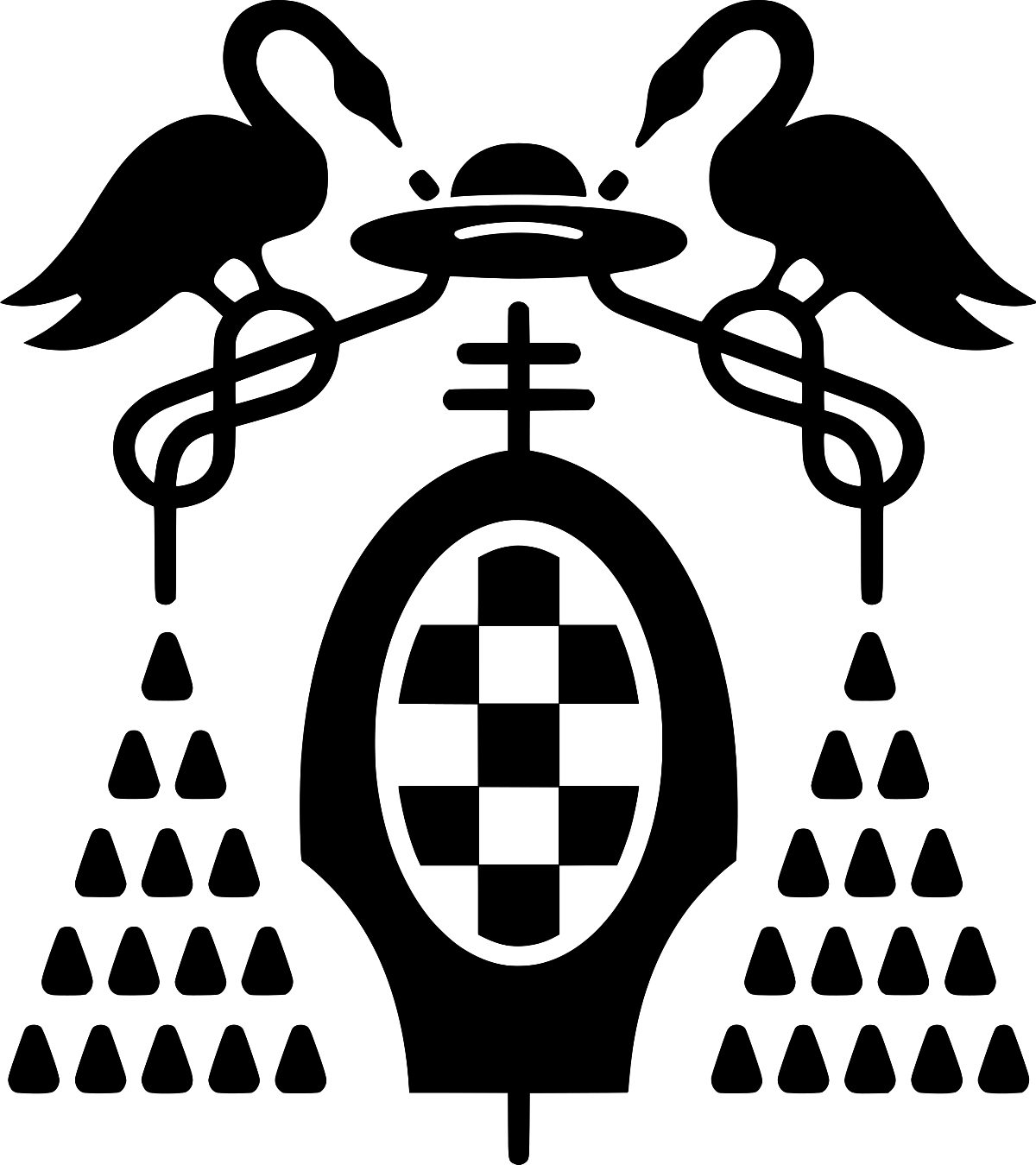
**Conocimiento y Razonamiento Automatizado**

*Analizador sint´actico de oraciones*

Grado en Ingenier´ıa Informa´tica

Universidad de Alcala´



Pablo Garc´ıa Garc´ıa

Alvaro Jesu´s Mart´ınez Parra´ Alejandro Raboso Vindel

27 de abril de 2023

**´Indice general**

1. **Introduccio´n y objetivos 2**
   1. Ana´lisis de oraciones . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
   2. Simplificacio´n de oraciones . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2. **An´alisis sint´actico de oraciones 3**
   1. Diccionario . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   2. Grama´tica . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
3. **Mejoras 4**
   1. Traduccio´n a ingl´es . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   2. Conexio´n entre Prolog y Python . . . . . . . . . . . . . . . . . . . . . . . . . 4
   3. Arbol sint´actico en L´ ATEX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
4. **Informacio´n relevante 6**
   1. Requisitos y dependencias . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
   2. Detalles . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6



**Cap´ıtulo 1**

# Introduccio´n y objetivos

## 1.1. An´alisis de oraciones

## 1.2. Simplificacio´n de oraciones

**Cap´ıtulo 2**

# An´alisis sint´actico de oraciones

## 2.1. Diccionario

Para poder realizar el análisis sintáctico de las oraciones es necesaria la creación de un diccionario con todas las palabras de la lengua que se vayan a usar.

