

Gramáticas lógicas - Sintaxis

- Regla de la gramática (producción):

$izq \rightarrow der1, der2, \dots, derN.$

- Cada uno de los símbolos de la gramática puede tener variables para diferentes usos.
- Se consumen símbolos de la entrada utilizando sintaxis de listas:

$aaaa \rightarrow [W], bbbb.$

- Podemos insertar código prolog en las producciones poniéndolo entre llaves:

$aaaa(W) \rightarrow [W], bbbb(W), \{number(W)\}.$

- Para reconocer una frase se utiliza el símbolo principal de la gramática y se le llama como un predicado prolog cualquiera, por ejemplo:

$analisis(F, [el, gato, come, pescado], []).$

- Los únicos parámetros obligatorios son los dos últimos, habitualmente se añade un primer parámetro que se vincula a la representación de la frase.

Gramáticas lógicas - Gramática 1

- Reconocimiento de frases sencillas (SN+V+CMPL)
- Aplicamos análisis descendente para generar la gramática
- Determinamos la estructura de cada uno de los elementos y escribimos producciones para cada uno de ellos (reglas y terminales).
- Añadimos un lexicón sencillo que nos permite reconocer las frases ejemplo
- Esta gramática solo reconoce, no genera ninguna representación
- Variantes: subordinadas de relativo y concordancia de género y número

Gramática 1

```
analisis(X,Y):- asercion(X,Y).
```

```
asercion --> sn, verb,compl.
```

```
compl --> [].
```

```
compl --> prep,sn.
```

```
compl--> sn.
```

```
sn--> npr.
```

```
sn--> det,n.
```

```
verb--> [W],{verbo(W)}.
```

```
npr--> [W],{npropio(W)}.
```

```
n--> [W],{nombre(W)}.
```

```
det--> [W],{determ(W)}.
```

```
prep--> [W],{prepo(W)}.
```

```
npropio(clara).
```

```
npropio(maria).
```

```
npropio(juan).
```

```
npropio(barcelona).
```

```
nombre(hombre).
```

```
nombre(profesor).
```

```
nombre(libro).
```

```
determ(un).
```

```
determ(el).
```

```
verbo(esta).
```

```
verbo(rie).
```

```
verbo(piensa).
```

```
verbo(habla).
```

```
verbo(lee).
```

```
prepo(en).
```

```
prepo(con).
```

```
prepo(de).
```

Gramática 1 - Relativo

asercion --> sn, rel, verb, compl.

rel--> prel, verb, compl.

compl --> [].

compl --> prep, sn.

compl--> sn.

sn--> npr.

sn--> det, n.

verb--> [W], {verbo(W)}.

npr--> [W], {npropio(W)}.

n--> [W], {nombre(W)}.

det--> [W], {determ(W)}.

prep--> [W], {prepo(W)}.

prel--> [W], {pronomrel(W)}.

npropio(clara).

npropio(maria).

npropio(juan).

npropio(barcelona).

nombre(hombre).

nombre(libro).

determ(un).

determ(el).

prepo(en).

prepo(con).

prepo(de).

pronomrel(que).

verbo(esta).

verbo(rie).

verbo(piensa).

verbo(habla).

Gramáticas lógicas - Gramática 2

- Restringimos la gramática para que acepte frases sólo con los complementos adecuados
- Aumentamos la información del lexicón incluyendo en los verbos los complementos que aceptan
- Modificamos la gramática para que las producciones comprueben la validez de los complementos de la frase

Gramática 2

```
analisis(X,Y):- asercion(X,Y).
```

```
asercion --> sn, verb(X),compl(X).
```

```
compl([])--> [].
```

```
compl([])--> sn.
```

```
compl([arg(X)|Y])--> prep(X),sn,compl(Y).
```

```
sn-->npr.
```

```
sn-->det,n.
```

```
verb(A)--> [W],{verbo(W,A)}.
```

```
npr--> [W],{npropio(W)}.
```

```
n--> [W],{nombre(W)}.
```

```
det--> [W],{determ(W)}.
```

```
prep(W)--> [W],{prepo(W)}.
```

```
npropio(clara).
```

```
npropio(maria).
```

```
npropio(juan).
```

```
npropio(barcelona).
```

```
nombre(hombre).
```

```
nombre(profesor).
```

```
nombre(libro).
```

```
determ(un).
```

```
determ(el).
```

```
prepo(en).
```

```
prepo(con).
```

```
prepo(de).
```

```
verbo(piensa,[arg(en)]).
```

```
verbo(esta,[arg(en)]).
```

```
verbo(rie,[]).
```

```
verbo(habla,[arg(de),arg(con)]).
```

```
verbo(lee,[]).
```

Gramática 2 - Ejemplo

asercion

clara

habla

de

un

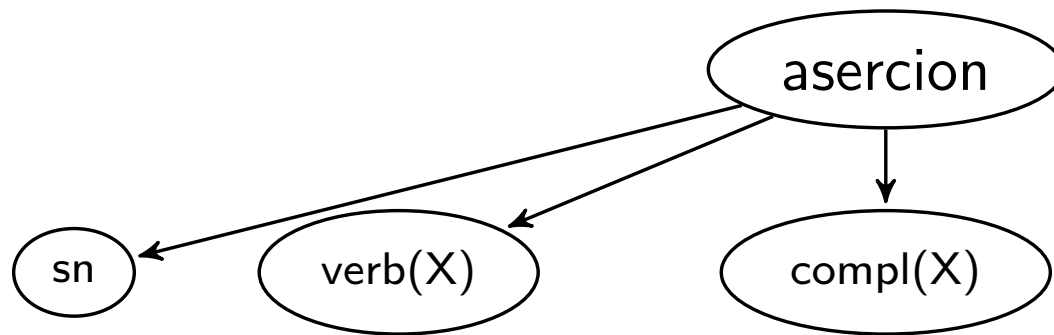
profesor

con

juan

□

Gramática 2 - Ejemplo



clara

habla

de

un

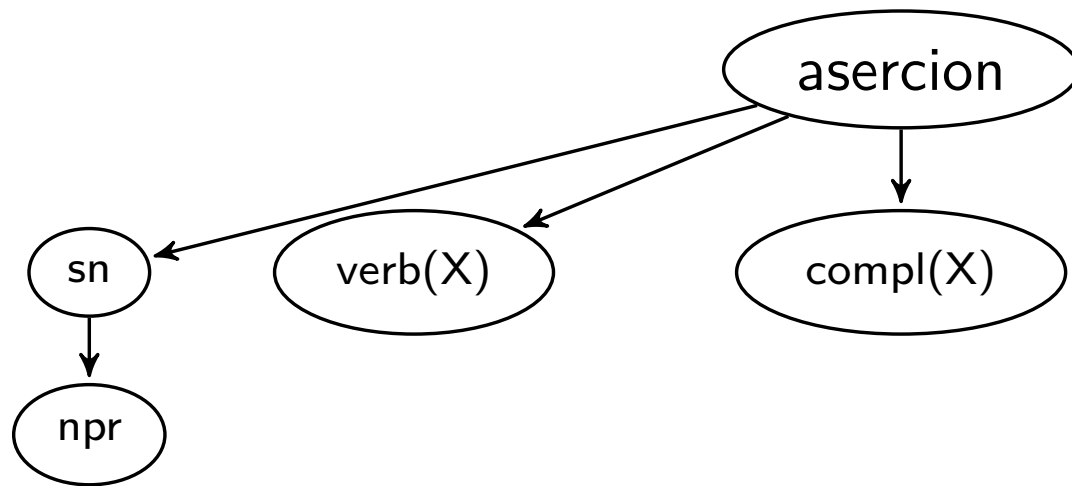
profesor

con

juan

□

Gramática 2 - Ejemplo



clara

habla

de

un

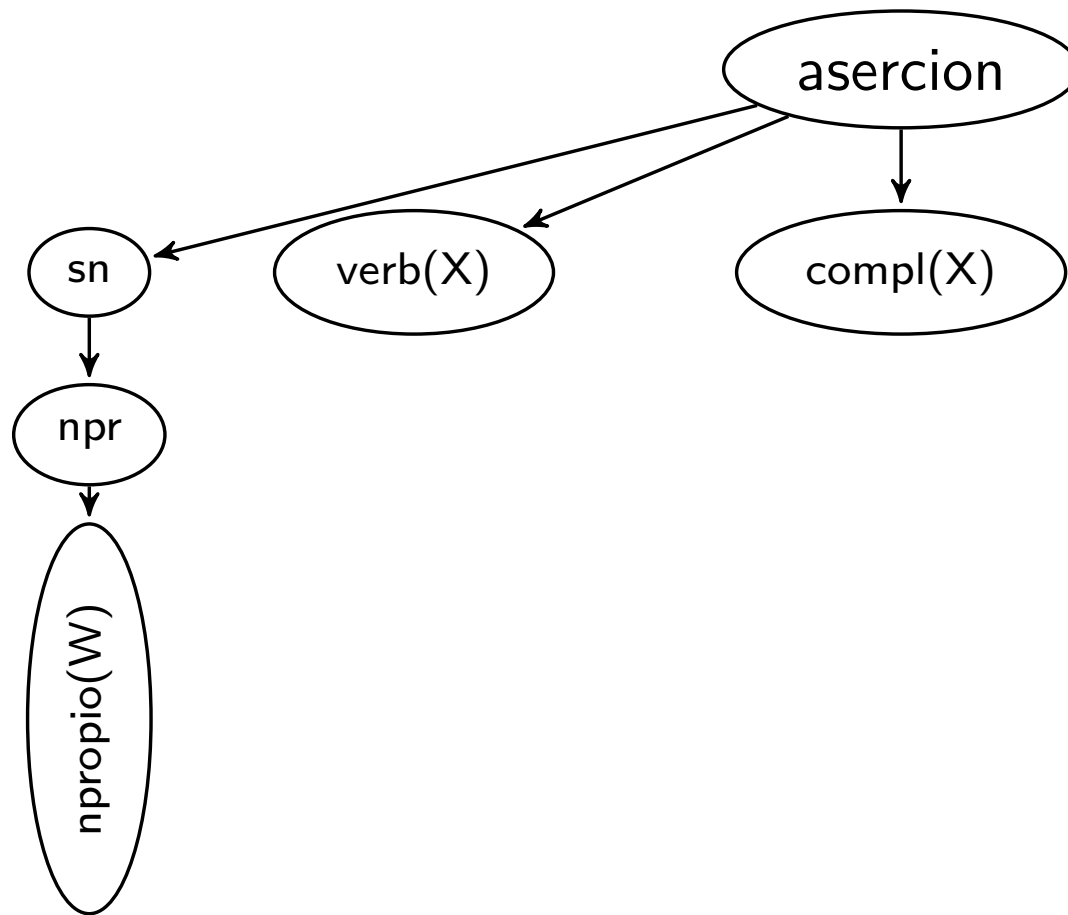
profesor

con

juan

[]

