17
29

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

ANDREW, RECONOCEDOR DE LENGUAJE NATURAL

T E S I S

QUE PARA OBTENER EL TITULO DE

M A T E M A T I C O

P R E S E N T A

NORMA ANGELICA MEJIA MONASTERIO

MEXICO D.F. 1990

FALLA DE ORIGEN

UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Contenido

INTRODUCCION.

I. Analizadores y gramáticas de casos.
1.1 El lenguaje natural.
1.1.1 Analizadores sintácticos.
1.1.1.1 Analizadores Top Down y Bottom Up.
1.1.1.2 Analizadores de redes de transición recursivas.
1.1.1.3 Analizadores de redes de transición aumentadas.
1.1.1.4 El analizador sintáctico de Andrew.
1.1.2 Gramáticas de casos.

II. El Sistema.
2.1 Descripción del sistema.
2.1.1 El modelo lingüístico.
2.1.2 La memoria.
2.1.3 Los operadores.
2.1.4 La interfaz.
2.2 El funcionamiento del sistema.
2.2.1 Los operadores de diálogo.

III. El modelo lingüístico utilizado.
3.1 Las categorías gramaticales.
3.2 La estructura semántica de la frase.
3.2.1 El sintagma nominal.
3.2.2 Los casos de la estructura semántica.
3.2.2.1 Descripción de casos.

IV. La memoria del sistema.
4.1 Los conocimientos.
4.1.1 Los conocimientos predefinidos.
4.1.2 Los conocimientos adquiridos.
4.2 La organización del vocabulario.
4.2.1 El nivel sintáctico.
4.2.2 El nivel semántico.
4.3 Estructuras de representación de frases.
4.3.1 La estructura sint.
4.3.2 La estructura prd.
4.3.3 La estructura ora.

V. El análisis de las frases.
5.1 El Analizador
5.1.1 El análisis de un sintagma nominal.
5.1.2 El análisis del predicado de una frase.
5.1.2.1 Procedimiento VERBO_ESTADO.
5.1.2.2 Procedimiento VERBO_ACCION.
5.1.2.3 Procedimiento VERBO_PROCESO.
5.1.2.4 Un ejemplo de análisis.

VI. Los otros operadores.
6.1 El Generador.
6.1.1 El procedimiento GENERA_SINT.
6.1.2 El procedimiento GENERA_PRD.
6.1.3 Ejemplo de generación de una frase.
6.2 El Contestador.

VII. Codificación en Prolog.
7.1 Codificación del vocabulario.
7.2 Codificación de la frase.
7.3 Codificación de los Operadores.
7.3.1 Codificación del ANALIZADOR.
7.3.2 Codificación del CONTESTADOR.
7.3.3 Codificación del GENERADOR.

VIII. Interfaz con el usuario
8.1 Interfaz con el usuario.

CONCLUSIONES.

APENDICE A.

APENDICE B.

APENDICE C.

BIBLIOGRAFIA

INTRODUCCION.

La inteligencia artificial es una de las áreas más importantes de las ciencias de la computación. La mayor parte de los trabajos realizados en esta área tienen que ver con el desarrollo de sistemas computacionales que simulan el comportamiento humano.

Una de las ramas más apasionantes de la inteligencia artificial es el estudio del lenguaje natural. Desde los primeros años de la aparición de las computadoras, personas especializadas en psicología, lingüística e informática, se han conjuntado para estudiar el problema del procesamiento del lenguaje natural mediante computadoras. Este problema abarca distintos aspectos que van desde el estudio del proceso del pensamiento humano, pasando por los formalismos de tipo lingüístico, hasta las técnicas meramente computacionales. Dentro de estos dos últimos aspectos se sitúa la llamada lingüística computacional, en donde podemos enmarcar el presente trabajo.

La tarea que nos ocupa tiene que ver sobre todo con un aspecto de la lingüística computacional, precisamente el de reconocer sintáctica y semánticamente frases del lenguaje natural. Nuestro objetivo central, fué el desarrollo de un analizador sintáctico-semántico, de frases aisladas del Español, para generar una representación interna basada en la gramática de casos definida por Fillmore [1].

Nuestro trabajo se inspira directamente en los sistemas de reconocimiento y comprensión de historias desarrollados por C. Loyo [3] y G. Sabah [4] en los años setenta. La idea fue la de iniciar la reprogramación de algunas partes de estos sistemas utilizando el lenguaje PROLOG. Esto con el objeto de poder contar con una herramienta de apoyo que nos permita trabajar, más adelante, aspectos de razonamiento dentro de un diálogo.

Para dar un marco de presentación más acabado a la programación de este analizador sintáctico-semántico, desarrollamos por encima de él, un sistema de preguntas y respuestas que muestra algunas de las ventajas del uso de las gramáticas de casos como modelo de representación.

El funcionamiento del sistema permite reconocer conjuntos de frases que pueden armar o no, una historia. El analizador reconoce cada frase de manera aislada y le asocia una estructura semántica al interior del sistema. Esta misma estructura es posteriormente utilizada para dar respuesta a las preguntas del usuario sobre la información contenida en una frase.

Es importante enfatizar que, por el momento, el sistema no maneja el contexto en el que están inmersas las frases, sino que considera a cada una ellas como aislada de las demás. Sin embargo, se ha previsto ya, dentro del sistema, la posibilidad de establecer una liga entre los distintos personajes y conceptos de una historia, a través de una organización semántica del vocabulario. Esto define un marco para hablar de los personajes dentro de un contexto y posteriormente pensar en una extensión hacia el manejo de la temporalidad y el tratamiento contextual de los pronombres.

El universo de trabajo del sistema está inicialmente definido para el relato de historias de ciencia ficción, y es precisamente de una de estas historias, "El hombre del bicentenario", del escritor Isaac Asimov, de la que toma su nombre : Andrew.

Para situar el tipo de analizador desarrollado para nuestro nuestro sistema, damos en el capítulo I una breve descripción de los tipos clásicos de analizadores y describimos el modelo de gramáticas de casos utilizado.

En el capítulo II, damos una descripción general del sistema de preguntas y respuestas en el que está inmerso el analizador y hablamos de los principales operadores que dicho sistema utiliza para el establecer el diálogo con el usuario.

En seguida, en el capítulo III, describimos a grandes rasgos el modelo lingüístico utilizado por el sistema, el cual incluye la definición de categorías gramaticales y el tipo de relaciones semánticas utilizadads para el análisis de las frases.

En el capítulo IV, damos una descripción detallada de la estructura que utilizamos para la representación del vocabulario y de las frases.

En el capítulo V, explicamos detenidamente el funcionamiento del operador principal, el analizador de frases. A continuacion, en el capítulo VI, damos una explicación sobre el funcionamiento de otros operadores utilizados en el diálogo, como son el operador que genera las respuestas a las preguntas del usuario , y el que genera a partir de una estructura semántica, una frase en Español.

En el capítulo VII, damos ejemplos de codificación en Prolog de algunos de los predicados más importantes utilizados por estos operadores y finalmente en el capítulo VIII explicamos el funcionamiento de la interfaz entre el usuario y el sistema.

Al final de este trabajo aparecen una serie de apéndices en los que mostramos ejemplos de diálogo y algunas otras de las opciones del sistema.

I. Analizadores y gramáticas de casos.

1.1 El lenguaje natural.

Una de las principales características, que distingue al ser humano, es su capacidad de comunicación. Con la aparición de las máquinas computadoras y su posterior desarrollo, surgieron un buen número de intentos de imitar no sólo ésta, sino otras características del ser humano, como, por ejemplo, la visión. Desde entonces, han surgido diversos trabajos enfocados al desarrollo de modelos para manejar el lenguaje con el que nos comunicamos, es decir el lenguaje natural.

Sin embargo, aún no existen trabajos cuya comprensión del lenguaje natural rivalice o esté cerca de rivalizar con la del ser humano, excepto en historias de ciencia ficción, en las que aparecen robots que se comunican perfectamente en lenguaje natural e incluso en varios diferentes idiomas. El robot Andrew, creado por el escritor Isaac Asimov, es un buen ejemplo de este tipo de robots.

Comprender el lenguaje natural, no es fácil. Se requiere no sólo de conocimientos lingüísticos del lenguaje, sino de un cierto conocimiento del mundo relacionado al tópico que se esté discutiendo.

El problema de reconocimiento de lenguaje natural, puede ser dividido en tres áreas :

- fonética-acústica,
- sintáctica-morfológica, y
- semántica-pragmática.

La primera de estas áreas se encarga de tomar los sonidos y traducirlos a palabras. La segunda toma estas palabras y establece la forma sintáctica basada en la pronunciación, mientras que la tercera trata de entender el significado de acuerdo a dicha sintaxis.

El problema que nos ocupa está inmerso en las últimas dos áreas, y, en particular, dentro del análisis a nivel sintáctico y semántico, de frases dadas por escrito.

Andrew, el sistema que hemos construído, constituye un ejemplo de realización, en el lenguaje de programación Prolog, de un analizador sintáctico-semántico de frases. Las frases son dadas en lenguaje natural, y Andrew las analiza para generar una representación semántica, que le permite posteriormente, contestar a una serie de preguntas simples que el usuario puede hacer con respecto a dichas frases. Con preguntas simples, nos referimos a preguntas cuya respuesta está contenida en frases aisladas, ya que el sistema no reconoce el contexto en que cada frase se desenvuelve, y preguntas del estilo "porqué" no son manejadas.

Para la generación de la representación semántica de frases, nuestro sistema se basa en el modelo de gramáticas de casos, que describiremos más adelante.

En las siguientes secciones daremos una breve descripción de los diferentes y más importantes analizadores que han sido y siguen siendo utilizados. Explicaremos así mismo, el tipo de modelo semántico que Andrew utiliza para representar las frases.

1.1.1 Analizadores sintácticos.

En computación, el proceso que se encarga de inferir una estructura sintáctica a partir de cadenas gramaticales de palabras, es conocido como analizador sintáctico o parser. Es precisamente el diseño de este tipo de analizadores, un trabajo que ha estado ocupando a mucha de la gente que se dedica al área de comprensión de lenguaje natural, en los últimos años.

Un analizador sintáctico de lenguaje natural, puede ser visto como un algoritmo que basado en un conjunto de reglas, genera una estructura gramatical para una frase dada. Esto ocurre, siempre y cuando, la frase sea correcta de acuerdo a dichas reglas. Ahora

bien, ese conjunto de reglas, constituye una gramática. Una gramática es una manera de formalizar el conocimiento sobre la estructura de un lenguaje, por medio de una serie de reglas que son capaces de reconocer frases de dicho lenguaje.

Un ejemplo de este tipo de gramática está dado en la figura 1.1. En dicha figura mostramos una gramática que representa un pequeño fragmento del Español.

```
FRASE  ->  FN FV
FV     ->  VERBO
FV     ->  VERBO FN
FN     ->  DET SUST
FN     ->  S_PRO
DET    ->  la
SUST   ->  fruta
VERBO  ->  come
S_PRO  ->  Andrew
```

Fig. 1.1.
Ejemplo de gramática.

En esta figura, la primera regla, especifica que una frase (FRASE), está dada por la unión de una frase nominal (FN), con una frase verbal (FV). A su vez una frase verbal (FV), está dada por un verbo (VERBO) y una frase nominal (FN). Una de las reglas subsecuentes especifica que un verbo (VERBO), está dado por "come".

En las reglas de la gramática, intervienen dos tipos de nodos:

Nodos no-terminales : Dados por identificadores en mayúsculas. Este tipo de nodos, forman parte de las reglas, pero no aparecen en la frase en Español, como por ejemplo : VERBO.

Nodos terminales : Dados por identificadores en minúsculas. Estos nodos definen a las palabras que aparecen en la frase en Español, como por ejemplo : come.

Este tipo de gramática es llamado libre del contexto, porque las reglas establecen como reemplazar un nodo no-terminal, sin ninguna referencia al contexto en que dicho nodo se encuentra.

Ahora bien la gramática en sí, sólo constituye una descripción del lenguaje en cuanto a su estructura sintáctica. Pasaremos ahora a describir algunos de los tipos más importantes de analizadores sintácticos de frases.

1.1.1.1 Analizadores Top Down y Bottom Up.

El problema a resolver para un analizador sintáctico, consiste en que, dada una cadena de palabras que representan una frase, debe descomponer esta cadena en una estructura sintáctica asociada a dicha frase.

La idea básica del análisis Bottom-Up es la de usar las reglas generativas de la gramática, a la inversa, es decir de izquierda a derecha. Para aclarar el proceso que sigue un analizador de este tipo, veamos el ejemplo dado en la figura 1.2. En dicha figura, mostramos el funcionamiento de un analizador Bottom-Up, basado en la gramática que dimos en la figura 1.1. La frase analizada en el ejemplo es la siguiente :

"Andrew come"

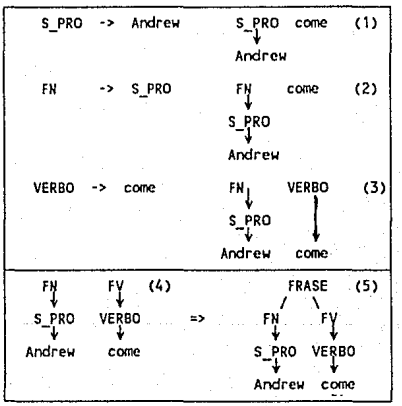


Fig. 1.2.
Funcionamiento de análisis Bottom-Up para la frase :
"Andrew come"

Basados en la figura anterior, describimos a continuación los pasos que sigue el analizador Bottom-Up :

- El analizador comienza a recorrer la frase de izquierda a derecha, y encuentra la palabra "Andrew".

- Trata de hacer coincidir esta palabra con el lado derecho de alguna de las reglas de la gramática. Encuentra así la regla, **S_PRO -> Andrew.**

- Reemplaza a la palabra "Andrew", por el nodo no-terminal S_PRO, generando así la estructura : "S_PRO come" (1).

- Busca ahora una regla cuyo lado derecho sea el nodo no-terminal S_PRO. Encuentra la regla : "FN -> S_PRO"

- Reemplaza al nodo no-terminal S-PRO por el nodo no-terminal FN (Frase Nominal), generando la estructura dada en (2).

- Busca ahora una regla cuyo lado derecho esté dado por el nodo FN. Esta vez no encuentra ninguna regla con esta característica, por lo que sigue analizando la frase, y encuentra la palabra "come".

- Aplicando este proceso consistentemente, el analizador genera la estructura sintáctica que se muestra al final de la figura, en (5).

En resumen, el proceso seguido por un analizador Bottom-Up, parte de buscar en la frase, una subestructura, que concuerde con el lado derecho de alguna de las reglas de la gramática, para después utilizar el lado izquierdo de dicha regla, en la generación de la estructura sintáctica. Este proceso se repite, hasta que la estructura sintáctica de la frase está completamente definida.

Ahora bien, este tipo de análisis tiene problemas cuando una frase o parte de ella, puede coincidir con dos reglas diferentes. En estos casos, el analizador intenta aplicar, una a una, las reglas alternativas hasta que encuentra una combinación que le de una estructura correcta a la frase.

Al encontrar un problema de este tipo, puede resultar útil el conocer el tipo sintáctico asociado a aquellas palabras de la frase, que ya han sido analizadas. Sin embargo , en los analizadores Bottom-Up, no es posible regresar para obtener esta información. En estos casos, el modelo Top-Down es más eficiente.

El modelo Top-Down, funciona de una manera muy parecida al modelo Bottom-Up, sólo que en sentido inverso. El procedimiento que siguen estos analizadores, consiste en tomar la regla inicial de la gramática y a partir de ésta, ir aplicando las reglas siguientes para generar la estructura de la frase.

En la figura 1.3., mostramos un ejemplo del funcionamiento de un analizador Top-Down, basado en las reglas de la gramática que definimos en la figura 1.1. La frase analizada en el ejemplo es la siguiente :

"Andrew come"

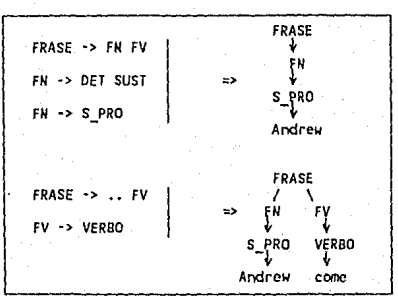


Fig. 1.3.
Funcionamiento de análisis Top Down para la frase :
"Andrew come"

Veamos los pasos que sigue el analizador Top-Down, en el ejemplo de la figura 1.3.

- Al comenzar el análisis, el analizador toma la primera regla de la gramática (FRASE -> FN FV), y trata de aplicarla.

- Toma el primer nodo de la parte derecha de dicha regla, que es FN y busca una regla cuyo lado izquierdo esté dado por este nodo.

- Encuentra la regla : FN -> DET SUST.

- Toma el primer nodo de esta regla, que es DET y, a su vez, busca una regla cuyo lado izquierdo esté dado por este nodo.

- Encuentra la regla : DET -> la.

- Como el lado derecho de esta regla es un nodo terminal, busca dicho nodo (la), en la frase que se está analizando.

- Dicha frase es "Andrew come", por lo que no encuentra el determinante "la", y regresa a buscar otra definición de una frase nominal, encontrando la regla : FN -> S_PRO.

- Siguiendo el análisis encuentra que el nodo terminal "Andrew", si aparece en la frase, por lo que da por terminado el análisis de la frase nominal (FN), y pasa a hacer el análisis de la frase verbal (FV).

- Siguiendo un proceso similar, el analizador genera la estructura sintáctica que se muestra en la figura.

En resumen el proceso seguido por un analizador Top-Down, consiste en aplicar las reglas de la gramática hasta ir encontrado nodos terminales que coincidan con las palabras de la frase.

Aunque los analizadores Top-Down funcionan bien para algunas frases funcionan mal para otras, en las que los analizadores Bottom-Up lo hacen mejor.

1.1.1.2 Analizadores de redes de transición recursivas.

En la sección anterior describimos a los analizadores Bottom-Up y Top-Down. La tendencia en general ha sido la de utilizar los analizadores Top-Down. En esta sección describiremos a los analizadores de red de transición, que son un tipo de analizador Top-Down.

En general, una red de transición, consiste de un conjunto de estados conectados por arcos. Cada arco representa una transición entre dos estados. Una red de transición, puede ser vista como un patrón para reconocimiento o generación de secuencias de palabras.