Este diccionario lo crearemos con las palabras separadas por su tipo, si son nombres propios, sustantivos, adjetivos, etc. Es necesario guardar las palabras con su tipo para poder realizar un análisis correcto debido a que será necesario saber en qué orden están en las oraciones para así conocer la estructura que tendrá la oración.

Para el diccionario español se ha seguido la siguiente estructura:  
%Nombres propios

np(‘IRENE’).

%Nombres

n(‘mesa’).

%Verbos

v(‘estudia’).

A parte del diccionario con las palabras de la lengua española, se ha tenido que crear uno con las palabras de la lengua inglesa. Esto se debe a que necesitaremos poder aplicar la traducción a las oraciones por lo que en este diccionario tendremos las palabras del diccionario español junto a su traducción al inglés. Gracias a esto podremos realizar la traducción de las palabras directamente nos aparezcan en las oraciones.

Para este diccionario de traducciones se ha seguido la siguiente estructura:  
%Nouns

n(‘mesa’,’table’).

%Verbs

v(‘estudia’,’studies’).

Y así sucesivamente para el resto de las palabras necesarias para todas las oraciones que se quieran analizar o traducir.

**2.2. Gram´atica**

Para la realización de la gramática necesaria para poder realizar el análisis de las oraciones se han tenido que crear diferentes términos:

Para empezar primero se ha debido crear el diccionario necesario que contenga las palabras de las oraciones a analizar. Se ha desarrollado como se ha podido explicar en el apartado anterior por lo que de esta forma ya no solo contamos con las palabras necesarias sino con sus términos siendo estos los determinantes, nombres, nombres propios, etc.

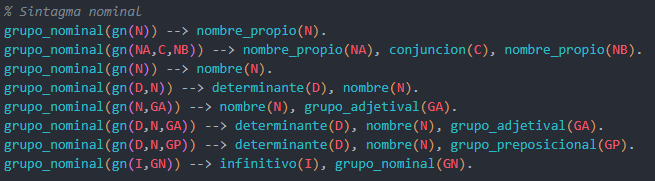
Con esto ya tendríamos el nivel más bajo del árbol de constituyentes. Para poder trabajar con el diccionario que hemos creado junto con la gramática correspondiente a los grupos sintácticos, se ha tenido que dar una equivalencia entre los tipos que tienen las palabras del diccionario a nuestros nodos terminales del árbol. Esto se realiza para que cuando llegue a uno de nuestros nodos terminales se guarde esa palabra y se pueda continuar con el análisis del resto del árbol.

Esto se ha realizado de la siguiente forma:  
Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media

Después, se han tenido que crear los grupos sintácticos que estarían compuestos de estos términos del diccionario por lo que agregamos los predicados que compongan los grupos nominales, adjetivales, verbales, etc.

Esto solo se puede hacer conociendo como se estructura la lengua española ya que es necesario saber como se forma estos grupos sintácticos. Por ejemplo, un grupo nominal se puede componer solo de un nombre propio, o también podría componerse de dos nombres propios unidos por una conjunción (José y Héctor) o por ejemplo un determinante seguido de un nombre y un adjetivo (el niño guapo). Entonces se han realizado todas las posibles combinaciones que tienen los distintos grupos sintácticos en el lenguaje.

La siguiente imagen muestra la estructura de cómo se han realizado los grupos nominales:  
****

Además de contar con las combinaciones necesarias para todos los grupos, en algunos casos se han tenido que realizar grupos auxiliares para poder realizar un análisis más correcto de algún tipo de oración más compleja.

Por ejemplo, hemos tenido que crear un grupo nominal compuesto que sería para los casos en los que habría un grupo nominal junto a una oración subordinada de relativo ya que todo el grupo contaría como un sintagma nominal pero que estaría compuesto de otro sintagma nominal dentro.

Este grupo nominal compuesto tendría la siguiente estructura:



Al igual que le pasa al grupo nominal compuesto, se ha tenido que realizar otro grupo auxiliar para los verbos. Este grupo sería exactamente igual que el nominal solo que esta vez estaría compuesto por un sintagma verbal junto a una oración subordinada de relativo.

**Cap´ıtulo 3**

# Mejoras

**3.1. Traducci´on a ingl´es**

## 3.2. Conexio´n entre Prolog y Python

## 3.3. Arbol sint´actico en L´ ATEX

Como mejora, se ha optado por generar una mejor visualizacio´n del ´arbol de sintaxis que la ofrecida por el archivo draw.pl. Para ello, hemos elegido LATEX como lenguaje que nos permitiera realizar el dibujo del ´arbol en tiempo de ejecuci´on dada la sintaxis de Prolog. Las salidas de esta mejora son los diferentes ´arboles que se han ido mostrando a lo largo de este documento.