Gramática 2 - Ejemplo



clara

habla

de

un

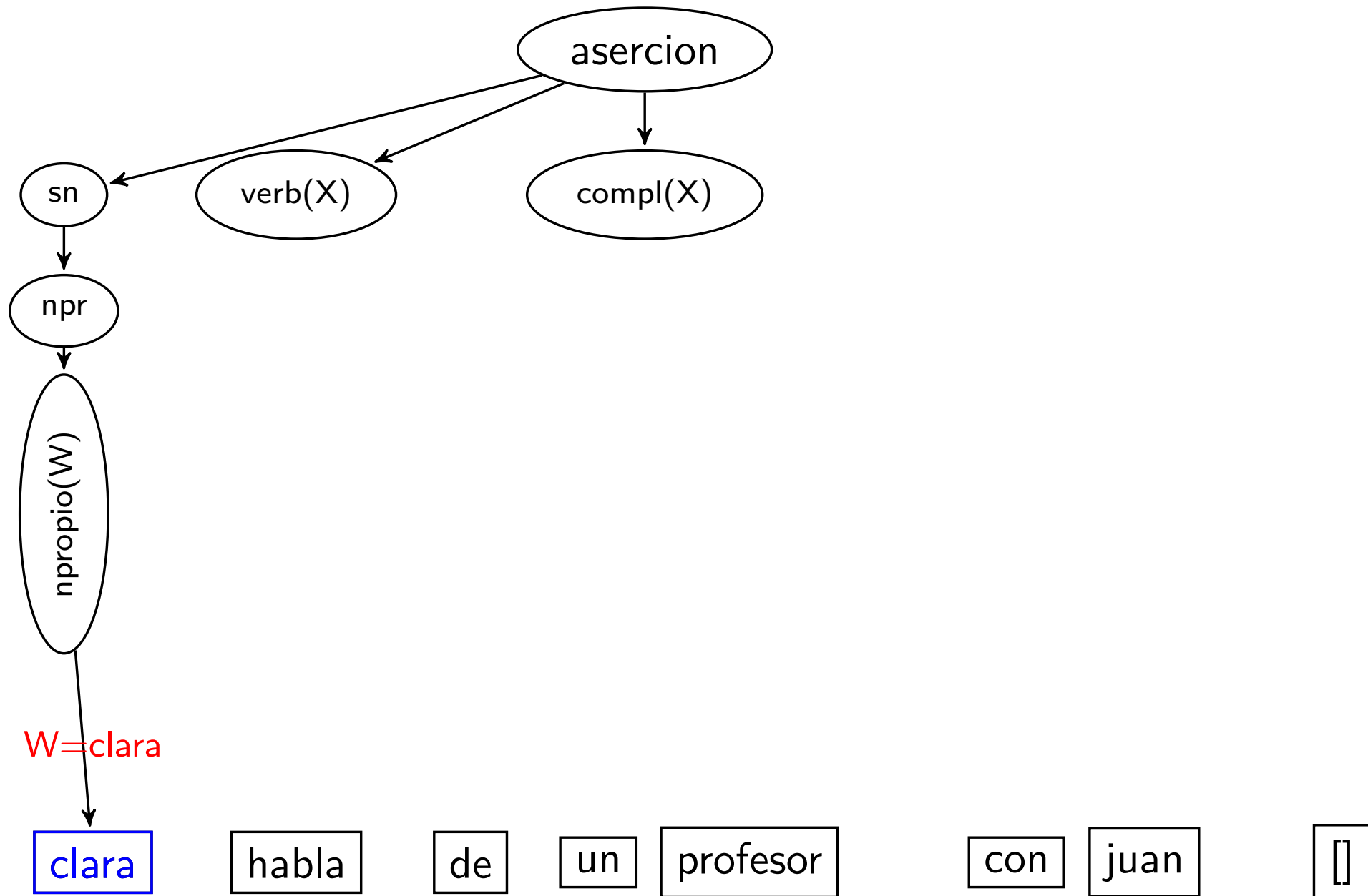
profesor

con

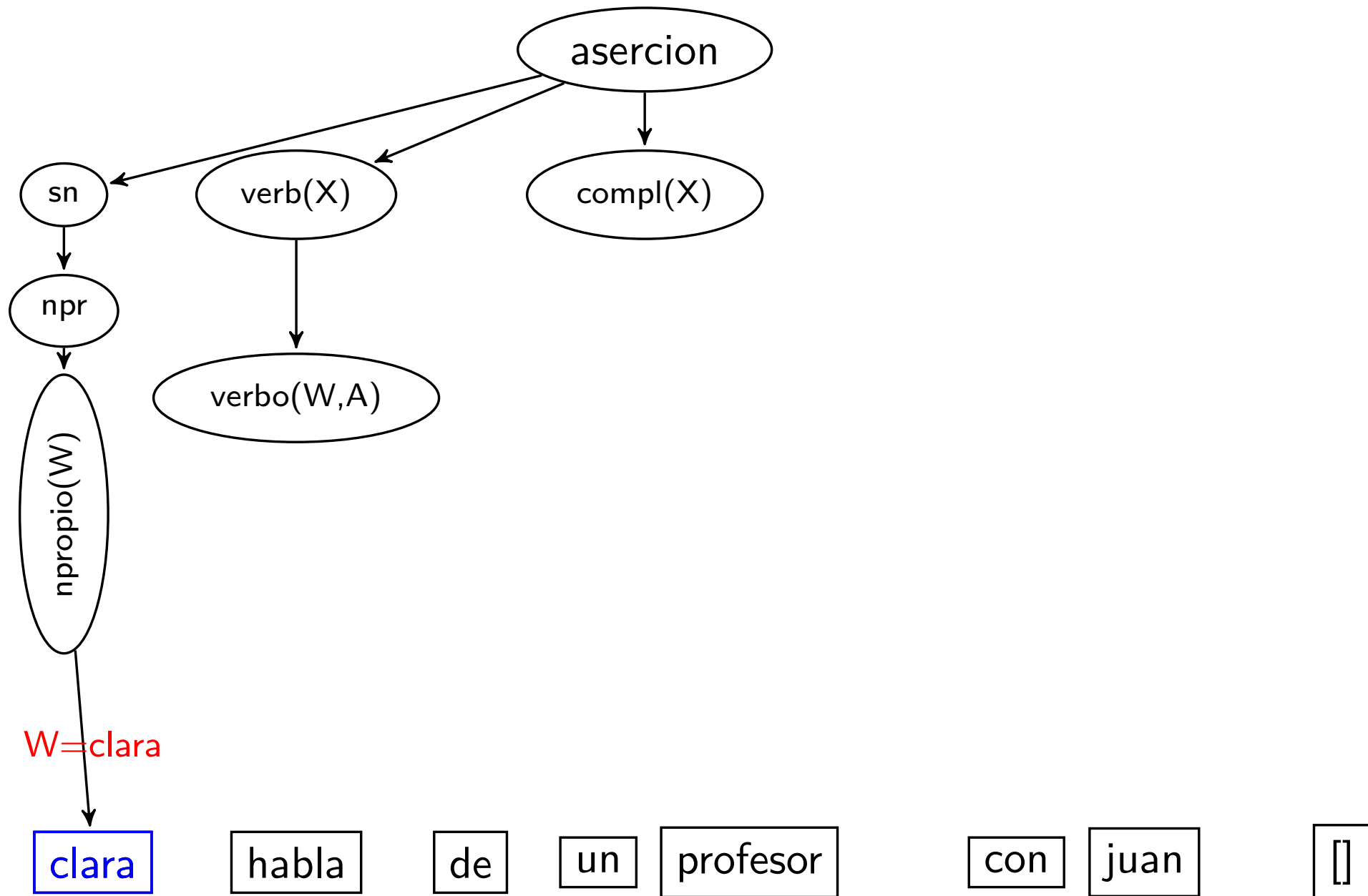
juan

□

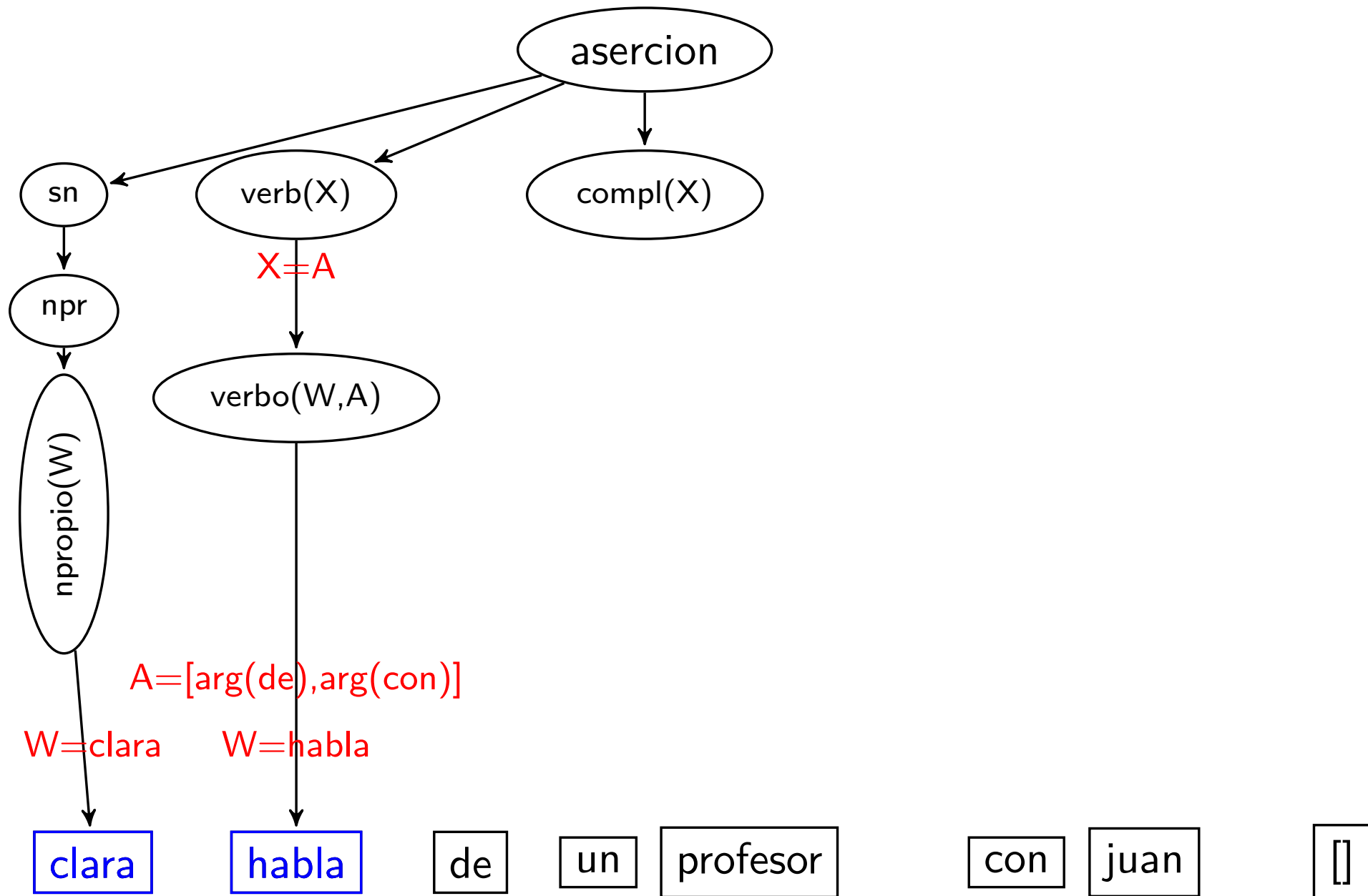
Gramática 2 - Ejemplo



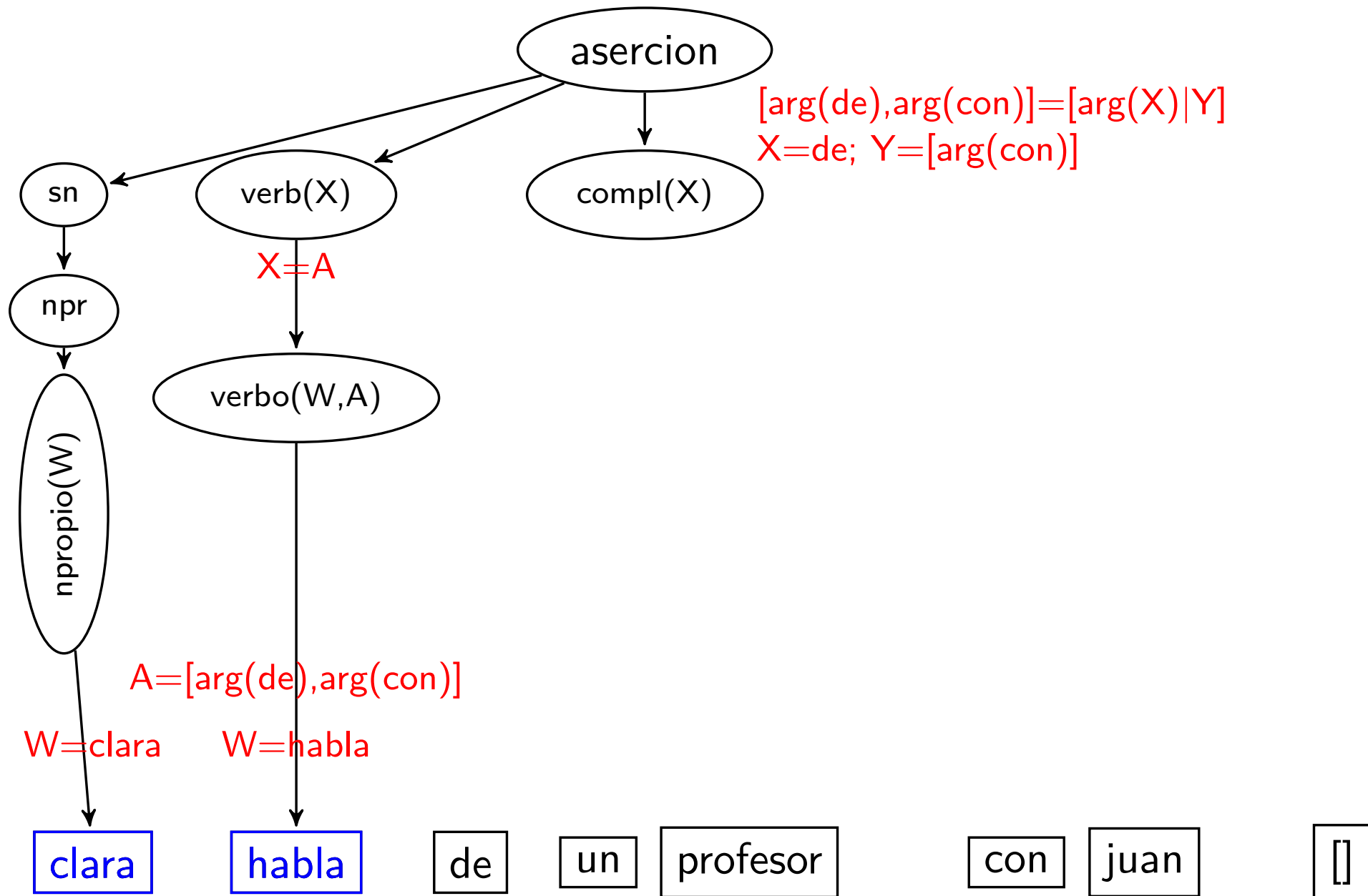
Gramática 2 - Ejemplo



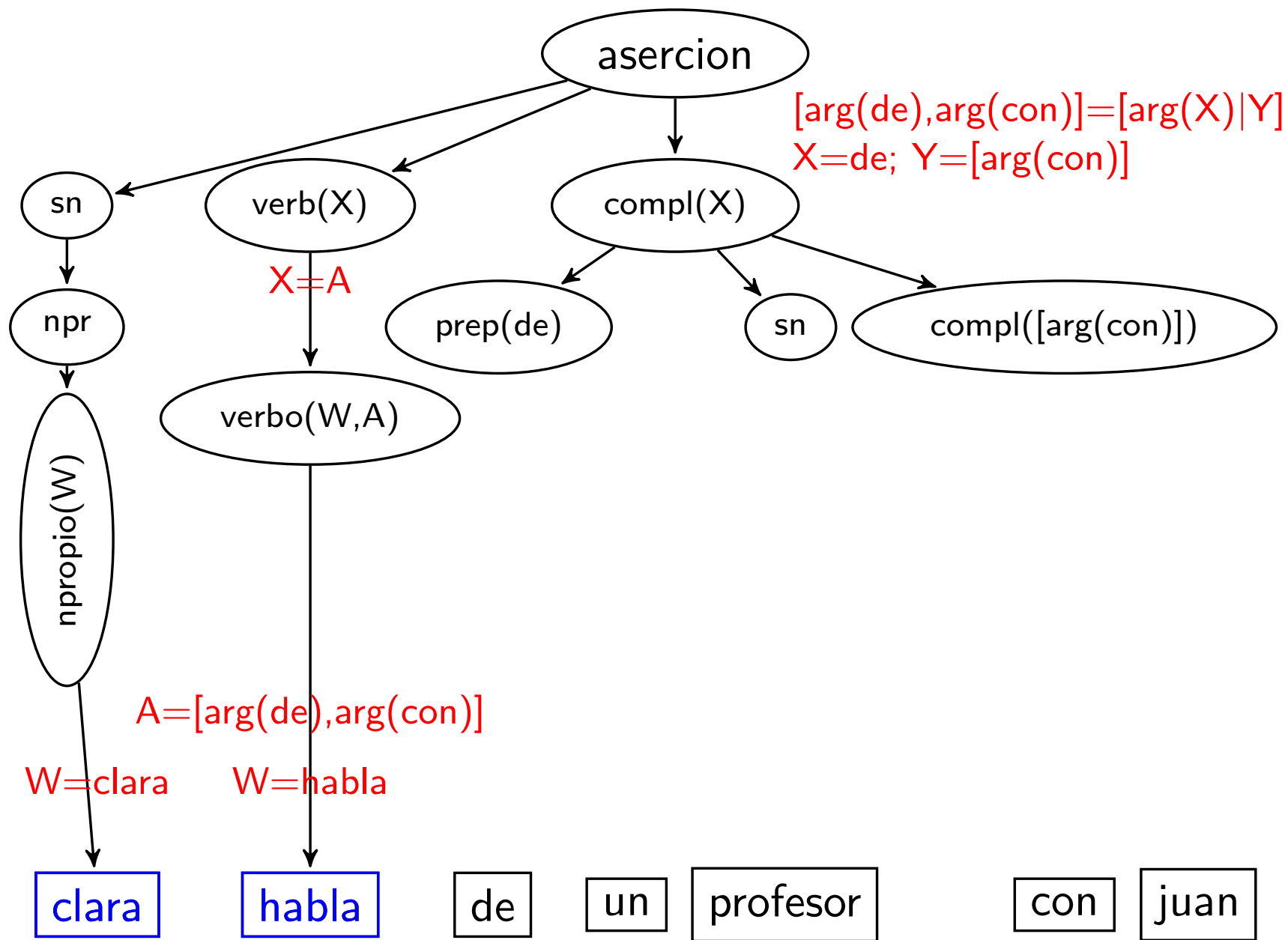
Gramática 2 - Ejemplo



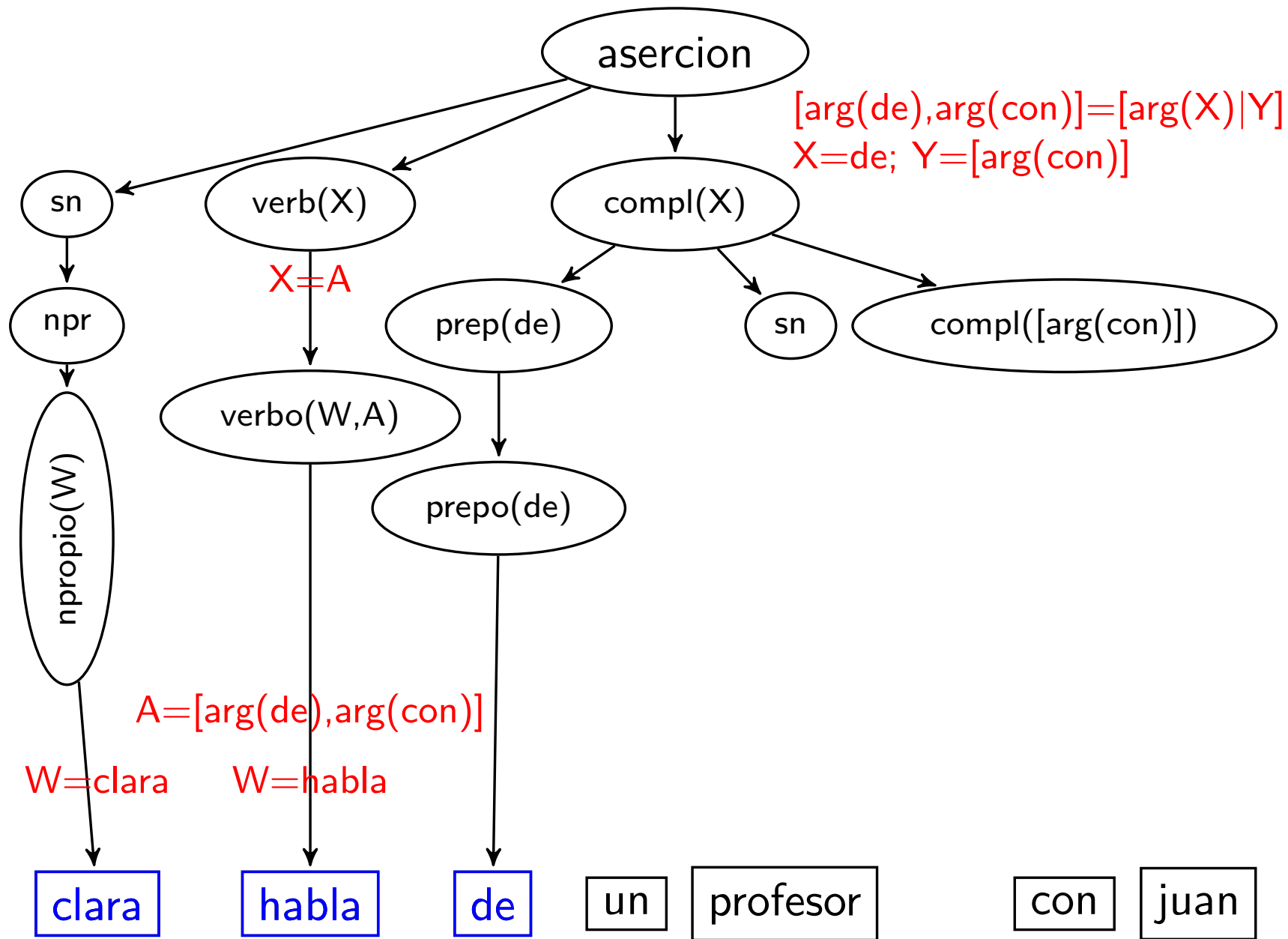
Gramática 2 - Ejemplo



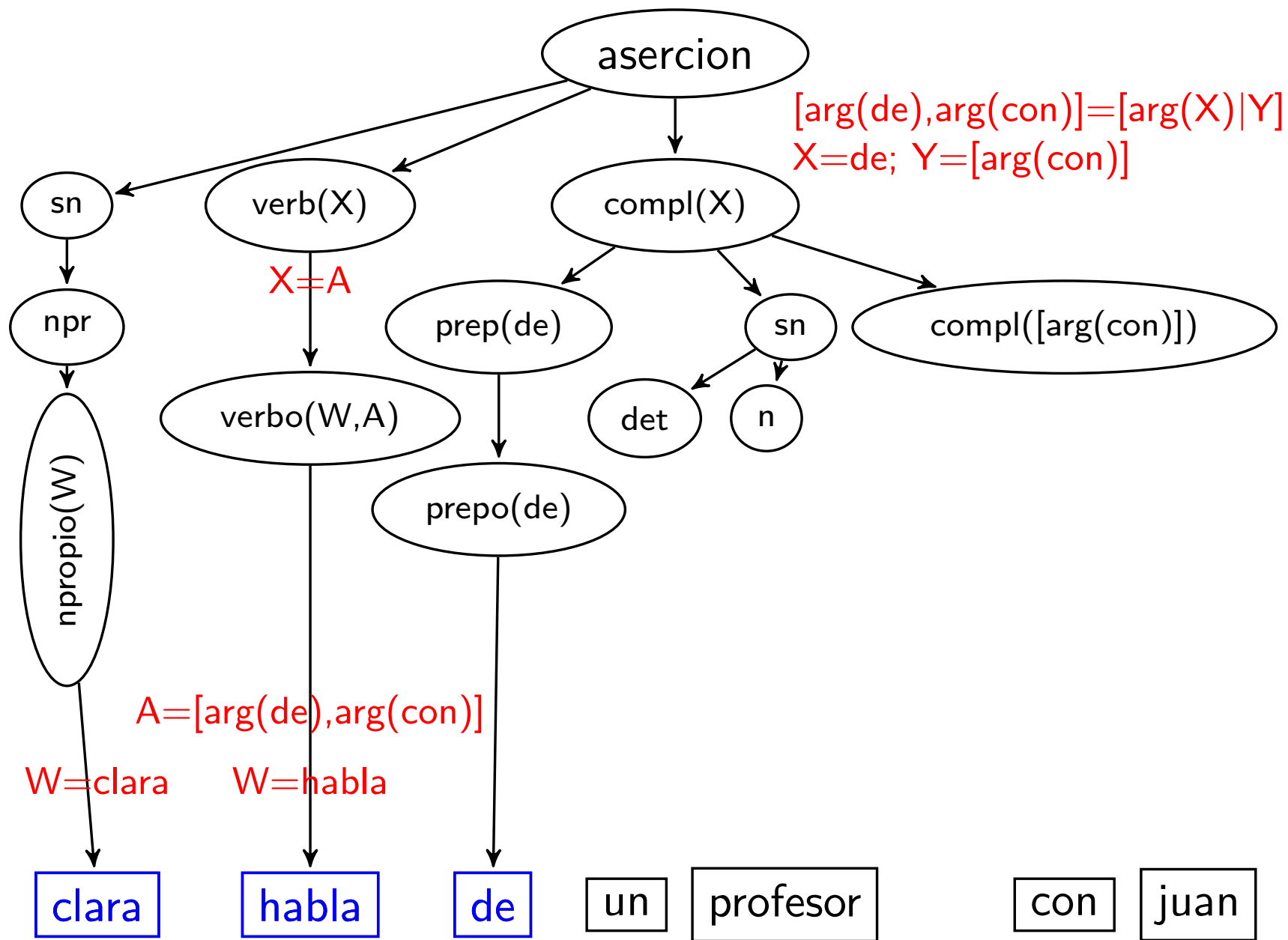
Gramática 2 - Ejemplo



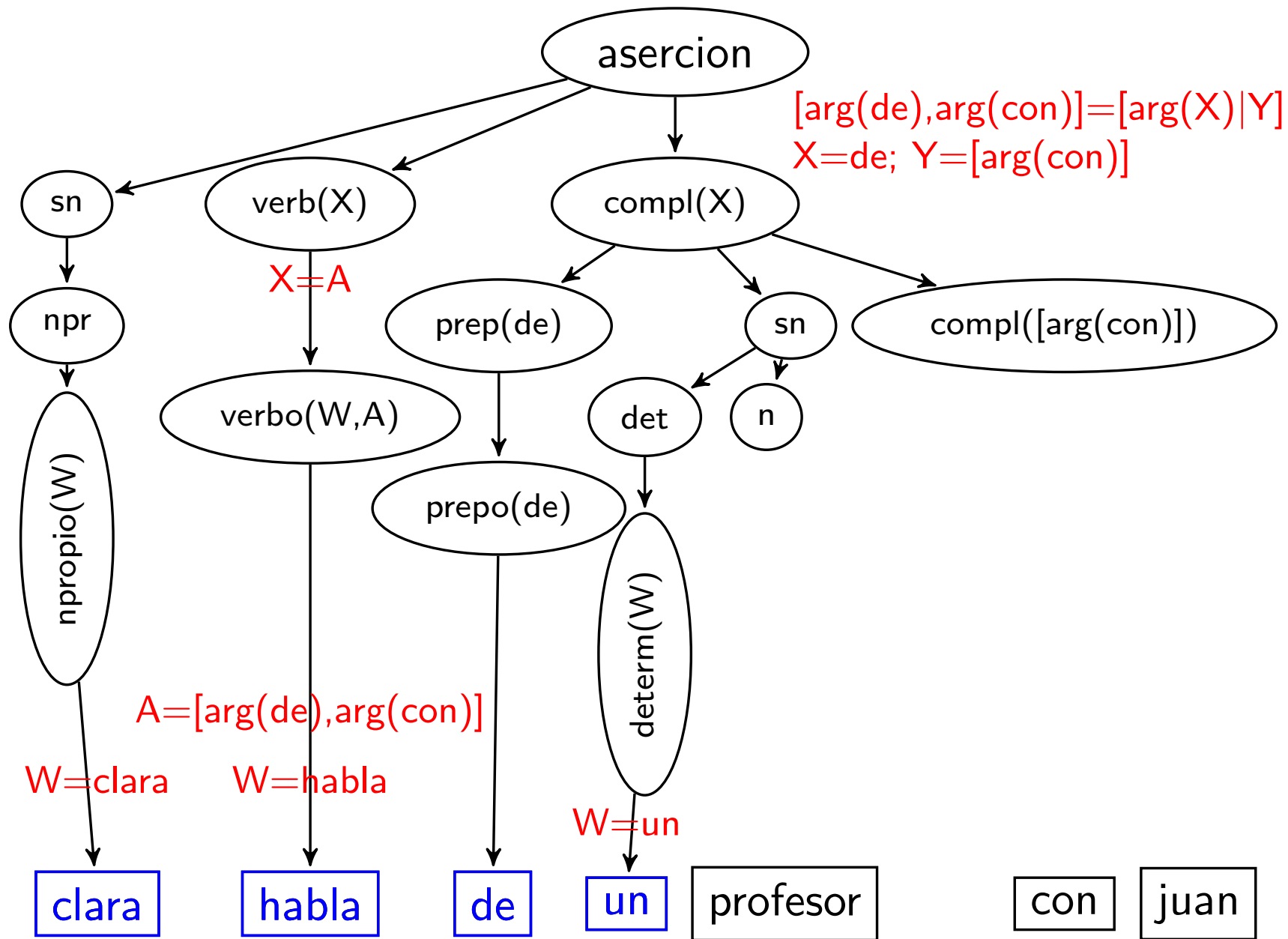
Gramática 2 - Ejemplo



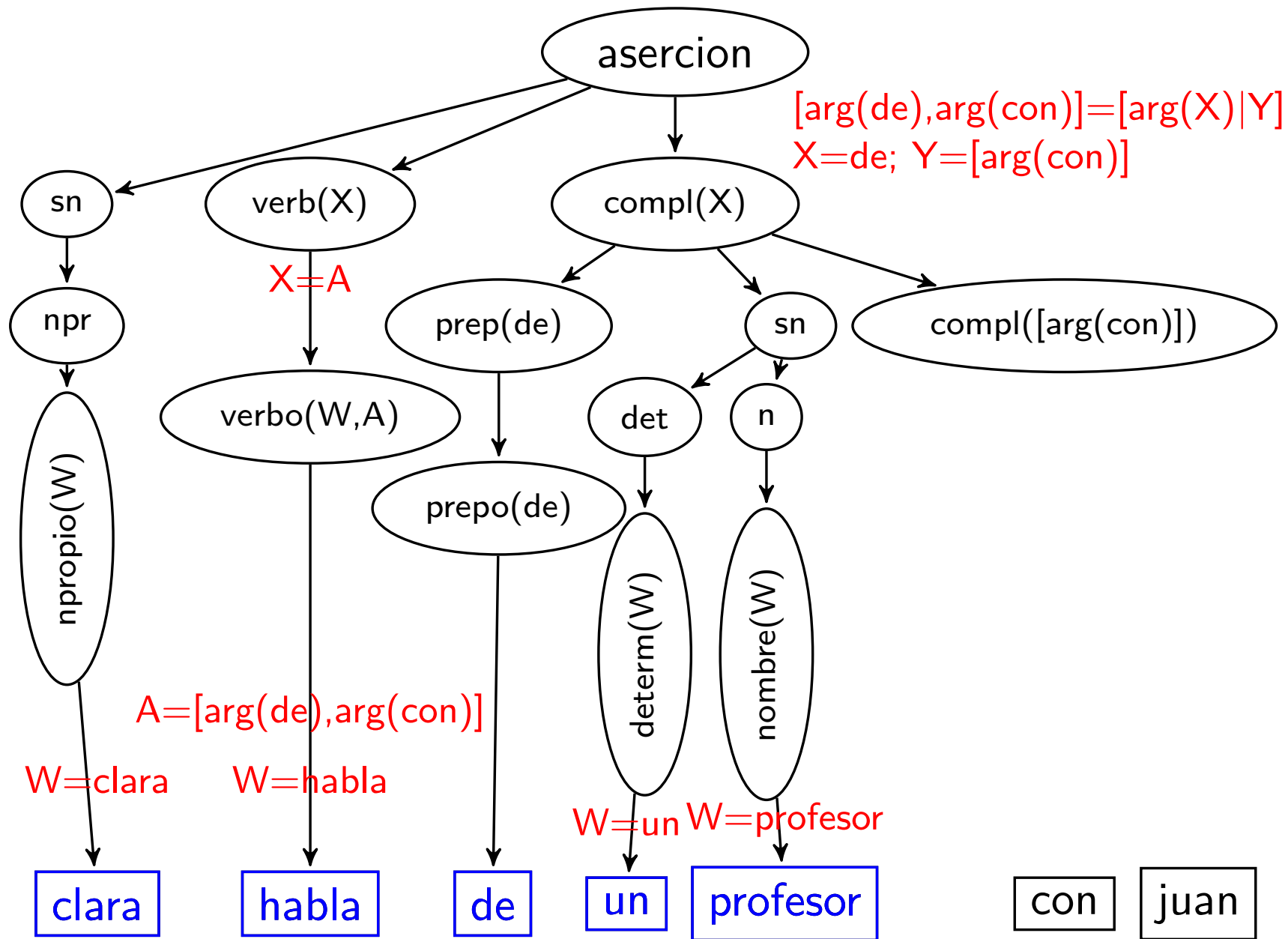
Gramática 2 - Ejemplo



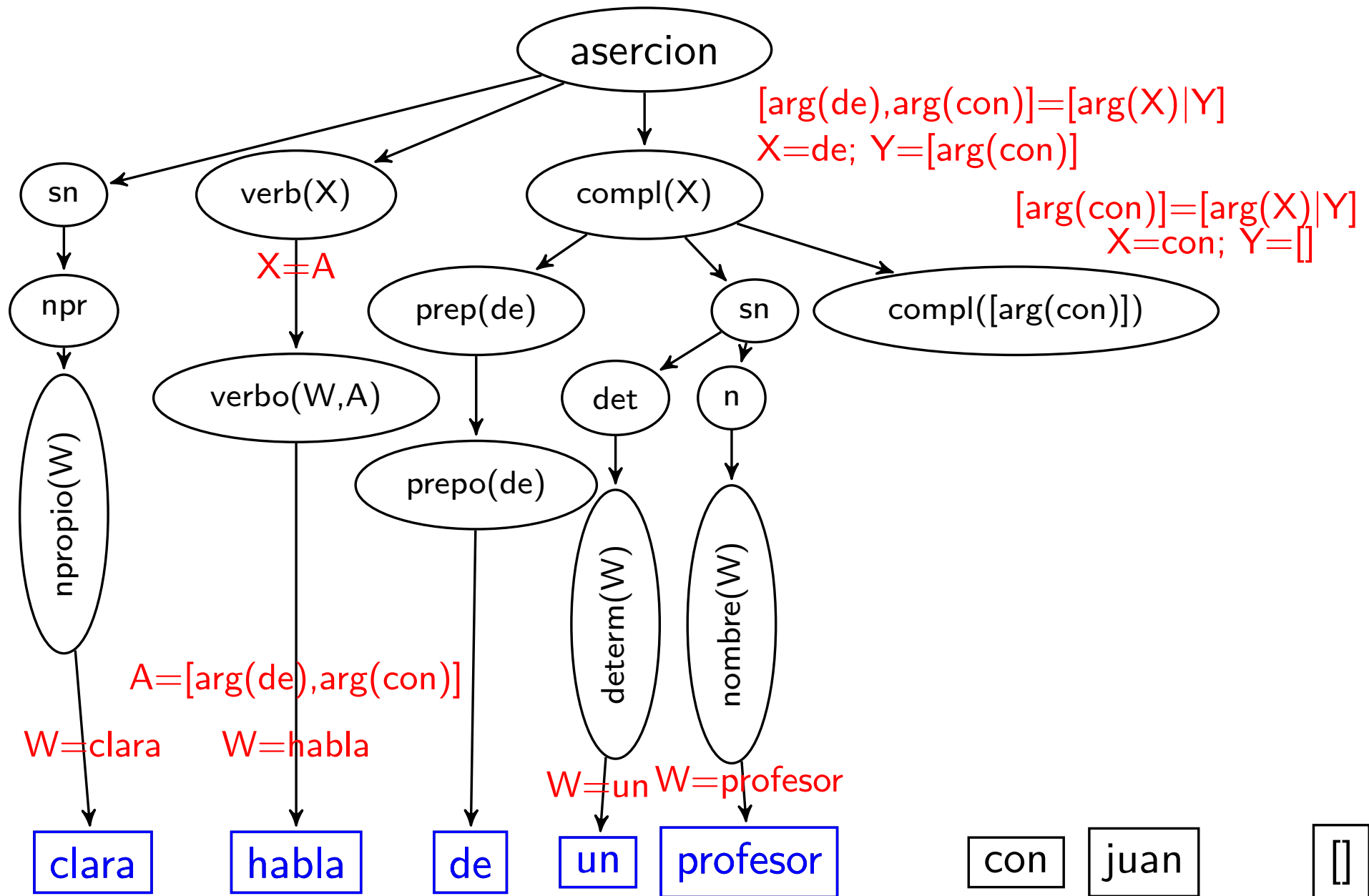
Gramática 2 - Ejemplo