Ahora bien un analizador de red de transición, cuenta con la red misma, y con un diccionario en el que cada palabra del vocabulario, aparece asociada a su tipo sintáctico. Un elemento de este diccionario sería por ejemplo :

SUST -> robot.

El analizador utiliza el patrón, definido por la red, recorriendo las transiciones de estado a estado. El recorrido debe comenzar en un nodo inicial, y debe terminar en un nodo terminal.

En la figura 1.4, mostramos un ejemplo de una red de transición.

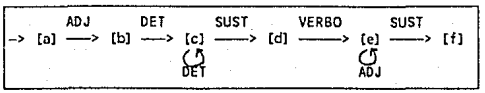


Fig. 1.4.
Ejemplo de una red de transición.

Siguiendo el patrón definido por esta red, podemos, por ejemplo, reconocer la siguiente frase :

" Feliz, el otro robot come frutas".

Para ello, el recorrido comienza en el estadio inicial : [a], y continúa, por la siguiente serie de arcos y nodos:

	->	[a]	
ADJ	->	[b]	(feliz)
DET	->	[c]	(el)
DET	->	[c]	(otro)
SUST	->	[d]	(robot)
VERBO	->	[e]	(come)
SUST	->	[f]	(frutas)

Un tipo de red de transición, son las redes de transición recursivas, de las que hablaremos en seguida.

Una red de transición recursiva, es básicamente una red de transición, en la que, los arcos pueden estar etiquetados con categorías sintácticas compuestas, es decir, la etiqueta de una arco puede ser, a su vez, la etiqueta de otra red. Cuando en el recorrido de una red recursiva, un arco está etiquetado de esta manera, dicho recorrido continúa por la subred que dicho arco define.

Otra diferencia, entre una red simple y una recursiva, consiste en la definición de un nuevo tipo de arco, que se denomina arco de "salto". Este tipo de arco puede ser recorrido sin necesidad de encontrar una categoría gramatical que lo defina, y permite pasar de un nodo a otro incondicionalmente.

En la figura 1.5, mostramos un ejemplo de una red de transición recursiva.

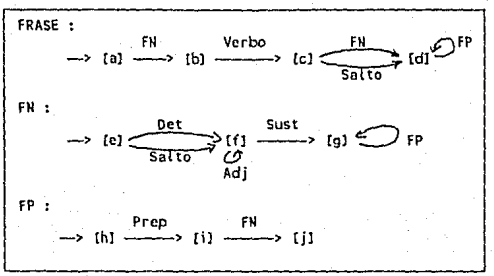


Fig. 1.5.
Ejemplo de una red de transición recursiva.

En esta figura, podemos observar, por ejemplo, que el arco que une a los nodos [a] y [b], es el identificador (FN) de una subred que representa a un sintagma nominal.

Observamos también que para pasar del arco [e] al [f], es requerido un determinante (Det), o bien se puede realizar un salto incondicional a dicho nodo.

Veamos un ejemplo del tipo de análisis que realiza un analizador de red de transición recursiva.

Tomemos la siguiente frase :

"El feliz robot vió un auto con sus ojos"

En la figura 1.6, mostramos el proceso que se sigue para analizar esta frase.

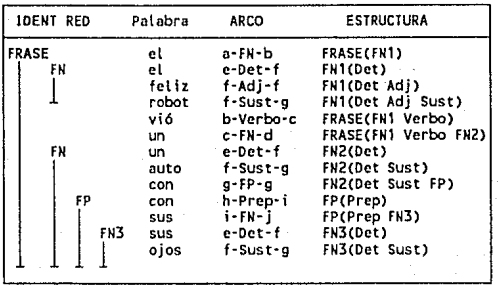


Fig. 1.6.
Funcionamiento de un analizador de red de transición recursiva para la frase: "El feliz robot vió un auto con sus ojos".

En el ejemplo que mostramos en la figura 1.6, el analizador, se sitúa en el nodo inicial [a], de la red **FRASE**, y a partir de él comienza a recorrer la red. Al mismo tiempo comienza a analizar la frase de izquierda a derecha y palabra por palabra. De esta manera, el analizador sigue por el arco **FN** (frase nominal) que, a su vez, define a la subred **FN**.

Ahora bien, la primera palabra de la frase es el determinante "el", por lo que el recorrido de dicha subred pasa del nodo [e], al nodo [f], a través del arco **Det**.

La siguiente palabra de la frase es el adjetivo "feliz", por lo que el recorrido de la subred **FN**, continúa en el nodo [f], a través del arco **Adj**. La siguiente palabra es el sustantivo "robot", y el analizador pasa del nodo [f] al nodo [g], a través del arco **Sust**. La palabra que viene enseguida es el verbo "vió", sin embargo, en este momento la red **FN** sólo puede reconocer una frase preposicional (FP), por lo que el recorrido de dicha red, concluye, y el analizador continúa con la red FRASE en la que se encontraba. El siguiente arco de esta red define a la categoría **Verbo**, que coincide con la categoría de la palabra "vió".

De esta manera el análisis continúa, hasta que ya han sido analizadas todas las palabras de la frase, y el analizador llega a un estado final.

Las redes de transición recursivas, tienen el mismo poder formal de una gramática libre de contexto, pero pueden ser trabajadas de una manera más conveniente.

A continuación describiremos un último tipo de analizador, el analizador de una red de transición aumentada.

1.1.1.3 Analizadores de redes de transición aumentadas.

El proceso que siguen los analizadores de redes de transición aumentadas para analizar y generar la estructura sintáctica de una frase, es casi igual al que siguen los analizadores de redes de transición recursivas. Sin embargo, introducen, entre otros, los siguientes adecuaciones :

a) Definición de rasgos
b) Condiciones y acciones.
c) Clasificación de arcos.

A continuación describiremos y daremos ejemplos que ayuden a aclarar cada uno de estos nuevos conceptos. En lo subsecuente, utilizaremos la abreviación ATN para las redes de transición aumentadas, por sus siglas en inglés (Augmented Transition Network).

a) Definición de rasgos.

Uno de los problemas sintácticos que aparecen al analizar las palabras de una frase, es la concordância de número entre las mismas. Por ejemplo en la frase : "El niños corre", existe un error de concordancia numérica entre el sustantivo "niños" y el verbo "corre".

Para manejar este problema, las gramáticas definidas por una ATN utilizan la definición de rasgos para las categorías sintácticas que aparecen en los arcos de la red. En la figura 1.7, mostramos un ejemplo de los tipos de rasgos que se asocian a estas categorías sintácticas.

CATEGORIA	RASGO	VALORES
Sustantivo	Número	Singular,Plural
Pronombre	Número Persona	Singular,Plural 1a,2a,3a

Fig. 1.7.
Ejemplo del tipo de rasgos que se definen para las categorías sintácticas en una ATN.

Así mismo, en las ATN, se asocian rasgos a las palabras del vocabulario. En la figura 1.8., vemos un ejemplo, en el que aparece un pequeño fragmento de un diccionario que típicamente se maneja en una ATN.

PALABRA	CATEGORIA	RASGOS
robot	Sustantivo	
el	Determinante	Número : singular
los	Determinante	Número : plural
correr	Verbo	Modo : infinitivo

Fig. 1.8.
Ejemplo del tipo de diccionario que se utiliza en una ATN.

b) Condiciones y acciones.

Las condiciones restringen las circunstancias bajo las cuales un arco puede ser tomado. Por su parte las acciones ejecutan operaciones, para construcción de estructuras, y marcado de rasgos.

De acuerdo a los rasgos que cada palabra y cada categoría tienen definidos, podemos asociar acciones en los arcos de una red, que se encarguen de checar la concordancia de número entre las palabras, por ejemplo, entre sustantivos y

determinantes. En la figura 1.9, mostramos un fragmento de una ATN, dado por una subred que analiza un sintagma nominal. En esta subred vemos la manera en que se concuerda el número de las palabras, utilizando acciones.

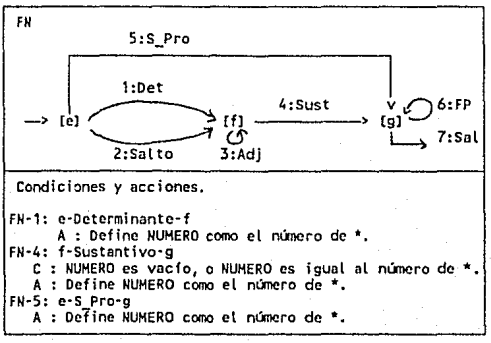


Fig. 1.9.
Ejemplo del manejo de acciones y condiciones en una ATN.

En el ejemplo de la figura 1.9, observamos el uso de las condiciones y acciones. En la primera parte de la figura, mostramos la red que representa la gramática asociada a un sintagma nominal. En esta red, la etiqueta de cada arco, contiene además de la etiqueta en sí, un número que distingue al arco, de los demás. Este número es usado en la segunda parte de la figura, en la que se muestran las acciones y condiciones asociadas a algunos nodos de la red.

Las acciones, en la segunda parte de la figura, se denotan con una "A", mientras que las condiciones se denotan con una "C". La primera de las acciones se refiere al nodo **Det** de la red, el cual es denotado como : **FN-1**. Dicha acción, consiste en definir un registro denotado como NUMERO, en el que se coloca el número del determinante. Si, por ejemplo dicho determinante fuera "el", cuyo número está definido como singular, la acción definiría el registro NUMERO, y le asignaría el valor de singular.

Para el nodo **Sust**, se definen una condición y una acción. La condición restringe el paso por dicho arco a alguna de las dos siguientes condiciones:

- o bien, el registro NUMERO es vacío, lo cual indicaría que ningún determinante fue encontrado, antes del sustantivo,

- o bien, el número del sustantivo, coincide con el número que estaba definido en el registro NUMERO, es decir con el número del determinante.

Por otra parte, la acción definida para este nodo consiste en colocar en el registro NUMERO, el número del sustantivo.

Una acción similar se lleva a cabo con el arco **S_pro.**

En este ejemplo, aparece el concepto de registro. Un registro es una estructura de datos para almacenamiento de información. Este tipo de estructura es utilizada en las ATN, para guardar información no sólo morfológica y sintáctica sino semántica, asociada a las características de las palabras de la frase y a la frase misma.

En la figura que acabamos de describir aparece un nuevo tipo de arco, el arco **Sal.** A continuación describiremos la clasificación de los arcos en una ATN.

c) Clasificación de arcos.

En las ATN, los arcos se clasifican de la siguiente manera :

- Arcos de salto.
- Arcos de salida.
- Arcos de categoría.

- Arcos de salto.

Este tipo de arco corresponde al que definimos para las redes de transición recursivas, y el recorrido se puede hacer a través de ellos sin importar la categoría sintáctica de la palabra que se esté analizando.

- Arcos de salida.

En la figura 1.9, el arco **7:Sal,** es de este tipo. Estos arcos sustituyen al nodo que representa al estado final de la red.

Una ATN puede tener definidos varios de estos arcos, y asociados a los mismos pueden estar definidas una serie de acciones y condiciones que serán utilizadas al terminar el recorrido de la red.

- Arcos de categoría.

Estos arcos son los que hemos venido usando desde que definimos las redes de transición, y representan a las categorías sintácticas de la gramática.

Aunque no es fácil, en general, el codificar las transiciones de una red de transición aumentada, los analizadores de este tipo de redes constituyeron el enfoque más popular, para el análisis sintáctico del lenguaje natural en los años setentas, debido a que permiten un análisis más fino y poderoso.

1.1.1.4 El analizador sintáctico de Andrew.

El analizador sintáctico de nuestro sistema, puede considerarse inmerso en el modelo de redes de transición aumentadas.

El proceso que sigue consiste en analizar una frase palabra por palabra , y de izquierda a derecha. Para reconocer cada una de las palabras de la frase, se basa, en una serie de reglas que constituyen la gramática y que le permiten reconocer las frases válidas, y desechar las inválidas.

El analizador guarda la información referente a género, y número de las palabras, para lanzar en algún momento, procesos que se encargan de validar la concordancia de estos atributos, para las diferentes palabras. Así mismo, es capaz también de manejar la recursividad que se presenta en las frases del tipo "El robot, que está feliz, corre", en la que observamos la aparición de una subfrase , dentro de la frase. Esto es posible, debido a que nuestro analizador es recursivo.

Ahora bien, al mismo tiempo que realiza el análisis sintáctico, el analizador se encarga de ir generando la estructura que representará internamente a la frase. Dicha estructura es una estructura semántica, que describiremos con detalle en el capítulo III.

En realidad, el analizador de nuestro sistema, es como ya habíamos mencionado, un analizador sintáctico-semántico, cuya descripción daremos en el capítulo V.

Para ubicar la parte semántica del analizador, describiremos en la siguiente sección, el modelo semántico de representación de frases, que estamos utilizando.

1.1.2 Gramáticas de casos.

Describiremos ahora, el modelo que Andrew utiliza para representar semánticamente a las frases del sistema. El modelo que utilizamos es el de las gramáticas de casos que fueron diseñadas por Fillmore [1], en 1968. Este enfoque constituye un ejemplo de como el análisis sintáctico puede combinarse con el análisis semántico para reconocer una frase.

Como mencionamos anteriormente, el analizador sintáctico de nuestro sistema utiliza, para efectuar su análisis, una serie de reglas gramaticales. Sin embargo, la estructura que genera para una frase, define una serie de relaciones semánticas y prácticamente elimina los elementos sintácticos.

Para comprender el tipo de análisis que se lleva a cabo, en este modelo, consideremos algunos ejemplos. Tomemos las siguientes frases:

El robot duerme en la casa.	(1)
El robot es de Isaac.	(2)
El robot corre.	(3)

Aunque estas oraciones tienen en común el contar con un sujeto y un predicado, el rol semántico que el sujeto juega en cada una de ellas, es diferente.

- En la oración (1), el "robot" es la fuente de un proceso, dado por el "trabajar" en un lugar : "la casa".

- En la oración (2), el "robot", juega el papel de ser el objeto de una pertenencia, cuya fuente está dada por "Isaac".

- Finalmente en la oración (3), el "robot" es el agente, de la acción de "correr".

La diferencia de roles semánticos, no es exclusiva del sujeto de la frase. Veamos, por ejemplo, las siguientes frases :

El robot fue visto por Isaac.
El robot fue visto por la casa.

Podemos observar que en el predicado de estas frases, aparece un sintagma introducido por la preposición "por", sin embargo el rol semántico que cada uno de estos sintagmas representa, es diferente. En la primera, "Isaac" juega el papel de fuente de la "visión", mientras que en la segunda, la "casa", es el lugar en que se desarrolla el proceso de la "visión".

Una manera de tratar la diferencia de roles semánticos, fue dada por Fillmore en 1968, en un artículo llamado "The case for case", [1]. En él, Fillmore argumentaba que era posible definir una serie de casos, que representaran los distintos roles semánticos que los sintagmas pueden jugar en una frase, y que muchos fenómenos lingüísticos podían ser explicados en términos de una serie de principios generales para la combinación de casos; para ello, postuló un modelo en el que definía los siguientes casos : AGENTE, DATIVO, OBJETO e INSTRUMENTO.

Así mismo definió una serie de reglas generales, que permitían determinar cual, de estos roles, jugaba cada sintagma en una frase, y postuló que los verbos, podían ser agrupados en clases que en general aceptaban distintos tipos de casos.

Más tarde Fillmore publicó una versión modificada, en la que agregaba nuevos casos. A continuación describimos cada uno de estos casos:

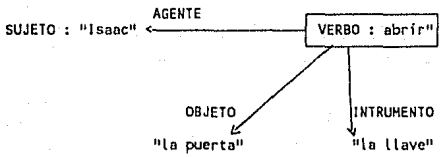
AGENTE : Sujeto animado que lleva a cabo una acción.
INSTRUMENTO : Fuerza u objeto inanimado, con el que se realiza una acción.
DATIVO : Objeto afectado por una acción.

PRODUCTO : Objeto resultante de una acción.
LUGAR : Lugar u orientación.
OBJETO : Cualquier otra cosa.

Las siguientes frases, son ejemplos de la clasificación de los sintagmas de una frase, de acuerdo a estos casos.

-"Isaac abrió la puerta con la llave"

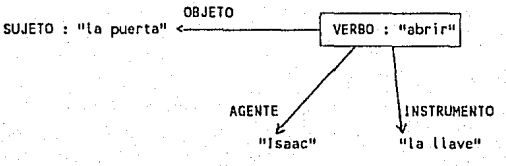
En esta frase, el sujeto , "Isaac", representa al AGENTE de la acción de abrir, por su parte, el sintagma, "la puerta", representa al OBJETO de dicha acción, mientras que "la llave", juega el papel del INSTRUMENTO. Veamos un esquema del verbo y sus casos :



-"La puerta fue abierta por Isaac con la llave"

Esta frase expresa lo mismo que la anterior. La diferencia es que la frase anterior está dada en voz activa, mientras que ésta, está dada en voz pasiva.

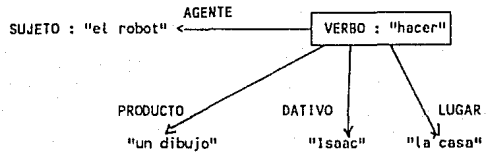
En este ejemplo, el sujeto, "la puerta", juega el papel de OBJETO. Por su parte el sintagma "Isaac" representa el caso de AGENTE, mientras que "las llaves" , juegan el rol de INSTRUMENTO. El diagrama de relaciones semánticas para esta oración, es igual al que vimos en el ejemplo anterior :



En las dos frases anteriores, podemos apreciar que aunque su estructura sintáctica es diferente, en realidad quieren decir lo mismo, por lo que al realizar un análisis basado en el modelo de gramáticas de casos, las estructuras semánticas generadas son iguales.

-"El robot hizo un dibujo para Isaac en la casa"

En esta frase, el sujeto, "el robot", es el AGENTE de la acción de "hacer"; el PRODUCTO de dicha acción es "un dibujo". "Isaac", por su parte, representa al DATIVO de la misma, mientras que "la casa", representa el lugar en que dicha acción se desarrolla.



Una de las características más interesantes de las gramáticas de casos es que el número posible de relaciones semánticas es muy pequeño, aunque, bien a bien, no existe un acuerdo común en cuantos o cuales casos se deben contemplar.

A partir del trabajo de Fillmore, surgieron nuevos desarrollos de este tipo. Uno de los más importantes fue el trabajo de Simmons y Celce [2], en 1973. Ellos definieron un nuevo conjunto de casos, y, basados en dichos casos, hicieron un programa de computación que generaba frases en Inglés.

