   "3 move"],
["says",    "3 say"],
["feels",   "3 feel"],
```

and so on.

While "touch" must be set apart because of the $-es$ suffix, all other stems behave in the same way, and writing a simple list of rules like the previous one loses the pattern, and, moreover, is a long and tedious process.

The pattern can be captured using *variables*. All the rules seen so far consist of two parts:

$$[string_1, string_2]$$

and the rule for "touches" will be, accordingly,

8 ⟨*English morphological rules* 4a⟩+≡ (3a) ◁7 9a▷

```
// S-FORMS OF VERBS

  ["touches", "3 touch"],
```

All "regular" verbs can be analyzed with a single *tripartite* rule. Such rules have the form the form

$$[string_1, string_2, string_3]$$

where $string_3$ is a list of variables, semicolon-separated:

$$"var_1; var_2; ...; var_n"$$

and each var_i group has the following structure:

$$variable\_name = value_1, value_2, ..., value_n$$

The variables thus defined will appear in $string_1$ and $string_2$, and will be able to assume each of the values declared.

Let's look directly at the s-form rule:

9a ⟨*English morphological rules* 4a⟩+≡ (3a) ◁8 9b▷

```
["#STEMs#",
 "#3 STEM#",
 "STEM=happen, move, say, hear, feel, want, think,\
       know, exist, see, live, die"],
```

This rule has a variable list consisting of only one member, so no semicolon is needed.

The variable name is the string `STEM`. The rule compiler will build one separate rule for each different value of the variable, so that this single rule will be equivalent, after compilation, to the list of rules above.

The $string_3$ part of the rule (the variable list) is a single string, so it will be enclosed in double quotes and, if we need to span it on several line, like in this case, we'll have to put a "continuation backslash".

In this rule, the string `STEM` is identified as a variable by the fact that it is declared as such in the third part of the rule. We could have defined the rule like this:

```
["#goess#",
 "#3 goes#",
 "goes=happen, move, say, hear, feel, want, think,\
       know, exist, see, live, die"],
```

Though perverse, it would have (probably) worked (the variable is called "goes"). Of course, it is a good rule to choose variable names:

- which reflect their function, and
- which do not correspond to words in either the source or target languages, to avoid unexpected behaviour on the part of the system.

2.4 Complex variables

Variables can also be used in a more complex way.

Having defined the "s-form" of the verbal primes, let's now look at the past tense. For idiosyncratic forms, the simple two-part rule will be enough:

9b ⟨*English morphological rules* 4a⟩+≡ (3a) ◁9a 10a▷

```
// VERBS: Past Tense

  ["knew", "PRET know"],
  ["saw", "PRET see"],
```

and for regular verbs we can define a variable STEM as for the S-FORM:

10a ⟨*English morphological rules* 4a⟩+≡ (3a) ◁9b 10b▷

```
["STEMed",
 "PRET STEM",
 "STEM=happen,touch,want,exist"],
```

There are, however, sub-regularities which we cannot capture with variables as used so far. For example, we can simply write:

```
["thought", "PRET think"],
["felt",    "PRET feel"],
```

treating "felt" and "thought" as whole forms. This would do. But suppose we want to say that "felt" and "thought" consist of a stem $fel-$ and $though-$ plus a suffix $-t$[1].

We cannot use a variable `STEM` like this:

```
["STEMt",
 "PRET STEM",
 "STEM=though,fel"],
```

because we want the analyzed forms to be `PRET feel`, `PRET think`, and not `PRET fel` or `PRET though`.

We can achieve what we want using a complex variable stem, defined as in the following rule:

10b ⟨*English morphological rules* 4a⟩+≡ (3a) ◁10a 11▷

```
["STEM2t",
 "PRET STEM1",
 "STEM=think|though,  feel|fel"],
```

[1] In this example, a simple list of two rules would perhaps be easier and faster. However, we will see later many cases in which we do need perform a similar analysis.

We will recall that a variable is defined as:

$$variable\_name = value_1, value_2, ..., value_n$$

Each $value_i$ item can be a string, or a *list of alternant strings* separated by the pipe character (`|`). In the previous rule, the variable `STEM` is defined as:

```
STEM  =  think|though,  feel|fel
```

and each $value_i$ item has two alternatives: `think | thought` and `feel | fel`.

In the $string_1$ and $string_2$ part of the rule, reference to the n-th alternative is made by $variable\_name_n$. The alternants correspond to each other: when `think` is substituted for `STEM1`, `though` is substituted for `STEM2`, and when `feel` is substituted for `STEM1`, `fel` is substituted for `STEM2`.

So, the rule

```
["STEM2t",
 "PRET STEM1",
 "STEM=think|though,  feel|fel"],
```

corresponds to (and is translated by the rule compiler into) the following to rules:

```
["thought", "PRET think"],
["felt",    "PRET feel"],
```

The possibility of defining alternative values which correspond each other will be very useful when defining rules which translate English pattern into the corresponding Spanish ones, because it will allow to set up correspondences between single points points in the pattern.

The following rule analyzes the past tense of verbs which take a $-d$ suffix, with or without change in the stem:

11 ⟨*English morphological rules* 4a⟩+≡ (3a) ◁10b 12a▷

```
["STEM1d",
 "PRET STEM2",
 "STEM=move|move, sai|say, hear|hear, ha|have,\
       live|live, coul|can, die|die"],
```

and corresponds to the following list of rules:

```
["moved", "PRET move"],
["said", "PRET say"],
["heard", "PRET hear"],
["had", "PRET have"],
["lived", "PRET live"],
["could", "PRET can"],
["died", "PRET die"],
```

In the same way, we can analyze the ING-form of verbs with only two rules: a rule without alternative values will suffice for the cases in which $-ing$ is added to the stem without modifications:

12a ⟨*English morphological rules* 4a⟩+≡ (3a) ◁11 12b▷

```
// Verbs: ING-forms

  ["STEMing",
   "ING STEM",
   "STEM=happen, say, feel, think, touch, \
         be, do, hear, see, exist"],
```

while alternants will be used with stem-modifying verbs:

12b ⟨*English morphological rules* 4a⟩+≡ (3a) ◁12a 13a▷

```
  ["STEM1ing",
   "ING STEM2",
   "STEM=mov|move, hav|have, dy|die, liv|live"],
```

We are almost done with the English morphology component. We still need to implement a sort of "affix hopping" rule, because it will be useful, in the translation to Spanish, to have the negation in front of the analyzed $inflected-form + STEM$ form:

13a ⟨*English morphological rules* 4a⟩+≡ (3a) ◁12b 13b▷

```
// Verbs: "Affix-hopping"

   ["INFL do not",
    "do not INFL",
    "INFL=3, PRET"],

   ["INFL be not",
    "do not INFL be",
    "INFL=PRET p,  PRET 3,  PRET,  1,  3,  p"],

   ["PAST MOD not",
    "do not PAST MOD",
    "MOD=can; PAST=PRET, 0"],
```

in the last rule, two variables have been defined, separated by a semicolon.

2.5 The "in_my_body-hack"

Only a few rules are needed for nouns. I have found it easier to implement the singular-plural alternation directly into the syntactic component, because Spanish shows the same alternation, and the alternation has syntactic consequence on the sentence level (because of subject-verb agreement), so we need anyway to set up separate patterns for masculine and feminine, singular and plural.

Now, a rather tricky situation is generated by the fact that in this variety of Spanish, *su* can mean "your" and "their": "esta persona siente algo *en su cuerpo*" and "usted siente algo *en su cuerpo*" will have to be distinguished for a right English translation. To achieve this, the collocation "in my/your/their body" is set apart both in English and in Spanish, and will be referred to expressely in a sentence level rule.

13b ⟨*English morphological rules* 4a⟩+≡ (3a) ◁13a 14a▷

```
   ["in PART1 POSS1 body", "LOC-PART2-POSS2-BODY",
    "POSS=your|2, their|3;\
     PART=this part of|TP, the same part of|SP, a part of|OP, \
          one part of|OP, 0|N"],
```

Then we will need to distinguish absolute uses of the primes BODY, PART, KIND from their possessed use (*a kind (part) of someone (something)*, and *my (your, someone's) body*). In the following rule, when one of the strings `body`, `bodies` are preceded by one of the strings `my`, `your`, `'s`, are transformed into, respectively, `BODY-OF` and `BODYs-OF`; in this way, a noun-phrase matching rule will be able to match only absolute uses (and transform. for example "a big body" into "un cuerpo grande"); a special rule will transform "someone's body" into "el cuerpo de alguien" *after* the recognition of basic noun phrases is completed.

14a ⟨*English morphological rules* 4a⟩+≡ (3a) ◁13b 14b▷

```
// Possession

   ["POSS1 bod-NUM1",
    "POSS2 BODY-NUM2-OF",
    "POSS=my|my, your|your, 's|0; -NUM=y|0,ies|s"],
```

A similar rule is needed to take PART OF and KIND OF from absolute uses of PART and KIND:

14b ⟨*English morphological rules* 4a⟩+≡ (3a) ◁14a 14c▷

```
   ["PK1-NUM of",
    "PK2-NUM-OF",
    "PK=part|PART, kind|KIND; -NUM=s,0"],
```

English allolexes *noone*, *nothing* are recognized by the following rule (which works only when *noone* and *nothing* are subjects):

14c ⟨*English morphological rules* 4a⟩+≡ (3a) ◁14b 14d▷

```
// noone and nothing as subjects

   ["noone", "anyone do not"],
   ["nothing", "anything do not"],
```

the following rules takes care of *people* with a selectional quantifier, which must be traduced into Spanish using *personas*, and not **gente*:

14d ⟨*English morphological rules* 4a⟩+≡ (3a) ◁14c 15a▷

```
// people as allolex of person in plural

   ["Q of these people",
    "Q of these persons",
    "Q=one,two,few,some,many"],
```

The following rule takes care of variants in expressions involving time primes:

15a ⟨*English morphological rules* 4a⟩+≡ (3a) ◁14d 15b▷

```
// variants of time primes

   ["at some times", "sometimes"],
   ["at many times", "many times"],
```

and the last rule introduces an empty complementizer when the string `know` is followed by the string `wh`:

15b ⟨*English morphological rules* 4a⟩+≡ (3a) ◁15a

```
// empty complementizer in sentences like:
// someone knows what this thing is

   ["Vcon WH",
    "Vcon COMPL wh",
    "Vcon=say,know; WH=wh, how"]
```

This empty complementizer will be needed in the rule for the valence of the prime KNOW. Note how simply this rule can be defined: we do not need to include all possible forms KNOW can assume (*knows, knew, known*), because these will have already been recognized by the previous rules, and the string which indicates the inflected form (`PRET`, `3`, `p`, `ING`) comes *before* the string representing the stem. And we are hepled by the fact that wh-words in English start with the string `wh`.

As the previous rule was the last one of the rule list, there is no comma at the end.

3 The Spanish morphology component

After having gone in detail through the English morphology rules, we can look more briefly at the equivalent Spanish module. There will be almost no new machinery in the rule system.

The list of rules which compose the Spanish morphological analyzer are stored in a variable called `spaCO_morph`:

15c ⟨*Spanish morphology* 15c⟩≡ (3b)

```
// SPANISH MORPHOLOGY

var spaCO_morph =  [
  ⟨Spanish morphological rules 16a⟩
];
```

The first rule is a simple orthographical regularization, which will permit to state the morphological rules for some verbs more succintely:

16a ⟨*Spanish morphological rules* 16a⟩≡ (15c) 16b▷