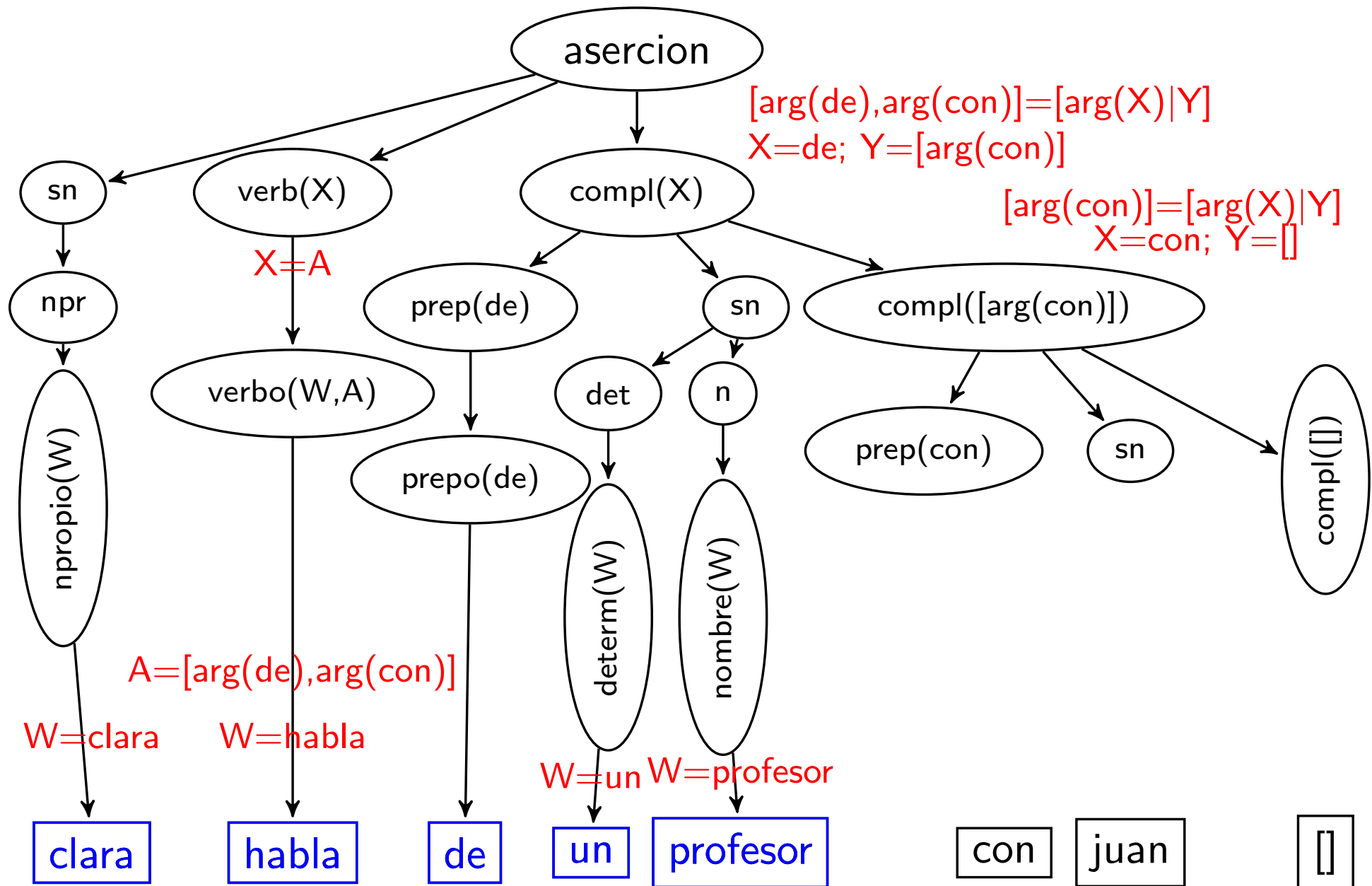
Gramática 2 - Ejemplo



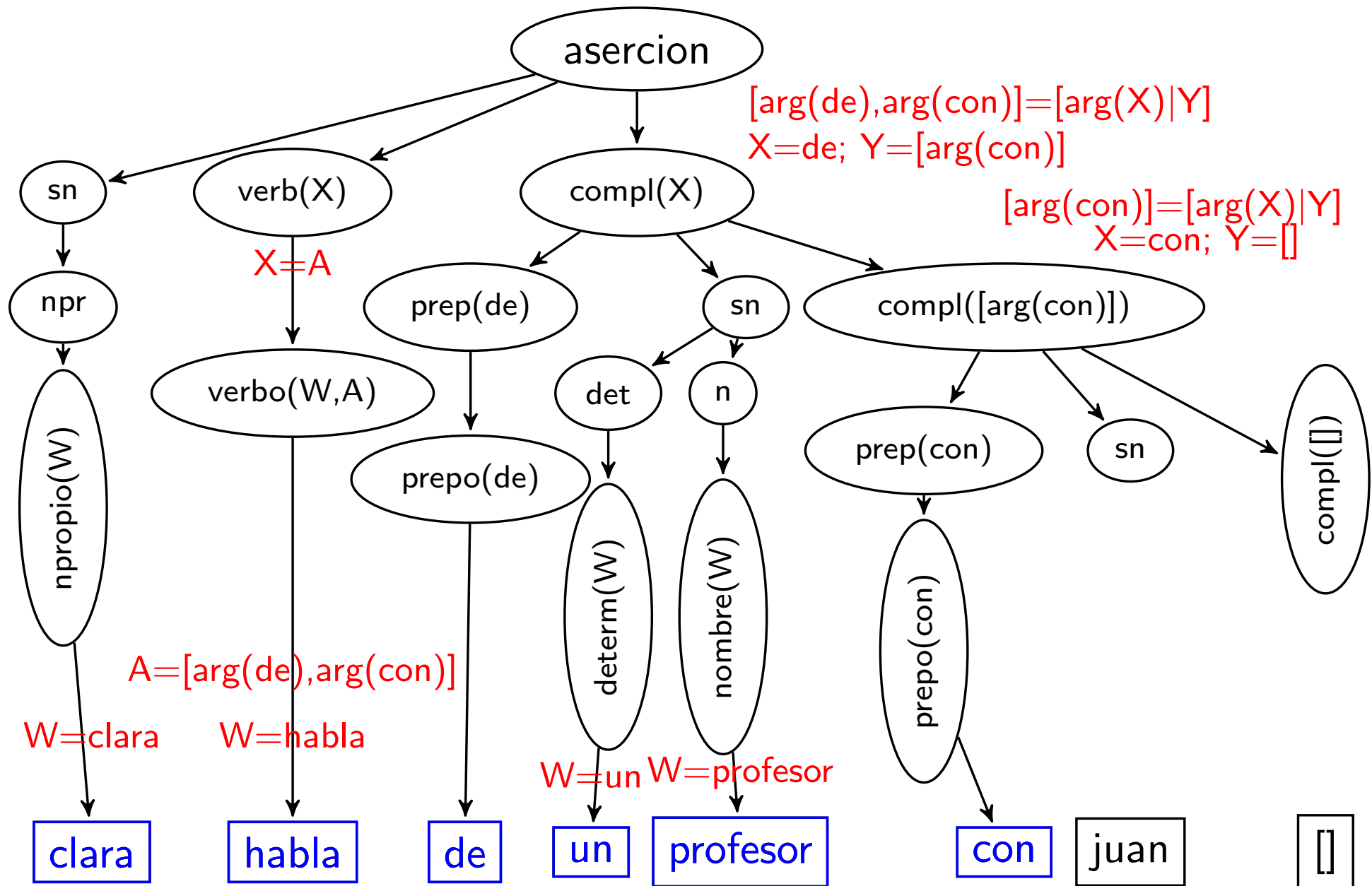
Gramática 2 - Ejemplo



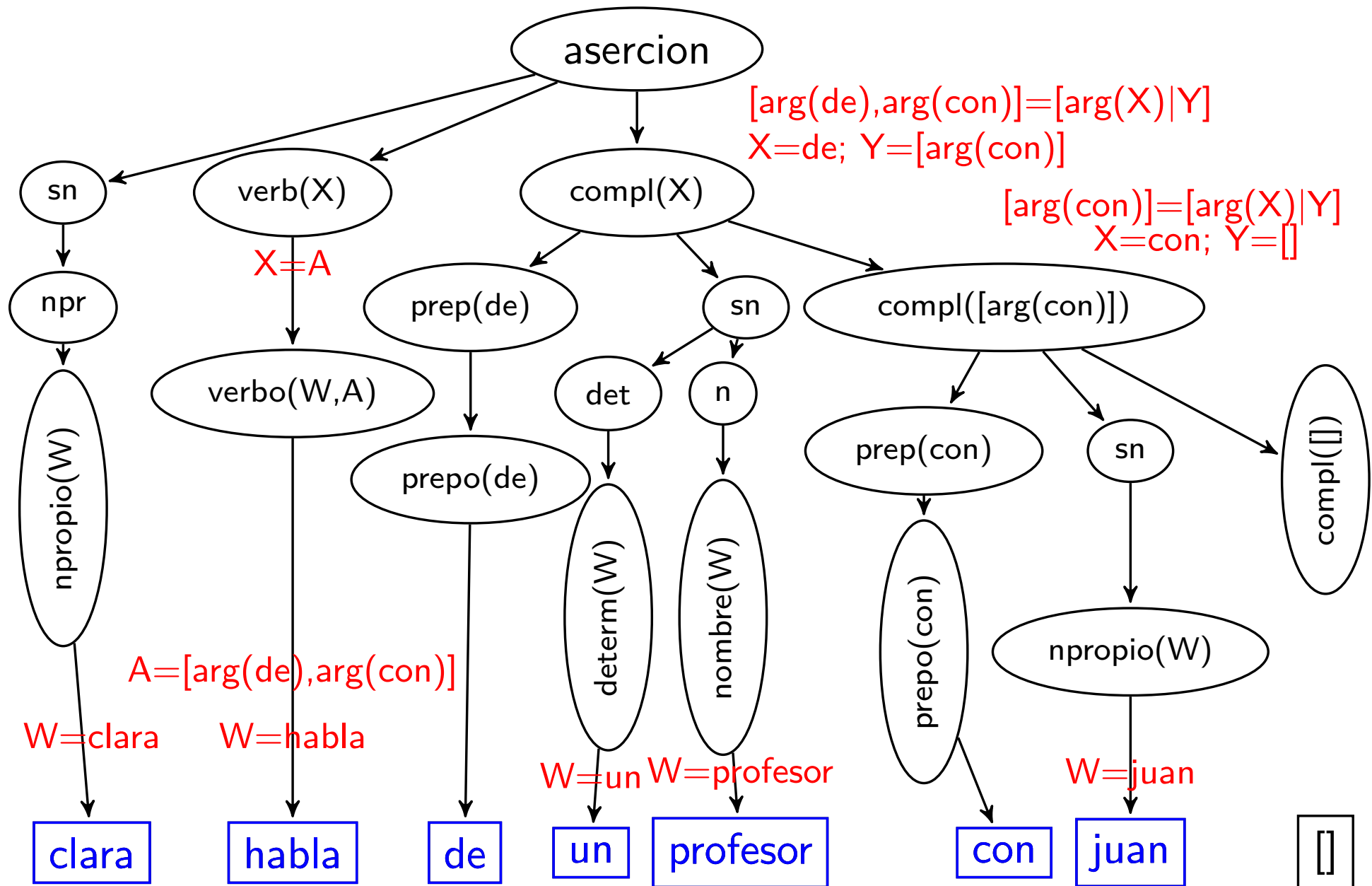
Gramática 2 - Ejemplo



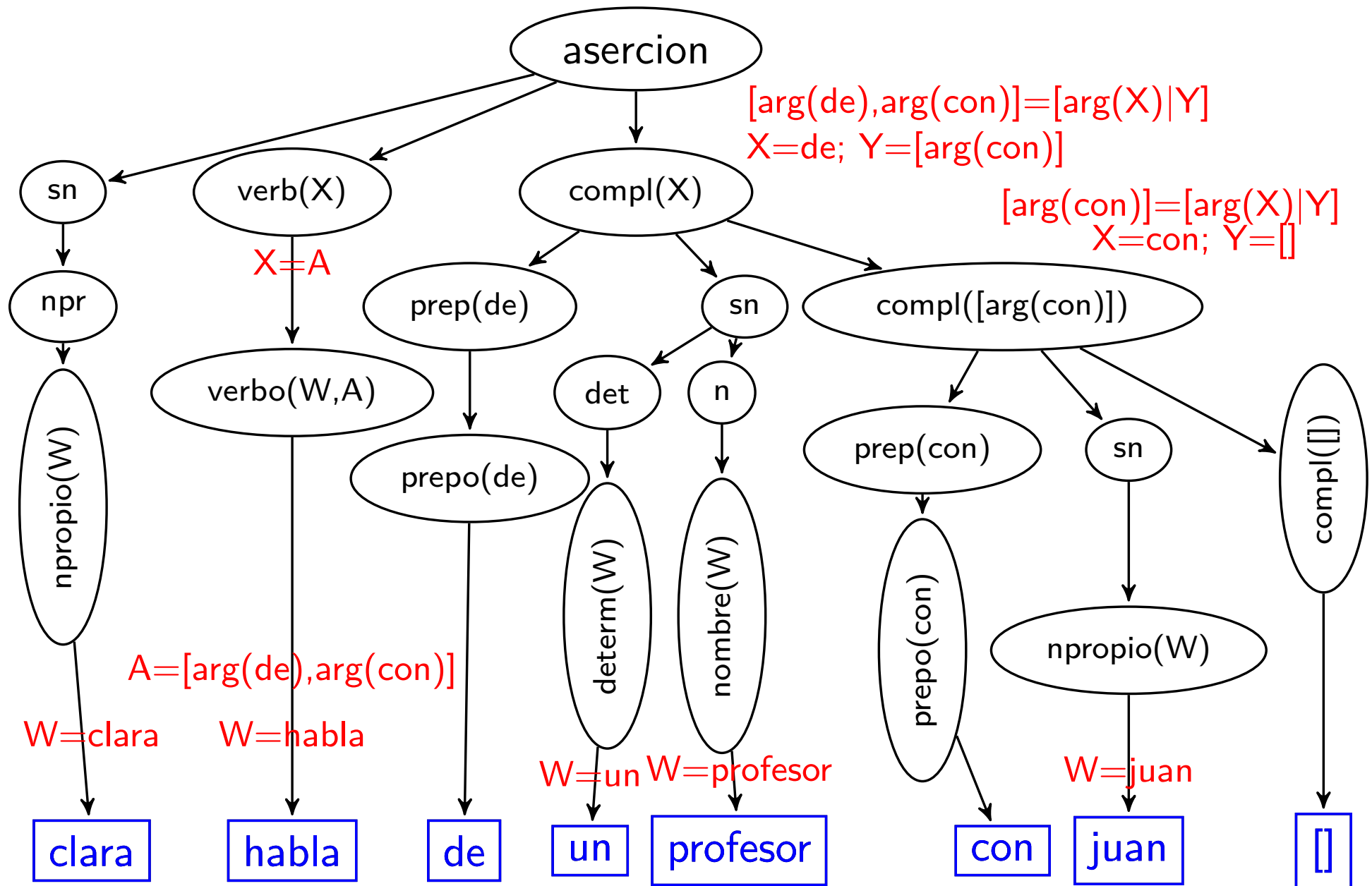
Gramática 2 - Ejemplo



Gramática 2 - Ejemplo



Gramática 2 - Ejemplo



Gramáticas lógicas - Gramática 3

- Vamos a crear una representación de las frases que reconocemos (semántica)
- Usamos como base la lógica de predicados e identificamos los constituyentes con elementos de CP_1
 - Verbos \longrightarrow predicados de aridad adecuada
 - Nombres propios \longrightarrow constantes
 - Nombres comunes \longrightarrow predicados unarios
