

Gramáticas Formales para el Lenguaje Natural

Evaluación individual – Julio 2014

HPSG

- a. Defina el concepto de jerarquía de tipos.
- b. Que propiedades deben cumplir las restricciones de los tipos en la jerarquía de tipos para que la herencia de restricciones sea adecuada?
- c. En que archivo de lkb puso las restricciones de género número y persona?
- d. Escriba la regla para los modificadores.
- e. Escriba la entrada léxica para la palabra "y" y la regla de conjunción. Tenga en cuenta los rasgos de valencia. Simplifique el rasgo AGR considerando únicamente los del primer argumento.
- f. Escriba las entradas léxicas para las palabras de la oración:
Pedro bebe lentamente un café.

Nota: Se deben considerar los rasgos HEAD, AGR, SPR, COMPS y MOD.

Solucion

- a.
Una jerarquía de tipos es un orden parcial $\langle \mathbf{Tipos}, \sqsubseteq \rangle$ que satisface:
 - **Tipos** tiene un máximo T
 - Para todo $S \subseteq \mathbf{Tipos}$, si S tiene cotas inferiores entonces existe una única mayor cota inferior.
- b.
 - El tipo de una TFS que expresa las restricciones de un tipo t debe ser el mismo tipo t
 - Las restricciones sobre un tipo deben ser subsumidas por las de sus ancestros
 - Todo rasgo debe ser introducido en un único punto de la jerarquía
 - Las restricciones expandidas (locales + heredadas) deben estar bien formadas
- c.
Puede estar en el léxico en cada entrada, a través la jerarquía o en las reglas.

d.

```
head-modifier-rule-1 := phrase &
[ HEAD #head,
  VAL [ SPR <#spr>,
        COMPS #comps,
        MOD #mod ],
  ARGS <
    #h &
    [ HEAD #head,
      VAL [ SPR <#spr>,
            COMPS #comps,
            MOD #mod ] ],
    phrase &
```

[VAL[SPR <>, COMPS <>, MOD #h] >].

e.

y-word := word &
[HEAD conj,
VAL [SPR <>,
COMPS <phrase, phrase>,
MOD <>]]

conj-rule := phrase &
[HEAD #head,
VAL #val,

ARGS <
#arg1 &
[HEAD #head,
VAL #val],

word &
[HEAD conj,
VAL.COMPS <#arg1, #arg2>],

#arg2 &
[HEAD #head,
VAL #val]
>]

f.

Pedro bebe lentamente un cafe.

pedro-word := word &
[HEAD noun & [AGR [NUM sg, GEN masc, PER ter]],
VAL [SPR <>, COMPS <>, MOD <>] ,
ORTH "Pedro"].

bebe-word := word &
[HEAD verb & [AGR [NUM sg, PER ter]],
VAL [SPR <phrase & [HEAD noun]>, COMPS < phrase & [HEAD noun]>, MOD <>] ,
ORTH "bebe"].

lentamente-word := word &
[HEAD adv & ,
VAL [SPR <>, COMPS <>, MOD <phrase & [HEAD verb]>] ,
ORTH "lentamente"].

un-word := word &
[HEAD det & [AGR [NUM sg, GEN masc]],
VAL [SPR <>, COMPS <>, MOD <>],
ORTH “un”].

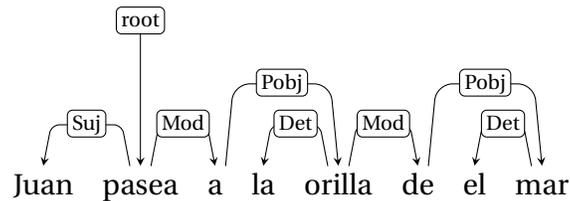
cafe-word := word &
[HEAD noun & [AGR [NUM sg, GEN masc, PER ter]],
VAL [SPR <phrase & [HEAD det]>, COMPS <>, MOD <>],
ORTH “cafe”].