Los casos definidos por Simmons y Celce, son los siguientes:

- CAUSAL-ACTOR
- TEMA
- LUGAR
- FUENTE
- META

El primer caso : CAUSAL-ACTOR, engloba a los casos INSTRUMENTO y AGENTE de Fillmore. El caso TEMA, es utilizado para cubrir una variedad de relaciones semánticas, incluyendo al caso OBJETO de Fillmore.

Ahora bien, para generar la estructura semántica que representa a las frases en nuestro sistema, nos basamos en las de gramáticas de casos. Para ello utilizamos los casos definidos en los trabajos de C. Loyo [3], y G. Sabah [4], quienes a su vez utilizaron, entre otros, los trabajos de Fillmore , Simmons y Celce-Murcia [5].

En el capítulo III, daremos una descripción de la manera en que clasificamos a los verbos y del tipo de relaciones semánticas o casos que ligan a los sintagmas nominales de la frase, con el verbo de la misma.

II. El Sistema.

Como explicamos en la introducción, el principal objetivo de nuestro trabajo consiste en el desarrollo de un analizador sintáctico-semántico de frases en lenguaje natural, basado en el modelo de gramáticas de casos.

Con el objeto de probar los resultados que ofrece nuestro analizador, desarrollamos un sistema de preguntas y respuestas en lenguaje natural, en el que nuestro analizador es precisamente quien se encarga del análisis de las frases. Este sistema es llamado Andrew, y a continuación daremos una descripción del mismo.

2.1 Descripción del sistema

Andrew es un sistema que permite al usuario dar una o más frases en español para posteriormente contestar a una serie de preguntas que dicho usuario realize, con respecto a las frases dadas. Para hacer ésto, el sistema analiza y traduce cada frase a una representación interna adecuada, que posteriormente será usada para dar respuesta a las preguntas del usuario.

Es importante aclarar que Andrew considera a las frases que le da el usuario, como frases aisladas, y no establece ninguna relación entre ellas. Sin embargo, existe una cierta complejidad dentro del tipo de frases que Andrew es capaz de reconocer.

Los siguientes son ejemplos del tipo de frases, que Andrew es capaz de reconocer :

"El robot está feliz",
"El niño dió las frutas a Isaac",
"El feliz robot de Isaac corre por la casa con sus piernas de metal",

"El hombre, que estaba en la cocina, sirvió la comida a sus amigos con los cubiertos",
"El robot pensaba que Isaac había estado feliz ".
"Los amigos del robot estuvieron ayer en el laboratorio".
"Las frutas fueron comidas por el niño".

Como podemos observar, el sistema es capaz de reconocer algo más que frases simples, aceptando, inclusive, frases en voz pasiva.

Cada una de estas frases, es analizada independientemente de las demás, y es traducida a una estructura semántica que la representa en el sistema. Un esquema general de la estructura semántica que el sistema utiliza para representar una frase, está dado en la figura 2.1.

La manera en que definimos una frase en nuestro sistema, es como sigue :

" Una frase es una aserción, formada por un sujeto y un predicado, que a su vez, están dados por una serie de sintagmas nominales. Dichos sintagmas están ligados al calificativo de la frase, por un conjunto de relaciones semánticas, llamados CASOS de la frase".

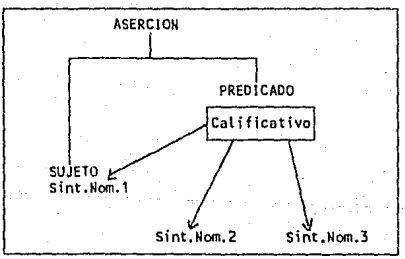


Fig. 2.1.
Esquema de la estructura semántica de una frase.

Ahora bien, el tipo de preguntas que Andrew puede responder, son preguntas directas, es decir, preguntas que de alguna manera piden información que está contenida en la representación de las frases dadas. En seguida, mostramos algunos ejemplos del tipo de frases interrogativas que Andrew reconoce :

"Cómo está el robot?",
"A quién dió las frutas el niño?",
"De quién es el robot?",
"Con qué corre el robot?",
"Qué pensaba el robot?",
"Cuándo estuvieron los amigos del robot en el laboratorio?",
"Quiénes estuvieron en el laboratorio?",
"Es cierto que el robot corre por la casa?",
"Cuál hombre sirvió la comida?".

Las frases interrogativas son analizadas de la misma manera en que se analizan las frases declarativas, y, al igual que a éstas, el sistema les genera una estructura semántica. Sin embargo cuando el sistema reconoce una frase como interrogativa, ejecuta además una serie de procedimientos que se encargan de buscar, entre las frases ya dadas, alguna que constituya o contenga una respuesta adecuada. En caso de existir, dicha respuesta se traduce al español y se presenta al usuario.

El sistema es capaz de establecer una relación entre frases que giran alrededor de un mismo concepto, lo cual le permite contestar a preguntas como : "Qué sabes del robot?". Sin embargo no maneja pronombres, por lo que no establece relacionas semánticas entre las frases, que le permitan hacer razonamientos deductivos o contestar a preguntas del tipo de "Porqué ...?".

El relacionar las frases, para posteriormente responder preguntas sobre las mismas , es un proceso muy complejo que en este trabajo no se realiza, sin embargo, los modelos de representación que hemos utilizado pueden servir como base para una segunda versión, en la que éste y otros problemas que surgen al reconocer el lenguaje natural, sean considerados.

Nuestro sistema está constituído por cuatro componentes principales:

- Modelo lingüístico
- Memoria
- Conjunto de operadores
- Interfaz con el usuario

A continuación, y, a manera de introducción, describiremos brevemente cada uno de estos componentes.

2.1.1 El modelo lingüístico.

El modelo lingüístico del español que utilizamos dista de representar una gramática completa del lenguaje. En realidad, manejamos únicamente, los elementos lingüísticos necesarios para el análisis sintáctico y la interpretación semántica, de las frases que el sistema es capaz de reconocer.

El modelo lingüístico comprende :

a) Un conjunto de reglas gramaticales que permiten reconocer y generar las frases en el sistema.

b) Un modelo semántico de estructura interna, que permite representar a una frase dentro del sistema.

c) Un conjunto de categorías gramaticales que definen una clasificación sintáctica de las palabras que componen el lenguaje y que son utilizadas durante el análisis.

En el siguiente capítulo describiremos, con más detalle, cada una de estas categorías, así como el modelo semántico de representación de frases.

2.1.2 La memoria.

Definimos como memoria del sistema, al conjunto de conocimientos con los que el sistema cuenta para poder, en determinado momento, entablar un diálogo con el usuario. La memoria está estructurada en dos partes :

- Vocabulario.
- Conjunto de representaciones internas de frases.

El vocabulario es el conjunto de conceptos que Andrew conoce y utiliza para comunicarse con el usuario.

Además del vocabulario, la memoria contiene el conjunto de estructuras semánticas que representan, dentro del sistema, a cada una de las frases dadas por el usuario.

En el capítulo IV, explicaremos la manera en que el vocabulario es representado en nuestro sistema.

2.1.3 Los operadores.

Los operadores del sistema, son un conjunto de procedimientos que actúan sobre la memoria, para entablar el diálogo con el usuario y responder a sus preguntas.

Los operadores del sistema son de dos tipos :

- Aquellos que enriquecen la memoria, y
- aquellos que explotan los conocimientos de dicha memoria.

En el primer tipo se encuentran los operadores, que agregan conocimiento a la memoria, como por ejemplo, el analizador sintáctico-semántico, que crea estructuras semánticas a partir de las frases dadas por el usuario, y agrega estas estructuras a la memoria.

En el segundo tipo, se encuentran los operadores que únicamente toman información de la memoria sin modificarla. Ejemplos de este tipo de operadores son, él que convierte estructuras semánticas de la memoria a frases en español, o él que responde a una pregunta.

Entre las principales tareas de los operadores del sistema, se encuentran las siguientes :

a) Recepción de las frases y conceptos, dados por el usuario.

b) Análisis y generación de estructuras semánticas para las frases que da el usuario.

c) Traducción de las estructuras semánticas a frases en español.

d) Generación de respuestas para las preguntas que el usuario hace con respecto a la información dada.

e) Búsqueda y exposición de los conocimientos con los que Andrew cuenta, acerca de algún concepto, en particular.

f) Aprendizaje de nuevas palabras.

En la siguiente sección hablaremos de los principales operadores de Andrew, así como de su funcionamiento.

2.1.4 La interfaz.

El sistema presenta una interfaz, que permite al usuario entablar el diálogo con Andrew de una manera sencilla. Esta interfaz presenta un menú, que permite las siguientes opciones :

a) dialogar con el sistema, por medio de preguntas y respuestas,

b) crear y editar archivos de frases,

c) traducir archivos de frases, a su correspondiente representación interna,

d) aprender nuevos conceptos,

e) desplegar la estructura semántica generada para una frase,

f) desplegar los conocimientos con los que Andrew cuenta sobre algún concepto en particular.

En el capítulo VIII, describiremos esta interfaz.

2.2 El funcionamiento del sistema.

Como acabamos de mencionar, nuestro sistema cuenta con un menú. De las opciones que dicho menú presenta, la más importante es la que permite el diálogo con el sistema.

Al nivel del diálogo, nuestro sistema puede ser visto como un conjunto de operadores que actúan sobre la memoria. A continuación veremos cuales son los principales operadores del sistema.

2.2.1 Los operadores de diálogo.

Los siguientes, son los operadores que intervienen en el diálogo entre Andrew y el usuario :

- **MONITOR** : Este operador dirige el funcionamiento del sistema de acuerdo al tipo de las frases dadas por el usuario.

- **ANALIZADOR** : Este operador se encarga de analizar las frases dadas en español, y de generar una estructura semántica que las represente en el sistema.

- **GENERADOR** : Este operador efectúa el proceso inverso, al del ANALIZADOR, es decir, genera a partir de una estructura semántica, la frase en español.

- **CONTESTADOR** : Este operador se encarga de generar respuestas a las preguntas que se hacen durante el diálogo.

- **AUXILIARES** : Un conjunto de operadores que se encargan de efectuar algunas otras funciones auxiliares para el diálogo.

En la figura 2.2, mostramos un diagrama de la manera en que estos operadores interaccionan. Como podemos observar, el **MONITOR**, es el operador que se encarga de dirigir las acciones a realizar, dependiendo del tipo de las frases dadas por el usuario. Dicho tipo puede ser declarativa o interrogativa.

Cuando el usuario elige la opción de dialogar con el sistema, el operador **MONITOR** se activa y llama al operador de lectura de frases, que está contenido en el

conjunto de AUXILIARES del sistema. Una vez que se ha leido una frase, el **MONITOR**, llama al **ANALIZADOR** para que convierta dicha frase a una representación interna adecuada.

Después de ésto, y dependiendo del tipo de frase llama a los operadores adecuados.

Si, por ejemplo, la frase es interrogativa, el MONITOR llama al **CONTESTADOR**, para que busque la respuesta y posteriormente, llama al **GENERADOR**, para que genere dicha respuesta en español.

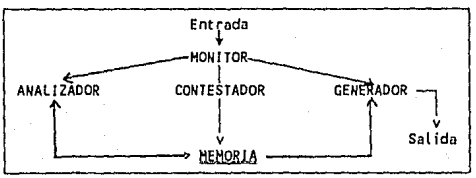


Fig. 2.2.
Operadores de diálogo del sistema.

En la figura 2.2, observamos también la manera en cada operador actúa sobre la memoria. El **ANALIZADOR**, toma de la memoria, las categorías gramaticales que definen cada una de las palabras de la frase y genera una estructura semántica para ésta que a su vez pasa a formar parte de la memoria del sistema. Por su parte, el **CONTESTADOR** busca en la memoria, las posibles respuestas a las preguntas hechas por el usuario, y genera la respuesta de forma adecuada, guardando temporalmente la representación interna de las mismas en la memoria. Finalmente el **GENERADOR** toma dichas estructuras de la memoria y la traduce al español para presentarlas al usuario.

Además de los operadores que permiten el diálogo entre el usuario y el sistema, Andrew cuenta con una serie de operadores que se encargan de manejar las diferentes opciones que el menú del sistema ofrece. En los siguientes capítulos describiremos únicamente los operadores más importantes que son el ANALIZADOR, el GENERADOR y el CONTESTADOR.

III. El modelo lingüístico utilizado.

El modelo lingüístico utilizado por nuestro sistema se divide en dos partes. La primera consiste en definir las diferentes categorías gramaticales que Andrew utiliza para clasificar las palabras del vocabulario. La segunda parte describe las estructuras semánticas que utilizamos para representar las frases dentro del sistema.

Pasaremos ahora a definir con más detalle estas dos fases del modelo lingüístico.

3.1 Las categorías gramaticales.

Las categorías gramaticales son de suma importancia en el análisis de las frases.

Como mencionamos en el capítulo anterior, en nuestro sistema, manejamos las siguientes categorías :

- Sustantivo
- Determinante
- Pronombre
- Calificativo
- Adverbio

La categoría sustantivo, engloba a todas las palabras que se refieren a un concepto determinado.

Por su parte la categoría de determinante, contiene a todas aquellas palabras que de alguna manera determinan al sustantivo en un sintagma. La categoría de determinante contiene, a su vez, un conjunto de subcategorías. A este conjunto pertenecen los determinantes demostrativos, posesivos, indefinidos, numerales y artículos. Veamos algunos ejemplos de estos determinantes :

- Demostrativos : esta, estos, esos, aquel, aquellos,...
- Posesivos : su, sus, mi, mis, tu, tus,....
- Indefinidos : algún, ningún, cierto, mismo, otro,...
- Numerales : dos, tres,...
- Artículos : el, la, los, las, un, una, unas, unos,...

En cuanto a los pronombres, nuestro sistema únicamente maneja el pronombre relativo "que" y algunos pronombres interrogativos, pues no hacemos uso del contexto que rodea a la oración y no manejamos la sustitución de un pronombre personal por el sustantivo al que reemplaza.

En lo que respecta al pronombre relativo "que", el sistema los reconoce como liga entre el concepto central de un sintagma y una subfrase que lo flexiona, como por ejemplo, en el sintagma, "El niño, que es feliz", en la que el pronombre relativo "que", liga a la subfrase "es feliz", con el sintagma "el niño".

El calificativo, representa la afirmación de una cualidad o atributo, del concepto central del sujeto de la frase. Si pensamos que una cualidad asociada a un sustantivo, puede ser dada por el hecho de estar o ser de tal o cual manera, podemos

definir tres tipos de calificativos :

- Preposición
- Adjetivo calificativo
- Verbo

Los siguientes ejemplos muestran el uso de estos calificativos:

- "El robot está en el campo",
- "El robot está feliz",
- "El robot está cansado",

Nuestro sistema hace uso de la mayoría de las preposiciones, por ejemplo : a, con, bajo, de, desde,...

Por lo que respecta a los adjetivos calificativos, el sistema tiene definidos inicialmente algunos adjetivos más o menos comunes. Por comodidad, en el resto del texto nos referiremos a los adjetivos calificativos, simplemente como adjetivos.

Finalmente viene lo que consideramos la palabra más importante de la oración, el verbo. Esto se debe a que de acuerdo al tipo de verbo que funcione como núcleo del predicado en una oración, podemos determinar las relaciones semánticas que se presentan entre el verbo y los diferentes sintagmas nominales de la frase.

En nuestro sistema, consideramos tres tipos de verbos :

1) Verbos de ACCION (correr, bailar, hacer,..)
2) Verbos de ESTADO (estar, ser, parecer,..)

3) Verbos de PROCESO (pensar, ver, creer,..)

Esta clasificación de los verbos, facilita la tarea de encontrar el tipo de relación semántica que liga al verbo, con cada uno de los sintagmas nominales de la frase. Más adelante, en este mismo capítulo, describiremos el tipo de casos o relaciones semánticas que admite, cada uno de estos verbos.

Es conveniente aclarar que, en realidad, los verbos que definimos como verbos de estado, no sólo representan un estado, como ocurre, por ejemplo, en la frase "El robot es feliz", sino que pueden también definir una relación de pertenencia, como, por ejemplo, en la frase "El robot es de Isaac", o inclusive una relación de coincidencia con otro estado, por ejemplo , "El niño parece triste", en la que no es un hecho que el niño esté triste, sino que únicamente lo parece. A pesar de estos diferentes usos de este tipo de verbos, decidimos trabajarlos como verbos de estado, por ser ésta la característica que, más comúnmente, los distingue.

Para terminar, nuestro modelo lingüístico, utiliza la categoría de ADVERBIO. Los adverbios son palabras que califican a los adjetivos, a los verbos o a ellos mismos. Andrew tiene definidos, en su vocabulario, a los adverbios más utilizados en un diálogo. En seguida mostramos, la manera en que estos adverbios son clasificados :

Adverbios de:

- modo (bien, rápido, despacio,..),
- tiempo (ahora, ayer, mañana, hoy,..),
- lugar (aquí, cerca, lejos, dentro,..),
- cantidad (mucho, demasiado, poco, nada, muy..),

Existen otro tipo de adverbios como son, los adverbios de negación, sin embargo nuestro sistema no maneja la negación en las frases por lo que este tipo de adverbios no son reconocidos en el sistema.

El sistema tiene dos maneras de aprender nuevas palabras. O bien el usuario define una serie de nuevos conceptos, o bien, si durante el diálogo, alguna palabra no es conocida, el sistema pide su definición y la agrega al vocabulario, continuando después con el diálogo.

3.2 La estructura semántica de la frase.

En el capítulo I, explicamos que el modelo en él que nuestro sistema se basa para generar la estructura semántica que representa a las frases en el sistema, es el de las gramáticas de casos.

En el modelo de las gramáticas de casos, una frase es representada por medio de una estructura semántica, en la que el verbo está relacionado semánticamente con cada uno de los diferentes sintagmas nominales que aparecen en la frase.

En esta sección describiremos la manera en que se manejan los sintagmas nominales dentro de nuestro sistema. Así mismo, explicaremos cada una de las relaciones semánticas o casos, que Andrew utiliza en el análisis de las frases y en su conversión a una estructura semántica.

3.2.1 El sintagma nominal.

Un sintagma nominal, es un sintagma que necesariamente contiene un sustantivo, que a su vez puede ser precedido por un determinante. Por ejemplo :

- "el robot",
- "sus robots",
- "dos robots",

son diferentes determinaciones del sustantivo robot.

En un sintagma puede aparecer más de un determinante, o bien ninguno. Por ejemplo :

- "el otro robot",
- "Isaac".