- Modificamos el lexicón para incluir la representación de cada elemento
- Añadimos a la gramática las variables que nos permitirán recuperar la representación y construirla

Gramática 3 - I

```
analisis(F,X,Y):- asercion(F,X,Y).
```

```
asercion(F) --> sn(S), verb(S,X,F),compl(X).
```

```
compl([])--> [].
```

```
compl([arg(X,0)|Y])--> prep(X),sn(0),compl(Y).
```

```
compl([arg(nulo,0)|Y])--> sn(0),compl(Y).
```

```
sn(S)--> npr(S).
```

```
sn(S)--> det,n(S).
```

```
verb(S,A,F)--> [W],{verbo(W,S,A,F)}.
```

```
npr(W)--> [W],{npropio(W)}.
```

```
n(F)--> [W],{nombre(W,_,F)}.
```

```
det--> [W],{determ(W)}.
```

```
prep(W)--> [W],{prepo(W)}.
```

Gramática 3 - II

```
npropio(clara).  
npropio(maria).  
npropio(juan).  
npropio(barcelona).
```

```
nombre(libro,K,libro(K)).  
nombre(hombre,K,hombre(K)).  
nombre(profesor,K,profesor(K)).
```

```
determ(un).  
determ(el).
```

```
prepo(en).  
prepo(con).  
prepo(de).
```

```
verbo(rie,S, [], reir(S)).  
verbo(piensa,S, [arg(en,0)], pensar_en(S,0)).  
verbo(habla,S, [arg(de,0), arg(con,01)], comunica(S,0,01)).  
verbo(habla,S, [arg(con,0), arg(de,01)], comunica(S,01,0)).  
verbo(esta,S, [arg(en,0)], locativo(S,0)).  
verbo(lee,S, [arg(nulo,0)], leer(S,0)).
```

Gramática 3 - Frases

```
| ?- analisis(F,[juan,esta,en,barcelona],[ ]).
```

```
F = locativo(juan,barcelona) ?
```

```
yes
```

```
| ?- analisis(F,[juan,piensa,en,maria],[ ]).
```

```
F = pensar_en(juan,maria) ?
```

```
yes
```

```
| ?- analisis(F,[el,libro,esta,en,barcelona],[ ]).
```

```
F = locativo(libro(_A),barcelona) ?
```

```
yes
```

```
| ?- analisis(F,[juan,lee,un,libro],[ ]).
```

```
F = leer(juan,libro(_A)) ?
```

```
yes
```

```
| ?- analisis(F,[el,hombre,habla,de,juan,con,maria],[ ]).
```

```
F = comunica(hombre(_A),juan,maria) ?
```

```
yes
```

```
| ?- analisis(F,[el,hombre,rie],[ ]).
```

```
F = reir(hombre(_A)) ?
```

```
yes
```

```
| ?- analisis(F,[el,profesor,piensa,en,un,libro],[ ]).
```

```
F = pensar_en(profesor(_B),libro(_A)) ?
```

```
yes
```


