Universidad Nacional Autónoma de México

Facultad de Ciencias

GENERACION AUTOMATICA DE DESCRIPCIONES BOTANICAS

T E S I S

Que para obtener el título de:

M A T E M A T I C A

P r e s e n t a :

IBONE BROSA CURCO

México, D. F. 1991

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

INDICE.

Introducción 2

Capítulo I. Generación de Textos

1. Lenguaje Natural 5
2. Generación de Textos 5
3. Autores y propuestas 9

Capítulo II. Descripciones Botánicas

1. Descripciones botánicas y su sintaxis 16
2. Sintaxis utilizada en el programa 24

Capítulo III. Estructura Principal del Programa

1. La base de datos 33
2. El plan 36
3. El diccionario 39
4. El atn 44

Capítulo IV. Manual de Utilización del Programa

1. Instalación y funcionamiento del programa 57
2. Ejemplos 60

Conclusiones 69

Bibliografía 77

INTRODUCCION

En los últimos años, los trabajos acerca del tratamiento del lenguaje natural han rebasado los estrechos márgenes de los sistemas de consulta a base de datos, incursionando en nuevas áreas de aplicación potencial como son, el análisis y la generación de textos. La utilidad eventual de dichos sistemas puede ser importante para la manipulación, extracción y utilización de conocimiento manejado en forma escrita.

El presente trabajo fue concebido como un módulo de salida del sistema basado en conocimiento SEMIB (Sistema Experto Multimodal para Identificación Botánica) de ayuda a la identificación de familias de plantas, que se está desarrollando en la Facultad de Ciencias UNAM. Su función consiste en transcribir en lenguaje natural, bajo las normas establecidas por los botánicos, los resultados de la determinación generada por SEMIB. Utilizamos el lenguaje Prolog para la programación de los distintos algoritmos.

El contar con un sistema capaz de generar reportes de descripciones botánicas en forma automática puede ser de gran utilidad tanto para los expertos en esa área como para los que no lo son, esto se debe a que, la redacción de estas descripciones suelen tomar mucho tiempo. Además existe una gran divergencia de opiniones en cuanto al orden, como en la sintaxis que se debe seguir, lo cual dificulta en ocasiones la comprensión de los textos.

En el capítulo I, se hace una breve presentación del Lenguaje Natural. Definimos la generación de textos, así como, las dificultades que ésto representa, y por último los trabajos realizados sobre éste tema y las diferentes propuestas que existen para una mejor redacción y comprensión de textos.

En el capítulo II, se da una introducción a las descripciones botánicas y la sintáxis que se tomará en cuenta a lo largo del trabajo, lo cual es necesario para un mejor entendimiento del proceso de generación. En el capítulo III describirá y analizarán las distintas partes que componen el programa. Finalmente, el capítulo IV, proporcionará al usuario una guía de utilización del programa y mostrará algunos ejemplos.

CAPITULO I

GENERACION DE TEXTOS

I.1 Lenguaje Natural.

Si bien la vista es uno de los sentidos más impresionantes, el lenguaje natural es probablemente el que nos identifica como seres humanos.

El lenguaje lo utilizamos para comunicarnos, ya sea de forma hablada o escrita. Existen reglas y definiciones del propio lenguaje y según estas reglas podemos saber si una oración pertenece al lenguaje y si tiene significado dentro del mismo.

Para hacer una buena formalización del lenguaje necesitamos conocer a fondo su sintaxis y su semántica.

Entender el lenguaje natural es difícil, requiere del conocimiento lingüístico de un lenguaje en particular y del conocimiento del mundo relativo al tema en discusión.

Escribir lenguaje natural es una invención más reciente y todavía juega un papel menos central que el lenguaje hablado, pero el entendimiento del lenguaje escrito es más fácil que el hablado, dado que, en éste último necesitamos conocimiento adicional para manejar el "ruido" de las ambigüedades de la señal auditiva.

I.2 Generación de textos.

Si bien la generación automática de textos surgió desde los inicios de la Lingüística Computacional con los trabajos en traducción automática de los años cincuenta, su estudio fue relegado a segundo plano dada la complejidad del tema y de la necesidad de dominar primero la fase de comprensión y análisis del lenguaje natural.

A fines de los años setenta y durante toda la década siguiente el tema de la generación, volvió a atraer la atención de los investigadores del área.

El proceso de generación de textos puede ser visto exactamente como el opuesto del entendimiento del lenguaje. Una estructura que representa alguna información debe ser "mapeada" dentro de una cadena válida en el lenguaje [illegible]eado.

Los investigadores están menos familiarizados con los problemas de la generación del lenguaje. Aunque existen investigaciones que sugieren que la misma información puede usarse tanto para la interpretación como para la generación (Kay 1979, Wilensky 1981, Winograd 1983).

Un generador debe ser capaz de construir la mejor producción para una situación dada, seleccionando entre muchas posibles opciones que involucran un amplio margen de fuentes de conocimiento.

Un analizador no suele preguntarse porqué se eligió mejor una forma que otra. Ahí es donde la investigación en interpretación debe describir las limitaciones de las opciones posibles para poder determinar más eficientemente la opción escogida, la investigación en generación deberá especificar porque una opción es mejor que otras en diversas situaciones.

Las elecciones que un generador de lenguaje debe encarar incluyen aquellas opciones que involucran el contenido y la forma textual de lo que se va a decir y las elecciones en la transformación del mensaje así determinado en lenguaje natural :

a) Qué información destacar

b) Cuál es la información más pertinente

c) Cómo iniciar el discurso o texto

d) Cómo ordenar sus partes

e) Cómo concluir el texto

f) Qué palabras usar

g) Cómo agruparlas en oraciones.

Si se ha de generar texto conectado (y no solamente oraciones simples), los problemas de estructura del texto y la coherencia del mismo son particularmente importantes. Se necesita la habilidad para determinar el orden de las oraciones de un texto. El escritor planea el marco general de referencia, a partir del cual se producen las oraciones individuales.

El proceso de generación puede concebirse como el resultado de la interacción de varias componentes :

1. Componente estratégica. Recibe una meta global compuesta por dos sub_componentes cuya interacción mutua es crucial para la elección de la información relevante y de la estrategia de organización de texto:

   i) La componente semántica. Que elige la información relevante.

   ii) La componente estructural. Que elige la estrategia organizacional.

2. Componente táctica. Quien genera el texto en lenguaje natural.

La mayoría de los trabajos en generación se han enfocado hacia la componente táctica. Estos van, desde la traducción directa de una representación formal subyacente hasta el desarrollo y la

representación de criterios para tomar decisiones acerca del vocabulario como parte de un diccionario. Estos trabajos tratan normalmente de la generación de oraciones simples.

Sobre la componente estratégica se ha trabajado en tres problemas principalmente :

a) Conocimiento necesario para la generación

b) La planeación para determinar un acto de habla apropiado

c) Una organización textual.

La generación de texto se basa en dos hipótesis fundamentales sobre producción de un texto :

1) No tienen porqué coincidir el cómo se almacena la información de memoria y en cómo una persona describe esa información.

2) La gente tiene nociones preconcebidas acerca de la forma en la cual se puede describir la información (implica la existencia de uno o más principios de organización de textos).

Otro punto importante, es el uso de lo que se llama la focalización, fenómeno común en todo tipo de discursos.

Todos, ya sea conciente o inconcientemente, centramos nuestra atención en varios conceptos u objetos a lo largo del proceso de lectura, escritura, al hablar o escuchar. En todas estas modalidades, el fenómeno de focalización aparece en múltiple niveles del discurso.

El uso de la focalización facilita el procesamiento de los participantes en una conversación y proporciona restricciones sobre las posibilidades de lo que puede decirse. Las restricciones de focalización abarcan a toda la base de conocimiento, produciendo un subconjunto que contiene los rubros sobre los cuales se puede hablar.

El uso de focalización proporciona un método manejable computacionalmente para producir un discurso coherente y asegura la conectividad del discurso.

I.3 Autores y propuestas.

Los primeros sistemas de generación contaban con :

1) El uso de texto archivado. Requiere que el diseñador del sistema enumere todas las preguntas que el sistema debe ser capaz de responder y escribir las respuestas a estas preguntas a mano y así archivarlas como un todo y recuperarlas cuando se necesite.

2) "Patrones". Son frases construidas por el diseñador con unas "ranuras" que pueden ser instanciadas con palabras y frases diferentes, dependiendo del contexto. Un problema con este tipo de diseño es que la yuxtaposición de frases completas frecuentemente resulta torpe, o bien, un texto ilegal.

Ambos métodos son útiles en situaciones donde se requiere de un rango limitado de generación porque el sistema puede ser tan elocuente como el diseñador al hablar.

Investigaciones posteriores en generación del lenguaje natural, que abarcan algunos de estos temas, pueden dividirse dentro de las siguientes áreas de investigación :

a) Componentes tácticas.

b) Planeación y generación.

c) Conocimiento necesario para generación.

Uno de los primeros sistemas de generación sobre componentes tácticas es el de Simmons y Slocum(1972) fue utilizado para generar

oraciones en inglés para redes semánticas. Más tarde Goldman (1975) desarrolló un sistema para redes de dependencia conceptual. Más recientemente McDonald (1980), su trabajo abarca problemas de la componente lingüística, que consiste en tres módulos diferentes: el diccionario, la gramática y el controlador.

Uno de los primeros artículos en manejar la necesidad de la planificación en la generación de textos fue el de Cohen y Perrault (1979). En él los autores proponen una metodología para integrar los actos del habla en un sistema de planificación, de modo de conectar los objetivos del sistema con el lenguaje que genera. Su propuesta consiste en manejar los actos del habla como operadores, dentro de un sistema de planificación. En ese trabajo no consideraron los detalles de la generación real de texto contentándose en producir listas de actos de habla que llevarían a cabo el objetivo final de comunicación.

Otro trabajo importante en esta misma dirección fue el de Appelt (1985). En él, el autor retoma las ideas de Cohen y Perrault examinando la interacción entre planeación y generación en todos los estados del proceso de generación. Estudia la forma de cómo generar el texto final a partir de las descripciones y de los actos del habla.