Algunas veces entre el determinante y el sustantivo, o bien, después del sustantivo, encontramos adjetivos o frases preposicionales, que lo modifican, como podemos ver en los siguientes ejemplos:

- "el robot feliz" (1)
- "el robot de Isaac" (2)

La aparición de un adjetivo o de una preposición en el sintagma nominal, define una inflexión para el sustantivo. Un sintagma nominal que cuenta con una o más inflexiones puede ser visto como una frase en la que el verbo "ser", no está explícito pero se sobreentiende. Por ejemplo, los sintagmas (1) y (2), pueden ser vistos como las siguientes frases :

- "el robot es feliz" (3)
- "el robot es de Isaac" (4)

Las sustantivos pueden, también, ser flexionados por una frase como en los siguientes ejemplos :

- "los niños cansados de caminar" (5)
- "el robot que ve las máquinas" (6)

En el ejemplo (5), la subfrase : "cansados de caminar", es una inflexión del concepto del sintagma, "niños". Dicha inflexión es, a su vez, considerada como una frase :

"los niños están cansados de caminar" (7)

Como apreciamos en los ejemplos (3),(4) y (7), el sistema representa a las diferentes inflexiones de un sintagma, como frases ligadas a dicho sintagma. Para generar esta representación de una manera eficiente, varios de los procedimientos del analizador son recursivos.

En la figura 3.1, mostramos un diagrama de la manera en que se manejan las inflexiones en un sintagma nominal. Dicho diagrama se refiere en particular a la representación del sintagma : "El feliz robot de Isaac".

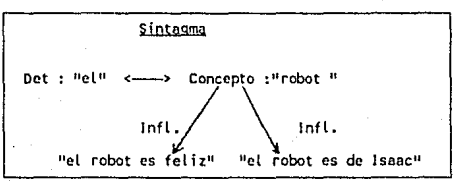


Fig. 3.1.
Ejemplo de representación de inflexiones para el sintagma :
"el feliz robot de Isaac".

En el capítulo V , veremos como, cada vez que el analizador del sistema encuentra una inflexión en un sintagma, se genera una nueva estructura semántica para representar a dicha inflexión.

3.2.2 Los casos de la estructura semántica.

Basado en los trabajos de C. Loyo [3] y G. Sabah [4], nuestro sistema utiliza siete casos, para definir las relaciones semánticas existentes entre los diferentes sintagmas nominales de una frase, y el verbo de la misma. Dichos casos son los siguientes :

- FUENTE
- AGENTE
- TEMA
- DESTINO

- AGENTE SECUNDARIO
- LUGAR
- TIEMPO

La aparición de estos casos en una frase, depende del tipo de verbo de la misma. Como vimos en la sección 1, en nuestro sistema, los verbos se clasifican como verbos de :

- Acción
- Proceso
- Estado

A continuación, describiremos el tipo de relación semántica que define cada uno de estos casos, para posteriormente dar ejemplos de los diferentes casos que cada tipo de verbo acepta.

3.2.2.1 Descripción de casos.

En la figura 3.2, mostramos la descripción de cada uno de los casos que nuestro sistema utiliza, junto con una frase en la que un ejemplo del caso que se describe, aparece subrayado.

CASO	DESCRIPCION
Fuente	Paciente de un proceso o un estado. ("El niño ve el libro")
Agente	Ejecutante de una acción. ("El robot corre")
Tema	Objeto sobre el que se ejecuta una acción o proceso. ("Los niños comen las frutas")
Destino	Receptor de una acción. ("Isaac dió el libro al robot")
Agente secundario	Instrumento con que se realiza una acción. ("Isaac comió con las manos"
Lugar	Entorno en que se desarrolla una acción, proceso o estado. ("El robot estuvo en la casa")
Tiempo	Tiempo en que se desarrolla una acción, proceso o estado. ("Isaac vino ayer")

Fig. 3.2.
Descripción de los casos utilizados en el sistema.

Cabe aclarar que el caso TIEMPO, no es utilizado en el sistema para manejar temporalidad, es decir, para definir si algún hecho tomó lugar antes o después de otro. Este caso es utilizado para guardar información respecto al momento en que se desarrolla alguna frase independientemente de las demás.

Como ya dijimos, el tipo de casos que pueden aparecer en una frase, depende del tipo de verbo de la misma. Veamos con detalle que casos acepta cada tipo de verbo.

Verbos de estado. (ser, estar, parecer)

Los casos que acepta un verbo de estado son los siguientes :

- FUENTE
- TEMA

- LUGAR
- TIEMPO
- AGENTE SECUNDARIO

A continuación daremos ejemplos de frases cuyo verbo es un verbo de estado, y de los diferentes casos definidos en dichas frases.

- "El robot era muy feliz"

En este caso el sujeto de la frase, "el robot", es considerado la FUENTE del estado, mientras que el adjetivo "feliz" es considerado como el CALIFICATIVO de la frase y el verbo "ser" simplemente es utilizado para dar el tiempo de la frase, que en este caso es copretérito. Por último el adverbio de cantidad, "muy", es considerado como un atributo del calificativo.

- "El robot está en la casa"

En este caso el sujeto de la frase, "el robot", es considerado la FUENTE del estado; el CALIFICATIVO está dado por la preposición "en" y el sintagma "la casa", ocupa el caso de LUGAR. Una vez más, el verbo de la frase "estar", es utilizado únicamente para obtener el tiempo de la frase, que en este caso es presente.

- "El robot es de Isaac"

En este ejemplo, el verbo "ser", introduce una relación de pertenencia entre "Isaac" y "el robot" y, en este caso, sí juega el papel de CALIFICATIVO de la frase. Por su parte, el sintagma "el robot" es considerado como el TEMA de la pertenencia mientras que "Isaac", es considerado como la FUENTE de la misma.

- "Andrew es un robot"

En este caso el sujeto de la frase "Andrew", es considerado la FUENTE del "ser", mientras que el sintagma "un robot", juega el rol de TEMA, por lo que el verbo "ser" es considerado como el CALIFICATIVO de la frase.

En el caso de los verbos de estado, el caso de AGENTE SECUNDARIO no se maneja en su papel de instrumento, sino únicamente como acompañante. Al final de esta sección hablaremos de este caso.

Verbos de acción. (correr, comer, bailar, hacer,...)

Este tipo de verbos, aceptan los siguientes casos :

- AGENTE
- TEMA
- DESTINO
- LUGAR
- TIEMPO
- AGENTE SECUNDARIO

A continuación veremos algunos ejemplos de frases en las que aparece un verbo de acción.

- " El robot corría ayer por la casa"

En este ejemplo, el sujeto de la frase, "el robot", es considerado como el AGENTE de la acción de "correr". El sintagma "la casa", es considerado como el

LUGAR en que dicha acción se realiza, y el verbo "correr" es definido como el CALIFICATIVO de la frase. Por último el adverbio de tiempo "ayer", se define como el TIEMPO en que se ejecuta la acción de "correr".

- "Isaac dió los libros al robot con sus manos"

En este ejemplo el CALIFICATIVO de la frase, está dado por el verbo "dar". El sujeto de la misma, "Isaac", es el AGENTE de la acción de dar. El sintagma "los libros" es considerado como el TEMA de dicha acción, mientras que el sintagma "el robot", es considerado como el DESTINO de la misma. Finalmente el sintagma "sus manos", es definido como el AGENTE SECUNDARIO, por ser el objeto del que el agente se vale para efectuar la acción de "dar".

- "La fruta fué comida por el niño"

Esta es una frase en voz pasiva. El verbo "comer", es considerado el CALIFICATIVO de la frase. El sujeto de la misma, "la fruta", es considerado como el TEMA de la acción de "comer", mientras que "el niño", juega el papel de AGENTE de dicha acción.

Verbos de proceso. (dormir, ver, creer, ...)

Este tipo de verbo, acepta los siguientes casos :

- FUENTE
- TEMA

- LUGAR
- TIEMPO
- AGENTE SECUNDARIO

Veamos algunos ejemplos.

- "Isaac durmió ayer en la sala"

En esta frase, "Isaac", el sujeto, es considerado como la FUENTE del proceso de "dormir". Dicho verbo es definido como el CALIFICATIVO de la frase, mientras que el adverbio de tiempo "ayer" y el sintagma "la casa", dan respectivamente, el TIEMPO y el LUGAR en que el proceso de "dormir" se desarrolla.

- "El robot vió que las frutas eran grandes"

En este caso "el robot", es la FUENTE del proceso de "ver", mismo que es el CALIFICATIVO de la frase. Como podemos observar, dicho verbo introduce a una frase, "las frutas eran grandes". Esta frase es analizada y definida como el TEMA de la visión.

El procedimiento que el sistema utiliza para decidir el tipo de relación semántica que liga al verbo de una frase, con cada uno de sus sintagmas, consiste en analizar primero el tipo de verbo de la frase, y posteriormente analizar cada sintagma y la preposición que lo introduce en la frase.

Existe un tipo de sintagma nominal en particular, que escapa a nuestra definición de casos. Veamos un ejemplo :

- "El robot está con Isaac"

En esta frase, el sintagma "Isaac", está introducido por la preposición "con". Esta preposición, en general, introduce a sintagmas que funcionan como AGENTES SECUNDARIOS, sin embargo, en esta frase, "Isaac" no juega el papel de instrumento, sino de acompañante. Debido a la falta de una caso que represente a este sintagma, en nuestro sistema, manejamos a este tipo de sintagmas como agentes secundarios, aunque su sentido semántico no sea precisamente el indicado por dicho caso.

IV. La memoria del sistema.

En el capítulo II, mencionamos que la memoria de Andrew, es decir los conocimientos con los que cuenta, están dados por su vocabulario y por una serie de estructuras semánticas, que representan a las frases que han sido dadas por el usuario.

Daremos, en este capítulo, una descripción del tipo de conocimientos que Andrew necesita y utiliza para establecer la comunicación con el usuario. Explicaremos también la manera en que se representa el vocabulario en nuestro sistema y finalmente, describiremos las estructuras de datos que utilizamos para representar las frases dentro del sistema.

4.1 Los conocimientos.

Como ya sabemos, Andrew, establece un diálogo con el usuario. Para ello, es necesario que cuente con una serie de conocimientos previamente definidos, que le permitan la comunicación inicial. Sin embargo, al igual que cualquiera que participe en un diálogo, Andrew puede ir aprendiendo nuevos conceptos y propiedades de los mismos, en el transcurso de dicho diálogo.

La memoria de Andrew, contiene conocimientos de dos tipos :

a) Los conocimientos-predefinidos
b) Los conocimientos-adquiridos.

4.1.1 Los conocimientos predefinidos.

Es evidente que un cierto número de conocimientos predefinidos, es necesario para que nuestro sistema sea capaz de manipular las frases y la información que el usuario le da en algún momento.

Estos conocimientos predefinidos , están dados por un conjunto de palabras, a las que consideramos el vocabulario inicial, y por una serie de propiedades de dichas palabras.

Así pues, este vocabulario inicial comprende :

a) Las diferentes categorías gramaticales que describimos en el capítulo anterior.
b) Una serie de palabras utilizadas comúnmente en español como artículos, preposiciones, el verbo auxiliar "haber" y algunos adverbios.

Además de estas palabras que de alguna manera están ya fijas en la memoria de Andrew, tenemos algunos verbos más o menos comunes, como los verbos ser y estar, y un conjunto de sustantivos que comúnmente aparecen en las historias de ciencia ficción, que constituyen el universo de trabajo de Andrew.

4.1.2 Los conocimientos adquiridos.

Los conocimientos adquiridos, están dados por la información que el usuario da al sistema durante el diálogo. Los conocimientos que la memoria puede adquirir son de dos diferentes tipos :

a) Nuevos conceptos
b) Nuevas frases

Los nuevos conceptos que son dados a Andrew, durante el diálogo , son colocados en el vocabulario. Por su parte, las frases son guardadas en la memoria, en forma de estructuras semánticas.

4.2 La organización del vocabulario.

En nuestro sistema, el vocabulario está definido mediante una estructura de árbol, en el que cada nodo representa un concepto, que a su vez tiene asociado un conjunto de propiedades que lo definen, y un conjunto de atributos morfológicos de la palabra que dicho nodo representa.

La estructura de árbol que representa al vocabulario del sistema, está constituída por dos niveles principales : un nivel sintáctico y un nivel semántico.

4.2.1 El nivel sintáctico.

En este nivel se encuentran todas aquellas palabras del árbol cuyo papel es el de agrupar a otras palabras, de acuerdo a las categorías gramaticales a las que pertenecen. Estas palabras no aparecen en el transcurso de la conversación con el usuario, de hecho, Andrew hace una distinción entre categorías y palabras del vocabulario. Por ejemplo, Andrew distingue la categoría ARTICULO de la palabra "artículo". En la figura 4.1, mostramos un diagrama del nivel sintáctico del árbol que representa al vocabulario de nuestro sistema.

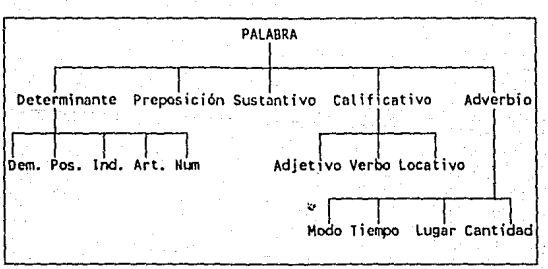


Fig. 4.1.
Nivel sintáctico del árbol que representa al vocabulario.

4.2.2 El nivel semántico.

Además del nivel sintáctico, el árbol que representa al vocabulario, contiene un nivel semántico. En este nivel cada nodo del árbol define un concepto, es decir, contiene información del tipo semántico. Los elementos del árbol, a este nivel, están jerarquizados de acuerdo a su significado.

En la figura 4.2, mostramos un diagrama del tipo de jerarquización que realizamos a nivel semántico.

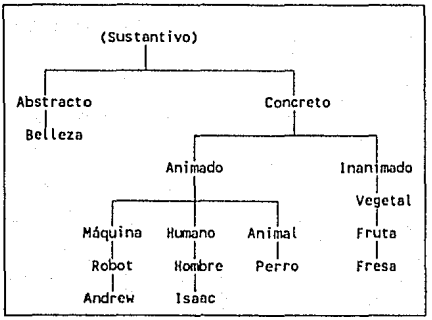


Fig. 4.2.
Nivel semántico del árbol que representa al vocabulario del sistema.

Cada nodo del árbol que representa a un concepto, tiene asociado un conjunto de propiedades semánticas, que son utilizadas para generar la estructura semántica que representa a la frase dentro del sistema.

Veamos un ejemplo. La palabra "casa", tiene asociada la propiedad de ser un lugar. Esta propiedad es utilizada por el analizador del sistema al definir los casos que aparecen en una frase. De esta manera, al encontrar que el sustantivo "casa" tiene asociada la propiedad de ser un lugar, el sintagma que lo contiene será un candidato a representar el caso de LUGAR.

Es importante aclarar, que en nuestro sistema, el árbol es utilizado únicamente a nivel sintáctico. El manejo del nivel semántico, incluyendo la herencia de propiedades semánticas, no se realiza en esta versión del sistema. Sin embargo la estructura de árbol que utilizamos, permite que en el futuro pueda ser manejado dicho nivel, sin cambiar el diseño del sistema, lo cual sería util para manejar razonamiento deductivo.

Ahora bien, además del árbol sintáctico y semántico que representa al vocabulario, el sistema cuenta con un diccionario en el que se encuentran definidas las palabras del vocabulario junto con una serie de atributos morfológicos.

El árbol y este diccionario están relacionados entre sí, es decir existe una liga entre algunos nodos del árbol y su entrada en el diccionario.

De esta manera, cuando el sistema analiza una frase, cada palabra de la misma es buscada en el diccionario, para posteriormente ir al nodo del árbol que representa a dicha palabra y recorrer el árbol hasta encontrar la categoría sintáctica a la que la palabra pertenece.

En el diccionario, las palabras tienen asociados, una serie de atributos que se refieren básicamente a la morfología de la palabra , es decir, a la manera en que la misma se construye de acuerdo a género y número. Estos atributos son usados para reconstruír una palabra a partir de su raíz.

Uno de estos atributos es el género. Para representar el tipo de las palabras de acuerdo a su género, el sistema maneja las siguientes clasificaciones:

- **dob** (**dob**le) : palabra cuya terminación varía dependiendo del género, por ejemplo "niño", "niña",

- **unf** (**ún**icamente **f**emenino) : palabra cuyo género solo puede ser femenino, por ejemplo, "mesa",

- **unm** (**ún**icamente **m**asculino) : palabra cuyo género solo puede ser masculino, como "libro",

- **inv** (**inv**ariante) : palabra cuya forma es independiente del género, por ejemplo "robot",

- **pro** (**pro**pio) : sustantivo propio.

Otro de los atributos morfológicos, es el número. Este puede ser singular o plural. Volviendo al ejemplo, de la palabra "casa", ésta tiene asociados los atributos unf, pues es una palabra cuyo género sólo puede ser femenino, mientras que su número es, por omisión, singular.

En la figura 4.3, vemos la representación de la palabra "casa" en el vocabulario del sistema.

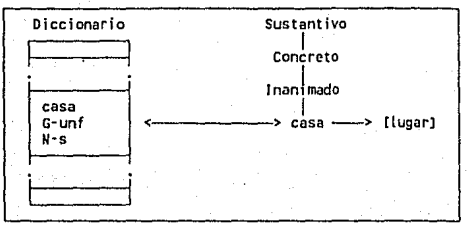


Fig. 4.3.
Representación de la palabra "casa" en el vocabulario y su liga dentro del árbol.

Es de esta manera como representamos el vocabulario que Andrew puede manejar para comunicarse con el usuario. Más adelante, en el capítulo 7, veremos la manera en que esta representación se codifica en el lenguaje de programación Prolog.

4.3 Estructuras de representación de frases.

En el capítulo anterior, describimos el modelo semántico que el sistema utiliza para generar la representación de una frase. En esta sección, describiremos las estructuras de datos que utilizamos para manejar dicha representación. Dichas estructuras son las

siguientes :

a) estructura sint (sintagma nominal),
b) estructura prd (predicado),
c) estructura ora (oración).

A continuación describimos, cada una de estas estructuras.

4.3.1 La estructura sint.

Utilizamos la estructura sint, para representar un sintagma nominal. De esta manera, cada sintagma nominal de la frase, será analizado y traducido a una estructura de tipo sint.