```
    ["ORTO1",
     "ORTO2",
     "ORTO=senti'|sinti', hice|hize, hicieron|hizieron, comigo|con mi'"
  ],
```

3.1 The verb system

Now we can begin the analysis of the verb forms. First, the two idiosyncatic first person singular forms of *ser*, *estar* and *ir*[2]:

16b ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁16a 16c▷

```
    ["STEM1", "1 STEM2",
     "STEM=estoy|esta', #soy#|#es#, voy|va"],
```

We'll take the third person singular of the indicative present as the unmarked stem form, to which we will reduce all other forms by analysis. So, "quieren" will be analyzed as `p quiere` (where `p` means "third person plural"), "quise" as `1 PRET quiere`.

Third person plurals of indicativer present are simply formed by suffixing $-n$ to the stem form; we first allow for the exeptional "son" (they are) in a separate rule, then use a variable `STEM` for the general case:

16c ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁16b 17▷

```
    //3rd person plural

    ["#son#", "#p es#"],

    ["STEMn",
     "p STEM",
     "STEM=siente, puede, mueve, piensa, hace, oye, tiene, muere,\
           vive, sabe, quiere, esta', va, existe, dice, conoce, pasa"],
```

[2]GO is not a prime; the verb *ir* is present in the grammar because of the periphrastic future *ir a + INF*.

In the parallel grammar for "mainland" Spanish, in which we have to allow for "tú"-forms in second person singular, a simple extension to this rule handles the second person, which is obtained by suffixing $-s$ to the stem[3]:

```
["STEM-INFL1",
 "-INFL2 STEM",
 "STEM=siente, puede, mueve, piensa, hace, oye, tiene, muere,\
       vive, sabe, quiere, esta', va, existe, dice, conoce, \
       pasa; \
 -INFL= s|2, n|p"],
```

Now for first person singular. It adds *-o* to the truncated stem, so we must define a `STEM` variable with two alternative values. We also need a separate rule for the idiosyncratic form "sé" ("I know") (stem *sabe-*):

17 ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁16c 18▷

```
//1st person singular

["STEM1o",
 "1 STEM2",
 "STEM=piens|piensa, oig|oye, dig|dice, hag|hace, teng|tiene,\
       muer|muere, viv|vive, pued|puede, sient|siente,\
       quier|quiere, muev|mueve, teng|tiene, \
       toc|toca, exist|existe, conozc|conoce, #ve|#ve"],

 ["se'", "1 sabe"],
```

[3]The hyphen before the variable INFO is simply part of the variable's name – the variable is actually -INFO. It has no particular computational status; it's simply "syntactic sugar"

Indicative imperfect tense is needed because it is the default past tense for some verbs; with the other verbs it translates the English progressive. We need two separate rules for the two different conjugation patterns.

These rules introduce a new possibility for variables with alternative values, namely, the possibility of introducing a *zero value*, written `0`.

When the character `0` is found in the value list of a variable (both in simple value lists as in lists with alternants), it is substituted with the empty string by the rule compiler (so it will match "nothing". It is useful for facultative elements).

18 ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁17 19▷

```
// imperfect tense

 ["STEM1-INFL1",
  "-INFL2 IMPF STEM2",
  "STEM=hac|hace, mov|mueve, viv|vive, mor|muere, ten|tiene, \
        sent|siente, quer|quiere, #v|#ve, o|oye, sab|sabe, \
        pod|puede, ten|tiene, dec|dice, hab|hay, exist|existe, \
        conoc|conoce; \
   -INFL=i'an|p, i'a|0"],


["STEM1-INFL1",
 "-INFL2 IMPF STEM2",
 "STEM=pensaba|piensa, tocaba|toca, estaba|esta', era|es, iba|va; \
  -INFL=n|p, 0|0"],
```

The last rule is equivalent to the following list of rules:

```
["pensaban", "p IMPF piensa"],
["pensaba",  "IMPF piensa"],
["tocaban", "p IMPF toca"],
["tocaba", "IMPF toca"],
...
```

Note that it is important that the second variable, `INFL`, be defined as

```
-INFL=n|p, 0|0
```

rather than

```
-INFL=0|0, n|p
```

because, in the latter case, the rule would be compiled as:

```
["pensaba",  "IMPF piensa"],
["pensaban", "p IMPF piensa"],
["tocaba", "IMPF toca"],
["tocaban", "p IMPF toca"],
...
```

It is easy to imagine what would happen when the input text contains "pensaban": the first rule would match the first seven characters of `pensaba`*n*, analyzing it into `IMPF piensa`*n*.

So it is a good rule to always put the *longest match first* (alternatively, in this particular case, we could have used word boundaries), and the zero value of variables as the last one.

Now for the simple past. We want to write some rules which analyzes a past tense into a string `PRET` + *stem*. The first such rule recognizes the stem-accented pattern (that of *hice*, *quise*), which is well represented among the primes:

19 ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁18 20a▷

```
//simple past

["STEM1-INFL1",
 "-INFL2 PRET STEM2",
 "STEM=hiz|hace, sup|sabe, estuv|esta', tuv|tiene, quis|quiere,\
       dij|dice; \
  -INFL=e|1, ieron|p, o|0"],
```

then we can allow for the tonic pattern, in which we must distinguish the first conjugation from the others:

20a ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁19 20b▷

```
["STEM1-INFL1",
 "-INFL2 PRET STEM2",
 "STEM=pens|piensa,pas|pasa,toc|toca; \
  -INFL=e'|1,o'|0,aron|p"],

["STEM1-INFL1",
 "-INFL2 PRET STEM2",
 "STEM=viv|vive,mur|muere,sint|siente, conoc|conoce, mov|mueve; \
  -INFL=ieron|p, i'|1,io'|0"],
```

"hear" is also apart:

20b ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁20a 20c▷

```
["V1",
 "V2 PRET oye",
 "V=#oi'#|#1#, oyo'|0, oyeron|p"],
```

The other tenses of Spanish are treated in the same way. Here are present and imperfect subjunctive:

20c ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁20b 21a▷

```
// present subjunctive
["STEM1INFL1",
 "INFL2 SUBJ STEM2",
 "STEM=piense|piensa, toque|toca, saiba|sabe, quiera|quiere,\
       sienta|siente, haga|hace, vea|ve, exista|existe, \
       oiga|oye, pase|pasa, mueva|mueve, viva|vive, muera|muere,\
       pueda|puede, tenga|tiene, est|esta', conozca|conoce, #sea|#es;\
  INFL=n|p, 0|0"],


// imperfect subjuntive
["STEM1INFL1",
 "INFL2 PSUBJ STEM2",
 "STEM=pensase|piensa, supiese|sabe, quisiese|quiere, sentise|siente,\
       hiciese|hace, vese|ve, sintiese|siente, conociese|conoce,\
       existiese|existe,oyese|oye, pasase|pasa, moviese|mueve, \
       viviese|vive, muriese|muere, pudiese|puede, estuviese|esta',\
       fuese|es, tuviese|tiene, tocase|toca; \
  INFL=n|p,0|0"],
```

The rule for infinitive will also take care of eventual clitic object pronoun, analyzing a form like e.g. "hacerle" into `INF le hace`.

21a ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁20c 21b▷

```
// infinitive
 ["STEM1rCLIT",
  "INF CLIT STEM2",
  "STEM=pensa|piensa, toca|toca, sabe|sabe, quere|quiere, \
         senti|siente, hace|hace, #ve|#ve, existi|existe, \
         oi'|oye, pasa|pasa, move|mueve, vivi|vive, mori|muere,\
         pode|puede, tene|tiene,  esta|esta', habe|hay, \
         deci|dice, conoce|conoce, #se|#es;\
    CLIT=les,le,me,lo,se,0"],
```

As for gerund, the rule is straightforward:

21b ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁21a 21c▷

```
// gerund
 ["STEM1ndo",
  "GER STEM2",
  "STEM=pensa|piensa, toca|toca, sentie|siente, hacie|hace,\
        #vie|#ve,  existie|existe, oye|oye, pasa|pasa, \
        move|mueve, vivie|vive, murie|muere, pudie|puede, \
        tenie|tiene, dicie|dice, conoci|conoce"],
```

Now for some "hacks". First of all, the free pronoun "yo" is used for disambiguation, in verbal paradigms whose third persons are the same than their first persons (*hacía, hiciese*). As we analyze "hago" into `1 hace` and "hice" into `1 PRET hace`, we want to analyze "yo hacía" into `1 IMPF hace`. The following rule will do the trick:

21c ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁21b 22a▷

```
// First person with "yo"
 ["yo NEG AFF",
  "NEG 1 AFF",
  "NEG=no,0; AFF=IMPF,SUBJ,PSUBJ"],
```

This is one of the cases in which Travis specifications must still be fully implemented. In this grammar, *yo* must be mentioned each time before an ambiguous first/third person form, while it should be only once, the first time, and then omitted until subject changes. The lack of a textual level causes this inconsistency, which will have to be addressed in successive versions of the work.

Now let's account for the verb whose imperfect tense is the unmarked past tense. The string `IMPF`, which has been generated by the analysis of the surface form of these verbs, is changed into `PRET`. So, for example, "yo tenía" becomes, first of all, `yo IMPF tiene`, then `IMPF 1 tiene`, and, finally, `PRET 1 tiene`. We can also see that it is important that this rule be ordered after the previous "hack".

22a ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁21c 22b▷

```
//imperfect as unmarked past tense
 ["IMPF Vstat",
  "PRET Vstat",
  "Vstat=quiere, tiene, esta', piensa, sabe, vive, es, puede, conoce,\
        hay"],
```

And, as a further hack, translation rules need for an eventual clitic object immediately *before* the verb stem, so we write:

22b ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁22a 22c▷

```
//clitic unclimbing
 ["CLIT PERS AFF",
  "PERS AFF CLIT",
  "AFF=PRET,PSUBJ,SUBJ,IMPF,0; \
   PERS=1,3,p,0; \
   CLIT=les,le,me,lo,se"],
```

in this way, for example, "le hago" becomes first `le 1 hace` and then `1 le hace`. After having analyzed the imperfect as the default past tense form of inherently durative verbs, we now need to account for the other cases of imperfect. I will tentatively render a Spanish imperfect with an English progressive form, although the status of the progressive form in NSM is still to be defined.

22c ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁22b 23a▷

```
 ["IMPF", "PRET PROG"],
```

Before we can move to nouns, we still need to account for the only Spanish reflexive verb which has the privilege of being a prime, that is, *moverse*. When the subject is first person, the reflexive clitic needs to be changed into "me", in all other cases, it has the form "se":

23a ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁22c 23b▷

```
// reflexive
["1 AFF AUX me",
 "1 AFF AUX se",
 "AFF=PRET,PSUBJ,SUBJ,IMPF,0; \
  AUX=quiere INF,  puede INF,  va a INF,  0"],
```

3.2 The noun system

Let us now briefly look at some nominal rules.

First of all, we analyze a preposition fused with an article:

23b ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁23a 23c▷

```
["#al#", "#a el#"],
["#del#", "#de el#"],
```

and we add the definite article to the prime PEOPLE, which in Spanish is normally "la gente":

23c ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁23b 23d▷