Gramáticas lógicas - Gramática 4

- Para completar la representación de las frases incluimos la cuantificación
- Cambiamos la representación de los determinantes para que sean estos los que indiquen el tipo de cuantificación
- Añadimos las variables adecuadas a la gramática para que las variables que aparecen en todos los elementos de la representación se correspondan

Gramática 4 - I

```
analisis(F,X,Y):- asercion(F,X,Y).
```

```
asercion(F) --> sn(K,S2,F), verb(K,X,S1), compl(X,S1,S2).
```

```
compl([],S,S)--> [].
```

```
compl([arg(X,K)|Y],S1,S)--> prep(X), sn(K,S2,S), compl(Y,S1,S2).
```

```
compl([arg(nulo,K)|Y],S1,S)--> sn(K,S2,S), compl(Y,S1,S2).
```

```
sn(K,F,F)--> npr(K).
```

```
sn(K,S2,F)--> det(K,S1,S2,F),n(K,S1).
```

```
verb(S,A,F)--> [W],{verbo(W,S,A,F)}.
```

```
npr(W)--> [W],{npropio(W)}.
```

```
n(K,F)--> [W],{nombre(W,K,F)}.
```

```
det(K,S1,S2,F)--> [W],{determ(W,K,S1,S2,F)}.
```

```
prep(W)--> [W],{prepo(W)}.
```


Gramática 4 - II

```
npropio(clara).  
npropio(maria).  
npropio(juan).  
npropio(barcelona).
```

```
nombre(libro,K,libro(K)).  
nombre(hombre,K,hombre(K)).  
nombre(profesor,K,profesor(K)).
```

```
determ(el,K,S1,S2,e(K,and(S1,S2))).  
determ(un,K,S1,S2,e(K,and(S1,S2))).  
determ(los,K,S1,S2,a(K,implies(S1,S2))).  
determ(todo,K,S1,S2,a(K,implies(S1,S2))).
```

```
prepo(en).  
prepo(con).  
prepo(de).
```

```
verbo(rie,S,[],reir(S)).  
verbo(piensa,S,[arg(en,0)],pensar_en(S,0)).  
verbo(habla,S,[arg(de,0),arg(con,01)],comunica(S,0,01)).  
verbo(habla,S,[arg(con,0),arg(de,01)],comunica(S,01,0)).  
verbo(esta,S,[arg(en,0)],locativo(S,0)).  
verbo(lee,S,[arg(nulo,0)],leer(S,0)).
```

Gramática 4 - Frases

```
| ?- analisis(F,[el,hombre,rie],[ ]).
```

```
F = e(_A,and(hombre(_A),reir(_A))) ?
```

```
yes
```

```
| ?- analisis(F,[el,profesor,piensa,en,un,libro],[ ]).
```

```
F = e(_B,and(profesor(_B),e(_A,and(libro(_A),pensar_en(_B,_A)))))) ?
```

```
yes
```

```
| ?- analisis(F,[el,hombre,habla,de,juan,con,maria],[ ]).
```

```
F = e(_A,and(hombre(_A),comunica(_A,juan,maria))) ?
```

```
yes
```

```
| ?- analisis(F,[todo,hombre,piensa,en ,un,libro],[ ]).
```

```
F = a(_B,implies(hombre(_B),e(_A,and(libro(_A),pensar_en(_B,_A)))))) ?
```

```
yes
```

```
| ?- analisis(F,[todo,libro,esta,en,barcelona],[ ]).
```

```
F = a(_A,implies(libro(_A),locativo(_A,barcelona))) ?
```

```
yes
```

```
| ?- analisis(F,[todo,libro,habla,de,un,hombre],[ ]).
```

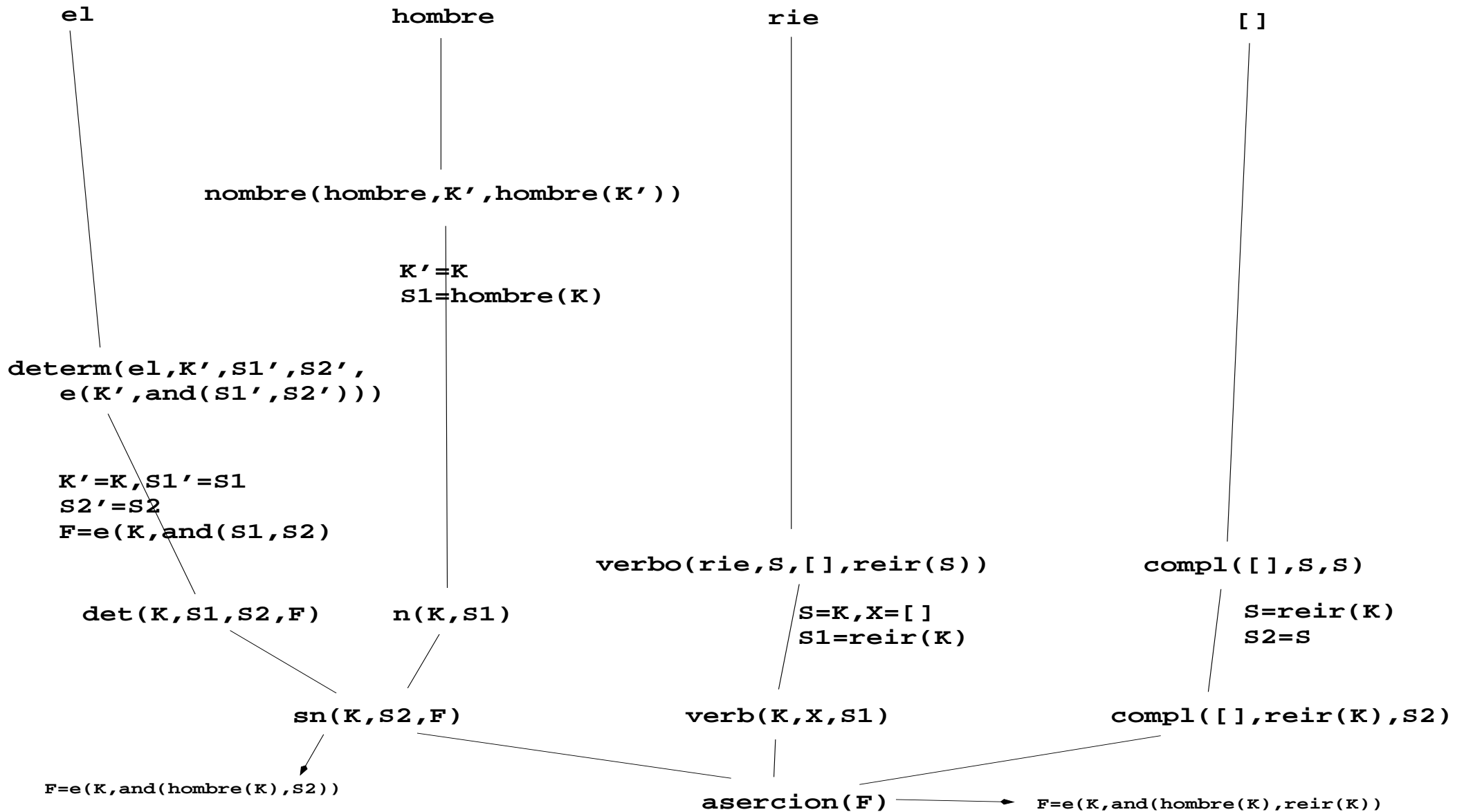
```
no
```

```
| ?- analisis(F,[todo,libro,habla,de,un,hombre,con,un,profesor],[ ]).
```

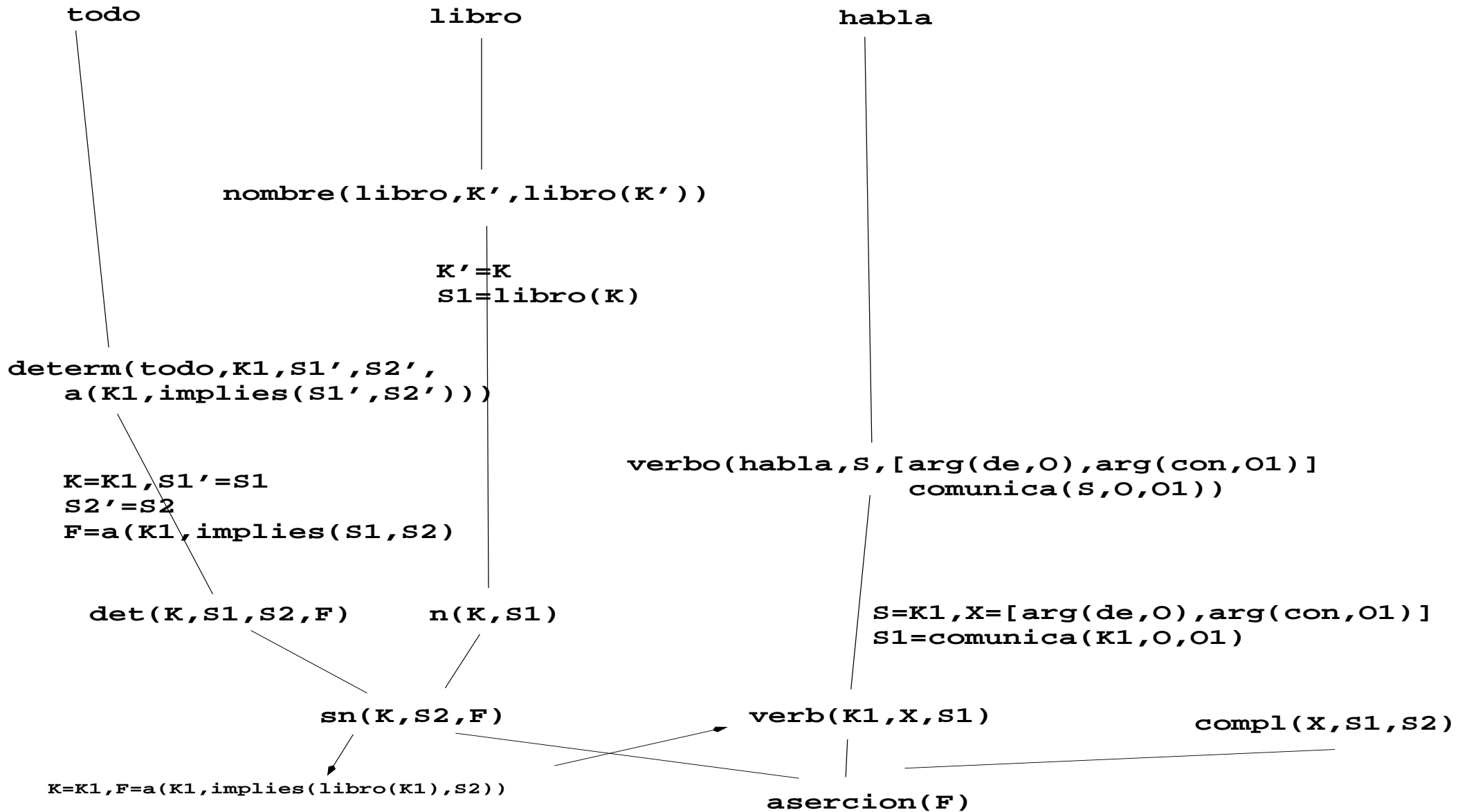
```
F = a(_C,implies(libro(_C),e(_B,and(hombre(_B),  
e(_A,and(profesor(_A),comunica(_C,_B,_A))))))) ?
```

```
yes
```