Una linea alterna a la anterior está representada por el trabajo de McKeown (1985), en el que el autor considera la generación de textos más extensos al mismo tiempo que reduce la cobertura de los objetivos lingüísticos de los textos a generar. McKeown se plantea el manejo de estrategias con tres metas de comunicación: definir, comparar y describir objetos almacenados en una base de datos. Para ello utiliza una representación de tipo "guión" para organizar el

discurso, dejando a un lado el manejo explícito de los actos del habla.

En un artículo reciente, E. Hovy (1990), compara las aproximaciones de Appelt y de McKeown desde el punto de vista del nivel de integración de las fases de planificación y de generación. La estrategia del primero es la de integrar ambas fases dentro de un mismo proceso continuo, donde el planificador maneja las restricciones sintácticas de la misma forma que trata todas las demás restricciones, tales como el punto focal o la falta de conocimiento acerca del oyente; la única diferencia en este caso es que las restricciones sintácticas tienden a aparecer más tarde en el proceso de planificación. La estrategia del segundo consiste en mantener separadas ambas fases, llevándose primero a cabo la de planificación y luego la de generación. En este caso no hay lugar para replanificaciones en el curso del proceso de generación; esto lleva a los planificadores a no manejar ningún tipo de información sintáctica, sino solamente información sobre los tópicos escogidos y el orden de las oraciones.

Hovy, a su vez propone una tercera alternativa que denomina planificación de compromiso limitado, que consiste en retrasar la planificación hasta que lo necesite el proceso de generación. En este esquema, el planificador sólo necesita ensamblar un conjunto parcial suficiente de instrucciones del generador para que la componente de realización empiece a trabajar, y pueda después, continuar a planear conforme la componente de realización requiera de nuevas instrucciones; de esta forma se mantienen separadas las tareas de

planeación y de realización, lo que permite al módulo de planeación tomar en cuenta oportunidades o problemas sintácticos no previstos.

De los trabajos antes citados los de Cohen-Perrault y Appelt se sitúan en la problemática de generación de párrafos en condiciones de un diálogo entre dos interlocutores; ésto les lleva a considerar una planificación muy sofisticada en términos de actos de habla. McKeown, por su parte se sitúa en una perspectiva muy diferente, lo que se propone es traducir el conocimiento almacenado implícitamente en una base de datos según las necesidades del usuario, ya sea generar una descripción, una definición o una comparación. En este caso es posible asignar a cada uno de los objetivos linguísticos una estructura de plan prototípico sin necesidad de tomar en cuenta los problemas de la intercomunicación entre dos agentes interactuantes.

Otras aproximaciones a la investigación de la generación del lenguaje natural han enfatizado el tipo de conocimiento necesario para generar descripciones apropiadas.

Swartout (1981) examinó este problema en el contexto de un sistema de consulta médica. Su principal preocupación, sin embargo, fue en la representación del conocimiento y no en el proceso de generación.

Meehan (1977) estaba también interesado en el problema del conocimiento necesario para generación como parte de su trabajo en producir historias cortas sobre planes de personas para ejecutar metas.

Ambos trabajos abarcan temas importantes en la representación del conocimiento, reconociendo que los sistemas son limitados en lo que pueden decir.

Por el lado de la generación de texto encontramos que Mann y Moore (1981) fueron dos de los primeros en estar interesados en los problemas que surgen en la generación de cadenas de "oraciones múltiples".

Un trabajo anterior en generación de texto, es el de Goguen, Linde y Weiner (1980). Ellos también están interesados en la estructura del texto. Proponen una gramática de interpretación la cual indica qué orden de preposiciones es posible, captura la jerarquía de la estructura del texto y el núcleo de la gramática. Más aún , también incorporaron la noción de foco de atención.

Stevens y Steinberg (1981), hacen un análisis de texto para instrucciones sobre plantas de propulsión.

Forbus (1981), proponen un sistema que usa proceso de simulación cualitativa para proporcionar explicaciones de este tipo.

Jensen et al (1981), propuso el desarrollo de un sistema capaz de generar cartas de negocios standard. Están particularmente interesados en la generación de texto coherente y sugieren usar predicados como causa y efecto para ayudar en la selección de conectivos textuales apropiados. Asumen, sin embargo, que el contenido de las cartas y la asignación de predicados han sido ya determinados.

Los trabajos realizados hasta el momento en el campo de generación nos muestran, por un lado, lo complejo que puede llegar a ser el manejo del lenguaje natural y por el otro, nos muestran que

esto es posible siempre y cuando se trabaje en un dominio muy delimitado y específico para que el proceso de lenguaje natural resulte satisfactorio.

CAPITULO II

DESCRIPCIONES BOTANICAS

En la primera parte de este capítulo daremos una explicación de las descripciones botánicas y sus diferencias en cuanto al estilo, así como, algunos ejemplos de su sintaxis. En la segunda parte definiremos una sintaxis y el orden de redacción, para el funcionamiento del programa.

II.1 Descripciones botánicas y su sintaxis.

La clasificación de las plantas es en gran medida subjetiva. Pocos estudiosos de la ciencia vegetal han alcanzado conclusiones unificadas en cuanto a la clasificación de los miembros del reino Plantae. Además la clasificación se encuentra frecuentemente alterada por el descubrimiento de nuevos hechos.

Para que sus esfuerzos sean reconocidos, todo clasificador tiene la obligación de delimitar categorías taxonómicas, llamadas taxa, de acuerdo a la legislación del Congreso Internacional de Botánica (Stafleu, 1972) que establece que: cualquier planta individual pertenece a una especie, toda especie pertenece a un género, todo género a una familia, toda familia a un orden, todo orden a una clase y toda clase a una división.

Sin embargo, hay que hacer notar, que existen distintos sistemas de clasificación en la actualidad, lo cual demuestra elocuentemente la divergencia de opiniones en la interpretación de los datos que relacionan evolutivamente las plantas entre sí. Estas divergencias de opiniones se dan por la diversidad y complejidad evolutiva de las plantas (Bold, Alexopoulos y Delevoyras 1980).

Existen también distintas formas de redactar la información obtenida de la clasificación, el orden de los datos y el estilo de redacción varian según el autor, para algunos es más importante nombrar unas características en primera instancia, mientras que otros las nombran en una segunda o bien no las nombran.

La descripción de la flor casi siempre coincide en ser la más extensa, esto es porque sus atributos son de suma importancia para la clasificación de la planta en una cierta familia.

Para hacer más clara esta diferencia de estilos, veremos como ejemplo a la familia *Compositae*, en dos versiones distintas.

La descripción de la flor no se tomará en cuenta por la razón antes mencionada (sumamente extensa), pero vale la pena aclarar que existe una diferencia muy grande de una versión a otra. Esta diferencia es basicamente que la primera es menos detallada que la segunda y esta última da información sobre inflorescencia mientras que la primera ni siquiera nombra ese sujeto.

Familia Compositae.

(Por Jerzy Rzedowski)

> Plantas herbáceas o arbustivas, rara vez arbóreas o trepadoras; hojas opuestas o alternas, en ocasiones todas radicales, sin estípulas; flores ..... ; fruto en forma de aquenio, que a menudo lleva en su extremo superior el vilano; semilla sin endosperma. Las cabezuelas ..... .

Familia Compositae.

(Por Jose Luis Villaseñor Rios)

Hierbas anuales o perennes, arbustos, bejucos o árboles, en ocasiones con jugo lechoso, glabros o variadamente pubescentes o glandulares, dioicos (rara vez poligamo-dioicos), monoicos o bisexuales. Tallos generalmente rollizos, a veces alados o inclusive aplanados. Hojas alternas, opuestas o verticiladas, algunas veces basales, simples, pinnadas o palmadamente lobuladas, divididas o compuestas, enteras o diversamente dentadas, pecioladas o sésiles, la lamina en ocasiones decurrente, auriculada o envainante, en algunas formas xerófitas aciculiforme o reducida a escamas o espinas, sin estípulas, aunque a veces presentes unas pseudoestípulas. Inflorescencia ..... . Cabezuelas ..... . Fruto un aquenio (cipsela), con una sola semilla y con un embrión recto y sin endospermo; en algunas especies el fruto es drupáceo o una baya, inclusive un atrículo por la fusión del aquenio con la pálea, las filarias u otra parte de la cabezuela ; el pericarpio por lo general rígido. Vilano coronado al aquenio, persistente o deciduo o ausente, constituido por cerdas, aristas o escamas, en ocasiones con un carpóforo conspicuo.

La primera descripción como ya mencionamos anteriormente es más compacta, no solo en la descripción de flor sino en la descripción en general.

La primera diferencia que encontramos es que la primera descripción empieza con la forma de vida de la planta (herbáceas o arbustivas, rara vez arbóreas o trepadoras) mientras que en la segunda las primeras características que nombra son sobre la duración de la planta (anuales o perennes) y a continuación hace referencia a la forma de vida.

La forma de referirse a la planta en cuestión también cambia. La primera es de la forma: plantas herbáceas o arbustivas en cuyo caso el sujeto es la palabra plantas, mientras que en el segundo caso el sujeto pasa a ser las palabras hierbas, arbustos, bejucos o árboles.

La segunda definición describe a continuación de la forma de vida, otros atributos como son el indumento y el sexo de la planta. La primera no describe estas características.

El tipo de tallo de la planta se menciona en la segunda descripción y en la primera no.

Claramente se puede observar que el orden de redacción sobre los atributos no es el mismo, nos encontramos que en el primer ejemplo, las características de las cabezuelas se describen después de haber descrito el fruto, mientras que el segundo lo describe antes. Esto es a lo que nos referimos con diferente peso a las características. Para unos autores es mas importante destacar ciertos sujetos o atributos, como lo vemos en el ejemplo anterior.

En cuanto a la sintaxis, existe también una gran divergencia. En el primer ejemplo vemos claramente que la información sobre los sujetos es separada por medio de un punto y coma (;).

ej/

Plantas herbáceas ..... ; hojas ....