```
["DE gente", "DE la gente"],
```

then we will implement the "in_my_body-hack" corresponding to the English one (see section 2.5, p. 13), which will allow to disambiguate occurrences of "en su cuerpo" as either "in *your* body" or "in *their* body", depending on the subject ("esta persona siente algo en su cuerpo" must become "this person feels something in *their* body", while "usted siente algo en su cuerpo" must be translated as "you feel something in *your* body"):

23d ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁23c 24a▷

```
["en PART1 su cuerpo",
 "LOC-PART2-2-BODY",
 "PART=esta parte de|TP, la misma parte de|SP, una parte de|OP,\
       0|N"],
```

after this, we can distinguish absolute uses of "cuerpo", "parte" and "cosa" from their uses as modifiers (part/kind of something; someone's body), by setting apart these last uses. In Spanish, "someone's" body is "cuerpo de alguien", thus it has the same syntax as "tipo/parte de algo":

24a ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁23d 24b▷

```
// type of, part of, X's body
["PK1-NUM de",
 "PK2",
 "PK=parte|PARTE-NUM-DE,  tipo|TIPO-NUM-DE,  cuerpo|CUERPO-NUM-DE; \
  -NUM=s,0"],


["POSS-NUM cuerpo-NUM",
 "POSS-NUM CUERPO-NUM-DE",
 "POSS=mi,su; \
  -NUM=s,0"],
```

Now we must account for gender and number inflection in adjectives, which concord:

- with the nominal head in a substantive phrase: "cosa buen-*a*", "sitios buen-*os*, etc.
- with the subject when used as predicates: "esta cosa es buen-*a*", "estes sitios son buen-*os*", etc.

We simply separate the suffix from the stem, and so we will be able to refer to both separately when we will define an adjectival predicate sentence (see section 4.12.1, p. 59)

24b ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁24a 24c▷

```
["ADJINFL",
 "ADJ -INFL",
 "ADJ=buen,mal,verdader,cercan,lejan,pequen'; \
  INFL=os,as,o,a"],
```

the last rule of the Spanish section, like the one of the English section, sets an empty complementizer which will be useful to distinguish the "wh-sentence" valence of KNOW ("yo sé *quiénes son estas personas*):

24c ⟨*Spanish morphological rules* 16a⟩+≡ (15c) ◁24b

```
["Vcon WH",
 "Vcon COMPL WH",
 "Vcon=sabe,dice; \
  WH=quie'n,cua'ndo,do'nde,que'"]
```

4 The Spanish-English grammar

The Spanish-English grammar is strored in a javascript variable called `spaCO_eng_gr`. We can divide it into several subparts:

25a ⟨*Spanish English grammar* 25a⟩≡ (3b)

```
//_________________________
// SPANISH - ENGLISH SYNTAX
//=========================

var spaCO_eng_gr = [
⟨Maybe Rule 25b⟩
⟨Time Modifiers 26a⟩
⟨Substantive Phrases 28⟩
⟨Locative Expressions 42b⟩
⟨Standalone It 43c⟩
⟨Preposition Plus Substantive Phrase 42d⟩
⟨Valence Rules 46⟩
⟨Verb Phrase 54a⟩
⟨Sentence Rules 57a⟩
];
```

As in the morphology components, each rule has one of the two forms:

$$[Spanish\_string, \quad English\_string]$$

$$[Spanish\_string, \quad English\_string, \quad veriables]$$

Here is the very first rule:

25b ⟨*Maybe Rule* 25b⟩≡ (25a)

```
["tal vez", "maybe"],
```

It stands before time adverbial rules, in which *vez* will match the English prime "time".

4.1 Phrase markers

We begin by translating *time adverbials*, starting with *occurrence adverbials* such as "once" or "many times".

In the following rule, the notation `<*TIMES*>` at the beginning of the two strings means: if you recognize the pattern specified in the rule, substitute it with the English equivalend and add a "TIMES" phrase marker to the pattern. Successive rules will be able to refer to any such string simply as `(*TIMES*)`.

To add a phrase marker to a pattern, we write the phrase marker at the beginning of both string of the rule, enclosed between `<*` and `*>`, like in the following rule:

26a ⟨*Time Modifiers* 26a⟩≡ (25a) 26b▷

```
// TIME MODIFIERS
// **************

  ["<*TIMES*> TIME1",
   "<*TIMES*> TIME2",
   "TIME=una vez|once, dos veces|twice, pocas veces|a few times, \
         algunas veces|at some times, muchas veces|many times, \
         todas las veces|all the times, siempre|always"],
```

in translating from Spanish to English, the input string "todas las veces" will become `<TIMES* all the times*>`.

When, in some later rules, we need to refer to a `TIMES` phrase as a whole, we can write `(*TIMES*)`, enclosing the phrase marker between `(*` and `*)`. Note that such a phrase becomes like a "black box" for subsequent rules, which can refer to it as a whole but not access its individual components.

Here are the other rules for time modifiers:

Duration

26b ⟨*Time Modifiers* 26a⟩+≡ (25a) ◁26a 27▷

```
  ["<*TDUR*> por TIME1",
   "<*TDUR*> for TIME2",
   "TIME=mucho tiempo|a long time, mucho tiempo|a very long time,\
         un tiempo|some time, poco tiempo|a short time,\
         poco tiempo|a very short time"],

  ["<*TDUR*> en un momento",
   "<*TDUR*> in a moment"],
```

Location in time

27 ⟨*Time Modifiers* 26a⟩+≡ (25a) ◁26b

```
["<*TPAST*> hace T1",
 "<*TPAST*> T2 ago",
 "T=mucho tiempo|a long time, mucho tiempo|a very long time, \
    un tiempo|some time,  poco tiempo|a short time, \
    un momento|a moment"],

["<*TSUCC*> TIME1 despue's",
 "<*TSUCC*> TIME2 after",
 "TIME=mucho tiempo|a long time, mucho tiempo|a very long time, \
       un tiempo|some time, poco tiempo|a short time, \
       poco tiempo|a very short time, un momento|a moment, 0|0"],

["<*TPRED*> TIME1 antes",
 "<*TPRED*> TIME2 before",
 "TIME=mucho tiempo|a long time, mucho tiempo|a very long time,\
       un tiempo|some time, poco tiempo|a short time,\
       poco tiempo|a very short time, un momento|a moment, 0|0"],

["<*TPUNCT*> TIME1",
 "<*TPUNCT*> TIME2",
 "TIME=en ese tiempo|at that time, al mismo tiempo|at the same time,\
       en este tiempo|at this time, \
       en este otro tiempo|at this other time, \
       en otro tiempo|at another time, \
       en un tiempo|at some time, \
       en muchos tiempos|at many times, a veces|sometimes, \
       a veces|at some times,\
       a un tiempo|at one time, #ahora#|#now#"],
```

Table 1: Features of Noun Phrases

Category	animate: `a`	inanimate: `i`	time: `t`	place `p`
GENDER	masculine: `m`	feminine: `f`		
NUMBER	singular: `s`	plural: `p`	collective: `c`	

4.2 Substantive phrases

We will have to distinguish various types of substantive phrases because of their syntactic behaviour:

1. Gender: masculine and feminine;
2. Number: singular, plural and also collective (the prime PEOPLE has different syntactic behaviours in Spanish, where it behaves as a singular, and English, where it is plural, so it will have to be set apart);
3. Category:
   (a) Animate and Inanimate (this last distinction is important, for example, in order to distinguish "veo algo" from "veo a alguien");
   (b) Time;
   (c) Place;
   (d) the prime SIDE, which combines with the preposition "on" in English.

We will mark a noun phrase with the symbol `N`, followed by three characters which indicate the value assumed by the three features:

So, for examples, we will write `Nasf` for "animate singular feminine", or `Nimp` for "inanimate masculine plural".

Several groups of rules are needed for substantive phrase:

28 ⟨*Substantive Phrases* 28⟩≡ (25a)
⟨*N Of This Kind* 30a⟩
⟨*Someone and Something* 29a⟩
⟨*Noone and Nothing* 31⟩
⟨*Noun Phrases* 32⟩
⟨*Body Kind Part* 39a⟩
⟨*Selector* 41a⟩
⟨*The Same* 41b⟩

First of all, let's provide for "this someone" and "this something", which, although marginal both in English and in Spanish is nonetheless useful in many definitions:

29a ⟨*Someone and Something* 29a⟩≡ (28) 29b ▷

```
// this someone, this something
  ["<*Nams*> DET1 alguien",
   "<*Nams*> DET2 someone",
   "DET=este mismo|this same, el mismo|the same, el otro|the other, \
        este otro|this other, este|this"],

  ["<*Nims*> DET1 algo",
   "<*Nims*> DET2 something",
   "DET=este mismo|this same, el mismo|the same, el otro|the other,\
        este otro|this other, este|this"],
```

Now for SOMEONE and SOMETHING in their more straightforward uses (e.g. "someone else", "something good"):

29b ⟨*Someone and Something* 29a⟩+≡ (28) ◁29a 30b ▷

```
// NP: something, someone
   ["<*Nims*> Q1 DETN1 INT1 AP1",
    "<*Nims*> Q2 DETN2 INT2 AP2",
    "DETN=algo|something; \
     INT=muy|very, 0|0; \
     AP=grande|big, pequen' -o|small, buen -o|good, mal -o|bad, \
        verdader -o|true, otro|else, ma's|more, asi'|like this, \
        cercan -o|near, lejan -o|far, (*Ac.*)|(*Ac.*), 0|0; \
     Q=mucho ma's|much more, mucho|much, ma's|more, \
       un poco de|a little, 0|0"],
```

the preceding rule tries to capture all the combinations of SOMEONE and SOMETHING in a substantive phrase: the variables `Q`, `INT` and `AP` stand, respectively, for *quantifier, intensifier* and *adjective*.

Among the values that the variable `AP` can assume, there is `(*Ac.*)`. This is a reference to a segment like "of this kind", "of the same kind", "of another kind" , which is defined by the following two rules:

30a ⟨*N Of This Kind* 30a⟩≡ (28)

```
// N + de este tipo
  ["<*Acs*> de DET1 tipo",
   "<*Acs*> of DET2 kind",
   "DET=un|one, este|this, el mismo|the same, el otro|the other,\
        otro|other"],

  ["<*Acp*> de DET1 tipos",
   "<*Acp*> of DET2 kinds",
   "DET=estos|these, los mismos|the same, dos|two, pocos|a few, \
    muchos|many, #todos los#|#all#"],
```

In the substantive phrase rule, we have used the notation `(*Ac.*)` to refer to both these phrases: in a phrase reference, we can use the character "point" (`.`) to match any letter or digit, as in regular expressions. So, for example, if we need to refer to any substantive phrase, we can simply write `(*N...*)`, while we can refer to all plural substantive phrases as `(*N..p*)`.

The same reference `(*Ac.*)` is used in the SOMEONE rule:

30b ⟨*Someone and Something* 29a⟩+≡ (28) ◁29b

```
["<*Nams*> DETN1 INT1 AP1",
 "<*Nams*> DETN2 INT2 AP2",
 "DETN=alguien|someone; INT=muy|very, 0|0; \
  AP=grande|big, pequen' -o|small, \
     buen -o|good, mal -o|bad, verdader -o|true, otro|else, \
     ma's|more,  asi'|like this, cercan -o|near, lejan -o|far, \
     (*Ac.*)|(*Ac.*),  0|0"],
```

As for the allolexes “anyone” and “anything”in negative sentences, they will be rendered in Spanish by “nadie” and “nada”:

31 ⟨*Noone and Nothing* 31⟩≡ (28)