Para llevar esto a cabo, nos hemos aprovechado de lo explicado previamente en la Seccio´n 3.2, an˜adiendo un archivo llamado Prolog2LaTeX.py que contiene diversas funciones, siendo main la u´nica que necesitamos llamar para generar nuestro dibujo. Como en el programa analizador.py pod´ıamos recoger los resultados de nuestras consultas en Prolog en forma de diccionario, pasaremos esta respuesta a la funcio´n main previamente nombrada y esta hara´ todo el trabajo.

Expliquemos ahora el funcionamiento general de Prolog2LaTeX.py. En este fichero tenemos tres funciones importantes, que se ejecutan en el siguiente orden en la funci´on main: parser, compile, y show.

parser: esta funci´on recibe la respuesta de Prolog en el siguiente formato **PONER EJEMPLO**, y la convierte a la sintaxis que utiliza un objeto \Tree del paquete qtree de LATEX. Como la sintaxis es la siguiente [.P1 [.HIJO1 P1 [.HIJO1 HIJO1 P1] [.HIJO2 HIJO1 P1] ] [.HIJO2 P1] [.HIJO3 P1] ], no tenemos ma´s que sustituir con cuidado par´entesis y comas, por corchetes y espacios. Pondremos este objeto dentro de un documento de clase standalone para que aparezca so´lo el a´rbol. Adem´as, para poder ver de manera resaltada las palabras clave como gn o gadj entre otras, se an˜ade un diccionario de forma que se pueda personalizar el contenido que aparecer´a

CAP´ITULO 3. MEJORAS 3.3. ARBOL SINT´ ACTICO EN L´ ATEX

en su lugar en el a´rbol. Una vez tenemos el archivo de LATEX listo, lo escribimos en un fichero y pasamos a la siguiente funci´on.

compile: como ya tenemos nuestro documento listo, debemos compilarlo para poder visualizarlo. Para ello, invocamos al compilador pdfLATEX con el archivo que parser ha escrito (podr´ıa haberse optado por cualquier otro compilador), y este nos devolvera´ un archivo .pdf con el ´arbol listo para visualizar. Borra tambi´en archivos temporales.

show: al usuario se le ofrece la opcio´n de recibir el a´rbol como documento pdf o como imagen. Para ello, previamente, ha introducido como argumento -pdf o -img. Esta funcio´n se encarga de, en caso de que se quiera imagen llamar a la funcio´n toJPG que convierte el pdf a imagen jpg, y muestra el archivo final mediante el visor de pdf o ima´genes que el usuario haya definido en su sistema operativo como predeterminado.

**Ap´endice A**

# Informacio´n relevante

## A.1. Requisitos y dependencias

Para la realizacio´n de la pra´ctica se han usado una serie de librer´ıas y software externo, por lo que detallamos aqu´ı qu´e es necesario para poder ejecutarla de manera correcta, y qu´e versiones se recomiendan.

**Python**: se ha realizado con la versio´n 3.8.6, sin embargo no se descarta que no funcione en versiones superiores. Previamente se debera´n instalar una serie de dependencias mediante pip:

pip install pyswipl pip install unidecode pip install shutil pip install datetime pip install pdf2image

**Prolog**: debera´ utilizarse **obligatoriamente** la versi´on 8.4.2 de SWI-Prolog, pues con versiones superiores hemos encontrado incompatibilidades con pyswipl.

**LATEX**: deberemos tener instalada una distribuci´on de LATEX en nuestro equipo, en nuestro caso al estar trabajando en Windows hemos optado por MiKTEX (en nuestro caso versio´n 21.2), pero en general nos puede servir cualquiera que contenga el compilador pdfLATEX y que lo tenga an˜adido como variable de entorno. En caso de no ser esto posible o querer usar otro compilador, no habr´ıa ningu´n problema, pero deber´ıa modificarse la l´ınea donde se invoca a este en compile. Adem´as, durante la primera ejecucio´n deberemos permitir la instalacio´n de los paquetes necesarios como qtree.

## A.2. Detalles

Se han cubierto todos los objetivos solicitados, funcionando correctamente las oraciones, sin ningu´n error de implementaci´on.

APENDICE A. INFORMACI´ ON RELEVANTE´ A.2. DETALLES

Respecto al reparto de tareas, la pra´ctica se ha realizado por todos los integrantes del grupo a la vez en una llamada de Discord, aportando cada uno soluciones y alternativas a los problemas que se iban encontrando.

Para finalizar, las referencias consultadas han sido, los apuntes de la asignatura, el [manual de SWI Prolog,](https://www.swi-prolog.org/pldoc/doc_for?object=manual) y [StackOverFlow.](https://stackoverflow.com/)