La segunda en cambio separa estos sujetos por medio de un punto (.).

ej/

Hierbas anuales ... . Tallo generalmente rollizos ... .

Por lo general el orden en el que se mencionan a los sujetos es bastante convencional, este orden es: Plantas, Tallos, Inflorescencia, Flor y Fruto. El no nombrarlos no implica que no se siga este orden, como es el caso de Inflorescencia y Tallo en el primer ejemplo.

Por otra parte el lenguaje que se utiliza en las descripciones botánicas es un lenguaje muy particular. La redacción que se utiliza es principalmente a base de frases incompletas.

definición : frases incompletas.- No todos los elementos aparecen explícitos en la frase. Se omiten:

a) Los sujetos suficientemente expresados en las desinencias verbales.

b) El verbo copulativo, no porque se sobrentienda, sino porque el sujeto o el predicado llevan en sí la esencia de la frase.

c) El predicado y algunos elementos modificadores, como la preposición, artículo, adjetivo etc.

Tomando en cuenta esta definición, podemos a continuación hacer un análisis de las elipsis (palabras que se omiten) hechas en las descripciones. Para esto es necesario mostrar primero como lo escribiriamos sin omitir palabras y segundo la descripción tal y como la escribirían los botánicos. Tomaremos como ejemplo la Familia *Flacourtiaceae* :

(1) La familia flacourtiaceae consta de árboles o arbustos perennifolios. El tallo es glabro, pubescente tomentoso o viloso y algunas veces con espinas. Sus hojas son persistentes o caedizas; su disposición es alterna, raramente opuesta o verticilada; las hojas tienen una consistencia membranosa, coriácea o cartácea y generalmente presentan puntos pelúcidos y lineas; son a su vez penninervadas o reticuladas; el indumento es glabro, pubescente, tomentoso o velutinoso; el ápice es agudo o acuminado, el margen es glandular crenado, dentado o aserrado; ...

(2) (Por Candolle A.P 1824)

Arboles o arbustos perennifolios. Tallo glabro, pubescente, tomentoso o viloso, algunas veces con espinas. Hojas persistentes o caedizas; alternas, raramente opuestas o verticiladas; membranosas, coriáceas o cartáceas; generalmente presentando puntos pelúcidos y lineas;penninervadas o reticuladas;glabras pubescentes, tomentosas o velutinosas; con el ápice agudo o acuminado; márgen glandular crenado, dentado o aserrado; ...

Las elipsis se basan principalmente en los puntos descritos en la definición anterior:

a.- El sujeto es nombrado solamente una vez, es decir la primera frase consta del sujeto y las siguientes hacen una elipsis.

b.- Cuando el verbo copulativo es el verbo "ser" casi siempre se hace una elipsis, no se presentan muchos verbos en el transcurso de descripción pero en caso de presentarse, por lo general se escribe en gerundio.

ej/

presentando puntos pelucidos y lineas.

c.- Los artículos se nombran a veces, esto depende del estilo propio del autor.

ej/

... con el ápice agudo o acuminado; margen glandular
crenado, dentado o aserrado; ...

El atributo ápice está antecedido por el artículo "el" a diferencia de margen glandular crenado que no lleva un artículo antes del atributo margen.

Decimos que depende del estilo del autor, dado que, podríamos alterar los artículos de manera que quedara:

... ápice agudo o acuminado; con el margen glandular
crenado, dentado o aserrado; ...

o bien, haciendo una elipsis total de los artículos y la descripción tendría la misma información y la misma congruencia en los tres casos.

El caso de los sujetos lo podriamos ver como una regla general, en la cual no se antecede un artículo.

ej/

Hojas persistentes o caedizas; ...

en vez de:

Las hojas son persistentes o caedizas.

ó

Sus hojas son persistentes o caedizas.

Para esto existe una razón. Si le antecedemos un artículo al sujeto, necesariamente tendría que llevar un verbo, lo cual presenta una incongruencia con el estilo propio de las descripciones botánicas.

Se hace una elipsis de los atributos, es decir por lo general se escriben los valores del atributo sin decir explícitamente a que atributo nos estamos refiriendo.

ej/

... membranosas coriáceas o cartáceas; ...

sin nombrar al atributo *consistencia*, al cual nos estamos refiriendo.

Esto depende de la información que el valor tenga por sí solo, es decir, en el caso del atributo *ápice* es necesario nombrarlo dado que los valores *agudo o acuminado* no dan la información suficiente para saber que estamos hablando de ese atributo, en cambio los valores *membranosas, coriáceas o cartáceas* llevan implícitamente el nombre del atributo *consistencia*, es decir, la información que tienen por sí mismas es suficiente y necesaria para ser entendidas.

Esta divergencia de opiniones sobre el orden de los atributos y sobre la sintaxis de redacción, no serán contempladas en el programa.

Para nuestros fines tomaremos un orden y una sintaxis establecida.

II.2 Sintaxis utilizada en el progama.

Como ya mencionamos en la primera parte de este capítulo el orden, la sintaxis e información varian en forma muy notoria, por lo cual nos es necesario definir tanto un orden como una sintaxis en general.

A pesar, de que el orden para escribir las descripciones cambia según el autor, existe una cierta tendencia por un orden en particular, mismo orden que tomaremos en cuenta para nuestros fines y que se presenta a continuación :

Plantas	(forma de vida, hábito, habitat, latex, jugo_acuoso, resina, zarcillos, espinas, duración, textura).
Tallo	(tipo, posición).
Hojas	(presencia, duración, disposición, tipo textura, estípulas(presencia, tamaño, duración), venación, tricomas, condición_ pelúcida, peciolo, indumento, ápice, márgenes, base).

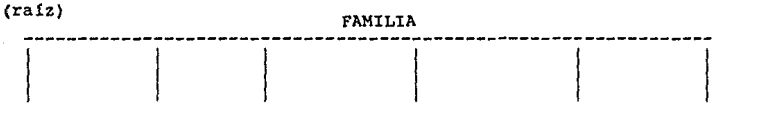
Inflorescencia	(tipo, posición).
Flor	(sexo, simetría, perianto (cáliz, corola) androceo (estambres (número, fusión, inserción, anteras(inserción, dehiscencia, orientación)), estaminodios), gineceo (ginóforo, carpelos, fusión, ovario, estilo, placentación, hojas_ carpelares).
Fruto	(consistencia, tipo, dehiscencia, semillas (cantidad, alas, plumas, arilo, estigma, partes), embrión(tipo,endospermo, cotiledones)).

Llamaremos a Plantas, Tallos, Hojas, Inflorescencia, Flor y Fruto *temas* o *sujetos principales*.

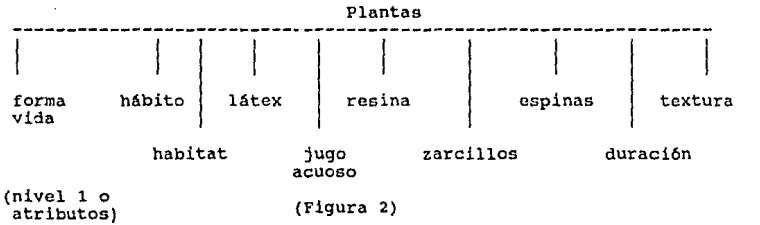
Las características supeditadas a estos temas, (forma de vida, habito, tipo, disposicion, textura etc..) los llamaremos *atributos*.

Por último las características supeditadas a estos atributos, (cáliz, corola, estambres etc..) los llamaremos *sub_atributos*.

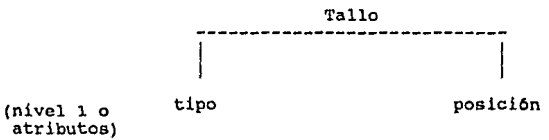
Para un mejor entendimiento representaremos esta información en un árbol:



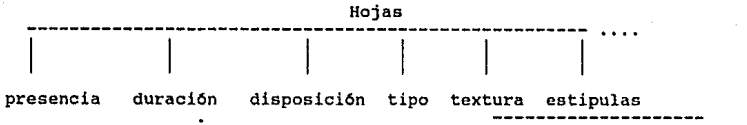
(Figura 1)



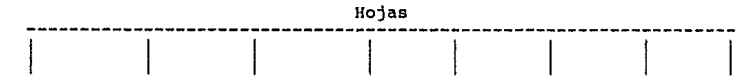
(Figura 2)



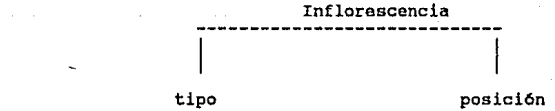
(Figura 3)



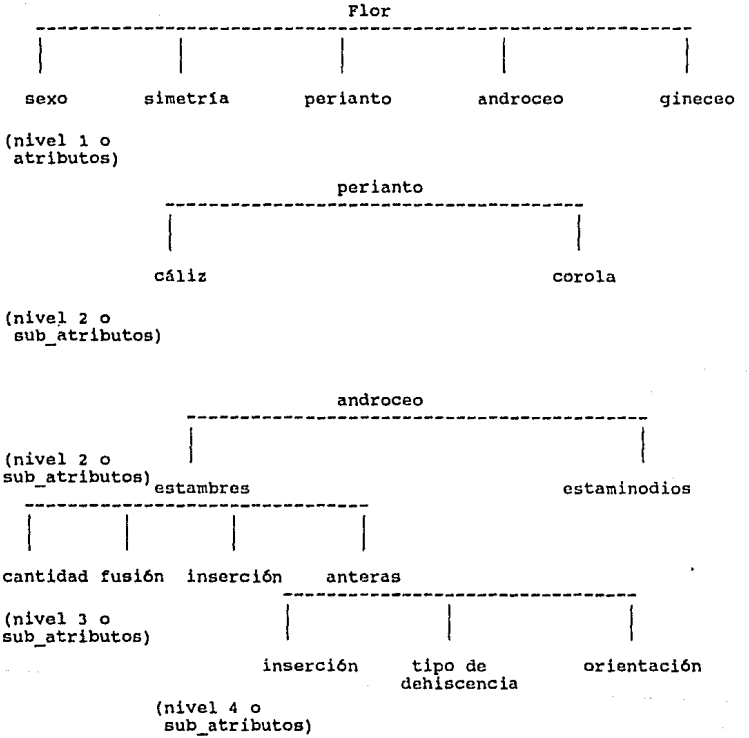
(continuación...)