```
// NP: nothing, noone
   ["<*NiNs*> DETN1 INT1 AP1",
    "<*NiNs*> DETN2 INT2 AP2",
    "DETN=nada|anything; INT=muy|very, 0|0; \
          AP=grande|big, pequen' -o|small, \
          buen -o|good, mal -o|bad, verdader -o|true, otro|else, \
          ma's|more,  asi'|like this, creca|near, lejos|far, \
          (*Ac.*)|(*Ac.*), 0|0"],

   ["<*NaNs*> DETN1 INT1 AP1",
    "<*NaNs*> DETN2 INT2 AP2",
    "DETN=nadie|anyone; \
     INT=muy|very, 0|0; \
     AP=grande|big, pequen' -o|small, buen -o|good, mal -o|bad, \
        verdader -o|true, otro|else, ma's|more, asi'|like this, \
        cercan -o|near, lejan -o|far, (*Ac.*)|(*Ac.*), 0|0"],
```

4.3 Substantives

After "someone" and "something", we can write rules for their nominal allolexes "thing", "person", as well as for the other "nouns" of NSM. We will need a separate rule for each type we have distinguished.

The following rule tries to capture the fullest possible substantive phrase headed by the plural form "personas", allolex of the primes SOMEONE and PEOPLE:

personas

32 ⟨*Noun Phrases* 32⟩≡ (28) 33a▷

```
//NP animate fem. plur
  ["<*Nafp*> ALL1 DET1 Q1 NP1 INT1 AP1 AUGM1",
   "<*Nafp*> ALL2 DET2 Q2 AUGM2 INT2 AP2 NP2",

  "ALL=#todas#|#all#,0|0; \
       DET=estas otras|these other,estas|these,\
           las mismas|the same,las otras|the other,otras|other,0|0; \
       Q=dos|two, algunas|some, muchas|many, pocas|few, 0|0; \
       INT=muy|very, 0|0; \
       AP=grandes|big, pequen' -as|small, buen -as|good, mal -as|bad, \
          verdader -as|true, cercan -as|near, lejan -as|far, \
          (*Ac.*)|(*Ac.*), 0|0; \
       NP=personas|persons; \
       AUGM=ma's|more,0|0"],
```

The singular will be defined after the plural, in order for the longer match (e.g. `presonas`) to be recognized before the shorter (e.g. `persona`).

In the same way are defined the maximal substantive phrases headed by the other substantive primes and their allolexes. All the following rules are very similar to each other; for every rule, just a few things need to be adapted to the same schema:

- the head noun;
- the features in the phrase marker, and
- the form of the adjectives and determiners.

cosas

33a ⟨*Noun Phrases* 32⟩+≡ (28) ◁32 33b▷

```
//NP inanimate fem. plur
   ["<*Nifp*> ALL1 DET1 Q1 NP1 INT1 AP1 AUGM1",
    "<*Nifp*> ALL2 DET2 Q2 AUGM2 INT2 AP2 NP2",
   "ALL=#todas#|#all#,0|0; DET=estas otras|these other,estas|these,\
        las mismas|the same,las otras|the other,otras|other,0|0; \
    Q=dos|two, algunas|some, muchas|many, pocas|few, \
      (*Ac.*)|(*Ac.*), 0|0; \
    INT=muy|very, 0|0; \
    AP=grandes|big, pequen' -as|small, buen -as|good, mal -as|bad, \
       verdader -as|true,  cercan -as|near, lejan -as|far, 0|0; \
    NP=cosas|things, partes|parts, palabras|words; \
    AUGM=ma's|more,0|0"],
```

gente

33b ⟨*Noun Phrases* 32⟩+≡ (28) ◁33a 34a▷

```
//NP animate fem. collective (gente)
   ["<*Nafc*> DET1 NP1 INT1 AP1",
    "<*Nafc*> DET2 INT2 AP2 NP2",
   "DET=esta otra|these other, esta|these, la misma|the same,\
        la otra|the other, otra|other, mucha|many, poca|few, \
        un poco de|some,  ma's|more, toda la|all, la|0; \
    INT=muy|very, 0|0; \
    AP=grande|big, pequen' -a|small, buen -a|good,\
        mal -a|bad, verdader -a|true, cercan -a|near, \
        lejan -a|far, (*Ac.*)|(*Ac.*), 0|0; \
    NP=gente|people"],
```

persona

34a ⟨*Noun Phrases* 32⟩+≡ (28) ◁33b 34b▷

```
//NP animate femm. sing.
   ["<*Nafs*> DET1 NP1 INT1 AP1 AUGM1",
    "<*Nafs*> DET2 AUGM2 INT2 AP2 NP2",
   "DET=esta otra|this other,esta|this,la misma|the same,\
        la otra|the other,otra|another, #una#|#a#, una|one; \
    INT=muy|very, 0|0; \
    AP=grande|big, pequen' -a|small, buen -a|good, mal -a|bad, \
       verdader -a|true, cercan -a|near, lejan -a|far, \
       (*Ac.*)|(*Ac.*), 0|0; \
    NP=persona|person; AUGM=ma's|more,0|0"],
```

palabra

34b ⟨*Noun Phrases* 32⟩+≡ (28) ◁34a 35a▷

```
//NP inanimate fem. sing.
   ["<*Nifs*> DET1 NP1 INT1 AP1 AUGM1",
    "<*Nifs*> DET2 AUGM2 INT2 AP2 NP2",
    "DET=esta otra|this other, esta|this, la misma|the same, \
        la otra|the other,otra|another, #una#|#a#, una|one; \
    INT=muy|very, 0|0; \
    AP=grande|big, pequen' -a|small, buen -a|good,\
       mal -a|bad, verdader -a|true, cercan -a|near, \
       lejan -a|far, (*Ac.*)|(*Ac.*), 0|0; \
    NP=cosa|thing,parte|part,palabra|word; AUGM=ma's|more,0|0"],
```

cuerpos, sitios Here the absolute use of the primes is captured:

35a ⟨*Noun Phrases* 32⟩+≡ (28) ◁34b 35b▷

```
//NP masch. plur.
   ["<*Nimp*> ALL1 DET1 Q1 NP1 INT1 AP1 AUGM1",
    "<*Nimp*> ALL2 DET2 Q2 AUGM2 INT2 AP2 NP2",
   "ALL=#todos#|#all#,0|0; \
    DET=estos otros|these other, estos|these, \
        los mismos|the same, los otros|the other, \
        otros|other, 0|0; \
    Q=dos|two,muchos|many, pocos|few, algunos|some, 0|0; \
    INT=muy|very, 0|0; \
    AP=grandes|big, pequen' -os|small, buen -os|good, mal -os|bad, \
       verdader -os|true, cercan -os|near, lejan -os|far, \
       (*Ac.*)|(*Ac.*), 0|0; \
    NP=cuerpos|bodies, tipos|kinds; \
    AUGM=ma's|more,0|0"],
```

sitios

35b ⟨*Noun Phrases* 32⟩+≡ (28) ◁35a 36a▷

```
   ["<*Nlmp*> ALL1 DET1 Q1 NP1 INT1 AP1 AUGM1",
    "<*Nlmp*> ALL2 DET2 Q2 AUGM2 INT2 AP2 NP2",
    "ALL=#todos#|#all#,0|0; \
     DET=estos otros|these other,estos|these,los mismos|the same, \
         los otros|the other, otros|other, 0|0; \
     Q=dos|two,muchos|many, pocos|few, algunos|some, 0|0; \
     INT=muy|very, 0|0; \
     AP=grandes|big, pequen' -os|small, buen -os|good, mal -os|bad, \
        verdader -os|true, cercan -os|near, lejan -os|far, \
        (*Ac.*)|(*Ac.*), 0|0; \
     NP=sitios|places; \
     AUGM=ma's|more,0|0"],
```

sitio

36a ⟨*Noun Phrases* 32⟩+≡ (28) ◁35b 36b▷

```
["<*Ntmp*> ALL1 DET1 Q1 NP1 INT1 AP1 AUGM1",
 "<*Ntmp*> ALL2 DET2 Q2 AUGM2 INT2 AP2 NP2",

 "ALL=#todos#|#all#,0|0; \
  DET=estos otros|these other, estos|these, \
      los mismos|the same, los otros|the other,\
      otros|other, 0|0; \
  Q=dos|two,muchos|many, pocos|few, algunos|some, 0|0; \
  INT=muy|very, 0|0; \
  AP=grandes|big, pequen' -os|small, buen -os|good, mal -os|bad, \
     verdader -os|true, cercan -os|near, lejan -os|far, \
     (*Ac.*)|(*Ac.*), 0|0; \
  NP=tiempos|times; \
  AUGM=ma's|more,0|0"],
```

partes

36b ⟨*Noun Phrases* 32⟩+≡ (28) ◁36a 37a▷

```
["<*NLmp*> ALL1 DET1 Q1 NP1 INT1 AP1 AUGM1",
 "<*NLmp*> ALL2 DET2 Q2 AUGM2 INT2 AP2 NP2",

 "ALL=#todos#|#all#,0|0; \
  DET=estos otros|these other, estos|these,los mismos|the same,\
      los otros|the other, otros|other, 0|0; \
  Q=dos|two,muchos|many, pocos|few, algunos|some, 0|0; \
  INT=muy|very, 0|0; \
  AP=grandes|big, pequen' -os|small, buen -os|good, mal -os|bad,\
     verdader -os|true, cercan -os|near, lejan -os|far, \
     (*Ac.*)|(*Ac.*), 0|0; \
  NP=lados|sides;\
  AUGM=ma's|more,0|0"],
```

sitio

37a ⟨*Noun Phrases* 32⟩+≡ (28) ◁36b 37b▷

```
//NP masch. sg.
   ["<*Nlms*> DET1 NP1 INT1 AP1 AUGM1",
    "<*Nlms*> DET2 AUGM2 INT2 AP2 NP2",

   "DET=este otro|this other, este|this, el mismo|the same,\
        el otro|the other, otro|another,  #un#|#a#, un|one, 0|0; \
    INT=muy|very, 0|0; \
    AP=grande|big, pequen' -o|small, buen -o|good, mal -o|bad, \
       verdader -o|true, cercan -o|near, lejan -o|far,\
       (*Ac.*)|(*Ac.*), 0|0;\
    NP=sitio|place;\
    AUGM=ma's|more,0|0"],
```

tiempo

37b ⟨*Noun Phrases* 32⟩+≡ (28) ◁37a 38a▷

```
["<*Ntms*> DET1 NP1 INT1 AP1 AUGM1",
 "<*Ntms*> DET2 AUGM2 INT2 AP2 NP2",

 "DET=este otro|this other,este|this,el mismo|the same, \
      el otro|the other, otro|another,  #un#|#a#, un|one, 0|0; \
  INT=muy|very, 0|0; \
  AP=grande|big, pequen' -o|small, buen -o|good,\
      mal -o|bad, verdader -o|true, cercan -o|near, \
      lejan -o|far, (*Ac.*)|(*Ac.*), 0|0;\
  NP=tiempo|time; AUGM=ma's|more,0|0"],
```

tipo, cuerpo

38a ⟨*Noun Phrases* 32⟩+≡ (28) ◁37b 38b▷