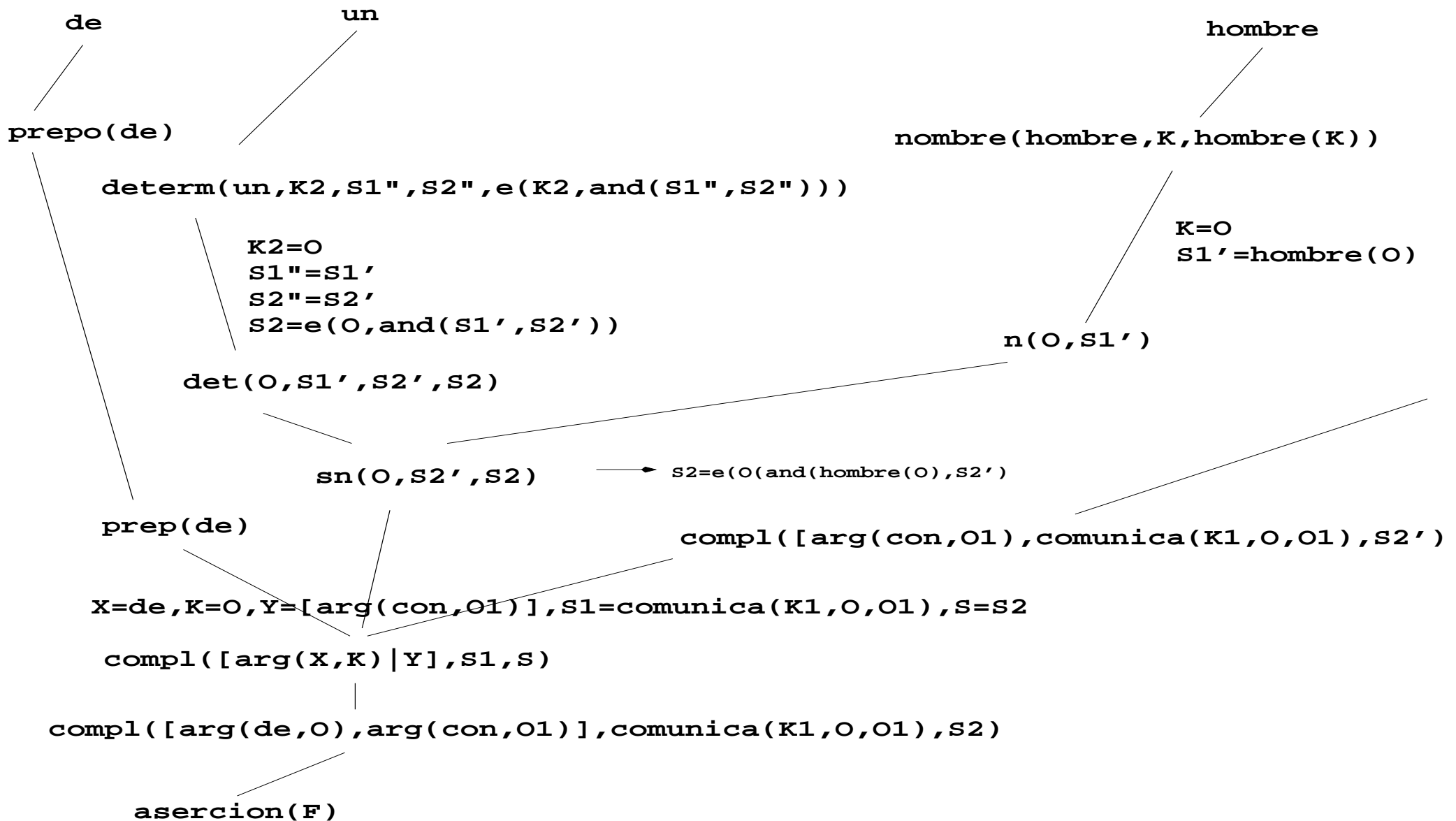
Gramática 4 - Ejemplo 1



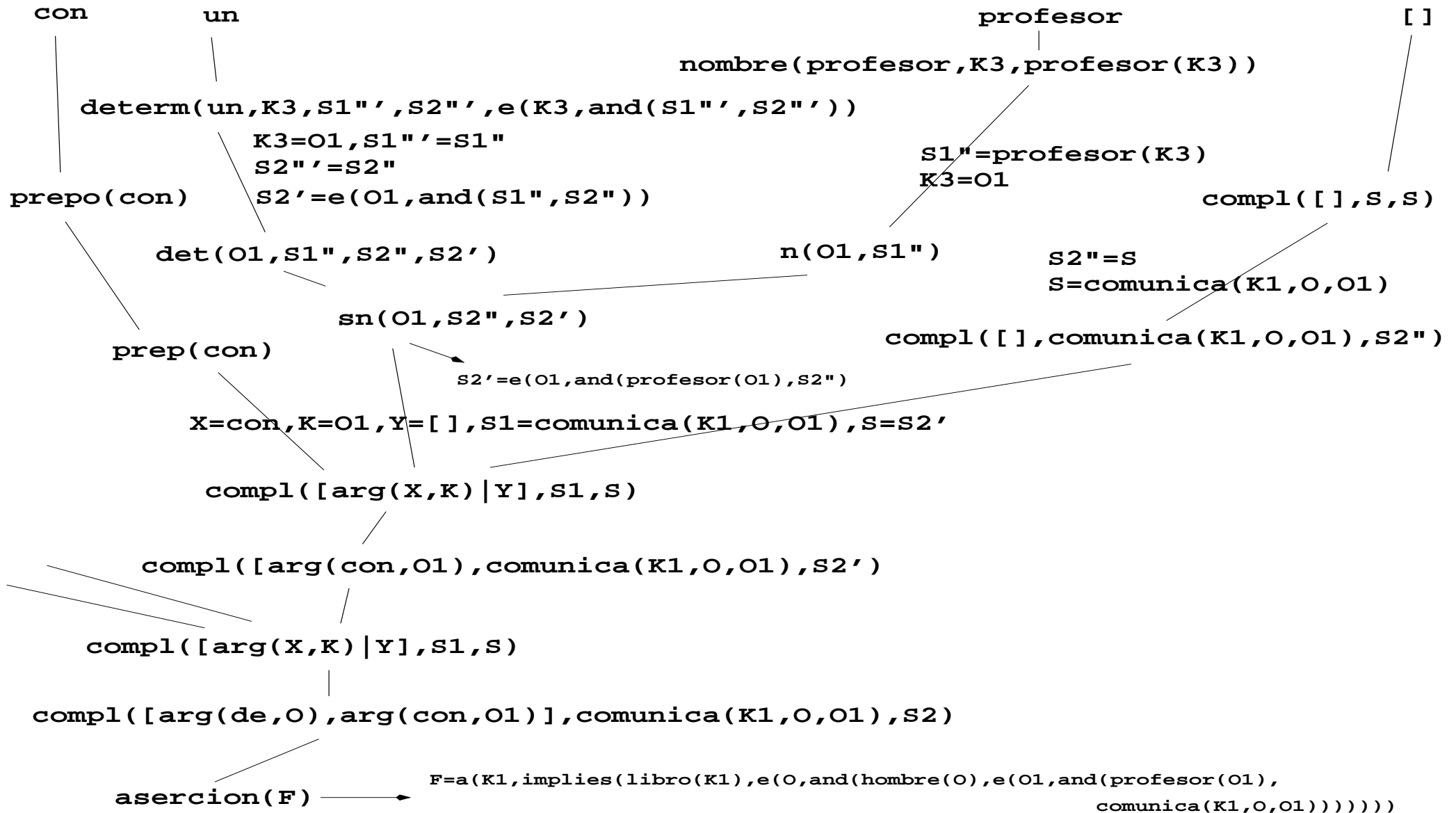
Gramática 4 - Ejemplo 2 (I)



Gramática 4 - Ejemplo 2 (II)



Gramática 4 - Ejemplo 2 (III)



Gramáticas lógicas - Gramática 5

- Completamos la gramática incluyendo información semántica
- Ahora añadimos restricciones semánticas a los argumentos de los complementos verbales
- Para ello aumentamos el lexicón con información semántica (nombres incluyen su tema, verbos indican los temas compatibles)

Gramática 5 - I

```
analisis(F,X,Y):- asercion(F,X,Y).
```

```
asercion(F) --> sn(K,S2,F,Sem1), verb(K,X,S1,asem(Sem1,Sem2)),  
               compl(X,S1,S2,Sem2).
```

```
compl([],S,S,[])--> [].
```

```
compl([arg(X,K)|Y],S1,S,[Sem1|Sem2])--> prep(X), sn(K,S2,S,Sem1),  
                                           compl(Y,S1,S2,Sem2).
```

```
compl([arg(nulo,K)|Y],S1,S,[Sem1|Sem2])--> sn(K,S2,S,Sem1),  
                                           compl(Y,S1,S2,Sem2).
```

```
sn(K,F,F,Sem)--> npr(K,Sem).
```

```
sn(K,S2,F,Sem)--> det(K,S1,S2,F),n(K,S1,Sem).
```

```
verb(S,A,F,AS)--> [W],{verbo(W,S,A,F,AS)}.
```

```
npr(W,S)--> [W],{npropio(W,S)}.
```

```
n(K,F,S)--> [W],{nombre(W,K,F,S)}.
```

```
det(K,S1,S2,F)--> [W],{determ(W,K,S1,S2,F)}.
```

```
prep(W)--> [W],{prepo(W)}.
```


Gramática 5 - II

```
npropio(clara,humano).  
npropio(maria,humano).  
npropio(juan,humano).  
npropio(barcelona,locativo).
```

```
nombre(libro,K,libro(K),inanimado).  
nombre(hombre,K,hombre(K),humano).  
nombre(profesor,K,profesor(K),humano).  
nombre(perro,K,profesor(K),animado).  
nombre(gato,K,gato(K),animado).  
nombre(camino,K,camino(K),locativo).  
nombre(pescado,K,pescado(K),inanimado).
```

```
determ(e1,K,S1,S2,e(K,and(S1,S2))).  
determ(un,K,S1,S2,e(K,and(S1,S2))).  
determ(los,K,S1,S2,a(K,implies(S1,S2))).  
determ(todo,K,S1,S2,a(K,implies(S1,S2))).
```

```
prepo(en).  
prepo(con).  
prepo(de).
```

```
verbo(rie,S,[],reir(S),asem(humano,[])).  
verbo(piensa,S,[arg(en,0)],pensar_en(S,0),asem(humano,[_])).  
verbo(habla,S,[arg(de,0),arg(con,01)],comunica(S,0,01),asem(humano,[_],humano))).  
verbo(habla,S,[arg(con,0),arg(de,01)],comunica(S,01,0),asem(humano,[humano,_])).  
verbo(esta,S,[arg(en,0)],locativo(S,0),asem(_,[locativo])).  
verbo(lee,S,[arg(nulo,0)],leer(S,0),asem(humano,[inanimado])).  
verbo(corre,S,[arg(en,0)],correr(S,0),asem(animado,[locativo])).  
verbo(come,S,[arg(en,0)],comer(S,0),asem(animado,[locativo])).  
verbo(come,S,[arg(nulo,0)],comer(S,0),asem(animado,[inanimado])).
```