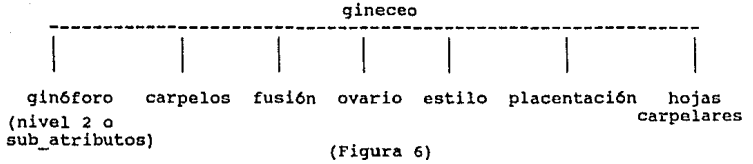


(Figura 4)

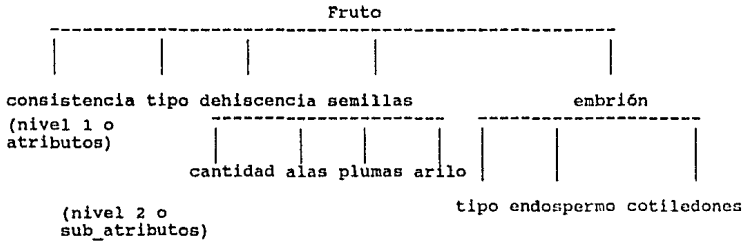


(Figura 5)


Flor
sexo
simetría
perianto
androceo
gineceo
(nivel 1 o atributos)
perianto
cáliz
corola
(nivel 2 o sub_atributos)
androceo
(nivel 2 o sub_atributos)
estambres
estaminodios
cantidad
fusión
inserción
anteras
(nivel 3 o sub_atributos)
inserción
tipo de dehiscencia
orientación
(nivel 4 o sub_atributos)



(Figura 6)



(Figura 7)

La descripción debe empezar con el nombre del sujeto al que nos vamos a referir y la primera letra en mayúscula.

El nombre del sujeto deberá ser nombrado solo una vez, al iniciar la oración.

Después de haber nombrado el sujeto empezaremos a dar su información siguiendo el orden establecido de sus atributos. Este orden se puede seguir facilmente haciendo el recorrido del árbol de izquierda a derecha.

Cada atributo contiene diferentes valores dependiendo de la familia a la cual nos estamos refiriendo. Estos valores son los que nos dan la información o características del atributo en cuestión

ej/

Plantas perennes.

La palabra *perenne* en este caso es el valor que toma el atributo *forma de vida* del sujeto *planta*.

Cada sujeto debe formar una oración por sí sola separada por un punto (.).

Cada atributo debe ir separado por un punto y coma (;) y por último, cada sub-atributo debe ir separado por una coma (,).

ej/

Plantas perennes.

Tallo glabro.

Hojas caedizas; alternas , opuestas o verticiladas.

Flores unisexuales; estambres libres, pocos y epipétalos.

En todos los casos se ve claramente que forman una oración por sí solas y que van separadas por un punto.

En el caso de hojas lo primero que decimos es que es *caediza*, que es un valor del atributo *duración*, sigue con los valores *alternas*, *verticiladas* que son valores al atributo *disposición* por lo que al cambiar de atributos es necesaria la separación por medio de un punto y coma.

Por último en el caso de flores, el punto y coma que divide al valor *unisexuales* del valor *libres* es el cambio de atributos del atributo *sexo* al atributo *estambres*. Las comas, separan a los sub_atributos; en este caso, a los sub_atributos *fusión*, *cantidad* e *inserción* respectivamente, terminando con una conjunción.

Cada familia puede constar de varios "géneros" y "especies". Cada género y especie tienen diferentes valores. Si queremos hablar de una familia en general tendremos que decir todos los valores que identifican a esa familia.

Para decir todos los valores que tenemos sobre un atributo la sintaxis que se debe usar es la misma que se muestra en el ejemplo anterior: los valores van separados por comas y la última característica con una disyunción (o).

CAPITULO III

ESTRUCTURA PRINCIPAL DEL PROGRAMA

En este capítulo mostraremos las dos fases que utiliza el programa para escribir las descripciones botánicas. La primera fase consiste en la planificación de la información. La segunda, es la generación de esta información.

Para hacer el plan correspondiente a una cierta familia, es necesario contar con una base de datos. Para la generación, es necesario la definición de un diccionario.

III.1 La Base de Datos.

La base de datos deberá tener toda la información necesaria para una buena descripción de la familia. Mientras más valores a atributos y más atributos tenga la base de datos, más detallada también será la descripción final.

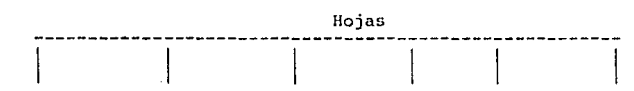
Esta base de datos podrá estar dada por el usuario o bien el programa constará de algunos datos de ciertas familias. Esto nos dará la facilidad de generar descripciones más completas, es decir, características de la familia que quizás el usuario no tenga la posibilidad de saber.

La base de datos deberá estar dada por el usuario con la siguiente sintaxis:

a) Nombre del atributo
b) Valores del atributo los cuales deberán estar representados en una lista y en caso necesario poner el adverbio correspondiente al valor.

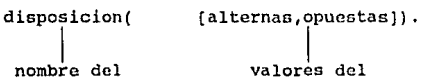
Estos datos van a estar representados internamente en una lista. (Las listas en PROLOG se representan con "corchetes cuadrados" al empezar y al finalizar la lista. Los valores van separados por comas)
ej/

Tomando en cuenta las ramas del árbol del sujeto *hojas*:



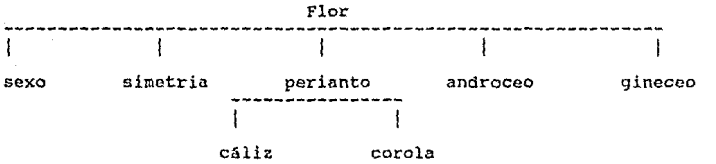
(Figura 8)

Sabemos que la profundidad de estas ramas corresponden a los atributos de primer nivel, lo cual correspondería a escribir la información de la base de datos de la siguiente forma:



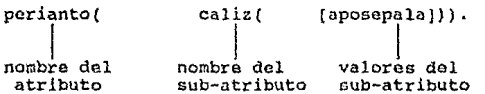
En caso de que existan sub-atributos o atributos de segundo nivel deberán seguir la misma sintaxis, con la diferencia de que la profundidad del árbol es un nivel más abajo, lo que significa que tomaremos en cuenta dos atributos.

ej\
Tomando en cuenta el sujeto *Flor*:



(Figura 9)

La forma de escribirlo en la base de datos quedaría entonces:

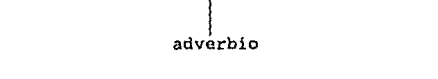


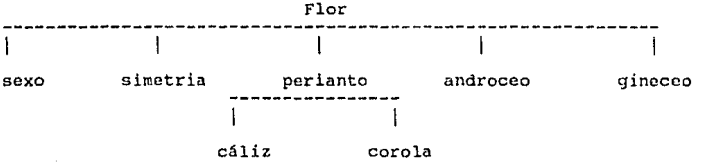
La sintaxis de los siguientes sub_atributos o atributos de tercer y cuarto nivel siguen la misma sintaxis, contemplando simplemente el nivel de profundidad y así saber cuántos atributos se encuentran involucrados.

En caso de querer tener en la descripción frases como:

" Hojas alternas, raramente opuestas o verticiladas ...."

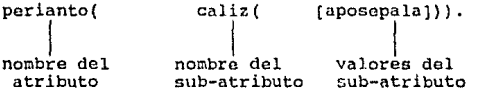
en donde se encuentra un adverbio (raramente) calificando a un valor, será necesario especificarlo en la base de datos de la siguiente forma:





(Figura 9)

La forma de escribirlo en la base de datos quedaría entonces:

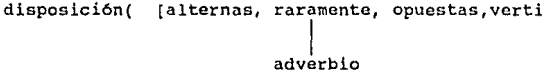


La sintaxis de los siguientes sub_atributos o atributos de tercer y cuarto nivel siguen la misma sintaxis, contemplando simplemente el nivel de profundidad y así saber cuántos atributos se encuentran involucrados.

En caso de querer tener en la descripción frases como:

" Hojas alternas, raramente opuestas o verticiladas ...."

en donde se encuentra un adverbio (raramente) calificando a un valor, será necesario especificarlo en la base de datos de la siguiente forma:



donde el adverbio debe de escribirse antes del valor al que nos estamos refiriendo.

La base de datos no necesariamente debe estar ordenada es decir no tiene que seguir el orden que establecimos en el capítulo anterior de las descripciones botánicas, dado que el sistema tiene una búsqueda secuencial.

III.2 El Plan.

Un plan de descripciones es el encargado de organizar la información que tenemos de la base de datos. Esta organización consiste en ordenar la información según el orden que establecimos en el capitulo anterior, creando así, una estructura que facilitará la fase de generación.

El usuario tiene la posibilidad de escoger, dentro de la base de datos establecida en el programa, la redacción particular de uno de estos temas o la información general de la familia. Existen entonces seis planes particulares para cada sujeto principal: (Plantas, Tallo, Hojas, Inflorescencia, Flor y Fruto) y un plan general, el cual llama a cada plan en particular.

Cada tema en particular contiene dos etapas. Para poder entender esto más claro se seguirán las etapas de un tema en particular (plantas). Cada tema tiene que seguir las mismas etapas para sus diferentes atributos:

1) Ver si en la base de datos existe alguna información sobre el tema (plantas) esto es, si existe algún predicado de la forma:

plantas(Atributo([Valores])).