```
["<*Nims*> DET1 NP1 INT1 AP1 AUGM1",
 "<*Nims*> DET2 AUGM2 INT2 AP2 NP2",

 "DET=este otro|this other, este|this, el mismo|the same,\
      el otro|the other, otro|another,  #un#|#a#, un|one, 0|0; \
  INT=muy|very, 0|0; \
  AP=grande|big, pequen' -o|small, buen -o|good, mal -o|bad, \
     verdader -o|true,  cercan -o|near, lejan -o|far, \
     (*Ac.*)|(*Ac.*), 0|0;\
  NP=tipo|kind, cuerpo|body; AUGM=ma's|more,0|0"],
```

parte

38b ⟨*Noun Phrases* 32⟩+≡ (28) ◁38a

```
["<*NLms*> DET1 NP1 INT1 AP1 AUGM1",
 "<*NLms*> DET2 AUGM2 INT2 AP2 NP2",

 "DET=este otro|this other,este|this,el mismo|the same, el otro|the other,\
      otro|another,  #un#|#a#, un|one, 0|0; \
  INT=muy|very, 0|0; \
      AP=grande|big, pequen' -o|small, buen -o|good,\
      mal -o|bad, verdader -o|true, cercan -o|near, lejan -o|far, \
      (*Ac.*)|(*Ac.*), 0|0;\
  NP=lado|side; AUGM=ma's|more,0|0"],
```

4.4 *Kind of* and *Part of*

After having recognized "simple" substantive phrases, we can embed them into a "higher level" substantive phrase headed by BODY, KIND or PART.

We will use whole noun phrases as alternative values of variables. In the following rule, `(*Na..*)` means "any animate substantive phrase".

39a ⟨*Body Kind Part* 39a⟩≡ (28) 39b▷

```
// NP's body
  ["<*Nimp*> NP1 CUERPOs-DE NP2",
   "<*Nimp*> NP3 BODYs-OF",
   "NP=los|(*Na..*)|(*Na..*), mis|0|my, sus|0|your"],

  ["<*Nims*> NP1 CUERPO-DE NP2",
   "<*Nims*> NP3 BODY-OF",
    "NP=el|(*Na..*)|(*Na..*), mi|0|my, su|0|your"],
```

In the following rule, we factor out the desinence *-os* from "algunos", "muchos", "pocos". The variable `NOUN` can have a whole NP as its value, or the bare substantives THING and PERSON.

39b ⟨*Body Kind Part* 39a⟩+≡ (28) ◁39a 40▷

```
// kinds of + NP
   ["<*Nimp*> Q1os TIPOs-DE NOUN1",
    "<*Nimp*> Q2 KINDs-OF NOUN2",
    "Q=much|many, poc|few, algun|some;\
     NOUN=(*N...*)|(*N...*), cosa|thing, persona|person"],

   ["<*Nimp*> Q1 TIPOs-DE NOUN1",
    "<*Nimp*> Q2 KINDs-OF NOUN2",
    "Q=dos|two,algunos|some,0|0; \
     NOUN=(*N...*)|(*N...*), cosa|thing, persona|person"],
```

In the KIND OF and PART OF rules, as in the rules above for other substantive phrases, we put the longest match (plural) first (kinds, parts). Following the plural, there come the rules for the singular. Here are all the rules:

40 ⟨*Body Kind Part* 39a⟩+≡ (28) ◁39b

```
// parts of + N

   ["<*Nifp*> Q1as PARTEs-DE NOUN1",
    "<*Nifp*> Q2 tp2-OF NOUN2",
    "Q=much|many, poc|few, algun|some;\
     NOUN=(*N...*)|(*N...*), cosa|thing, persona|person"],


   ["<*Nifp*> Q1 PARTEs-DE NOUN1",
    "<*Nifp*> Q2 PARTs-OF NOUN2",
    "Q=dos|two,algunos|some,0|0; \
     NOUN=(*N...*)|(*N...*), cosa|thing, persona|person"],

// kind of + NP

   ["<*Nimw*> DET1 TIPO-DE NOUN1",
    "<*Nimw*> DET2 KIND-OF NOUN2",
    "DET=un|a, un|one, 0|0;\
     NOUN=(*N...*)|(*N...*), cosa|thing, persona|person"],

// part of + NP

   ["<*Nifw*> DET1 PARTE-DE NOUN1",
    "<*Nifw*> DET2 PART-OF NOUN2",
    "DET=una|a, una|one, 0|0;\
     NOUN=(*N...*)|(*N...*), cosa|thing, persona|person"],
```

4.5 Selectors

Another "higher level" substantive phrase is the structure Quantifier + of + substantive phrase. The rules are straightforward:

41a ⟨*Selector* 41a⟩≡ (28)

```
// one/two... of Np

  ["<*Nifs*> una de (*Nifp*)",  "<*Nifs*> one of (*Nifp*)"],
  ["<*Nafs*> una de (*Nafp*)",  "<*Nafs*> one of (*Nafp*)"],
  ["<*Nims*> uno de (*Nimp*)",  "<*Nlms*> one of (*Nimp*)"],
  ["<*Nlms*> uno de (*Nlmp*)",  "<*Nlms*> one of (*Nlmp*)"],
  ["<*NLms*> uno de (*NLmp*)",  "<*NLms*> one of (*NLmp*)"],
  ["<*Ntms*> uno de (*Ntmp*)",  "<*Ntms*> one of (*Ntmp*)"],
  ["<*Nams*> una de (*Namp*)",  "<*Nams*> one of (*Namp*)"],

  ["<*Nifp*> Q1 de (*Nifp*)",
   "<*Nifp*> Q2 of (*Nifp*)",
   "Q=dos|two, pocas|few, algunas|some, muchas|many"],

  ["<*Nafp*> Q1 de (*Nafp*)",
   "<*Nafp*> Q2 of (*Nafp*)",
   "Q=dos|two, pocas|few, algunas|some, muchas|many"],

  ["<*Nimp*> Q1 de (*N[iltL]mp*)",
   "<*Nimp*> Q2 of (*N[iltL]mp*)",
   "Q=dos|two, ocos|few, algunos|some, muchos|many"],

  ["<*Namp*> Q1 de (*Namp*)",
   "<*Namp*> Q2 of (*Namp*)",
   "Q=dos|two, pocos|few, algunos|some, muchos|many"],
```

4.6 "lo mismo"

If the rules defined so far have not yet included the string "the same" in a substantive phrase, then it is used substantivally. This use is matched in Spanish by "lo mismo". First, we check whether it has a complement (e.g. "the same as me"):

41b ⟨*The Same* 41b⟩≡ (28) 42a▷

```
// the same as + NP
  ["<*Nins*> lo mismo que NP1",
   "<*Nins*> the same as NP2",
   "NP=(*N...*)|(*N...*), yo|me, usted|you"],
```

if it hasn't, we can match it as stand-alone:

42a ⟨*The Same* 41b⟩+≡ (28) ◁41b

```
// the same (substantive use)

  ["<*Nins*> lo mismo", "<*Nins*> the same"],
```

4.7 Locative Expressions

Locative expressions are parsed as `<*LOC*>` phrases in order for later rules to be allowed to incorporate them into the valency frame of a prime like $LIVE$ or BE_{loc}.

In locative expressions, the use of preposition is not always the same in both English and Spanish. We distinguish:

1. Local adverbs followed by *de* in Spanish and by nothing in English ("cerca de aquí" = "near here", "dentro de mí" = "inside me"):

42b ⟨*Locative Expressions* 42b⟩≡ (25a) 42c▷

```
// PREPOSITION + SUBSTANTIVE PHRASE
// ******************************
  ["<*LOC*> ADV1 de OBJ1",
   "<*LOC*> ADV2 OBJ2",
   "OBJ=#aqui'#|#here#, (*N...*)|(*N...*), mi'|me, usted|you; \
    ADV=arriba|above, debajo|below, cerca|near, dentro|inside"],
```

2. Local adverbs followed by *de* in Spanish and by *from* in English ("lejos de aquí" = "far from here"):

42c ⟨*Locative Expressions* 42b⟩+≡ (25a) ◁42b 43a▷

```
  ["<*LOC*> ADV1 de OBJ1",
   "<*LOC*> ADV2 from OBJ2",
   "OBJ=aqui'|here, (*N...*)|(*N...*), mi'|me, usted|you;\
    ADV=lejos|far"],
```

3. the previous two groups of locative elements can also be used absolutely ("this thing is far" = "esta cosa está lejos"):

42d ⟨*Preposition Plus Substantive Phrase* 42d⟩≡ (25a) 43d▷

```
  ["<*LOC*> ADV1",
   "<*LOC*> ADV2",
   "ADV=lejos|far, debajo|below, arriba|above, debajo|below, \
        cerca|near, dentro|inside"],
```

4. the prime SIDE requires "on" in English:

43a ⟨*Locative Expressions* 42b⟩+≡ (25a) ◁42c 43b▷

```
["<*LOC*> a (*NL..*)", "<*LOC*> on (*NL..*)"],
```

5. other locative expressions:

43b ⟨*Locative Expressions* 42b⟩+≡ (25a) ◁43a

```
["<*LOC*> aqui'", "<*LOC*> #here#"],

["<*LOC*> en otro sitio", "<*LOC*> somewhere else"],
["<*LOC*> en algun sitio", "<*LOC*> somewhere"],
```

4.8 Standalone "it"

43c ⟨*Standalone It* 43c⟩≡ (25a)

```
["<*Npms*> #eso#", "<*Npms*> #it#"],
```

4.9 Substantive Phrases with Prepositions

Prepositions are not primes, but are required by the combinatorics of predicates. The following rules embed the already recognized noun phrases into prepositinal phrases, which will combine with predicates according to the valence of the latter.

The following rules recognizes benefactives (do something good FOR SOMEONE), targets of emotions (feel something good TOWARDS SOMEONE), arguments (say something to someone ABOUT SOMETHING) and comitatives (do something WITH SOMEONE).

43d ⟨*Preposition Plus Substantive Phrase* 42d⟩+≡ (25a) ◁42d 44a▷

```
// more P + NP
  ["<*PREP3*> NEG1 PREP1 NP1",
   "<*PREP3*> NEG2 PREP2 NP2",
   "PREP=con|with|COMIT, de|about|ARG, para|for|BEN, hacia|towards|TARGET; \
    NP=(*Na..*)|(*Na..*),mi'|me,usted|you; \
    NEG=no|not, 0|0"],
```

With inanimate nouns "con" (and its English equivalent "with") mark instruments and not comitatives:

44a ⟨*Preposition Plus Substantive Phrase* 42d⟩+≡ (25a) ◁43d 44b▷

```
["<*PREP3*> NEG1 PREP1 (*N[ilL]..*)",
 "<*PREP3*> NEG2 PREP2 (*N[ilL]..*)",
 "PREP=con|with|INSTR, de|about|ARG, en|in|LOC, hacia|towards|TARGET;\
  NEG=no|not,0|0"],
```

Spanish "como" is set apart because it is followed by nominative pronouns and not by accusatives:

44b ⟨*Preposition Plus Substantive Phrase* 42d⟩+≡ (25a) ◁44a 44c▷

```
["<*MAN*> como NP1",
 "<*MAN*> like NP2",
 "NP=(*N...*)|(*N...*), yo|me, usted|you"],
```

Other rules match "well" as manner adverbial (know someone well) and "because of" + substantive;

44c ⟨*Preposition Plus Substantive Phrase* 42d⟩+≡ (25a) ◁44b 45▷