Como sabemos, un sintagma nominal tiene como elemento central a un sustantivo. Este sustantivo, va en general acompañado de una serie de determinantes que lo califican. Así mismo , puede tener una o más inflexiones como ocurre, por ejemplo, en el sintagma : " El feliz niño de Isaac". En este sintagma, el concepto central "niño", está flexionado por ser "feliz" y ser "de Isaac". Para representar a las inflexiones de un sustantivo, el sistema, genera una subfrase para cada inflexión que se presente y utiliza, a su vez, una estructura ora (oración), para representar cada subfrase.

Para representar internamente un sintagma nominal, utilizamos la estructura sint que se muestra en la figura 4.4.

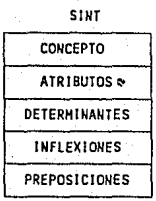


Fig. 4.4.
Estructura sint, para representación de sintagmas nominales.

En esta figura apreciamos los siguientes elementos:

CONCEPTO : Este elemento contiene al concepto central del sintagma nominal, por lo que es siempre un sustantivo.

ATRIBUTOS : Este elemento contiene al conjunto de atributos morfológicos del sintagma. Estos atributos, son género, número y persona, y están dados por los determinantes y el concepto del sintagma.

DETERMINANTES : En este elemento, están definidos los determinantes, que califican al sustantivo, es decir, al concepto del sintagma nominal.

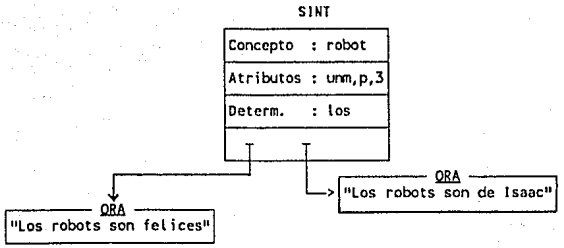
INFLEXIONES : Este elemento contiene, una lista de estructuras ora, que representan cada inflexión.

PREPOSICIONES : Este elemento contiene la lista de preposiciones, que introducen al sintagma.

Demos un ejemplo, sobre la representación de un sintagma en una estructura sint. Tomemos, por ejemplo, la frase :

- " Los felices robots de Isaac".

La estructura sint, correspondiente a este sintagma, se vería de la siguiente manera :



4.3.2 La estructura prd.

El predicado de una frase, está compuesto por uno o más verbos, acompañados de una serie de sintagmas nominales que definen los casos del predicado. La estructura prd, que representa al predicado de una frase, está compuesta por lo tanto de un elemento que representa a el o los verbos del predicado, y por una serie de estructuras sint que representan a los sintagmas nominales del mismo.

La figura 4.5, nos muestra la manera en que está constituída la estructura prd.

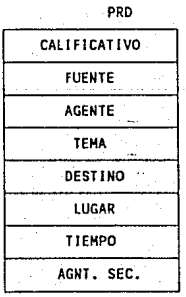


Fig. 4.5.
Estructura prd, para representación del predicado de una frase.

Podemos observar en esta figura, que el primer elemento de la estructura, contiene al calificativo del predicado, y los restantes elementos de la estructura, corresponden a los diferentes sintagmas del mismo, colocados de acuerdo a la relación semántica que los liga al calificativo. Los casos que no aparecen en la frase se dejan vacíos en esta estructura.

4.3.3 La estructura ora.

Finalmente, para representar la frase completa, utilizamos la estructura ora, cuya descripción observamos en la figura 4.6.

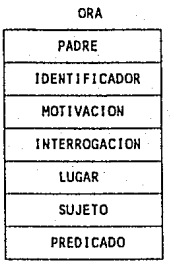


Fig. 4.6.
Estructura ora, para la representación de una frase.

Como podemos ver, la estructura ora, contiene los siguientes elementos :

PADRE : Este elemento es utilizado cuando la frase representa una inflexión para un sintagma de alguna otra frase. Contiene un apuntador a esa otra frase.

IDENTIFICADOR : Número que identifica a la frase.

MOTIVACION : Este elemento contiene un identificador del tipo de frase que representa. Este tipo puede ser decl,

por declarativa o int por interrogativa, y es utilizado por el operador **monitor**, que decide el tipo de acción a ejecutar con una frase.

INTERROGACION : Este elemento de la estructura, define, en caso de que la frase sea interrogativa, el caso del predicado, por él que se está preguntando. Así , si la pregunta se refiere a, "a quién", se define como objeto de la interrogación al caso DESTINO de la frase.

LUGAR : El siguiente elemento contiene una estructura sint, que representa al lugar en que se desarrolla la frase.

SUJETO : Este cuarto elemento es también una estructura sint, que define al sujeto de la misma.

PREDICADO : Este último elemento, es una estructura prd, que representa al predicado de la frase.

Podemos observar como, de esta manera quedan bien definidas cada una de las partes de una frase. Posteriormente veremos la manera en que cada una de estas estructuras se representa en el lenguaje de programación Prolog.

Veamos, para terminar, un ejemplo completo de la estructura semántica que Andrew genera para la siguiente frase:

"Los pequeños robots corrieron por la Luna"

El verbo de esta frase, "correr", es un verbo de acción, y es considerado como el CALIFICATIVO de la misma. Los siguientes, son los casos que aparecen en esta frase :

CALIFICATIVO : correr
- AGENTE -> sint ("los pequeños robots")
- LUGAR -> sint ("la Luna")

En la figura 4.7, vemos el diagrama de la estructura semántica que Andrew genera para esta frase.

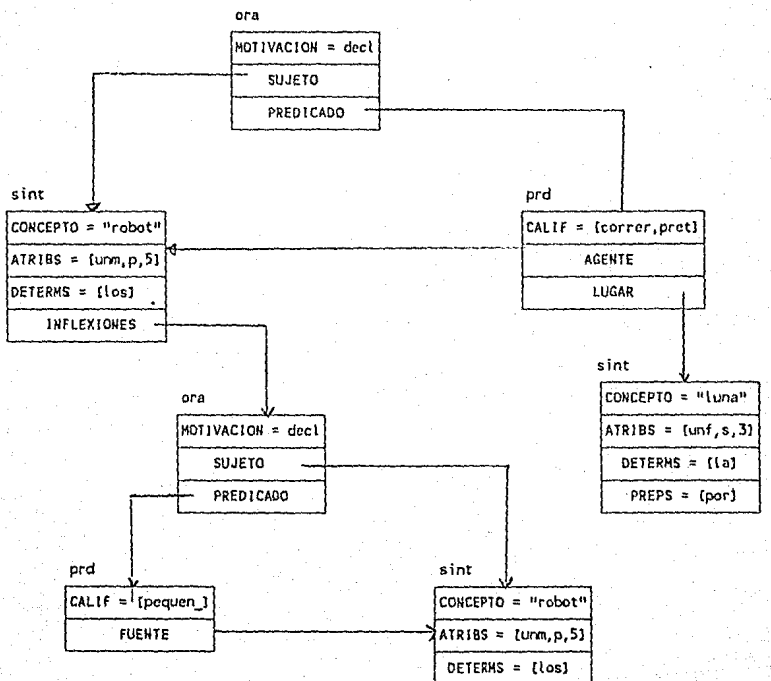


Fig. 4.7.

Estructura semántica de la frase :
"Los pequeños robots corrieron por la luna"

V. El análisis de las frases.

En el capítulo anterior, describimos la estructura semántica que nuestro sistema genera, para representar internamente a una frase. Esta representación interna es generada por el operador más importante del sistema, el ANALIZADOR. En el presente capítulo, daremos una descripción del análisis sintáctico y semántico, que el ANALIZADOR realiza sobre una frase, para generar la estructura semántica de la misma.

5.1 El Analizador

Para llevar a cabo el análisis de una frase, el ANALIZADOR, se basa en la estructura de la misma. Como ya hemos mencionado, la estructura de una frase está dada por un verbo, y un conjunto de sintagmas nominales, que se relacionan con el mismo. Por esta razón, el ANALIZADOR utiliza dos importantes procedimientos:

a) ANA_SINT (analiza un sintagma nominal),
b) ANA_PREDI (analiza el predicado de una frase).

Al mismo tiempo que efectúa el análisis de una frase, el ANALIZADOR, genera, una estructura ora que representa a dicha frase. Para hacer ésto, es necesario analizar cada uno de los sintagmas nominales de la frase, con el objeto de generar una estructura sint, para cada uno de ellos. Es precisamente el procedimiento ANA_SINT, quien se encarga de hacer ésto. Por otra parte el análisis del predicado, es llevado a cabo por el procedimiento ANA_PREDI, quien genera para dicho predicado, una estructura prd, que lo representa y llama a ANA_SINT cada vez que lo necesita.

Antes de que el ANALIZADOR entre en acción, el sistema utiliza una serie de procedimientos que se encargan de leer las frases ya sea de pantalla o de un archivo (dependiendo de la elección del usuario). El papel de estos procedimientos se puede resumir en dos puntos importantes:

a) Lectura de la frase , encargándose al mismo tiempo de :
- la eliminación de signos de puntuación de la frase,
- la conversión a minúsculas,
- la conversión de vocales acentuadas y ñ's, a una representación interna adecuada (á -> a_, ñ -> n_, etc); y
- la generación de una lista de las palabras, ya convertidas, de la frase.

b) Búsqueda en el vocabulario, de cada una de las palabras de la lista generada en el paso anterior, para a su vez agregar a dicha estructura los atributos de cada palabra, mismos que son tomados de los atributos morfológicos y semánticos que las palabras tienen definidos en el vocabulario.

El funcionamiento del ANALIZADOR se puede resumir mediante el diagrama de la figura 5.1.

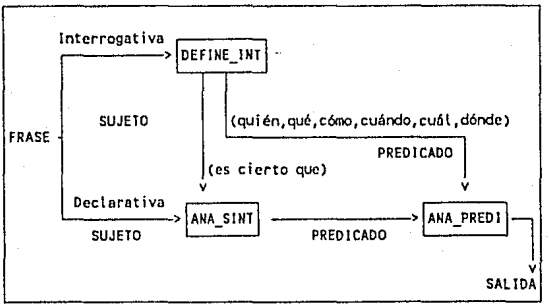


Fig. 5.1.
Diagrama del ANALIZADOR

Como podemos observar en este diagrama, la primera etapa del ANALIZADOR, consiste en determinar el tipo de frase que va a analizar. El sistema reconoce dos tipos de frases : declarativas e interrogativas.

El reconocimiento del tipo de frase, se lleva a cabo analizando la categoría gramatical a la que pertenecen las primeras palabras de la frase, como se muestra en la siguiente tabla.

TIPO DE LAS PRIMERAS PALABRAS	TIPO DE FRASE (EJEMPLO)
DETERMINANTE	Frase declarativa ("El niño corre")
PRONOMBRE INTERROGATIVO	Frase interrogativa ("Quién come")
PREPOSICION : seguida de PRON. INTERR.	Frase interrogativa ("En dónde estaba el robot") ("A quién dieron los libros")
seguida de DETERM. o SUST.	Frase declarativa ("Por el bosque el niño corre")
VERBO "ES" : es cierto que, es verdad que	Frase interrogativa ("Es verdad que el robot es feliz")

Las frases declarativas son manejadas bajo el supuesto de que el primer sintagma nominal de la frase define al sujeto de la misma, por lo que el ANALIZADOR llama al procedimiento ANA_SINT.

ANA_SINT se encarga de analizar el sujeto de la frase, y de generar la representación interna adecuada para el mismo. Esta representación interna está dada por la estructura sint, que describimos en el capítulo anterior.

Posteriormente, el ANALIZADOR llama al procedimiento ANA_PREDI, el cual define el calificativo de la frase y cada uno de los casos de la misma, generando así la estructura prd que representa a dicho predicado y estableciendo el caso correspondiente al sujeto ya analizado.

Finalmente se integra la información obtenida para sujeto y predicado y se reúnen las estructuras que los representan, en un estructura ora que representa a la frase completa.

Ahora bien, si la frase que llega al ANALIZADOR, es del tipo interrogativo, éste pasa por un proceso llamado DEFINE_INT, en el que se determina cuál es el caso por el que se está preguntando.

Por ejemplo si la frase comienza con las palabras, "a quién", entonces, se define como objeto de la interrogación al caso DESTINO. Esta información se guarda en la estructura ora.

Una vez determinado ésto, el ANALIZADOR, sabe que no hay sujeto que analizar, por lo que llama directamente al procedimiento ANA_PREDI.

Finalmente, al igual que para las frases declarativas, el ANALIZADOR integra la estructura prd generada por ANA_PREDI, en la estructura ora que representa a la frase.

A continuación describiremos el funcionamiento de los procedimientos que se encargan de analizar los sintagmas nominales y el predicado de la frase.

5.1.1 El análisis de un sintagma nominal.

El procedimiento ANA_SINT, es el responsable de analizar un sintagma nominal y de generar la estructura semántica que representará a dicho sintagma, dentro del sistema.

En la figura 5.2, mostramos el diagrama de ANA_SINT. Por claridad, no utilizamos del todo la notación de una ATN. En este diagrama y los que posteriormente aparecen en este capítulo, la etiqueta de los arcos indica la categoría sintáctica de la palabra que se está analizando, mientras que las casillas definen procedimientos.

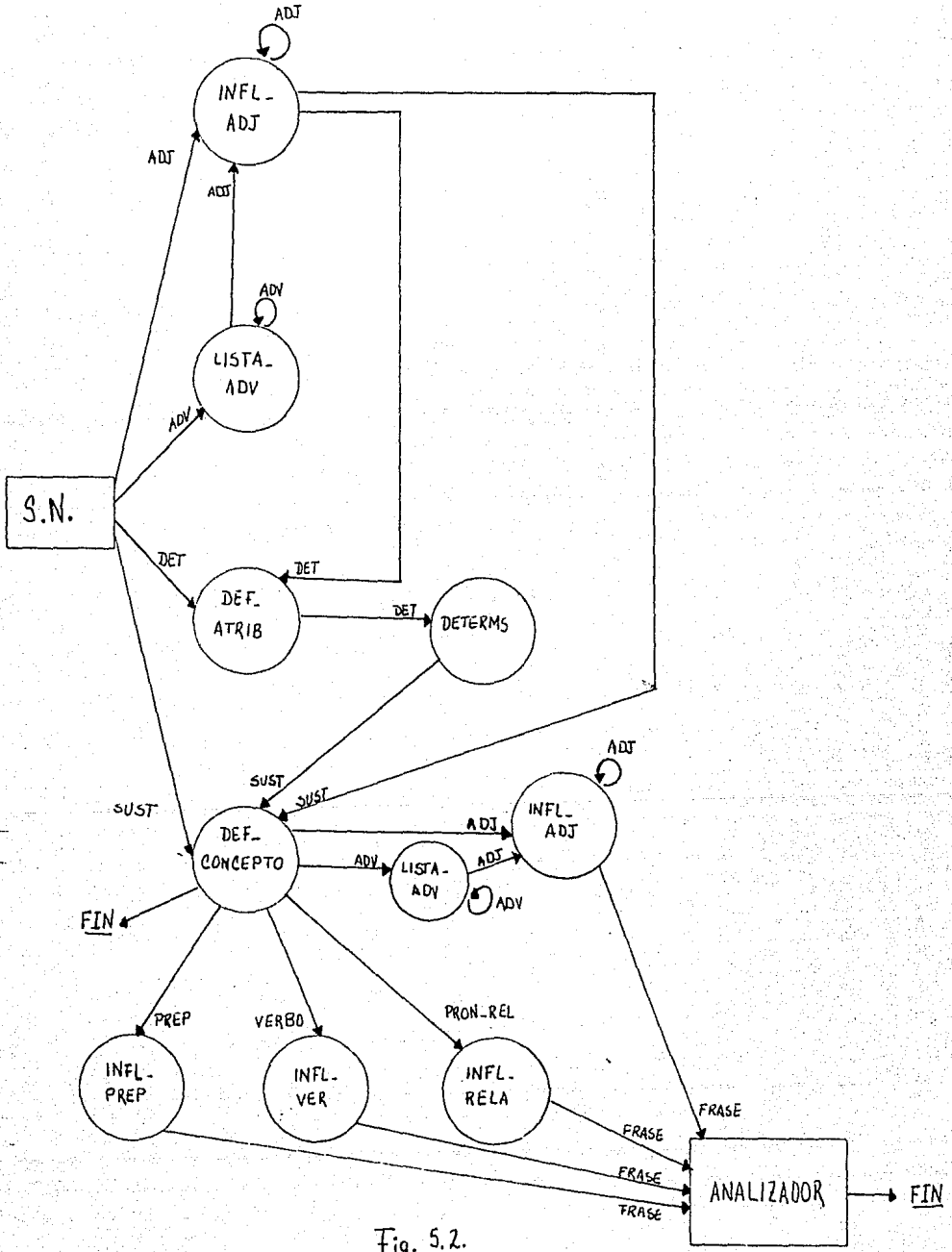


Fig. 5.2.

Procedimiento ANA_SINT

Los sintagmas que el procedimiento ANA_SINT acepta, son de la siguiente forma :

(Determ.) (Adverb.) (Adj.) Sustantivo (Adverb.) (Adj.) (Inflexión)

En esta expresión los elementos que se encuentran entre paréntesis son opcionales y repetibles. Observamos que la única palabra que por fuerza debe aparecer en un sintagma es el sustantivo.

Basados en esta expresión y en el diagrama de la figura 5.2, daremos la descripción del procedimiento ANA_SINT.

La entrada a este procedimiento es un sintagma nominal y la salida es una estructura sint que representa a dicho sintagma. Al recibir al sintagma, ANA_SINT crea una estructura sint vacía, cuya información irá actualizando en el transcurso del análisis.

Las acciones que se realizan de acuerdo al tipo de la palabra que se está analizando, son las siguientes :

1) DETERMINANTE :

En este caso el procedimiento ANA_SINT, llama al procedimiento DEF_ATRIB, para definir los atributos de género, persona y número del sintagma. Una vez hecho ésto se agrega el determinante a la estructura sint. Si existe más de un determinante, cada uno de ellos se agrega, a la lista de determinantes de la estructura sint.

Una vez hecho ésto, ANA_SINT espera encontrar un adverbio, un adjetivo o un sustantivo. En cualquier caso el proceso que sigue, está descrito en los incisos 2, 3 y 4, respectivamente.