Tenemos dos casos:

i) En caso de que no encontremos esta información, esto es, no exista un predicado con esa forma, se crean unas variables que se instancían con la constante "nil". Esta constante significa que la información es nula. La cantidad de variables dependerá de la cantidad de atributos que tenga el sujeto al que nos estamos refiriendo.

ej\

A1 = nil , A2 = nil , A3 = nil , A4 = nil ... A10 = nil, donde

A1 son los valores del atributo *forma de vida*,

A2 son los valores del atributo *hábito*,

A3 son los valores del atributo *habitat*, etc .

ii) Cuando haya información sobre el tema, pasamos a la siguiente etapa.

2) Ver qué atributos lo hacen verdadero; esto es, si existe algún atributo llamado forma de vida, hábito, habitat, latex, jugo_acuoso, resina, zarcillos, espinas, duración, textura, que es el mismo orden que establecimos.

Tenemos dos casos:

i) Que exista información de todos los atributos. En tal caso el plan del tema (plantas) será completo así como la información que se dará de ellos. Las variables en este caso se instancían con los valores correspondientes a cada atributo.

ii) Que exista información sólo en algunos atributos. En este caso el plan no será completo y las variables se instanciarán con el valor correspondiente al atributo o con la constante "nil" dependiendo de la información que se tenga.

Si por ejemplo la información que no tenemos es la de hábito que es la que ocupa el segundo lugar, las variables quedarán de la siguiente forma:

A1 = Valores del atributo *forma de vida*,

A2 = nil,

A3 = Valores del atributo *habitat* etc... .

Al terminar este proceso para cada sujeto principal o principal creamos un predicado y dos listas:

L1) El predicado llamado "sujetos" el cual contiene el sujeto en cuestión :

```
sujetos(plantas).
```

L2) La lista llamada " atributos" la cual está constituida de todos los atributos correspondientes al sujeto (plantas) :

```
atributos([ forma_de_vida, habito, habitat, latex, jugo_acuoso,
            resina, zarcillos, espinas, duración, textura ]).
```

L3) La lista llamada "valores" la cual está constituida de los valores de cada atributo :

```
valores([ A1, A2, A3,...., A10 ]).
```

Esta lista como ya hemos visto contiene la información que el usuario haya dado.

El orden de los valores es el orden establecido por el usuario, mismo orden que se mantendrá durante todo el proceso.

Para el plan general se siguen los diferentes planes particulares creando una lista general de sujetos, una lista general de atributos y una lista general de valores, esto es, se hace una concatenación de las listas particulares de cada plan.

III.3 El diccionario.

En el diccionario se almacena el conjunto de rasgos semánticos necesarios para la generación de las descripciones botánicas, así como, información sintáctica y estilo. Los rasgos semánticos se manejan en la parte correspondiente a los valores de cada atributo, los rasgos sintácticos, en los sujetos principales o temas, así como, en los atributos, sub_atributos y en los artículos. Por último el estilo se maneja en la parte correspondiente a los atributos, sub_atributos y en la parte de valores de cada atributo. De esta manera podemos dividir la información del diccionario en cuatro partes principales :

1) Información sobre sujetos o temas principales
2) Información de los atributos y sub_atributos
3) Información de los valores de cada atributo.
4) Información de los artículos.

y de dos listas :

a) Lista de adverbios
b) Lista de vocales

1) La información de los temas o sujetos principales son predicados que se componen de tres argumentos :

a) Nombre del sujeto

b) Género

c) Número

ej/

```
sujeto(planta,fem,sing).
sujeto(tallo,masc,sing).
```

2) La información de los atributos y sub_atributos son predicados que tienen cuatro argumentos :

a) Nombre del atributo o sub_atributo

b) Género,

c) Número

d) "Nombrar" el cual se refiere al estilo de las descripciones. Contiene a su vez dos argumentos:

i) Nombrarlo: en caso de que ese atributo se tenga que decir se pone la constante 'si', en caso contrario la constante 'no'

ii) Prep: que en caso de nombrarlo, hay que especificar con qué preposición se nombra. Si no se nombra hay que poner la constante 'no'.

ej/

```
atr(nombre,género,número,nombrar(Nombrarlo,Prep)).
atr(presencia,fem,sing,nombrar(no,no)).
atr(margen,masc,sing,nombrar(si,no)).
atr(apice,masc,sing,nombrar(si,con)).
```

Cuando tengamos que hablar de los atributos *presencia*, *margen* y *ápice* de la hoja teniendo esta información nos quedará una oración de este tipo:

" Hojas escamosas; margen entero; con el apice agudo. "

Esta información es necesaria dado que si no nombramos al *margen* y al *ápice* al decir sus valores no quedaría claro a que nos estamos refiriendo; en cambio el valor escamosas nos dice por sí sola que se refiere al atributo presencia.

El artículo (el) que acompaña al atributo ápice se explicará más adelante cuando se de la explicación de los procedimientos que sigue la generación (atn).

Los atributos del segundo nivel o sub_atributos, tienen casi la misma representación que los atributos a diferencia que son atributos compuestos: atr(sub_atributo), los cuales se descomponen en dos atributos sencillos y cada uno con sus rasgos sintácticos y su estilo correspondiente.

ej\

```
sub_atr(semillas(cantidad),semillas,fem,plural,nombrar(si,no),
                                cantidad,fem,sing, nombrar(no,no)) .
```

Los atributos del tercer y cuarto nivel o sub_atributos1 y sub_atributos2 respectivamente, tienen la misma representación que los anteriores, la diferencia es que se descomponen en tres y cuatro atributos sencillos y cada uno con su información correspondiente.

3) La información de los valores para cada atributo, son predicados que estan constituidos de tres argumentos :

i) Nombre del atributo.

ii) Una lista la cual contiene todos los valores posibles del atributo. En este argumento se manejan los rasgos semánticos de las descripciones.

iii) Un argumento llamado prep(Prep) en el cual se especifica la preposición que se debe de utilizar al nombrar los valores.

ej/

```
val(tipoi,[fasciculos,cimas,racimos],prep(en)).
```

El argumento *tipoi* se refiere al nombre del atributo *tipo* del sujeto *inflorescencia*, el siguiente argumento es la lista de los valores de ese atributo y por último tenemos el argumento prep(en) el cual significa que antes de nombrar algún valor tendrá que ir precedido por la preposición "en", teniendo asi como resultado final la siguiente oración:

" Inflorescencia en fasciculos, cimas o en racimos. "

4) La información de los artículos son predicados que tienen tres argumentos :

a) El nombre del artículo

b) Género

c) Número.

ej/

```
art(el, masc, sing).
```

Por otra parte, tenemos las listas de adverbios y de vocales.

a) La lista de adverbios, servirá para verificar si algún elemento de la lista de valores es un adverbio y así redactarlo de la manera correcta.

ej\

Supongamos que la base de datos tiene la información siguiente:

```
hojas(disposición([alternas,raramente,opuestas,verticiladas])).
```

La descripción final que se da es:

" Hojas alternas raramente opuestas o verticiladas. "

b) La lista de vocales, que contiene las cinco vocales existentes en el alfabeto. Esta lista está dada en código ASCII y se utilizará en caso de tener que cambiar el número de algún valor, es decir, cambiar de singular a plural o de plural a singular, dependiendo de cómo se encuentre en el diccionario.

La forma de hacerlo es por medio de estas vocales definidas, es decir, si la palabra termina en vocal como es el caso de opuesta, automaticamente la cambia a opuestas.

Esto le permite cierta flexibilidad al usuario.

III.4 El atn.

El atn normalmente se usa como reconocedor sintáctico, en ese caso, basta con definir las reglas sintácticas de una oración. En el caso de un atn como generador es necesario , además de tener definidas las reglas sintácticas, estar interactuando con el diccionario y con el plan.

El programa consta principalmente de cuatro atn's :

a) El atn general o de punto a punto

b) El atn de punto y coma a punto y coma

c) El atn de coma a coma y disyunción

d) El atn de coma a coma y conjunción.

y de otros, tales como:

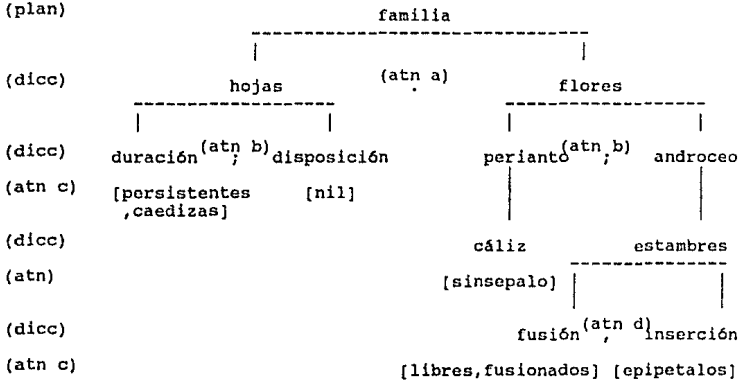
a) El atn del grupo nominal del sujeto

b) El atn del grupo nominal de los atributos.

El atn de punto y coma a punto y coma, asi como el de coma a coma (conjunción y disyunción) son llamados por el atn general. El atn del sujeto es llamado en el programa cada vez que se empieza una oración, el atn de atributos se ejecuta cada vez que empezamos la descripción de un nuevo atributo o sub_atributo.

La forma de interacción de las tres partes, se mantiene durante todo el proceso de generación. El plan nos mantiene informados en qué parte de la redacción nos encontramos. El diccionario nos da la posibilidad de verificar la sintaxis y la semántica de los sujetos, atributos, sub_atributos y valores. El atn es finalmente quien produce la estructura superficial final del texto.

Para poder entender esto claramente, supongamos que tenemos la siguiente información representada por medio de un árbol :



(Figura 10)

La figura 10 muestra claramente la forma de recorrido del árbol (plan), las veces que llamamos al diccionario asi como las llamadas a los atn's correspondientes, misma trayectoria que sigue el programa.

Como podemos observar (figura 10), la llamada al diccionario se presenta cada vez que cambiamos de tema y de atributo o sub-atributo, esto se debe a que para cada uno de ellos es necesario saber sus rasgos semánticos, sintácticos, así como el estilo. En el caso de los valores nos interesa unicamente que sean válidos, es decir, que pertenezcan al diccionario.