```
["<*MAN*> bien", "<*MAN*> well"],

["<*CAUS*> NEG1 por causa de (*N...*)",
 "<*CAUS*> NEG2 because of (*N...*)",
 "NEG=no|not,0|0"],

["<*CAUS*> por eso",
 "<*CAUS*> because of this"],
```

finally, there are rules for portmanteau "así" (like this), for standalone "this" and "before/after" + noun phrase:

45 ⟨*Preposition Plus Substantive Phrase* 42d⟩+≡ (25a) ◁44c

```
// like this and substantive this and it
  ["<*MAN*> asi'", "<*MAN*> like this"],

  ["<*Nims*> esto", "<*Nims*> this"],


// before/after + N
  ["<*TSUCC*> (*TSUCC*) de (*N[ip]..*)",
   "<*TSUCC*> (*TSUCC*) (*N[ip]..*)"],

  ["<*TPRED*> (*TPRED*) de (*N[ip]..*)",
   "<*TPRED*> (*TPRED*) (*N[ip]..*)"],
```

4.10 Valence

This section specifies the combinatorial properties of verbs. Each "verbal" prime can appear in a specific set of frames.

DO. The prime DO has the greatest range of possibilities: it combines with a comeplement (DO SOMETHING) and can have an animate patient argument (DO SOMETHING TO SOMEONE), an inanimate instrument (DO SOMETHING WITH SOMETHING), an animate comitative (DO SOMETHING WITH SOMEONE) and a benefactive (DO SOMETHING GOOD FOR SOMEONE).

In Spanish, an objective preverbal clitic may express the complement ("*lo* hago"), while a preverbal dative clitic obligatorily doubles the postverbal nominal patient ("*le* hago algo a esta persona"). If the patient is first or second person ("to me", "to you"), no postverbal prepositional phrase is needed in NSM ("esta persona me hace algo" and not "esta persona me hace algo a mí").

Accordingly, we will write two separate rules, one for third person patients and one for first and second person ones.

So far, all the variables with alternative values we have seen have only a choice between to values. In the next rule, we will use a three-value alternative for variables `PAT` (patient) and `COMPL` (complement).

- The first and third place of Variable `COMPL` refer to the two possible object positions in Spanish, respectively, the postverbal, occupied by a noun phrase, and the preverbal, occupied by the clitic "lo". Only one of the two can be occupied at the same time, so the variable has always a zero as its third option, when the first option is filled. The object can also be a `(*MAN*)` phrase ("this person did *like this*") or the prime MORE ("I want to do *more*").
- The alternants of the variable `PAT` set the right clitic to correspond with the right noun phrase: "le" with singular or collective, and "les" with plural noun phrases.

46 ⟨ *Valence Rules* 46⟩≡ (25a) 47a▷

```
// PREDICATE FRAMES (VALENCE)
// ************************

//DO

 ["<*V*> PAT2 COMPL3 hace COMPL1 a PAT1 Z K W",
  "<*V*> do COMPL2 to PAT3 Z K W",
  "COMPL=(*Ni..*)|(*Ni..*)|0, (*MAN*)|(*MAN*)|0, 0|it|lo, ma's|more|0;\
   PAT= (*Na.[sc]*)|le|(*Na.[sc]*), (*Na.p*)|les|(*Na.p*);\
   Z=(*COMIT*),0; \
   K=(*INSTR*),0; \
   W=(*BEN*),0"],
```

The second rule for DO takes care of the first- or second-person patient, and also of the case in which no patient is present:

47a ⟨ *Valence Rules* 46⟩+≡ (25a) ◁46 47b▷

```
["<*V*> PAT1 COMPL3 hace COMPL1 Z",
 "<*V*> do COMPL2 PAT2 Z",
 "COMPL=(*Ni..*)|(*Ni..*)|0, (*MAN*)|(*MAN*)|0, 0|it|lo, ma's|more|0;\
  PAT=me|to me, le|to you, 0|0;\
  Z=(*COMIT*),0"],
```

SAY. Another prime for which we must account for dative clitic pronouns is SAY. The following rule takes care of the cases in which SAY has nominal complements (SAY SOMETHING TO SOMEONE ABOUT SOMEONE/SOMETHING); a later rule will be written which recognizes the clausal valence (the " SAY WH-" and "SAY $CLAUSE_{that}$" constructions).

47b ⟨ *Valence Rules* 46⟩+≡ (25a) ◁47a 48a▷

```
//SAY

 ["<*V*> PAT2 dice COMPL1 a PAT1 Z K",
  "<*V*> say COMPL2 to PAT3 Z K",
  "COMPL=(*Ni..*)|(*Ni..*), (*MAN*)|(*MAN*), ma's|more;\
   PAT=(*Na.p*)|les|(*Na.p*), (*Na.[sc]*)|le|(*Na.[sc]*); \
   Z=(*ARG*),0; \
   K=(*INSTR*),0"],

 ["<*V*> PAT1 dice COMPL1 Z K",
  "<*V*> say COMPL2 PAT2 Z K",
  "COMPL=(*Ni..*)|(*Ni..*), (*MAN*)|(*MAN*), ma's|more;\
   PAT=me|to me, le|to you, 0|0;\
   Z=(*ARG*),0; \
   K=(*INSTR*),0"],
```

HEAR. The prime HEAR needs to distinguish an animate from an inanimate complement, because in Spanish, the animate complement is marked by the preposition "a" (by the way, I am not sure that " I hear someone" is admissible in NSM). The rule for inanimate object will also take care of objects expressed by clitic pronouns in Spanish, as in "me oye" and "le oigo":

48a ⟨*Valence Rules* 46⟩+≡ (25a) ◁47b 48b▷

```
//HEAR
  ["<*V*> oye a (*Na..*)",
   "<*V*> hear (*Na..*)"],

  ["<*V*> OBJ3 oye OBJ1 Z",
   "<*V*> hear OBJ2 Z",
   "OBJ= (*Ni..*)|(*Ni..*)|0, ma's|more|0, 0|me|me, 0|you|le, 0|0|0;\
    Z=(*ARG*),0"],
```

LIVE. Prime LIVE admits a "habitat" locational argument, and a comitative; it can also combine with the prime MORE[4]:

48b ⟨*Valence Rules* 46⟩+≡ (25a) ◁48a 48c▷

```
//LIVE
  ["<*V*> vive Z1 Y X",
   "<*V*> live Z2 Y X",
   "X=(*LOC*),0; \
    Y=(*COMIT*),0; \
    Z= ma's|more, 0|0"],
```

SEE. Like HEAR, SEE distinguishes an animate from an inanimate object:

48c ⟨*Valence Rules* 46⟩+≡ (25a) ◁48b 49a▷

```
//SEE
  ["<*V*> ve a (*Na..*)",
   "<*V*> see (*Na..*)"],

  ["<*V*> OBJ3 ve OBJ1",
   "<*V*> see OBJ2",
   "OBJ= (*N[liL]..*)|(*N[liL]..*)|0, ma's|more|0, 0|me|me, 0|you|le, 0|0|0"],
```

[4] [Wie01] talks of "living *more* with God.

FEEL. The prime FEEL is often used with a "target" argument (FEEL SOMETHING GOOD TOWARDS SOMEONE), although it is not yet clear whether this option is universal (quotation needed):

49a ⟨ *Valence Rules* 46⟩+≡ (25a) ◁48c 49b▷

```
//FEEL
  ["<*V*>  siente X1 ARG",
   "<*V*> feel X2 ARG",
  "X=(*Ni..*)|(*Ni..*),ma's|more, (*MAN*)|(*MAN*); \
   ARG=(*TARGET*),(*LOC*),0"],
```

HAPPEN. This prime has an optional dative or locative argument:

49b ⟨ *Valence Rules* 46⟩+≡ (25a) ◁49a 49c▷

```
//HAPPEN
  ["<*V*> DAT1 pasa",
   "<*V*> happen to DAT2",
   "DAT=me|me, le|you"],

  ["<*V*> DAT1 pasa a DAT2",
   "<*V*> happen to DAT3",
   "DAT=les|(*N..p*)|(*N..p*), le|(*N..[sc]*)|(*N..[sc]*)"],

  ["<*V*> pasa LC",
   "<*V*> happen LC",
   "LC=(*LOC*),0"],
```

HAVE. HAVE takes a non-animate substantive phrase as complement:

49c ⟨ *Valence Rules* 46⟩+≡ (25a) ◁49b 50a▷

```
//HAVE
  ["<*V*> tiene X1",
   "<*V*> have X2",
   "X=(*N[ilL]..*)|(*N[ilL]..*), ma's|more"],
```

TOUCH.

50a ⟨*Valence Rules* 46⟩+≡ (25a) ◁49c 50b▷

```
//TOUCH
  ["<*V*> toca a (*Na..*)",
   "<*V*> touch (*Na..*)"],

  ["<*V*> OBJ3 toca OBJ1",
   "<*V*> touch OBJ2",
   "OBJ= (*N[liL]..*)|(*N[liL]..*)|0, ma's|more|0, 0|me|me, 0|you|le, 0|0|0"],
```

MOVE, DIE, EXIST. These primes have no other arguments than the subject:

50b ⟨*Valence Rules* 46⟩+≡ (25a) ◁50a 50c▷

```
//MOVE
  ["<*V*> se mueve", "<*V*> move"],

//DIE
  ["<*V*> muere", "<*V*> die"],


//EXIST
  ["<*V*> existe", "<*V*> exist"],
```

LOCATIVE BE and IDENTIFICATIONAL BE. To these primes correspond two different lexemes in Spanish, "ser" (theme es-) and "estar". While "be in a place" is "estar en un sitio", "be like this" can be said in Spanish both "estar así" for temporaty states, and "ser así" for permanent ones. As the NSM BE+LIKE+THIS combination is used mostly in definitions, I select "ser" for the exponent of BE in this case. Accordingly,

50c ⟨*Valence Rules* 46⟩+≡ (25a) ◁50b 50d▷

```
//BE loc
  ["<*Vcop*> esta' ADV",
   "<*Vcop*> be ADV",
   "ADV=(*LOC*), (*COMIT*)"],

  ["<*Vcop*> es ADV",
   "<*Vcop*> be ADV", "ADV=(*MAN*)"],
```

Here we will have to add the be+-ing rule (defined in section 4.11, p. 54).

50d ⟨*Valence Rules* 46⟩+≡ (25a) ◁50c 51a▷
⟨*Progressive Aspect* 54b⟩

GOOD, BAD, BIG, SMALL. The "adjectival" primes, used predicatively, need copular BE both in Spanish and in English; in Spanish, moreover, they agree with the subject. The valence rule will recognize the stem *buen-, mal-pequeñ*, leaving aside the suffix, which will be cared for by a sentence rule. Also, prime BIG is not present here, because the Spanish equivalent "grande" does not show gender agreement, and will be handled by a later rule.

51a ⟨*Valence Rules* 46⟩+≡ (25a) ◁50d 51b▷

```
// BE + Adj (e BE+grande/NP)
  ["<*C*> #es# INT1 ADJ1",
   "<*C*> #be# INT2 ADJ2",
   "INT=muy|very,0|0; ADJ=buen|good, mal|bad, pequen'|small, \
        verdader|true, lejan|far, cercan|near, 0|0"],