2) ADVERBIO :

La función de un adverbio es la de calificar a un adjetivo, a un verbo o a otro adverbio. Por esta razón, cuando ANA_SINT encuentra un adverbio, espera que la siguiente palabra del sintagma sea un adverbio o bien un adjetivo. Los adverbios que encuentra son agregados a una lista temporal. Cuando finalmente encuentra un adjetivo, lo envía junto con la lista de adverbios al procedimiento INFL_ADJ que describimos en el inciso 3.

3) ADJETIVO :

Como mencionamos en el capítulo III, la aparición de un adjetivo ocasiona una inflexión ligada al sustantivo del sintagma. El procedimiento INFL_ADJ, se encarga de generar dicha inflexión, de la siguiente manera :

- Si el sustantivo, que es el concepto central del sintagma, aún no ha sido definido, simplemente agrega el adjetivo a una lista temporal en la que será almacenado hasta que se encuentre el sustantivo. En caso de que exista una lista de adverbios asociada al adjetivo, dicha lista a su vez, se agrega a la lista de adjetivos.

- Si el sustantivo ya fue definido, genera una frase cuyo sujeto está dado por dicho sustantivo, y cuyo predicado está dado por el verbo "ser" acompañado del adjetivo en cuestión. Por ejemplo, si el sintagma original es : "el robot feliz", la flexión generada es : "el robot es feliz".

INFL_ADJ, llama recursivamente al ANALIZADOR para que analice dicha frase y genere una estructura ora que la represente. Finalmente, INFL_ADJ liga la estructura ora generada a la estructura sint, que representa al sintagma que se está analizando, quedando así definida la inflexión.

SUSTANTIVO :

Cuando el procedimiento ANA_SINT, encuentra un sustantivo en el sintagma, llama al procedimiento GENERA_CONCEPTO que define a dicho sustantivo como el concepto central del sintagma en la estructura sint.

Una vez hecho ésto, ANA_SINT, espera que la siguiente palabra pertenezca a alguna de las siguientes categorías :

- ADVERBIO ("robot muy feliz"),
- ADJETIVO ("robot feliz"),
- VERBO PARTICIPIO ("robot cansado"),
- PRONOMBRE RELATIVO ("robot que corre"),
- PREPOSICION ("robot de Isaac").

Las acciones que toma el procedimiento ANA_SINT, en cada uno de estos casos, son las siguientes :

Adverbio y Adjetivo :

- Si la palabra que sigue, pertenece a las categorías de ADVERBIO o ADJETIVO, pasa a realizar los procesos definidos para estas categorías, en los incisos 2 y 3 respectivamente.

Verbo en participio :

- Por otro lado si llega un VERBO PARTICIPIO, ANA_SINT llama al procedimiento INFL_VER el cual, genera una frase cuyo sujeto está dado por el sustantivo que define al concepto central del sintagma y cuyo predicado está dado por el verbo "estar", acompañado del verbo en participio.
Si por ejemplo, el sintagma original es: "el robot cansado", se genera la frase : " el robot está cansado ". INFL_VER, llama recursivamente al ANALIZADOR que analiza dicha frase y genera una estructura ora que la represente.

Finalmente INFL_VER liga esta estructura ora a la estructura sint asociada al sintagma que está analizando.

Pronombre Relativo "que" :

- En caso de que después del sustantivo llegue el PRONOMBRE RELATIVO "que", ANA_SINT llama al procedimiento INFL_RELA que distingue dos casos :

a) El pronombre "que" está seguido de un verbo.
Un ejemplo de este caso está dado por el sintagma, "el niño que había estado corriendo". Como podemos observar la información introducida por el pronombre "que", se refiere o califica al sustantivo del sintagma , que en este caso es "niño".
Para manejar este caso, el procedimiento INFL_RELA,

genera una frase en la que el sujeto está dado por el sustantivo del sintagma y el predicado está dado por la información que la preposición "que" introduce. Para el ejemplo que mencionamos, la frase generada como flexión sería : "el niño había estado corriendo".
Una vez generada dicha frase, INFL_RELA llama al ANALIZADOR y liga la estructura ora generada por éste, debajo de la estructura sint, asociada al sintagma que se está analizando.

b) El pronombre "que" está seguido de un determinante o de un sustantivo.
Este caso se da, por ejemplo, en el sintagma : "los libros que el niño traía".
Para manejar estos casos, INFL_RELA, genera una frase en la que el sujeto y parte del predicado están dados por la información introducida por el pronombre "que", mientras que el sustantivo del sintagma es agregado al predicado como objeto directo. Para nuestro ejemplo, la frase generada sería : "el niño traía los libros".
INFL_RELA llama al ANALIZADOR para que analice la frase generada, y liga la estructura ora que éste produce, debajo de la estructura sint del sintagma que se está analizando.

Preposición "de" :
- Finalmente, si ANA_SINT encuentra la PREPOSICION "de", llama al procedimiento INFL_PREP, el cual genera una frase cuyo sujeto está dado por el concepto central del sintagma y cuyo predicado está dado por el verbo "ser" seguido de la preposición "de" y la información que ésta introduce.

Por el ejemplo para el sintagma "el robot de Isaac", se genera la siguiente frase : "el robot es de Isaac".
Para analizar la frase generada INFL_PREP llama recursivamente al ANALIZADOR, quien genera una estructura ora para la frase. Dicha estructura es finalmente asociada con la estructura sint que representa al sintagma.

Es importante aclarar que la información introducida por la preposición "de", sólo puede ser manejada como una relación que define una pertenencia. Otro tipo de casos, como "el robot de metal", no son aceptadas.

Es de esta manera como se analiza un sintagma nominal. Pasaremos ahora a describir el funcionamiento del procedimiento que analiza el predicado de una frase, el ANA_PREDI.

5.1.2 El análisis del predicado de una frase.

El proceso que sigue el procedimiento ANA_PREDI, en el análisis del predicado de una frase, consiste en reconocer al verbo de la frase, y, de acuerdo al tipo de dicho verbo, definir las relaciones semánticas que ligan a este último con los restantes sintagmas nominales de la misma, para finalmente, con esta información generar la estructura semántica prd, que representa al predicado dentro del sistema.

En la figura 5.3, mostramos un diagrama de este procedimiento.

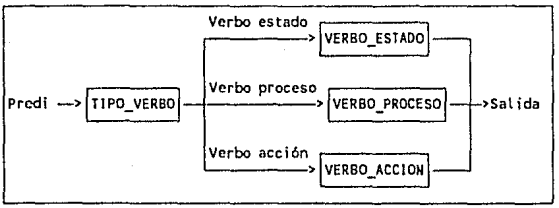


Fig. 5.3
Diagrama del procedimiento ANA_PREDI

El procedimiento ANA_PREDI recibe como entrada, el predicado de una frase, y por medio del procedimiento TIPO_VERBO, encuentra el tipo del verbo de dicho predicado.

Ahora bien, es posible que en una frase, aparezca más de un verbo, como ocurre por ejemplo, en la siguiente frase :

"El robot había estado corriendo".

En este caso el procedimiento TIPO_VERBO, define cual de los verbos es el que porta la información real del predicado y, basándose en él, define el tipo de análisis a seguir. Para el ejemplo que dimos, el verbo "corriendo", es quien realmente representa lo que "el robot" hace, por lo que TIPO_VERBO, determina a VERBO_ACCION como el análisis a seguir.

Dependiendo del tipo del verbo, ANA_PREDI llama a alguno de los siguientes tres procedimientos :

- VERBO_ESTADO
- VERBO_ACCION
- VERBO_PROCESO

Cada uno de estos procedimientos analiza el predicado de la frase para generar la estructura prd que lo representará en el sistema.

Para describir el funcionamiento del procedimiento ANA_PREDI, describiremos el funcionamiento de cada uno de estos tres procedimientos.

5.1.2.1 Procedimiento VERBO_ESTADO.

Como ya habíamos mencionado, los casos que el sistema reconoce, cuando el verbo de una frase es un verbo de estado, son : FUENTE, TEMA, LUGAR, TIEMPO y AGENTE SECUNDARIO.

En la figura 5.3., mostramos un diagrama que describe la manera en que el procedimiento VERBO_ESTADO analiza el predicado para definir cada uno de estos casos.

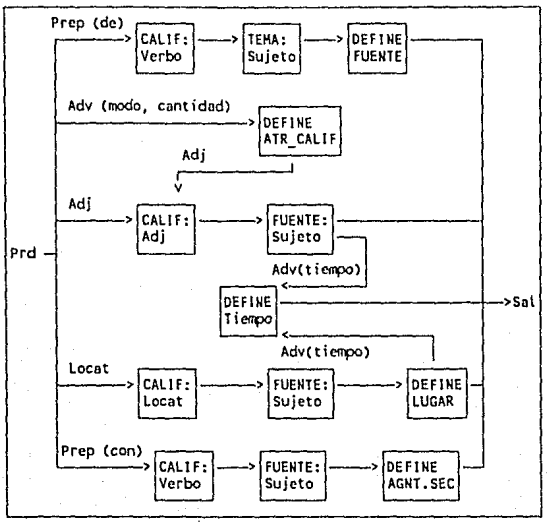


Fig. 5.4.
Diagrama del procedimiento VERBO_ESTADO

En la figura anterior podemos observar el tipo de palabras que pueden aparecer después de un verbo de estado en una frase. A continuación describiremos el proceso que sigue el procedimiento VERBO_ESTADO, dependiendo del tipo de la palabra que sigue al verbo.

Preposición DE :

Como mencionamos en la sección anterior, la preposición DE es aceptada por el sistema, cuando dicha preposición introduce una relación de pertenencia (el robot es de Isaac).

En este caso, VERBO_ESTADO define al verbo de la frase como el CALIFICATIVO de la misma, mientras que define al sujeto como TEMA de la pertenencia. Finalmente, por medio del procedimiento ANA_SINT, analiza el sintagma que sigue a la preposición DE, y lo define como la FUENTE de dicha pertenencia.

Adverbio de modo o cantidad :

Cuando este tipo de adverbios aparece en el predicado de una frase, su función es la de calificar al verbo de la misma o bien a un adjetivo. Por esta razón cuando el procedimiento VERBO_ESTADO encuentra uno de estos adverbios, lo define como ATRIBUTO del calificativo y pasa a buscar al adjetivo que dicho adverbio está calificando.

Adjetivo :

En este caso VERBO_ESTADO define al adjetivo como CALIFICATIVO de la frase, y al sujeto de la misma como la FUENTE de la cualidad dada por dicho adjetivo. Posteriormente puede aparecer un adverbio de tiempo, el cual define el caso TIEMPO.

Locativo :

Definimos un locativo como una preposición de lugar (en, por), o bien como un adverbio de lugar seguido de la preposición DE, (dentro de, arriba de,..) . Al encontrar un locativo, el procedimiento VERBO_ESTADO lo define como CALIFICATIVO de la frase, mientras que el sujeto es, a su vez, definido como la FUENTE del estar en el lugar introducido por dicho locativo. Finalmente, por medio del procedimiento ANA_SINT, se analiza el sintagma que el locativo introduce y se define como el caso de LUGAR.

Al igual que con los adjetivos, un adverbio de tiempo puede aparecer después del sintagma que define al LUGAR. En dicho caso se define dicho adverbio como el caso de TIEMPO.

Preposición CON :

Mencionamos en el capítulo III, que nuestro sistema no maneja ningún caso que defina semánticamente la idea de un acompañante. Este tipo de idea viene, en general, introducida por la preposición CON, como ocurre por ejemplo en la frase, "el robot estaba con Isaac".

En este caso ANA_SINT define al verbo de estado como el CALIFICATIVO de la frase, al sujeto de la misma como la FUENTE del estado, y al sintagma introducido por la preposición CON, como el AGENTE SECUNDARIO de dicho estado.

5.1.2.2 Procedimiento VERBO_ACCION.

Recordamos del capítulo III que los casos que el sistema acepta cuando el verbo de la frase es un verbo de acción son : AGENTE, TEMA, DESTINO, LUGAR, TIEMPO y AGENTE SECUNDARIO.

En el diagrama de la figura 5.5, describimos el funcionamiento del procedimiento VERBO_ACCION, para frases en voz activa.

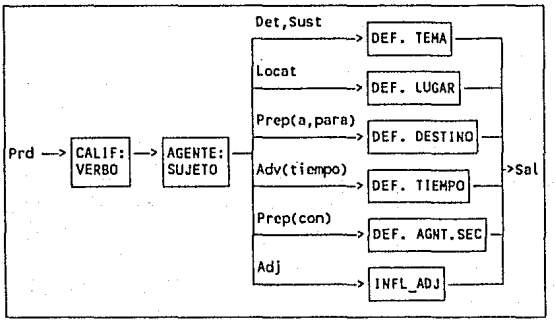


Fig.5.5.
Diagrama del procedimiento VERBO_ACCION

En el caso de las frases cuyos verbos son verbos de acción, el verbo se define siempre como el CALIFICATIVO de la frase.

EL diagrama de la figura 5.5, describe el funcionamiento de VERBO_ACCION para frases en voz activa, por lo que siempre se define al sujeto de la frase como el AGENTE de la misma.

En dicho diagrama podemos observar el tipo de palabras que el sistema acepta estén dadas, después de un verbo de acción en una frase. A continuación describiremos el proceso que sigue el procedimiento VERBO_ACCION, dependiendo del tipo de la palabra que sigue al verbo.

Determinante o Sustantivo :

Un ejemplo de este caso, está dado en la frase "El niño comió las frutas". En este caso, VERBO_ACCION llama al procedimiento ANA_SINT para que analice el sintagma introducido por el determinante o el sustantivo, y coloca la estructura sint que dicho procedimiento genera, como TEMA de la acción.

Locativo :

En este caso, VERBO_ACCION llama al procedimiento ANA_SINT para que analice el sintagma introducido por el locativo, y coloca la estructura sint que dicho procedimiento genera, como LUGAR de la acción. Un ejemplo de este caso está dado por la frase "El robot corre por la casa", en la que el sintagma "la casa" es considerado como el LUGAR en que se desarrolla la acción de "correr".

Preposición A,PARA :

En este caso, VERBO_ACCION llama al procedimiento ANA_SINT para que analice el sintagma introducido por la preposición A o PARA, y coloca la estructura sint que dicho procedimiento genera, como DESTINO de la acción. En la frase, "El niño da las flores a su madre", vemos un ejemplo en el que el sintagma "la madre", es el DESTINO de la acción de "dar".

Adverbio de TIEMPO :

En este caso, el procedimiento VERBO_ACCION define al adverbio de tiempo, como el TIEMPO en que la acción se desarrolla. Un ejemplo está dado por la frase sintagma "Isaac bailó ayer", en la que "ayer" es considerado como el caso TIEMPO.

Preposición CON :

En este caso, VERBO_ACCION llama al procedimiento ANA_SINT para que analice el sintagma introducido por la preposición CON, y coloca la estructura sint que dicho procedimiento genera, como AGENTE SECUNDARIO de la acción.

Adjetivo :

Cuando un adjetivo aparece en el predicado de una frase, no tiene una relación semántica con el verbo. Su función real, es la de calificar al sujeto de dicha frase. Por ejemplo en la frase "el robot corre feliz", el adjetivo feliz es una característica del sujeto, "el robot".

Por esta razón, cuando un adjetivo aparece en una frase cuyo verbo es un verbo de acción, el procedimiento VERBO_ACCION lo define como una inflexión al sujeto de la frase. Para ello, llama al procedimiento INFL_ADJ, que se encarga de generar dicha inflexión.

Es importante aclarar que el manejar de esta manera a los adjetivos, no es del todo adecuada, pues existe por ejemplo, una diferencia semántica entre la frase "El robot corre feliz" y la frase "El feliz robot corre". Sin embargo el sistema genera la misma representación para ambas frases y pierde información. Una manera de solucionar este problema podría ser la de generar en la estructura prd un campo para guardar al calificativo de un calificativo, además de la manejada para los adverbios.

Describiremos ahora los casos que aparecen en una frase en voz pasiva. La siguiente es un ejemplo de frase en voz pasiva : "La fruta fue comida por el niño".

Para manejar este tipo de frases el procedimiento VERBO_ACCION, define al sujeto de la frase como TEMA de la acción, mientras que el verbo de dicha frase es definido como CALIFICATIVO. Finalmente busca un sintagma introducido por la preposición POR, y llama al procedimiento ANA_SINT, para que lo analice y define a la estructura sint que este procedimiento genera, como el AGENTE de la acción de "comer".

5.1.2.3 Procedimiento VERBO_PROCESO.

En el capítulo 3, mencionamos que los casos que se reconocen cuando el verbo de una frase es un verbo de proceso, son los siguientes : FUENTE, TEMA, LUGAR, y TIEMPO.

En el diagrama de la figura 5.6, describimos el funcionamiento del procedimiento VERBO_PROCESO, para el análisis del predicado de una frase en voz activa.

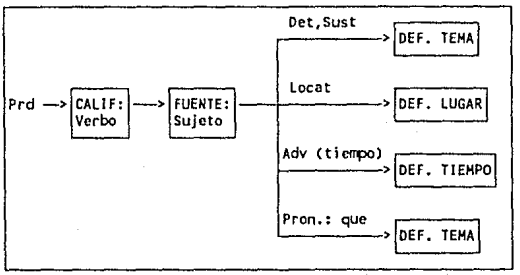


Fig.5.6.
Diagrama del procedimiento VERBO_PROCESO

Como podemos observar, en esta figura, el verbo siempre es definido como el CALIFICATIVO de la frase, mientras que el sujeto es considerado como la FUENTE del proceso dado por dicho verbo.

Ahora bien, de acuerdo al tipo de palabra que introduce a cada sintagma, el procedimiento VERBO_PROCESO define los siguientes casos :

Determinante o Sustantivo :

Un ejemplo de este caso, está dado por la frase "El niño ve las casas". En este caso, el procedimiento VERBO_PROCESO define al sujeto de la frase como la FUENTE del proceso y llama al procedimiento ANA_SINT para que analice el sintagma introducido por el determinante o por el sustantivo, para finalmente definir a la estructura sint generada por ANA_SINT, como el TEMA del proceso.

Locativo :

Cuando un locativo (preposición en, por; adverbio de lugar seguido de preposición DE), es encontrado, VERBO_PROCESO llama al procedimiento ANA_SINT, para que analice el sintagma que es introducido por el locativo. Finalmente define a la estructura sint generada por ANA_SINT, como el LUGAR en que se desarrolla el proceso.

Adverbio de tiempo :

En este caso, VERBO_PROCESO define al adverbio de tiempo, como el TIEMPO, en que se desarrolla el proceso.