La redacción final es generada por el atn :

1. Hacemos una llamada al atn general, recorremos la rama izquierda del primer sujeto, redactando la información del atributo por medio del atn de coma a coma terminando con una disyunción.
2. Recorremos la rama derecha del mismo sujeto en cuestión, en cuyo caso tenemos la constante nil, o bien, información nula. Al no tener más información regresamos al atn de punto a punto. En caso de tener información en la rama derecha, como es el caso del sujeto flores hacemos una llamada al atn de punto y coma a punto y coma.
3. Cuando la profundidad es a nivel de sub_atributos es necesario el atn de coma a coma con conjunción, es el caso de los aub_atributos estambres(fusión) y estambres(inserción).

De esta manera obtenemos la siguiente descripción :

" Hojas persistentes o caedizas.
Flores cáliz sinsépala; estambres libres o fusionados y epipétalos. "

A continuación presentaremos el algoritmo principal de la generación con el cual podremos generar automáticamente cualquier descripción de familia.

El predicado principal es :

```
empieza_generar(Sujetos,Atributos,Valores).
```

el cual está definido de la siguiente manera :

```
P1:  empieza_generar([],[],[]).

P2: empieza_generar([Sujetos|Rsujeto],[Atr|Ratr], [Val|Rval]):-
                atn(Sujeto,Atr,Val,no),
                empieza_generar(Rsujeto,Ratr,Rval).
```

es un predicado recursivo cuya función es, recorrer las listas, de tal forma que la información completa que tenemos se va dividiendo para asi llamar al predicado atn(Sujeto,Atr,Val,no) con el primer sujeto, la lista de atributos a ese sujeto y los valores de ese atributo.

Como vimos en el inciso de la creación del plan, la lista de valores puede o no tener información.

En caso de no tener información, la definición es como sigue:

```
E1: atn(Sujeto,[Atr|Ratr],[Val|Rval],Nombro) :-
                Val = nil,
                atn(Sujeto,Ratr,Rval,Nombro).
```

al no tener información que transmitir sobre el atributo, hacemos una llamada recursiva al siguiente atributo con sus valores correspondientes.

El segundo caso consta de tres definiciones, dependiendo del nivel de profundidad del árbol. Si el atributo es sencillo, atributos o si son atributos compuestos, sub_atributos1 o sub_atributos2.

```
E2: atn(Sujeto,[Atr|Ratr],[Val|Rval],Nombro) :-
        atr(Atr,G,N,nombrar(Nom,P)),
        dicc(Sujeto,Atr,Valores,Prep,Adverbios,Nombro),
        atn_nombrar(Atr,G,N,Nom,P),
        checar_pal(Val,Valores,Adverbios,Val1),
        contar(Val1,Total),
        atn_coma_coma(Val1,Total,Prep,no,Adv,Adverbios),
        ifthenelse((inf(Rval,C), C = no),
              (punto,
        (punto_y_coma,
         atn(Sujeto,Ratr,Rval,si))).
```

Las tres definiciones varían en la llamada del atributo correspondiente al diccionario y en la última parte correspondiente al *ifthenelso*. Explicaremos este caso y después mostraremos las diferencias existentes para evitar escribir las definiciones siguientes y asi no hacerlo repetitivo.

```
atr(Atr,G,N,nombrar(Nom,P)) :
```

Es un predicado que es parte del diccionario, lo que hace es instanciar la variable *Atr* con el atributo en cuestión y traer la información correspondiente a sus argumentos.

dicc(Sujeto,Atr,Valores,Prep,Adverbios,Nombro) :

trae la información de los valores del atributo, la lista de adverbios y verifica por medio de otro predicado se_nombro (Nombro , Sujeto) si el sujeto al que nos estamos refiriendo ya ha sido o no nombrado.

atn_nombrar(Atr,G,N,Nom,P) :

tiene a su vez cuatro definiciones:

1) Si se nombra pero ya ha sido nombrado anteriormente, no hace nada.
2) Si se nombra sin ninguna preposición, se escribe el atributo.
3) Si se nombra con alguna preposición, entonces escribe la sintáxis del grupo preposicional completo, esto es, la preposición, el artículo con el mismo género y número del atributo y a continuación escribe el atributo.
4) Si no se nombra.

checar_pal(Val,Valores,Adverbios,Val1) :

es un predicado recursivo que recorre la lista de valores correspondiente al atributo en cuestión y verifica por primera vez si la información que dio el usuario es correcta. La información es correcta siempre y cuando :

a) El valor pertenezca al diccionario, ya sea en plural o singular
b) El valor pertenezca a la lista de adverbios.

En caso de que el valor no pertenezca a ninguna de estas definiciones el programa continua con la descripción poniendo un caracter que identifica la posición del valor pero no lo escribe.

contar(Val1,Total) :
cuenta los valores que caracterizan al atributo en cuestión.

Hasta este punto hemos escrito el sujeto, el atributo (en caso de haber sido necesario), hemos verificado que la base de datos contenga información correcta y hemos contado sus valores. Lo que resta es escribir esta información correctamente, es decir, con la sintaxis y la puntuación correspondiente. Para esto contamos con el predicado:

```
atn_coma_coma(Val,Total,Prep,no,Adv,Adverbios) :
A1: atn_coma_coma([C|R],T,Prep,Decir,Adv,Adverbios) :-
                T = 1,
                decir_prep(Prep,Decir),
                escribe(C).
```

```
A2: atn_coma_coma([C,C1|R],T,Prep,Decir,Adv,Adverbios) :-
                T = 2,
                ifthenelse(miembro(C,Adverbios),
                      (escribe(C),
                       decir_prep(Prep,Decir),
                       escribe(C1)),
                (decir_prep(Prep,Decir),
                 Decir1 = no,
                 escribe(C),
                 put(32),
                 checa_letra(C1),
                 decir_prep(Prep,Decir1),
                 escribe(C1))).

A3: atn_coma_coma([C,C1|R],T,Prep,Decir,Adv,Adverbios) :-
                 T > 2,
                 ifthenelse(miembro(C,Adverbios),
                       (escribe(C),
                        T1 is T - 1,
                        atn_coma_coma([C1|R],T1,Prep,si,Adverbios)),
                  (decir_prep(Prep,Decir),
                   escribe(C),
                   put(44),
                   T1 is T - 1,
                   atn_coma_coma([C1|R],T1,Prep,si,Adverbios))).
```

el cual contempla las definiciones para los tres casos necesarios en la descripción:

A1 : cuando la lista de Valores tiene un solo elemento, en cuyo caso lo que corresponde es saber si el valor tiene que ir o no precedido por una preposición.

A2 : cuando la lista de Valores tiene dos elementos:

1) Si el primer elemento es un adverbio, escribimos el adverbio seguido del valor.
2) Si los dos elementos son valores, escribimos el primer valor, una disyunción y escribimos a continuación el segundo valor.

A3 : cuando tiene más de dos elementos:

1) Si el primero es un adverbio, lo escribimos, decrementamos la variable T en uno y hacemos una llamada recursiva con los nuevos valores obtenidos y el resto de la lista de valores.
2) Si el primero es un valor, lo escribimos seguido de una coma, decrementamos la variable T en uno y volvemos a hacer la llamada recursiva de igual forma que en el caso anterior.

En los tres casos se verifica si hay alguna preposición que debe preceder al valor. Esta preposición sólo se nombra en el primer valor y en el último.

ej/

Inflorescencia en fasciculos, cimas o en racimos.

Analizaremos a continuación las diferencias del algoritmo atn general :

El primer caso, el cual dimos la definición anteriormente, consta de un ifthenelse al final, de la siguiente forma:

```
ifthenelse((inf(Rval,C), C = no),
                    (punto,
              (punto_y_coma,
               atn(Sujeto,Ratr,Rval,si))).
```

Esta definición es para cuando nos encontramos en el primer nivel del árbol, es decir, a nivel atributo, en cuyo caso, es necesario verificar si existen otras ramas derechas :

a) si no existen, quiere decir que hemos terminado el árbol correspondiente al sujeto en cuestión, para lo cual es necesario un punto (.) y continuar con la rama derecha a nivel de temas o sujetos principales.

b) si existen, es necesaria la puntuación de punto y coma (;) y continuar con la rama derecha del árbol a nivel atributos.

El predicado encargado de saber si existen otras ramas derechas es :

```
inf(Valores,Verdadero_o_falso).
I1 : inf([],no).
I2 : inf([Val|Rval],C) :-
             Val = nil,
             inf(Rval,C).
I3 : inf([Val|Rval],si).
```

La segunda y tercera definición del atn son exactamente iguales y está definida de la siguiente manera :

```
Q1 : cuanta_inf(Sujeto,Atr,Ratr,Rval,C),
     ifthenelse(C > 0,
           (ifthenelse(C = 1,
                 escribe ('y'),
            coma)),
     (ifthenelse((inf(Rval,I), I = no),
           punto,
     punto_y_coma)),
      atn(Sujeto,Ratr,Rval,si).
```

La cual es para el caso de los sub_atributos, esto es, cuando estamos en una profundidad mayor.

cuanta_inf(Sujeto,Atr,Ratr,Rval,C)

cuenta las ramas derechas existentes del sub_atributo en cuestión. Al tener información sobre la cantidad, la definición siguiente contempla los casos :

a) Si la cantidad de atributos es mayor que cero, contemplamos los casos :

i) que sea igual a uno, en cuyo caso escribimos una disyunción, continuando con una llamada al atn con los argumentos correspondientes.

ii) cuando es mayor que uno, entonces ponemos una coma (,),lo cual quiere decir que continuamos hasta llegar al caso anterior.

b) Si la cantidad de atributos es igual a cero, tenemos de igual manera dos casos :

i) Si existen más ramas del árbol a nivel atributo, procede escribir un punto y coma (;) y hacemos de igual manera la llamada al atn para continuar redactando la información de la rama existente.

ii) Si no existen más ramas, escribimos un punto (.). Hacemos la llamada al atn, en caso de existir más ramas a nivel sujeto se sigue el mismo procedimiento, en caso contrario, se da por terminado el proceso.