//facultative adj: id adj is grande or is an NP, it
//remains outside the verb
```

KNOW, THINK and SAY. These primes (and also WANT in non-equi frames) can take a whole sentence as complement. Before defining the corresponding rules, we must account for the tense-mode shift caused by NOT+THINK, which require the subjunctive ("quiero que haga algo" and "quería que hiciese algo"). This rule is not optimal and needs to be revised.

51b ⟨*Valence Rules* 46⟩+≡ (25a) ◁51a 52a▷

```
// NOT + THINK
  ["no F-AFF TENSE1 MOD ASP piensa que NP S-AFF TENSE2",
   "no F-AFF TENSE1 MOD ASP piensa que NP S-AFF TENSE3",
   "F-AFF=1,3,n,0; \
    S-AFF=1,3,n,0; \
    TENSE=PRET|PSUBJ|PRET PROG va a INF, 0|SUBJ|va a INF;\
    NP=(*N...*),usted,eso; \
    ASP=PROG,0; \
    MOD=quiere INF, puede INF, 0; \
    THINK-WANT=piensa"],
```

Now we can account for the NP-arguments of KNOW and THINK (KNOW SOMETHING, KNOW SOMETHING ABOUT SOMEONE, THINK LIKE THIS, KNOW MORE):

52a ⟨*Valence Rules* 46⟩+≡ (25a) ◁51b 52b▷

```
// KNOW, THINK
  ["<*V*> X3 COGN1 X1 Y1",
  "<*V*> X4 COGN2 X2 Y2",
   "COGN=piensa|think, sabe|know; \
    X=(*Ni..*)|(*Ni..*)|0|0, m'as|more|0|0, (*MAN*)|(*MAN*)|0|0;\
    Y=(*ARG*)|(*ARG*),0|0"],
```

KNOW SOMEONE has a separate exponent for KNOW in Spanish, namely "conocer":

52b ⟨*Valence Rules* 46⟩+≡ (25a) ◁52a 52c▷

```
  ["<*V*> NP2 conoce MANN",
   "<*V*> know NP1 MANN",
   "NP=me|me, you|le; \
    MANN=(*MAN*),0"],

  ["<*V*> conoce MANN a NP1",
   "<*V*> know NP2 MANN",
   "NP=(*Na.s*)|(*Na.s*), (*Na.p*)|(*Na.p*);\
    MANN=(*MAN*),0"],
```

propositional KNOW, THINK and SAY can be accounted for by the following rules:

52c ⟨*Valence Rules* 46⟩+≡ (25a) ◁52b 53a▷

```
// KNOW THAT ~ WHETHER
  ["<*V*> sabe COMPL1",
   "<*V*> know COMPL2", "COMPL=que|that, si|whether"],

// KNOW, SAY + WH
  ["<*V*> Vcon1 COMPL",
   "<*V*> Vcon2 COMPL",
   "Vcon=sabe|know, dice|say"],

// THINK THAT
  ["<*V*> piensa que", "<*V*> think that"],
```

and the rule for "non-equi" WANT (which also needs revision):

53a ⟨*Valence Rules* 46⟩+≡ (25a) ◁52c 53b▷

```
// NON-EQUI WANT
  ["PRET quiere que NP1 NEG1 NP2 PSUBJ",
   "PRET want NP3 NEG2 to",
     "NP=(**N..p**)|p|(**N..p**), (**N..[sc]**)|0|(**N..[sc]**), \
         0|1|me, usted|0|you; \
      NEG=no|not,0|0"],

  ["quiere que NP1 NEG1 NP2 SUBJ",
   "want NP3 NEG2 to",
     "NP=(**N..p**)|p|(**N..p**), \
         (**N..[sc]**)|0|(**N..[sc]**), 0|1|me, usted|0|you; \
      NEG=no|not,0|0"],

  ["<*V*> quiere que","<*V*> quiere que"],
```

after the "non-equi want" rule, we can consider the simpler case where WANT has a substantive argument (WANT SOMETHING) and the WANT MORE construction;

53b ⟨*Valence Rules* 46⟩+≡ (25a) ◁53a

```
["<*V*> quiere NP1", "<*V*> want NP2",
   "NP=(*Ni..*)|(*Ni..*), ma's|more"],
```

4.11 Verb Phrase

After valence rules have applied, the bare stem of the verb has been inserted into a "verb phrase" with its following complements and arguments. These verb phrases are labeled `<*Vcop*>` (BE), `<*C*>`, (predicative adjective phrase with copula BE) and `<*V*>` (all other verbs).

Before this "bare verb phrase" come the auxiliaries and the abstract strings such as `PRET`, `p`, which signal the inflectional pattern.

These rules reflect the idiosyncrasies of both the English and the Spanish verb system. Three classes of verb phrases need to be distinguished because of their English agreement pattern:

1. The verb "be":
   - Has a separate form for first-person singular present (*I am*);
   - distinguishes between singular and plural in past tense (*was* vs. *were*);
2. All other verbs have a separ te third person singular form in present, except:
3. Modals *can* and *will*, as well as all other verbs in past tense, which do not inflect for person.

The following rules incorporate the auxiliaries and the past tense inflection into the verb phrase, and label the three classes differently, so that they can be taken apart by agreement rules:

54a ⟨*Verb Phrase* 54a⟩≡ (25a)
```
⟨Progressive Aspect 54b⟩
⟨Modals 55a⟩
⟨Past Tense 56⟩
```

Progressive. The Spanish be+gerund is translated with the be+ing English form. Note how the verb phrase is relabeled `<*Vcop*>`, because the head of the English verb phrase is now "be" and will inflect accordingly.

54b ⟨*Progressive Aspect* 54b⟩≡ (50d 54a)
```
//BE ING
  ["<*Vcop*> esta' GER (*V*)",
   "<*Vcop*> be ING (*V*)"],
```

"Equi" want. English modal verb "want" is conjugated like other verbs (wants, wanted), so "want to" + verb phrase is labeled as `<*V*>`. Note how this changes the inflectional class of a `(*Vcop*)`: if the "be" is preceded by "want", there will be no more first-person separate forms (like *I am*), and no more distinction in past tense between singular and plural (like in *was* vs. *were*).

Accordingly, the `<*C*>` phrase is relabeled as `<*Cv*>`. Sentence rules will inflect `<*Cv*>` like `<*V*>` (`<*C*>` cannot be relabeled simply as `<*V*>` because we need to match a `<*Cv*>`-phrases manifest subject-predicate agreement in gender and number in Spanish, so they must be distinguished from other verb phrases).

Here is the rule for modls:

55a ⟨*Modals* 55a⟩≡ (54a) 55b▷

```
// MODAL
// *****

["<*V*> MOD1 VP", "<*V*> MOD2 VP",
 "MOD=quiere INF|want to; VP=(*V*), (*Vcop*)"],

["<*Cv*> MOD1 VP", "<*Cv*> MOD2 VP",
 "MOD=quiere INF|want to; VP=(*C*)"],
```

Can, will. English modals "can" and "will" do not take third-person -s. So, verb phrases headed by such modal will be set apart by the following rule into yet another inflectional class:

55b ⟨*Modals* 55a⟩+≡ (54a) ◁55a

```
["<*Vi*> MOD1 VP",
 "<*Vi*> MOD2 VP",
 "MOD=puede INF|can, PRET PROG va a INF|would, va a INF|will; \
      VP=(*V*), (*Vcop*)"],

["<*Ci*> MOD1 VP",
 "<*Ci*> MOD2 VP",
 "MOD=puede INF|can, PRET PROG va a INF|would, va a INF|will;\
  VP=(*C*),(*Cv*)"],
```

Past tense. Like modals, past tense verbs do not inflect, except "be", which distinguishes singular (was) from pluras (were). So, `<*V*>` phrases are relabeled as `<*Vi*>`, like those with a modal, but `<*C*>` and `<*Vcop*>` phrases must remain in the particular "copular" inflectional class.

56 ⟨*Past Tense* 56⟩≡ (54a)

```
["<*Vi*> PRET VP", "<*Vi*> PRET VP",
 "VP=(*V*), (*Vi*)"],

["<*Vcop*> PRET (*Vcop*)",
 "<*Vcop*> PRET (*Vcop*)"],

["<*C*> PRET (*C*)",
 "<*C*> PRET (*C*)"] ,
```

4.12 Sentences

At sentence level, let us distinguish, first of all, the wh-identificational sentencs, used as complements of KNOW and SAY (I want to know *who this person is*). We distinguish three cases: collective (people), plural and singular subject, becasue of concordance rules:

57a ⟨*Sentence Rules* 57a⟩≡ (25a) 57b▷

```
// SENTENCE
// ********

// subject PEOPLE

["<*S*> NP1 COP2 COP1 NP3",
 "<*S*> NP2 NP3 COP3 COP1",
 "NP=quie'n|who|(*Nafc*); \
  COP=(*C*)|0|p, (*Cv*)|0|0, (*Ci*)|0|0"],


// plural subject
["<*S*> NP1 COP2 COP1 NP3", "<*S*> NP2 NP3 COP3 COP1",
 "NP=quie'nes|who|(*Na.p*), do'nde|where|(*Nl.p*), \
     cua'ndo|when|(*Nt.p*), que'|what|(*Ni.p*); \
  COP=(*C*)|p|p, (*Ci*)|p|0, (*Cv*)|p|0"],


// singular subject

["<*S*> NP1 COP2 COP1 NP3", "<*S*> NP2 NP3 COP3 COP1",
 "NP=quie'n|who|(*Na.s*), do'nde|where|(*Nl.s*), \
     cua'ndo|when|(*Nt.s*), que'|what|(*Ni.s*); \
  COP=(*C*)|0|3, (*Cv*)|0|3, (*Ci*)|0|0"],
```

After that, we can consider the general case.

first-person subject

57b ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁57a 58a▷

```
["<*S*> NEG1 1 VP",
 "<*S*> I NEG2 VP NEG3",
 "NEG=ya no|do not|anymore, no|do not|0, 0|0|0; \
  VP=(*V*),(*Vcop*),(*Vi*),(*Vt*)"],
```

singular subject. We can see in the following rule the "in-my-body"-hack at work. This is surely something which needs urgently to be replaced with something linguistically more meaningful:

58a ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁57b 58b▷

```
["<*S*> NP NEG1 VP1 BODYLOC1",
 "<*S*> NP NEG2 VP2 VP1 BODYLOC2 NEG3",
 "NP=(*N..s*); \
  NEG=ya no|do not|anymore, no|do not|0, 0|0|0;\
  VP=(*V*)|3,(*Vcop*)|3,(*Vi*)|0,(*Vt*)|0; \
  BODYLOC=LOC-N-2-BODY|LOC-N-3-BODY, \
          LOC-TP-2-BODY|LOC-TP-3-BODY, \
          LOC-SP-2-BODY|LOC-SP-3-BODY, \
          LOC-OP-2-BODY|LOC-OP-3-BODY, 0|0"],
```

PEOPLE as subject. The prime PEOPLE agrees as singular in Spanish and as plural in English, so it needs a separate rule:

58b ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁58a 58c▷