Pronombre relativo QUE :

En este caso el sintagma introducido por el pronombre "que", es una frase, por lo que VERBO_PROCESO llama al ANALIZADOR y coloca la estructura generada por este procedimiento como el TEMA de la frase.

Describiremos ahora los casos que aparecen en una frase en voz pasiva. La siguiente es un ejemplo de frase en voz pasiva : "La fruta fue vista por el niño". El sistema reconoce que una frase está dada en voz pasiva si el verbo de la misma es el verbo "ser" en alguna de sus conjugaciones, acompañado de otro verbo en participio.

Para manejar este tipo de frases el procedimiento VERBO_PROCESO, define al sujeto de la frase como TEMA de la acción, mientras que el verbo de dicha frase es definido como CALIFICATIVO. Finalmente busca un sintagma introducido por la preposición POR, y llama al procedimiento ANA_SINT, para que lo analice y define a la estructura sint que este procedimiento genera, como el FUENTE de la acción de "ver".

En el diagrama no incluímos el caso de las frases interrogativas. Esto es debido a que no quisimos complicar demasiado las cosas. Explicaremos brevemente a continuación lo que Andrew hace en este caso.

En general las frases interrogativas comienzan con un pronombre interrogativo (quién, cómo,..), o bien con una preposición seguida igualmente de un pronombre (a quién, en dónde,...). Lo que resta en una pregunta después de estas palabras, es una estructura de tipo predicado, excepto tal vez por las últimas palabras.

Por ejemplo, la frase : "A quién da las flores el robot?", puede ser vista como : " A quién -- da las flores -- el robot?", donde las primeras palabras expresan el caso por el que se está preguntando, es decir el DESTINO, las siguientes, son en cierta forma el predicado de la oración y las últimas representan lo que sería el sujeto en la frase que dió lugar a esta pregunta ("El robot da las flores a Isaac").

Así las cosas, Andrew, llama al procedimiento ANA_PREDI para generar la estructura de las palabras que efectivamente corresponden al predicado de la frase interrogativa y, por otra parte, genera la representación interna para el último sintagma de la frase el cual representa al sujeto de la misma.

Posteriormente, el proceso que se encarga de generar las respuestas a las preguntas que el usuario hace, utilizará la estructura generada para la frase interrogativa, en el momento de dar respuesta a la misma. En el siguiente capítulo describiremos el funcionamiento de éste y otros procedimientos.

5.1.2.4 Un ejemplo de análisis.

Daremos ahora un ejemplo de los pasos que sigue el analizador para generar la estructura de una frase, durante el análisis de la misma.

Tomemos la siguiente frase :

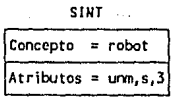
" El robot feliz vió la máquina."

Antes que nada, el ANALIZADOR, genera una estructura ora inicialmente vacía, que utilizará para representar internamente a la frase de entrada. Una vez hecho ésto, comienza a recorrer dicha frase de izquierda a derecha. Al encontrar que la primera palabra es un determinante, define a la motivación de la frase como declarativa, colocando esta motivación en la estructura ora.

Una vez hecho ésto, el analizador llama al procedimiento ANA_SINT, para interpretar al sintagma nominal que representa al sujeto de la frase. ANA_SINT, genera una estructura sint inicialmente vacía, que representará a dicho sintagma. Una vez hecho ésto, el determinante "el", es analizado en ANA_SINT, por el procedimiento DETERMINANTE, que define los atributos morfológicos de la frase, guardándolos en la estructura sint. Los atributos morfológicos en este caso son:

Número : singular (s),
Género : masculino (unm, por unicamente masculino),
Persona : tercera (3).

Después del determinante, ANA_SINT, encuentra un sustantivo : "robot", por lo que llama al procedimiento GENERA_CONCEPTO, que se encarga de generar el concepto de la estructura sint y de checar que exista concordancia de género y número entre el sustantivo y el determinante. En este momento esta estructura se ve de la siguiente manera :



ANA_SINT, encuentra en seguida la palabra "feliz", que es un adjetivo, por lo que genera la inflexión : "el robot es feliz", y llama al procedimiento ANA_PREDI, que genera una estructura ora, para esta subfrase.

Esta estructura prd, es colocada en la estructura sint, a manera de inflexión. Con ello termina el análisis del sujeto de la frase.

En este momento, el analizador encuentra un verbo por lo que llama al procedimiento GENERA_VERBOS, que analiza el o los verbos de la frase, decidiendo el tipo al que pertenecen. En este caso, la frase tiene un sólo verbo : "vió", que resulta ser un verbo de proceso. El analizador, llama entonces, al procedimiento ANA_PREDI, para continuar con el análisis del predicado. Este procedimiento genera una estructura prd vacía, y coloca en ella al verbo "ver", como calificativo de la frase. Así mismo, por tratarse de un verbo de proceso, coloca al sujeto como el caso FUENTE.

Posteriormente ANA_PREDI, encuentra al determinante "la". Esto indica que el siguiente sintagma, al no contar con una preposición como primera palabra, debe ser considerado como el caso TEMA del predicado. Así las cosas, llama al procedimiento ANA_SINT, para que genere la estructura sint , que represente a dicho sintagma y lo coloca como tema del predicado. En este momento, la frase termina, por lo que termina también la ejecución de ANA_PREDI.

Finalmente el analizador, coloca la estructura sint, que representa al sujeto, y la estructura prd, que representa al predicado, en la estructura ora que representa a la frase.

En la figura 5.7, podemos observar la estructura generada, en manera de árbol y en la figura 5.8, mostramos un diagrma más sintetizado de esta estructura.

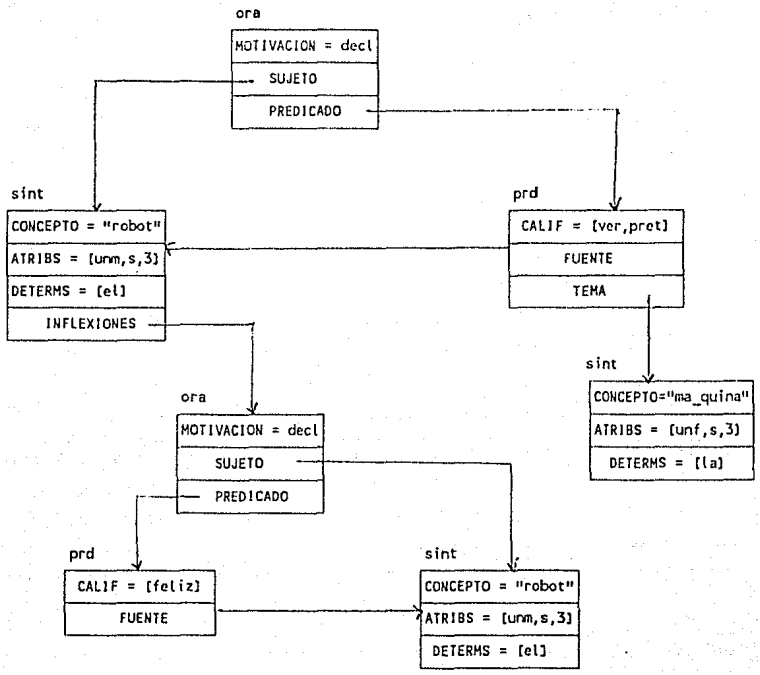


Fig. 5.7.

Estructura semántica de la frase :
"El robot feliz vió la máquina"

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

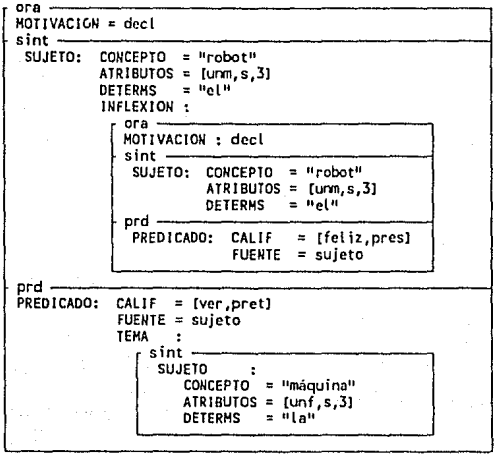


Fig. 5.8.
Estructura semántica generada para la frase :
"El robot feliz vió la máquina"

VI. Los otros operadores.

En el capítulo anterior describimos el funcionamiento del ANALIZADOR. En este capítulo describiremos el funcionamiento de dos de los operadores más importantes en el sistema, el GENERADOR y el CONTESTADOR.

6.1 El Generador.

Este operador realiza un proceso inverso al del ANALIZADOR. Su función es la de generar una frase en español, a partir de la estructura semántica que representa a dicha frase.

Como mencionamos en el capítulo IV, el sistema se vale de tres estructuras de datos para representar una frase. Estas estructuras son las siguientes : sint, prd y ora.

Dado que la representación interna de una frase está basada en estas tres estructuras, el GENERADOR utiliza dos procedimientos principales :

GENERA_SINT : que efectúa la traducción, de la representación interna de un sintagma, al español, y

GENERA_PRD : que efectúa la traducción, de la representación interna de un predicado, al español.

En la figura 6.1, mostramos un diagrama que describe el funcionamiento del GENERADOR.

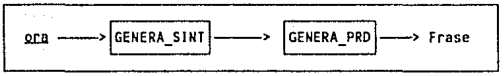


Fig. 6.1.
Diagrama del operador GENERADOR.

Como observamos en la figura 6.1 , el GENERADOR recibe como entrada la estructura ora que representa a la frase en el sistema. De dicha estructura toma la estructura sint que representa al sujeto de la frase, y llama al procedimiento GENERA_SINT para que traduzca dicha estructura al español. Finalmente toma de la estructura ora, la estructura prd que representa al predicado de la frase y llama al procedimiento GENERA_PRD para que traduzca su contenido.

De esta manera el GENERADOR da como salida la frase en español.

A continuación describiremos el funcionamiento de los procedimientos GENERA_SINT y GENERA_PRD.

6.1.1 El procedimiento GENERA_SINT.

Como ya habíamos mencionado, este procedimiento se encarga de traducir la representación interna de un sintagma nominal, al español.

En la figura 6.2, mostramos el diagrama de funcionamiento de GENERA_SINT.

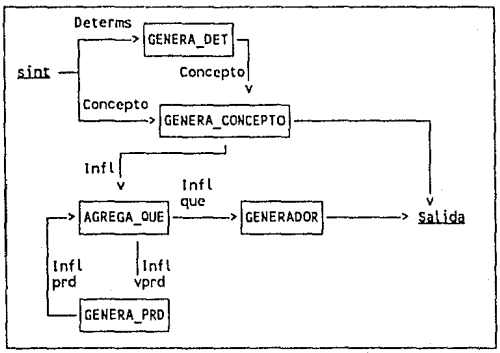


Fig. 6.2.
Diagrama del procedimiento GENERA_SINT.

En el diagrama de la figura 6.2, los arcos no definen categorías gramaticales, como ocurría en los diagramas del capítulo anterior. Ahora hemos utilizado los arcos para describir la existencia de los elementos de la estructura sint (concepto, atributos, determinantes, inflexiones) . Por su parte las casillas definen procedimientos.

Veamos ahora el funcionamiento de GENERA_SINT.

El primer paso consiste en ver si la estructura sint tiene definido algún determinante. De ser así se llama al procedimiento GENERA_DET, que coloca en la frase de salida, los determinantes.

Una vez definidos los determinantes, o en caso de que no existan, el procedimiento GENERA_SINT pasa a obtener los atributos del sintagma en cuanto a género, número y persona. Esta información es utilizada en lo que llamamos la generación del concepto.

Como ya habíamos mencionado, los sustantivos y verbos no se guardan en la forma en que el usuario los da al introducir una oración, sino que se guardan en su forma más básica, es decir, la raíz del sustantivo , o bien el infinitivo del verbo.

Por esta razón, los atributos del sintagma que, son los mismos que debe tener el sustantivo, son utilizados para generar a partir de la raíz, el sustantivo en la forma en que fue dado por el usuario.

Para llevar a cabo este proceso, GENERA_SINT se vale del procedimiento GENERA_CONCEPTO y, una vez que éste generó el sustantivo ya en español, éste último es agregado a la oración en español.

En este momento, a GENERA_SINT le resta únicamente traducir cada una de las inflexiones del sintagma. Para ello, agrega a la frase en español la palabra "que", para que sirve como conector entre el sujeto de la frase y la inflexión.

Como mencionamos en el capítulo III, un sustantivo puede tener diferentes inflexiones, cada una de las cuales el sistema maneja como frases asociadas al sintagma en el que el sustantivo se encuentra definido. Estas frases pueden tener a dicho sustantivo como SUJETO o bien como TEMA.

Cuando el sustantivo funciona como sujeto de la inflexión, GENERA_SINT llama al procedimiento GENERA_PRD para que genere en español al predicado de la inflexión. Dicho procedimiento se vuelve a llamar tantas veces como inflexiones existan en la estructura sint. Un ejemplo de inflexión de este tipo está dada por el sintagma "el robot feliz", cuya flexión es la frase "el robot es feliz".

Si en cambio, el sustantivo funciona como TEMA de la inflexión, GENERA_PRD coloca a dicho TEMA como vacío y llama recursivamente al GENERADOR para que genere una subfrase en español. Dicha subfrase se agrega a la frase original anteponiéndole la palabra "que". Un ejemplo de inflexión de este tipo está dada por el sintagma "los libros que el niño ve", cuya flexión está dada por la frase : "los libros son vistos por el niño".

Es de esta manera como el sistema traduce la estructura semántica de un sintagma nominal a un sintagma en español. Como podemos observar realmente la información está totalmente contenida en la representación semántica de la frase, por lo que la tarea de la generación se convierte en un proceso muy sencillo.

6.1.2 El procedimiento GENERA_PRD.

Este procedimiento se encarga de generar el predicado de una frase a partir de la estructura semántica prd.

En la figura 6.3, mostramos un diagrama del funcionamiento de este procedimiento.

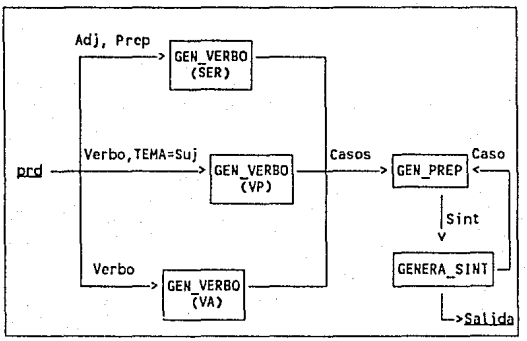


Fig. 6.3.
Diagrama del procedimiento GENERA_PRD.

En el diagrama observamos que la entrada del procedimiento GENERA_PRD, está dada por una estructura prd, que representa al predicado de una frase.
Como mencionamos en el capítulo III, el calificativo de una frase puede estar dado por un verbo, un adjetivo o un locativo. Así pues, para generar a partir de la estructura semántica prd, el predicado de una frase, en español, GENERA_PRD analiza el CALIFICATIVO que está definido en dicha estructura, por medio del procedimiento GEN_VERBO.

Si dicho CALIFICATIVO es un adjetivo o una preposición, GEN_VERBO añade a la frase en español, el verbo "ser" conjugado de acuerdo a los atributos del sujeto de la frase.

Si el CALIFICATIVO es un verbo y el sujeto de la frase está definido como el TEMA de la misma, sabemos que la frase fue dada en voz pasiva. De esta manera el procedimiento GEN_VERBO toma al verbo de la frase y lo conjuga en voz pasiva, agregándolo a la frase en español.

Finalmente, si el CALIFICATIVO es un verbo y el sujeto no es el TEMA de la frase, GEN_VERBO genera a dicho verbo de acuerdo a los atributos del sujeto, y lo coloca en la frase en español.

Una vez generado el verbo de la frase, GENERA_PRD, analiza cada uno de los casos definidos en la misma. Para ello llama al procedimiento GEN_PREP. Este procedimiento define la preposición que introduce al sintagma que representa a un caso, de acuerdo precisamente a dicho caso. Veamos un ejemplo. Si un sintagma está definido como el caso DESTINO, GEN_PREP genera la preposición "para", y la coloca en la frase en español. Posteriormente el procedimiento GENERA_SINT es llamado, para que genere el sintagma dado por el caso que se está analizando.

Este proceso se lleva a cabo tantas veces como casos estén definidos en la estructura semántica que representa al predicado de la frase.

6.1.3 Ejemplo de generación de una frase a partir de la estructura semántica ora.

Es momento ahora de dar un ejemplo en que mostremos la manera en que a partir de una lista ora, se genera una frase en español.

Supongamos que tenemos la representación interna para la siguiente frase:

"Los niños corrieron hacia la casa amarilla"

La figura 6.4., muestra el diagrama de la estructura semántica de esta frase.

```
ora
MOTIVACION = decl
sint
SUJETO:  CONCEPTO  = "niñ"
         ATRIBUTOS = [unm, p, 3]
         DETERMS   = "los"
prd
PREDICADO:  CALIF  = ["correr",pret]
            AGENTE = sujeto
            LUGAR  :
              sint
              CONCEPTO  = "casa"
              ATRIBUTOS = [unf, s, 3]
              DETERMS   = "la"
              INFLEXION :
                ora
                MOTIVACION = decl
                sint
                SUJETO:  CONCEPTO  = "casa"
                         ATRIBUTOS = [unf, s,3]
                         DETERMS   = "la"
                prd
                PREDICADO: CALIF = ["amarilla",pres]
                           FUENTE = sujeto
              PREPS :   "hacia"
```

Fig. 6.4.
Estructura semántica de la frase :
"Los niños corrieron en la casa amarilla"

El primer paso que realiza el GENERADOR es tomar de la estructura ora que representa a la frase, a la estructura sint que representa al sujeto de la misma, para enviarla al procedimiento GENERA_SINT.

GENERA_SINT, encuentra en dicha estructura al determinante "los" y lo agrega a la frase en español.

Después toma de la estructura sint el concepto "niñ" y junto con los atributos de la frase, lo manda al procedimiento GENERA_CONCEPTO, el cual genera el sustantivo "niños".

Finalmente GENERA_SINT busca alguna inflexión y al no encontrar ninguna, da por terminado el análisis de la estructura sint.

Hasta este momento tenemos ya parte de la frase en español :

"los niños"

El generador pasa ahora a analizar el predicado usando el procedimiento GENERA_PRD.