CAPITULO IV

MANUAL DE UTILIZACION DEL PROGRAMA

En la primera parte de éste capítulo explicaremos como se instala el programa y el funcionamiento de cada una de las opciones del menú. En la segunda parte se ilustrará el funcionamiento del programa con algunos ejemplos.

IV.1 Instalación y funcionamiento del programa.

Para el funcionamiento del programa es necesario contar con dos diskettes : el del intérprete de Prolog en su implementación de Arity/Prolog (versión 5.1), y el de los archivos menú.ari, plan.ari, dicc.ari, genera.ari que son los que contienen las definiciones generales y los archivos de las familias fam1.ari, fam2.ari, fam3.ari y fam4.ari.

Para la instalación basta con tener estos dos diskettes y copiarlos al disco, para lo cual es recomendable crear un subdirectorio en el disco duro con el nombre de genera. Una vez creado el subdirectorio, al trasladarse a él deberá aparecer el indicador del sistema operativo como:

C:\GENERA>

Se copian los diskettes a este subdirectorio. Una vez realizado lo anterior, el sistema estará instalado.

Para ejecutar el programa es necesario seguir los pasos siguientes :

Desde el sistema operativo de la máquina, se debe llamar al intérprete de Prolog:

C:\GENERA>api. <return>

A continuación aparecerá una pantalla con diferentes opciones y el indicador correspondiente del intérprete:

?-

Para cargar el programa de generación se teclea enseguida del indicador lo siguiente:

?-consult(genera). <return>

Con esta instrucción .:log cargará el archivo genera.ari a memoria y este a su vez carga los archivos del programa mencionados anteriormente. Cuando se termina de cargar el archivo genera, la llamada al programa despliega el Menú Principal a través de la siguiente pantalla:

```
                    M E N U   P R I N C I P A L

    Familia          (Escoger una familia para redactar)
    Información      (Dar la información de cierta familia)
    Salir            (Salir al sistema operativo)

  Teclea f, i o bien s:
  >
```

si tecleamos la letra f aparecerá el siguiente menú en pantalla:

```
                M E N U   D E   F A M I L I A S

     Flacourtiaceae                Gerenaiceae

     Compositae                    Verbenaceae

     Salir      (Salir menú principal)

    Teclea f, g, c o bien s:
    >
```

Al teclear cualquiera de las letras anteriores f,g ó c aparecerá el siguiente menú :

```
                        S U J E T O S
                  (Escoger el sujeto a redactar)

     Plantas                       Hojas

     Tallo                         Inflorescencia

     Flor                          Fruto

     información General           Salir (Menú Principal)

    Teclea p, t, h, i, f, fr, g o bien s:
    >
```

Cuando tecleamos cualquiera de estas opciones (a excepción de 's'), automaticamente aparecerá la descripción del sujeto que hayamos escogido en el menú de sujetos y de la familia escogida en el menú de familias.

IV.2 Ejemplos.

Mostraremos el funcionamiento del programa con tres ejemplos. El primero muestra una descripción de la Familia *Gereniaceae*, con el orden establecido en el programa y con la información dada por el usuario. El segundo, muestra una descripción de la Familia *Gereniaceae*, alterando el orden de los atributos, esto es, el plan de generación y con la misma información anterior, dada por el usuario. El tercero hace la descripción de la Familia *Flacourtiaceae* con el orden establecido y contemplando casi todos los atributos así como casi todos los valores de cada atributo. Solamente en el primer ejemplo y en el tercero describiremos paso a paso lo que el usuario y el programa van escribiendo.

Finalmente se expondrá una descripción de la Familia *Flacourtiaceae* escrita por un experto.

Ejemplo 1.

Supongamos que después de la aparición del menú principal tecleamos la letra i:

```
                M E N U   P R I N C I P A L

   Familia          (Escoger una familia para redactar)
   Información      (Dar la información de cierta familia)
   Salir            (Salir al sistema operativo)

  Teclea f, i o bien s:
  > i
```

Al hacer esto en la pantalla aparecerá el siguiente mensaje :

```
                    I N F O R M A C I O N

La sintaxis es de la siguiente forma:

atributo ([Valor1, Valor2,..., Valorn]).              ó
atributo (atributo ([Valor1,Valor2,...,Valorn])).     ó
atributo (atributo (atributo ([Valor1,..., Valorn]))).

Cuando aparezca ">" empieza a poner la información
siguiendo la sintaxis anterior y sobre el sujeto que
nos digan.

 Si no hay más información teclea <enter>.
```

Después de aparecer ésta pantalla de información sobre la sintaxis, tecleamos <enter> y el programa preguntará :

```
| Nombre de la familia:                                    |
```

a continuación podemos escribir el nombre de la familia ya sea con letras mayúsculas o minúsculas, dependiendo de cómo queramos que aparezca en la descripción.

```
| Nombre de la familia: FAMILIA GERENIACEAE <enter>        |
```

o bien tecleamos <enter> para saltar a la siguiente etapa:

```
| A CONTINUACION DAME LA INFORMACION DE : plantas          |
```

```
>
```

Y aparecerá el indicador para empezar a escribir la información del sujeto plantas de la familia *Gereniaceae* :

```
> duracion([anuales,bianuales,perennes])        <enter>
> forma_de_vida([herbaceas])                    <enter>
> <enter>
```

Al dar <enter> después del indicador, el programa da por hecho que no hay más información que transmitir sobre el sujeto en cuestión. Continua entonces con el siguiente sujeto:

```
| A CONTINUACION DAME LA INFORMACION DE : tallo |
```

```
> posicion([erecto,ascendente,rastrero])      <enter>
> <enter>
```

```
| A CONTINUACION DAME LA INFORMACION DE : hojas |
```

```
> estipulas([presentes])                      <enter>
> disposicion([opuestas,alternas])            <enter>
> peciolo([presente])                         <enter>
> <enter>
```

```
| A CONTINUACION DAME LA INFORMACION DE : inflorescencia |
```

```
> <enter>
```

```
| A CONTINUACION DAME LA INFORMACION DE : flor |
```

```
> gineceo(ovario([supero]))                                    <enter>
> gineceo(Hojas_carpelas([5]))                                 <enter>
> sexo([hermafroditas,actinomorfas,rara_vez,zigomorfas]).      <enter>
> perianto(caliz(sepalos([5])))                                <enter>
> perianto(corola(petalos([5])))                               <enter>
> androceo(estambres(cantidad([10])))                          <enter>
> <enter>
```

A CONTINUACION DAME LA INFORMACION DE : fruto

> tipof([esquizocarpo]) <enter>

> <enter>

A continuación aparecerá la siguiente redacción :

FAMILIA GERANIACEAE

Plantas herbaceas; anuales,bianuales o perennes.

Tallo erecto, ascendente o rastrero.

Hojas opuestas o alternas; estipulas presentes; peciolo presente.

Flores hermafroditas, actinomorfas o rara_vez zigomorfas;

sepalos 5 y petalos 5; estambres 10; ovario supero y hojas_

carpelares 5.

Fruto un esquizocarpo.

Al terminar de escribir la descripción oprimimos cualquier tecla y regresamos al menú principal.

Ejemplo 2.

Con el objeto de mostrar cómo quedaría la misma descripción del ejemplo 1 presentada en otro orden. Supongamos que el orden de los atributos es el mismo orden en el que estan escritos. De esta manera nos da la siguiente descripción :

```
                    FAMILIA GERANIACEAE

Plantas bianuales o perennes; herbaceas.

Tallo erecto, ascendente o rastrero.

Hojas estipulas presentes; opuestas o alternas; peciolo presente.

Flores ovario supero, hojas_carpelares 5;  hermafroditas,

actinomorfas o rara_ ·· zigomorfas; sepalos 5, petalos 5;
estambres 10.

Fruto un esquizocarpo.
```

El primer ejemplo es más claro, esto se debe a que el orden de redacción (mismo orden que utilizamos en el programa) describe a la planta de lo general a lo particular.

Ejemplo 3.

Supongamos que después de la aparición del menú principal tecleamos las letras f, f y g :

```
                M E N U   P R I N C I P A L

     Familia          (Escoger una familia para redactar)

     Información      (Dar la información de cierta familia)

     Salir            (Salir al sistema operativo)

    Teclea f, i o bien s:
    > f
```

```
                    M E N U   D E   F A M I L I A S

     Flacourtiaceae                    Gerenaiceae

     Compositae                        Verbenaceae


     Salir      (Salir menú principal)

    Teclea f, g, c o bien s:
    > f
```

```
                          S U J E T O S
                    (Escoger el sujeto a redactar)

     Plantas                           Hojas

     Tallo                             Inflorescencia

     Flor                              Fruto


    Información General                Salir (Menú Principal)

    Teclea p, t, h, i, f, fr, g o bien s:
    > g
```

En este caso, procede a redactar la información de la familia flacourtiaceae con la base de datos que se encuentra en el archivo (fam1.ari), dando como resultado :

FAMILIA PLACOURTIACEAE

Plantas arboreas o arbustivas; perennifolias.

Tallo glabro, pubescente, tomentoso o viloso.

Hojas persistentes o caedizas; alternas, raramente opuestas o verticiladas; membranosas, coriaceas o cartaceas; estipulas usualmente pequeñas y caducas; pinnado_nervadas o reticuladas; peciolo presente; glabras, pubescentes, tomentosas o velutinosas; con el apice agudo o acuminado; margenes glandular_crenados, dentados o aserrados; base aguda, atenuada, cordada o cuneada.

Inflorescencia en fasciculos, cimas, racimos, paniculas, corimbos, espigas o en flores_solitarias; axilares o terminales; pedunculos presentes; bracteas numerosas.

Flores bisexuales o unisexuales; actinomorfas; sepalos 3-8 o mas, contortos,imbricados o valvados y generalmente ausentes o abortivos; estambres comunmente numerosos y libres o connados; anteras biloculares,y dehiscencia longitudinal; hojas_carpelares 2-10, estilo apical, libre o unido, estigma capitado o lobulado, ovario súpero o semi_infero, unilocular y ovulos anatropos o anfitropos.