```
["<*S*> NP NEG1 VP1 BODYLOC1",
 "<*S*> NP NEG2 VP2 VP1 BODYLOC2 NEG3",
 "NP=(*N..c*); \
  NEG=ya no|do not|anymore, no|do not|0, 0|0|0; \
  VP=(*V*)|0,(*Vcop*)|p,(*Vi*)|0,(*Vt*)|0; \
  BODYLOC=LOC-N-2-BODY|LOC-N-3-BODY,
          LOC-TP-2-BODY|LOC-TP-3-BODY, \
          LOC-SP-2-BODY|LOC-SP-3-BODY, \
          LOC-OP-2-BODY|LOC-OP-3-BODY, 0|0"],
```

plural subject

58c ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁58b 59a▷

```
["<*S*> NP1 NEG1 NP2 VP1", "<*S*> NP3 NEG2 VP2 VP1 NEG3",
 "NP=(*N..p*)|p|(*N..p*), usted|0|you, 0|0|you; \
  NEG=ya no|do not|anymore, no|do not|0, 0|0|0; \
  VP=(*V*)|0,(*Vi*)|0,(*Vt*)|0,(*Vcop*)|p"],
```

4.12.1 Adjective and Noun predicates

Now for the copulas:

First person subject has person concordance in both English and Spanish:

59a ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁58c 59b▷

```
["<*S*> NEG1 1 COP NP1",
 "<*S*> I NEG2 COP NP2 NEG3",
 "NP=(*N..s*)|(*N..s*), muy grande|very big, grande|big, -o|0, -a|0; \
  COP = (*C*),(*Ci*),(*Cv*); \
  NEG=ya no|do not|anymore, no|do not|0, 0|0|0"],
```

All other perrsons are put together; separate rules are needed for the various inflectional classes. The first rule distinguishes between singular and plural form of the copula in English, so it willbe applied to groups of type `<*C*>`. A five-value alternant variable is used here: NP1 is the class of the subject (masculine or feminine, singular or plural or collective), NP2 in Spanish, together with NP5 in English, is the predicate after the copula, which can be another noun phrase concording in number with the subject, or the prime BIG[5], or the gender-number suffix of the adjective (its root is already incorporated into the phrase).

NP3 and NP4 are respectively the Spanish and English person-number suffixes which were stripped from the verb already by the morphological component.

59b ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁59a 60a▷

```
["<*S*> NP1 NEG1 NP4 (*C*) NP2",
 "<*S*> NP1 NEG2 NP3 (*C*) NP5 NEG3",
 "NP=(*N..p*)|(*N..p*)|p|p|(*N..p*), (*N..s*)|(*N..s*)|3|0|(*N..s*), \
     (*N..c*)|(*N..c*)|p|0|(*N..c*), \
     (*N..p*)|muy grandes|p|p|very big, \
             (*N..s*)|muy grande|3|0|very big, \
     (*N..c*)|muy grande|p|0|very big,\
     (*N..p*)|grandes|p|p|big, (*N..s*)|grande|3|0|big, \
     (*N..c*)|grande|p|0|big,\
     (*N.mp*)|-os|p|p|0, (*N.fp*)|-as|p|p|0, \
             (*N.ms*)|-o|3|0|0,\
     (*N.fs*)|-a|3|0|0, (*N.fc*)|-a|p|0|0;\
  NEG=ya no|do not|anymore, no|do not|0, 0|0|0"],
```

[5] Recall that it was not considered in the rule of the adjectival predication, because it agrees with its subject only in number.

Other rules are similar to the last one, only the inflectional class of the copula changes. The groups `<*Ci*>` represent the copular phrases with a modal – thus such phrases do not inflect for person in English, but do in Spanish (note the value of NP4, which alternates between `p` and `0` for Spanish, while NP5 (English) always remains `0`.

60a ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁59b 60b▷

```
["<*S*> NP1 NEG1 NP4 (*Ci*) NP2",
 "<*S*> NP1 NEG2 NP3 (*Ci*) NP5 NEG3",
 "NP=(*N..p*)|(*N..p*)|0|p|(*N..p*), (*N..s*)|(*N..s*)|0|0|(*N..s*), \
     (*N..c*)|(*N..c*)|0|0|(*N..c*),\
     (*N..p*)|muy grandes|0|p|very big, (*N..s*)|muy grande|0|0|very big, \
     (*N..c*)|muy grande|0|0|very big,\
     (*N..p*)|grandes|0|p|big, (*N..s*)|grande|0|0|big, (*N..c*)|grande|0|0|big,\
     (*N.mp*)|-os|0|p|0, (*N.fp*)|-as|0|p|0, (*N.ms*)|-o|0|0|0, \
     (*N.fs*)|-a|0|0|0, (*N.fc*)|-a|0|0|0;\
  NEG=ya no|do not|anymore, no|do not|0, 0|0|0"],
```

`<*Cv*>` are copular phrases with a modal like “want”. These phrases inflect, in English, like normal verbs:

60b ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁60a 61a▷

```
["<*S*> NP1 NEG1 NP4 (*Cv*) NP2",
 "<*S*> NP1 NEG2 NP3 (*Cv*) NP5 NEG3",
 "NP=(*N..p*)|(*N..p*)|0|p|(*N..p*), (*N..s*)|(*N..s*)|3|0|(*N..s*), \
     (*N..c*)|(*N..c*)|0|0|(*N..c*),\
     (*N..p*)|muy grandes|0|p|very big, (*N..s*)|muy grande|3|0|very big, \
     (*N..c*)|muy grande|0|0|very big,\
     (*N..p*)|grandes|0|p|big, (*N..s*)|grande|3|0|big, \
     (*N..c*)|grande|0|0|big,\
     (*N.mp*)|-os|0|p|0, (*N.fp*)|-as|0|p|0, (*N.ms*)|-o|3|0|0, \
     (*N.fs*)|-a|3|0|0, (*N.fc*)|-a|0|0|0;\
  NEG=ya no|do not|anymore, no|do not|0, 0|0|0"],
```

4.13 The Existential predicate

The last two rules of this group recognize the patterns of the existential predicates. Two rules are needed:

One for non present tenses, because the Spanish form inflects for number (*había/habían, habrá/habrán*):

61a ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁60b 61b▷

```
// THERE IS

["<*S*> NEG1 NP3 TMA1 MOD1 hay TEMP1 NP1 LC",
 "<*S*> NEG2 NP2 TMA2 MOD2 EXIST TEMP2 NP1 LC NEG3",
 "NEG=ya no|NO|anymore, no|NO|0, 0|0|0;  \
  TMA=PRET PROG va a INF|would, va a INF|will, PRET|PRET; \
  MOD=puede INF|can,0|0; \
  TEMP=nunca|ever, siempre|always, 0|0;\
  NP=(*N..s*)|0|0, (*N..c*)|p|0, (*N..p*)|p|p;\
  LC=(*LOC*),0"],
```

and a simpler one for present tense (which must allow for the English alternance *there is/there are*):

61b ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁61a 62▷

```
["<*S*> NEG1 hay TEMP1 NP1 LC",
 "<*S*> NEG2 NP2 EXIST TEMP2 NP1 LC NEG3",
  "NP=(*N..s*)|0, (*N..[cp]*)|p; LC=(*LOC*),0; \
   NEG=ya no|NO|anymore, no|NO|0, 0|0|0; \
   TEMP=nunca|ever, siempre|always, 0|0"],
```

4.14 Conjunctions

In a more mature grammar, which will define a textual level, the following rules will have to be rewritten. For the time being, a sentence grammar can allow itself a simple match-and-replace of the conjunctions in the two languages, because the syntactic patterns are the same:

62 ⟨*Sentence Rules* 57a⟩+≡ (25a) ◁61b

```
["LOC1 donde", "LOC2 where",
 "LOC=en el sitio|in the place, 0|0"],

["LOC1 cuando", "LOC2 when",
 "LOC=al tiempo|at the time, 0|0"],


["#si#", "#if#"],

["porque", "because"]
```

References

[GW02] Cliff Goddard and Anna Wierzbicka. *Meaning and universal grammar : theory and empirical findings*, volume 1. Benjamins, Amsterdam, 2002.

[Tra02] Catherine Travis. La metalengua semntica natural: The natural semantic metalanguage of spanish. In *Meaning and Universal Grammar. Theory and Empirical Findings. Volume I*, pages 173–242. John Benjamins, Amsterdam/Philadelphia, 2002.

[Wie01] Anna Wierzbicka. *What did Jesus mean?* Oxford University Press, 2001.

References

[GW02] Cliff Goddard and Anna Wierzbicka. *Meaning and universal grammar : theory and empirical findings*, volume 1. Benjamins, Amsterdam, 2002.

[Tra02] Catherine Travis. La metalengua semntica natural: The natural semantic metalanguage of spanish. In *Meaning and Universal Grammar. Theory and Empirical Findings. Volume I*, pages 173–242. John Benjamins, Amsterdam/Philadelphia, 2002.

[Wie01] Anna Wierzbicka. *What did Jesus mean?* Oxford University Press, 2001.

Defined Chunks

⟨*Body Kind Part* 39a⟩ 28, 39a, 39b, 40
⟨*English morphological rules* 4a⟩ 3a, 4a, 4b, 5, 6, 7, 8, 9a, 9b, 10a, 10b, 11, 12a, 12b, 13a, 13b, 14a, 14b, 14c, 14d, 15a, 15b
⟨*English-morphology* 3a⟩ 3a
⟨*Locative Expressions* 42b⟩ 25a, 42b, 42c, 43a, 43b
⟨*Maybe Rule* 25b⟩ 25a, 25b
⟨*Modals* 55a⟩ 54a, 55a, 55b
⟨*N Of This Kind* 30a⟩ 28, 30a
⟨*Noone and Nothing* 31⟩ 28, 31
⟨*Noun Phrases* 32⟩ 28, 32, 33a, 33b, 34a, 34b, 35a, 35b, 36a, 36b, 37a, 37b, 38a, 38b
⟨*Past Tense* 56⟩ 54a, 56
⟨*Preposition Plus Substantive Phrase* 42d⟩ 25a, 42d, 43d, 44a, 44b, 44c, 45
⟨*Progressive Aspect* 54b⟩ 50d, 54a, 54b
⟨*Selector* 41a⟩ 28, 41a
⟨*Sentence Rules* 57a⟩ 25a, 57a, 57b, 58a, 58b, 58c, 59a, 59b, 60a, 60b, 61a, 61b, 62
⟨*Someone and Something* 29a⟩ 28, 29a, 29b, 30b
⟨*spaCO-eng.js* 3b⟩ 3b
⟨*Spanish English grammar* 25a⟩ 3b, 25a
⟨*Spanish morphological rules* 16a⟩ 15c, 16a, 16b, 16c, 17, 18, 19, 20a, 20b, 20c, 21a, 21b, 21c, 22a, 22b, 22c, 23a, 23b, 23c, 23d, 24a, 24b, 24c
⟨*Spanish morphology* 15c⟩ 3b, 15c
⟨*Standalone It* 43c⟩ 25a, 43c
⟨*Substantive Phrases* 28⟩ 25a, 28
⟨*The Same* 41b⟩ 28, 41b, 42a
⟨*Time Modifiers* 26a⟩ 25a, 26a, 26b, 27
⟨*Valence Rules* 46⟩ 25a, 46, 47a, 47b, 48a, 48b, 48c, 49a, 49b, 49c, 50a, 50b, 50c, 50d, 51a, 51b, 52a, 52b, 52c, 53a, 53b
⟨*Verb Phrase* 54a⟩ 25a, 54a