2. Análisis en dependencias (20 puntos)

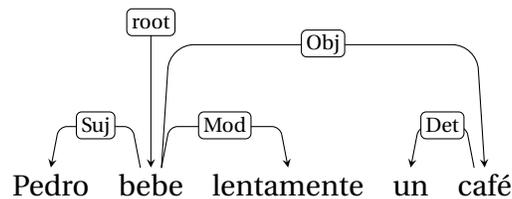
a. Dé un árbol de dependencias de las siguientes oraciones, utilizando el siguiente conjunto \mathcal{R} de relaciones de dependencia: $\mathcal{R} = \{Suj, Obj, Det, Pobj, Prep, Mod, Rcm\}$. No considere el punto final.

- a1. Juan pasea a la orilla del mar.
- a2. Pedro bebe lentamente un café.
- a3. Pedro perdió el libro que compró la semana pasada.

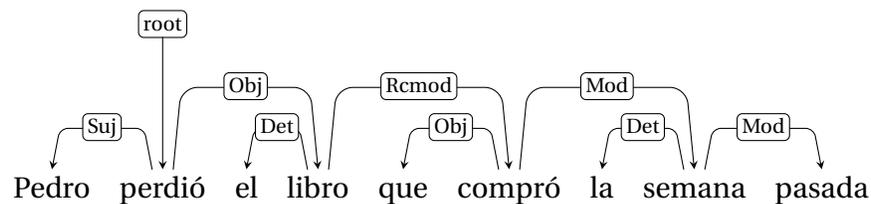
a1.



a2.



a3.



b. Para la oración a2 de la parte a realice el parsing en dependencias (indique en cada paso la transición y contenidos anterior y posterior del stack y de la cola, según el algoritmo de Nivre de parsing basado en transiciones).

1. SHIFT, $(root, Juan|(bebe, lentamente, un, café)) \rightarrow (root|Juan, bebe|(lentamente, un, café))$

2. STACK-HIJO(Suj):
 $(root|Juan, bebe|(lentamente, un, café)) \mapsto (root, bebe|(lentamente, un, café))$
3. STACK-PADRE(root):
 $(root, bebe|(lentamente, uncafé)) \mapsto (root|bebe, (lentamente|un, café))$
4. STACK-PADRE(Mod):
 $(root|bebe, lentamente|(un, café)) \mapsto ((root, bebe|lentamente), un|café)$
5. REDUCE:
 $((root, bebe|lentamente), un|café) \mapsto (root|bebe|, un|café)$
6. SHIFT:
 $(root|bebe, un|café) \mapsto ((root, bebe)|un, café)$
7. STACK-HIJO(Suj):
 $((root, bebe)|un, café) \mapsto (root|bebe, café)$
8. STACK-PADRE(Obj):
 $(root|bebe, café) \mapsto ((root, bebe)|café, \epsilon)$

c. En la oración a3, "libro" es argumento de 2 verbos. ¿Es posible indicar ambas relaciones en el árbol de dependencias? Discuta brevemente.

"libro" es complemento de objeto de "compró" y de "perdió". Ambas relaciones no se pueden indicar en el árbol de dependencias, porque todo nodo puede tener a lo sumo un padre.

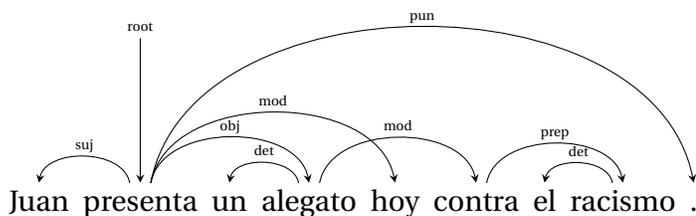
d. Defina la propiedad de proyectividad para un árbol de dependencias. Dé un ejemplo de árbol no proyectivo (una oración en español cuyo análisis genera naturalmente un árbol no proyectivo).