Fruto una capsula, baya, drupa o una samara_trialada; dehiscente o indehiscente; semillas pocas o numerosas, desiguales o comprimidas, arilo usualmente presente y endospermo generalmente copioso y carnoso; embrion recto o curvo, cotiledones anchos y frecuentemente cordados.

CONCLUSIONES

Para concluir, empezaremos por hacer un recuento de lo que a lo largo de esta tesis fueron los puntos más importantes : primero presentamos las diferencias sintácticas existentes en las descripciones botánicas, segundo la sintaxis y el orden que utilizamos para el algoritmo. Mostramos la forma de representación de la base de datos, así como, el plan que se sigue por medio de árboles. Por último se expuso el programa de generación.

A continuación mencionaremos los aspectos más importantes que caracterizan al sistema automatizado.

Con respecto a la implementación de los algoritmos, usamos Arity/Prolog versión 5.1 por ser una implementación que tiene intérprete y compilador. Tiene además las características esenciales de un lenguaje lógico y tiene definidos predicados intrínsecos de mucha utilidad como : el *name*, el *assert*, el *functor*, así como en particular tiene estructuras de control como el *ifthen*, *ifthenelse* y el *case*, lo cual nos da cierta independencia a nivel del "backtracking". Por otra parte, Prolog es un lenguaje expresivo para los algoritmos en lingüística computacional. Es un lenguaje de alto nivel donde facilmente podemos expresar operaciones con los simbolos (representado por átomos, cadenas, números) y estructuras (representado por listas de términos) sin tener la preocupación de cómo estan representados éstos conceptos internamente. Nos permite a su vez, representar información a un nivel muy abstracto por medio de un conjunto de "hechos", además de no existir resticciones en la

definición de predicados que se llaman así mismos (directa o indirectamente). Con repecto a la implementación del atn, prolog permite traducir los algoritmos de manera casi directa.

Con respecto a las limitaciones del programa, el orden de los atributos y sub_atributos, se encuentran fijos. Para cambiar éste orden es necesario hacerlo directamente en el programa, en la parte correspondiente al pl...

Por otro lado, el programa no contempla el uso de los acentos, la base de datos creada por el usuario, deberá entonces, prescindir de éstos.

Una extensión posible al programa sería contemplar los diferentes estilos, así como, generar descripciones más completas, sin hacer elipsis de sujetos, verbos y artículos principalmente. Para esto es necesario modificar este procedimiento por medio de cuatro casos:

1) Escribir los artículos que anteceden al sujeto
2) Escribir el sujeto cuantas veces sea necesario
3) Escribir los atributos con el artículo o preposición correspondiente.
4) Escribir el verbo correspondiente

Caso 1.

Para escribir el artículo que antecede al sujeto, será necesario especificarlo en el atn. Para esto habría que mandar escribir su artículo correspondiente, es decir, escoger un artículo definido o indefinido, con el mismo género y número del sujeto y a continuación escribir el artículo.

Esto no alteraría la estructura del programa dado que es la misma forma como se hace en los atributos que se nombran en la parte correspondiente al procedimiento atn_nombrar.

En caso que queramos que algunos sujetos estén acompañados de un artículo y otros no, será entonces necesario especificarlo en el diccionario, de la misma forma que se hace en los atributos, esto es, aumentar un argumento a los sujetos y especificar si queremos o no escribir el artículo que antecede al sujeto.

ej/

```
sujeto(hojas,fem,plural,decir_art(si))).
sujeto(tallo,masc,sing,decir_art(no)).
```

Teniendo esto en el diccionario, se procedería de igual manera que como se hace con los atributos. Verificar en el diccionario si el artículo se debe o no mencionar. En caso afirmativo tendríamos que buscar en el diccionario, qué artículo se instancia con el género y número del sujeto y mandarlo a un nuevo atn que escriba ese artículo a continuación del sujeto. Para esto tendría que alterarse también el procedimiento del atn para las mayúsculas, es decir, el artículo deberá empezar con la primera letra en mayúscula y el sujeto que se escribe a continuación en minúsculas. Habría también que especificar que después del artículo, viene el sujeto y a continuación el verbo 'ser'.

ej\

Las hojas son .....

En caso negativo no tendríamos mas que escribir el sujeto, de la misma forma que se hace actualmente.

Caso 2.

En el programa actual nombramos el sujeto sólo una vez, al iniciar la oración, aunque siempre lo tenemos presente, es decir, cada vez que escribimos un nuevo atributo o sub_atributo verificamos que pertenezca al sujeto en cuestión.

Para escribir el sujeto cuantas veces sea necesario, no existe mayor complejidad, por la razón antes mencionada. De esta manera, podriamos definir un contador que no exceda de cierto número de frases, sin contemplar de nuevo al sujeto o bien definiendo en el diccionario que atributos o sub_atributos nos permiten ser antecedidos por el sujeto.

Caso 3.

Este caso está contemplado en el programa sólo para algunos atributos. La forma de contemplarlo para todos los atributos se haría, especificando que palabra debe anteceder al atributo.

ej/

atr(disposición,fem,sing,nombrar(si,su)).

... su disposición es alterna, raramente opuesta o verticilada; ...

Cada vez que hablemos de un nuevo atributo tendríamos que ir al diccionario a verificar qué palabra debemos poner antes de escribir el nuevo atributo. El atn quedaría de la misma manera que está actualmente.

Caso 4.

Escribir el verbo correspondiente sería un poco más complicado.

Este caso se podría ver a su vez como dos casos. El primero cuando el verbo es el verbo "ser" y el segundo en los casos restantes.

El verbo, como podemos observar en el ejemplo escrito en Lenguaje Natural de la Familia *Flacourtiaceae*, algunas veces antecede al atributo y otras veces va después de éste. Esto depende de si el sujeto se nombró o no.

ej/

> ...; su disposición es alterna, raramente opuesta o verticilada; las hojas tienen una consistencia membranosa, coriácea o cartácea; ...

El atributo *disposición* antecede al verbo "es", en cambio, el atributo *consistencia* se escribe después del verbo "tener". Esto se debe a que disposición es el sujeto del verbo "es" y hojas es el sujeto del verbo "tienen".

Para el caso cuando se nombra el tema o sujeto principal, podríamos poner cualquier verbo, en particular, el verbo "tener". Como en el ejemplo anterior, lo que se podría hacer es definir un nuevo argumento para los atributos, donde esté contemplado este caso.

ej/

atr(consistencia,fem,sing,nombrar(si,una),verbo(tienen)).

Si el sujeto se nombra así como el atributo, a continuación tendríamos que verificar en el diccionario cuál sería el verbo correspondiente. Ahora bien, si el sujeto no se nombra entonces podríamos generalizar que el verbo correspondiente sería el verbo "ser", con el número correspondiente al atributo.

Por último, si ni el sujeto ni el atributo se nombran entonces el verbo sería el mismo verbo "ser" pero habría que verificar que número tiene el sujeto.

ej/

...; son a su vez penninervadas o reticuladas; ...

Como el verbo "son" se refiere al sujeto hojas, el cual está en plural, el verbo también tiene que ir en tercera persona plural.

Los resultados obtenidos hasta el momento para la generación automática de descripciones botánicas demuestra que existen aplicaciones del procesamiento del lenguaje natural de utilidad práctica, distintas a las clásicas consultas a las bases de datos. Aunque en este trabajo sólo se muestra la aplicación para las familias de plantas, es fácil pensar que se puede generalizar a otras ramas de la biología o bien a otro tipo de clasificación de las plantas como son el caso de género y especie. Para ello sería necesario diseñar un módulo general de planificación que permita una interacción más

flexible y poderosa con el usuario y en particular, que le permita a éste expresar las peculiaridades del estilo de los reportes que desea obtener. Por otra parte, será necesario extender la representación semántica del diccionario, así como el módulo generador con Redes de Transición Aumentadas (ATN) capaces de manejar los diferentes estilos de descripciones a partir del plan de generación.

BIBLIOGRAFIA

[1] ALLEN James. Natural Language Understanding, University of Rochester, Cummings Publishing Co., Inc, 1987

[2] APPELT Douglas, Planning English Referring Expressions, contenido en: Readings in Natural Language Processing, Grosz B. et. al., Morgan Kaufmann, 1986

[3] ARITY/PROLOG Reference and Technical Manual. Versión 5.1

[4] BOLD, H.C., ALEXOPOULOS, C.J. and T. DELEVOYRAS. Morphology of plants and fungi, New York, USA, Harper and Row. Publishers., 1980.

[5] BRATKO, Ivan. Prolog Programming for Artificial Intelligence, Addison-Wesley, 1986

[6] COHEN P. and PERRAULT R. Elements of a plan Based Theory of Speech Acts, contenido en: Readings in Natural Language Processing, Grosz B. et. al., Morgan Kaufmann, 1986

[7] GAZDAR G. and MELLISH C. Natural Language Processing in Prolog, Addison-Wesley, 1989

[8] HOVY Eduard H. Pragmatics and Natural Language Generation, Artificial Intelligence, 1990.

[9] McKEOWN Kathleen R. Text Generation, Cambridge University Press, 1985

[10] McKEOWN, Kathleen R. Discurse Strategies for Generating Natural- Language Text, contenido en: Readings in Natural Language Processing, Grosz B. et. al., Morgan Kaufmann, 1986

[11] MORENO Nancy P. Glosario Botánico Ilustrado, CECSA, 1984

[12] NILSSON Nils J. Principles of Artificial Intelligence, Berlin, Springer-Verlag 1982

[13] RZEDOWSKI Jerzy y RZEDOWSKI G.C. de, Flora fanerogámica del Valle de México, CECSA, México, 1979

[14] STERLING L. and SHAPIRO E. The art of Prolog, MIT Press, 1986

[15] VILLASEÑOR José L., Instituto de Biología, UNAM, México, 1987