Definición vista en clase:

Un grafo es proyectivo respecto a un orden lineal de sus nodos si, \forall arco $a \rightarrow b$ y nodo w , w ocurre entre a y b en el orden lineal si a domina (clausura transitiva de la relación arco) w .

Ejemplo visto en clase:

Juan presenta un alegato hoy contra el racismo.



la palabra "hoy", modificador de "presenta" se ubicó entre "alegato" y un modificador de la misma. El español permite distintas posiciones en la oración para algunos adverbios.

3. Gramáticas Categoriales Combinatorias (20 puntos)

a. Defina categorías sintácticas y términos semánticos para las palabras de las siguientes oraciones, utilizando como categorías básicas S y \mathcal{NP} . Coindice adecuadamente las componentes de las categorías siempre que sea necesario.

- a1. Pedro bebe lentamente café.
- a2. Pedro logró beber café.
- a3. Pedro prepara y María sirve comida.

| palabra | tipo sintáctico | tipo semántico |
|------------|---|--|
| Pedro | NP | Pedro' |
| María | NP | María' |
| café | NP | café' |
| comida | NP | comida' |
| bebe | $(S \setminus NP)/NP$ | $\lambda x.\lambda y.beber'(y,x)$ |
| prepara | $(S \setminus NP)/NP$ | $\lambda x.\lambda y.preparar'(y,x)$ |
| sirve | $(S \setminus NP)/NP$ | $\lambda x.\lambda y.servir'(y,x)$ |
| beber | $(S \setminus NP)/NP$ | $\lambda x.\lambda y.beber'(y,x)$ |
| lentamente | $(S \setminus NP)/NP \setminus (S \setminus NP)/NP$ | $\lambda P.\lambda u.\lambda v.lent'(P(u))(v)$ |
| logró | $(S \setminus NP_i)/(S \setminus NP_i)$ | $\lambda P.\lambda u.lograr'(P(u))$ |
| y | $(X \setminus X)/X$ | $\lambda P.\lambda Q.\lambda u.P(u) \wedge Q(u)$ |

b. Realice la derivación sintáctica de la oración a3.

Las derivaciones son bottom-up, comenzando por categorías preléxicas

$$\begin{array}{c}
 \frac{\frac{\frac{\text{Pedro}_L}{NP}}{S/(S \setminus NP)} \text{>T} \frac{\frac{\text{prepara}_L}{(S \setminus NP)/NP}}{S/(S \setminus NP)} \text{>C}}{S/NP} \text{>C} \frac{\frac{\frac{\frac{\frac{\text{Maria}_L}{NP}}{S/(S \setminus NP)} \text{>T} \frac{\frac{\text{sirve}_L}{(S \setminus NP)/NP}}{S/(S \setminus NP)} \text{>C}}{S/NP}}{((S/NP) \setminus (S/NP))/ (S/NP)} \text{>L}}{(S/NP) \setminus (S/NP)} \text{>A} \frac{\frac{\text{comida}_L}{NP}}{S/NP} \text{<A}}{S} \text{>A}
 \end{array}$$

c. Usando las categorías básicas S , \mathcal{NP} y \mathcal{N} dé un tipo sintáctico para cada una de las

ocurrencias de “que” en el siguiente texto, con eventual coindización de argumentos:

Sabiendo que el ómnibus tardaría en llegar, Juan empezó a leer el libro que Pedro le recomendó. El vecino que vive arriba también se lo había recomendado.

Fundamente en cada caso, indique tipo para las categorías contiguas si lo considera necesario.

1. que — S/S (suponemos que "sabiendo" toma S como complemento)
2. que — $(N_i \setminus N_i) / (S / NP_i)$ pronombre relativo objeto
3. que — $(N_i \setminus N_i) / (S \setminus NP_i)$ pronombre relativo sujeto