Este procedimiento toma el CALIFICATIVO de la frase, que está dado por "correr", y se da cuenta de que es un verbo en voz activa, por lo que genera el verbo "corrieron", basado en dicho verbo y en los atributos del sujeto y el tiempo de la frase. La frase generada hasta este momento tiene la siguiente forma :

"los niños corrieron"

Finalmente GENERA_PRD, pasa a analizar cada uno de los casos del predicado, encontrando, en este ejemplo, el caso de lugar, por lo que llama al procedimiento GEN_PREP, el cual por tratarse del caso LUGAR, genera la preposición "en", y llama a su vez al procedimiento GENERA_SINT el cual, genera la subfrase "la casa". Esta vez, GENERA_SINT encuentra una inflexión asociada al sustantivo "casa", por lo que llama a GENERA_PRD para generar la subfrase "la casa que es amarilla", misma que se añade a la frase final en español, que es :

"los niños corrieron en la casa que es amarilla".

De esta manera termina el análisis de la frase y el sistema llama a otros procedimientos que se encargan del despliegue de esta frase.

6.2 El Contestador.

Este es el procedimiento que se encarga de generar la respuesta a las preguntas realizadas por el usuario. Su funcionamiento depende en mucho, de la representación interna que el ANALIZADOR genera tanto para oraciones declarativas, como para oraciones interrogativas.

El dato más importante para generar una respuesta, es el tipo de caso por el que se pregunta. Este dato está contenido en el argumento INTERROGACION de la estructura semántica ora que fue generada para la frase y nos permite saber exactamente cual es la información que el usuario requiere de una frase.

En la figura 6.5, mostramos el diagrama del funcionamiento del CONTESTADOR.

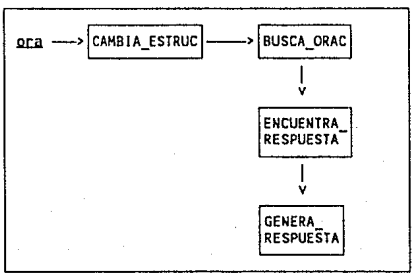


Fig. 6.5.
Diagrama del operador CONTESTADOR

Como podemos observar, el procedimiento GENERADOR, recibe como entrada a la estructura semántica ora , que representa a la pregunta.

El GENERADOR llama al procedimiento CAMBIA_ESTRUC. La función de este procedimiento consiste en hacer algunos cambios a la estructura ora que representa a la pregunta, de manera que dicha estructura pueda ser usada para compararse con las frases declarativas que el usuario ha dado.

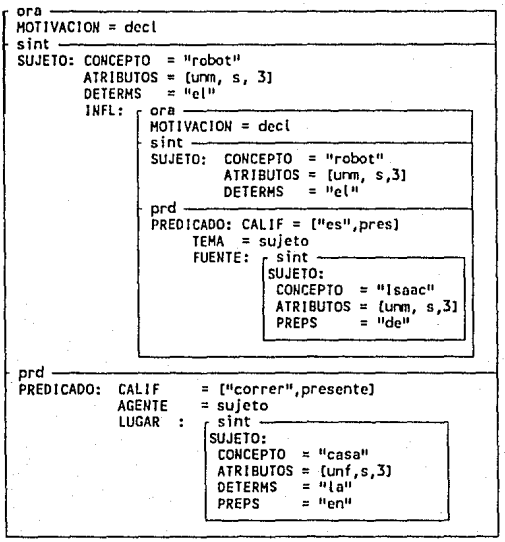
Un ejemplo del tipo de cambios que CAMBIA_ESTRUC realiza es el de cambiar la motivación de la frase, de "int" a "decl"

De esta manera, el sistema obtiene una estructura que contiene parte de la frase que contiene la respuesta que buscamos. Veamos un ejemplo, para aclarar esta idea.

Supongamos que, durante el diálogo, el usuario dió la siguiente frase :

"El robot de Isaac corre en la casa"

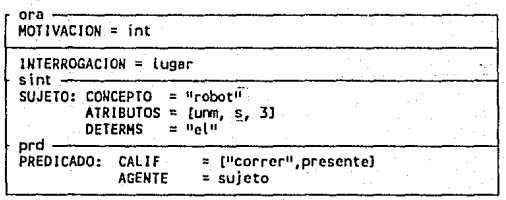
Andrew genera para esta frase la siguiente estructura semántica:



Supongamos ahora, que el usuario hace la siguiente pregunta :

"Dónde corre el robot?"

La estructura semántica que el sistema genera para la pregunta, es la siguiente :



Si comparamos las dos estructuras, podemos ver que son muy parecidas. De hecho la segunda estructura está sumergida en la primera y si a la segunda le cambiamos la

MOTIVACION que es int por decl, y eliminamos el argumento INTERROGACION, lo que obtendremos será una estructura sumergida dentro de la que representa a la oración declarativa.

La estructura de la interrogación no contiene toda la información que aparece en la frase declarativa, sino sólo algunas partes de ella. Si por otra parte, la frase interrogativa tuviese alguna información no contenida en la frase declarativa, como ocurre por ejemplo en la pregunta, "Dónde corre el feliz robot?", entonces dicha frase declarativa no sería considerada como candidata a dar la respuesta a esta pregunta.

Este ejemplo resume el principal proceso que lleva a cabo el procedimiento CAMBIA_ESTRUC, que una vez más, consiste en generar, a partir de la pregunta, una estructura que ya contenga cierta información de la respuesta. Posteriormente el CONTESTADOR llama al procedimiento BUSCA_ORAC que basado en la estructura modificada por CAMBIA_ESTRUC , selecciona de entre las frases declarativas que el usuario ha dado, a aquella que concuerda con dicha estructura.

Ahora bien, hasta este punto el CONTESTADOR ya tiene la estructura de la frase que contiene a la respuesta que está buscando. Ahora debe, efectivamente, encontrar la respuesta. Veamos un ejemplo para aclarar a que nos referimos.

Supongamos que el usuario pregunta sobre el "cómo" alguien hace algo. Esta información puede estar contenida en el calificativo de la frase o bien en alguna de las inflexiones del sujeto. El problema radica en encontrar cual o cuales de las inflexiones definen una cualidad del sujeto.

Para manejar este problema, el CONTESTADOR llama al procedimiento ENCUENTRA_RESPUESTA. El proceso que sigue este procedimiento, consiste en buscar primero la respuesta en el predicado y de no encontrarlo ahí buscar en algún otro lugar como el "lugar" general de la frase, o las inflexiones del sujeto.

Finalmente, el CONTESTADOR llama al procedimiento GENERA_RESPUESTA que define la estructura semántica de la frase que constituye la respuesta, misma que posteriormente será enviada al procedimiento GENERADOR para contestarle al usuario en español.

Hemos descrito en este capítulo el funcionamiento de dos de los más importantes operadores del sistema. En el siguiente capítulo describiremos la manera en que algunos de sus procedimientos fueron codificados en el lenguaje de programación PROLOG.

VII. Codificación en Prolog.

En este capítulo describiremos la manera en la que el vocabulario del sistema y las estructuras ora, sint y prd, son codificadas en el lenguaje de programación Prolog. Describiremos también la codificación en dicho lenguaje, de algunos de los más importantes procedimientos del sistema.

7.1 Codificación del vocabulario.

Para representar a las palabras del vocabulario, nos valemos de un predicado en Prolog al cual llamamos "pal" (por palabra), que consta de tres argumentos :

pal(P1,P2,P3).

En donde los argumentos son de la siguiente manera:

P1 : La raíz de la palabra que se quiere representar, donde por raíz entendemos a la parte de la palabra que no cambia con género, número o derivativos. Por ejemplo , la raíz de "niño" es "niñ". (En gramática española a esta raíz se le llama morfema).

P2 : El padre de la palabra, por ejemplo, para la palabra "hombre", tomamos al padre como la palabra "humano".

P3 : Una lista con propiedades morfológicas y semánticas de la palabra.

Como podemos ver, en el predicado "pal" , guardamos como primer argumento la raíz de la palabra. Esto lo hacemos por eficiencia, ya que de lo contrario, tendríamos que generar un predicado "pal" para cada definición de la palabra dependiendo de su género y

número. Sin embargo, de alguna manera debemos de saber el tipo morfológico de la palabra, y esto lo hacemos guardando los atributos morfológicos en el argumento P3 de dicho predicado, el cual nos da información sobre la manera en que la palabra varía con respecto a género y número. Por ejemplo para el género, como ya habíamos mencionado antes, utilizamos la siguiente clasificación : doble, únicamente femenino, únicamente masculino, invariante, propio.

De esta manera nos ahorramos una buena cantidad de predicados, aunque obviamente necesitamos una serie de reglas que nos permitan convertir las raíces de las palabras al plural o al singular o bien al femenino o masculino.

En cuanto a los verbos regulares, guardamos el infinitivo del verbo y contamos con una serie de reglas que nos permiten pasar de dicha raíz a un verbo conjugado de acuerdo a la persona, tiempo y número de la frase y viceversa.

Para los verbos irregulares, en cambio, este proceso no es posible debido a las variaciones que presentan en sus conjugaciones, por lo que para definirlos usamos una cláusula "pal" para cada tiempo y persona.

El segundo argumento de la lista pal, que define al padre de la palabra, es a su vez una lista, lo cual permitiría guardar más de un padre para manejar la herencia de propiedades de cada uno de ellos. En esta versión del sistema no manejamos herencia múltiple, sin embargo, la representación que definimos permite hacerlo sin necesidad de cambiar el diseño del sistema.

7.2 Codificación de la frase.

Para esta codificación nos hemos basado en estructuras de Prolog, y hemos utilizado un método posicional para manejar los diferentes casos de la frase.

Contamos con tres tipos básicos de estructuras :

- Lista sint (sintagma nominal).
- Lista prd (predicado).
- Predicado ora (oración).

A continuación describiremos cada una de estas estructuras :

- Lista sint.

Esta lista representa un sintagma nominal. La lista sint contiene cinco elementos :

[S1, S2, S3, S4, S5]

En esta lista los argumento son de la siguiente manera :

S1 : Una lista de dos elementos que contiene en la primera posición la raíz de la palabra que es el principal concepto del sintagma, y en la segunda el tipo de sustantivo que dicha palabra es de acuerdo a su género y número.

S2 : Una lista de tres elementos, en la que se definen el género, el número y la persona del sintagma.

S3 : Una lista que contiene a los determinantes del concepto central del sintagma.

S4 : Una lista que contiene el identificador de cada una de las inflexiones del sustantivo.

S5 : Una lista de preposiciones que introducen al sintagma.

Veamos un ejemplo de la lista sint que se genera para el sintagma nominal : "las mismas manos",

sint([[[mano,unf]],[unf,5,p],[las,mismas],[]])

Vemos en este ejemplo que el concepto principal del sintagma es el sustantivo "mano", cuyo atributo en cuanto a género es el ser "unf" (únicamente femenino), el cual coincide con el atributo "femenino" de los determinantes "las" y "mismas".

Observamos también que la persona del sintagma es la 5 (que corresponde al plural de la 3a. Persona, es decir a "ellos"), y que el número del mismo es plural ("p").

Nos damos cuenta de que no hay inflexiones en este caso pues el cuarto elemento de la lista sint es una lista vacía.

- Lista prd.

La lista prd, representa al predicado de una frase y cuenta con ocho elementos para representar al calificativo de la frase, y a cada uno de los casos que se presenten en la frase. Los elementos se encuentran en el siguiente orden :

[P1, P2, P3, P4, P5, P6, P7, P8]

A continuación describimos cada uno de estos argumentos :

P1 : Una lista que contiene al calificativo de la frase. Esta lista está compuesta por listas de dos elementos, que contienen a los verbos del predicado que pueden ser uno o más (como en "había estado pensando") y el modo de conjugación de cada uno (pues bien pueden ser diferentes como en la frase anterior). Los verbos en esta lista aparecen en infinitivo si son regulares, o bien en el tiempo y persona de la frase si son irregulares.

P2 : Este segundo elemento contiene la palabra "sujeto" si el sujeto actúa como fuente del proceso, de lo contrario es una lista vacía.

P3 : Esta es una lista sint que representa el caso agente si tal caso existe en el predicado o bien es simplemente la palabra "sujeto" si el sujeto es el agente.

P4 : Esta es una lista sint que representa al caso lugar.

P5 : Esta es una lista sint que representa al caso destino.

P6 : Esta es una lista sint que representa al caso tema, o bien es la palabra "sujeto", si el sujeto de la frase es el tema de la acción o proceso.

P7 : Esta es una lista sint que representa al caso agente secundario.

P8 : Esta es una lista de adverbios de tiempo, y representa precisamente al caso tiempo.

Demos ahora un ejemplo de predicado, tomemos la frase :

"estaba comiendo la fruta en la casa con sus manos"

la lista prd generada al analizar este predicado es la siguiente :

```
prd([ [[estaba,copre],[comer,ger]],
        sujeto,
        [],
        [ [[casa,unf]], [unf,3,s], [la], [en] ],
        [],
```

```
[ [[frut,dob]],[unf,3,s],[la],[] ],
[ [[mano,unf]],[inv,0,p],[sus],[con] ],
[] ]).
```

- Predicado ora.

Este predicado engloba a las dos listas anteriores para representar ya una frase completa. Los elementos de esta predicado son los siguientes :

ora (O1, O2, O3, O4, O5, O6, O7).

A continuación describiremos cada uno de estos argumentos :

O1 : Este elemento es utilizado por frases que representan inflexiones de otra frase.Contiene un número que es el identificador de esa otra frase.

O2 : Este elemento es un número y es utilizado como identificador de la frase.

O3 : Este es simplemente un término de Prolog que indica el tipo de oración de que se trata, y toma el valor "int" si la frase es una interrogación , o bien el valor "decl", si la frase es de tipo declarativo.

O4 : Esta es una lista que es vacía si la frase es declarativa, pero en caso de ser interrogativa , contiene dos elementos, que son un par de números que indican la posición que ocupa en la aserción el caso por el que se está preguntando.

O5 : Esta es una lista sint que guarda el lugar en que se desarrolla la acción o el proceso de la oración.

O6 : Esta es una lista sint que guarda el sujeto de la oración.

O7 : Este último elemento es una lista de tipo prd, que representa al predicado de la oración.

Para terminar, demos un ejemplo de la manera en que Andrew representa una frase completa:

"En la casa el feliz robot da la fruta a Isaac con sus manos."

```
oraC[ 1, 1, decl,
       [],
       [ [[casa,unf]],  [unf,3,s], [la], []],
       [ [[robot,inv]], [unm,3,s], [el], [ [[feliz,pres]] , sujeto,
                                                    [], [],[],[],[],[]  ]
       ],
      [ [dar,pres]], [], sujeto,
        [],
        [ [[isaac,pro]], [-,6], [], [] ],
        [ [[frut,dob]],  [unf,3,s], [la], [], [] ],
        [ [[mano,unf]],  [inv,0,p], [sus],[], [con] ],
        []
      ]
  ]).
```

Como podemos apreciar, el seguir esta lista, se torna difícil conforme la oración se va complicando. Por esta razón Andrew cuenta con una opción en la que el usuario puede obtener un listado de la estructura interna generada para una frase determinada, pero en una forma más clara en la que se especifica cada caso de la oración. A continuación mostramos el listado que Andrew genera para la frase anterior :

"En la casa el feliz robot da la fruta a Isaac con sus manos"

```
TIPO FRASE    : decl
LUGAR         :
      CONCEPTO      : [[casa,unf]]
      ATRIBUTOS     : [unf,3,s]
      DETERM.       : [la]

SUJETO        :
      CONCEPTO      : [[robot,inv]]
      ATRIBUTOS     : [unm,3,s]
      DETERM.       : [el]
      INFLEXION     :
            SUJETO        :
                  CONCEPTO      : [[robot,inv]]
                  ATRIBUTOS     : [unm,3,s]
                  DETERM.       : [el]
            PRD
                  CALIF(S)      : [[feliz,pres]]
                  FUENTE        : sujeto

PREDICADO     :
      VERBO(S)      : [[dar,pres]]
      AGENTE        : sujeto
      DESTINO       :
            CONCEPTO      : [[isaac,pro]]
            ATRIBUTOS     : [-,6]

      TEMA          :
            CONCEPTO      : [[frut,dob]]
            ATRIBUTOS     : [unf,s,3]
            DETERM.       : [la]

      AGNT.SEC.     :
            CONCEPTO      : [[mano,unf]]
            ATRIBUTOS     : [inv,0,p]
            DETERM.       : [sus]
```

De esta manera se aprecia más claramente la manera en que Andrew representa internamente una frase.

7.3 Codificación de los Operadores.

En esta sección describiremos la codificación de algunos de los procedimientos más importantes del sistema. Estos procedimientos son definidos como predicados de Prolog que cuentan con una serie de argumentos de entrada y salida.

7.3.1 Codificación del ANALIZADOR.

- Predicado ANALIZADOR.

Objetivo : Generar la estructura interna (representación semántica) de una frase en español.

Código :

```
analizador (Motiv) :-
      call(frase(Lista_pal)),
      abolish(frase/1),
      obten_subl(1,1,2,Lista_pal,Tipo_pal),
      tipo_motiv(Tipo_pal,Motiv),
      inicia_ora(Ora),
      coloca(1,3,Ora,Motiv,Ora_temp),
      llama_proced(Tipo_pal,Lista_pal,Ora_temp,Ora_sal,Resto),
      checa_resto_vacio(Resto),
      asserta(ora(Ora_sal).
```

Argumentos que regresa :

1. Tipo de motivación de la frase.

Descripción : Toma la lista de palabras que componen la frase dada por el usuario y obtiene el tipo de la primera palabra. Basado en este tipo, define el tipo de la frase (declaratiava o interrogativa), mediante el predicado tipo_motiv. Después llama al predicado inicia_ora, para generar la lista Ora, en la que se almacenará la estructura

interna de la frase y llama al predicado llama_proced, que se encarga de llamar a los predicados que convengan de acuerdo a la motivación de la frase, para finalmente regresar la estructura interna.

- Predicado LLAMA_PROCED.

Objetivo : De acuerdo a la motivación de la frase, llama a los procedimientos adecuados para analizar sintagmas nominales y predicado.

```
llama_proced (decl,Tipo,Oracion,Ora_ent,Ora_sal,Resto_final) :-
      llama_ana_sint (Tipo,Oracion,Ora_ent,Ora_temp,Resto_temp,4),
      obten_tipo (Resto_temp,Tipo_temp),
      llama_ana_sint (Tipo_temp,Resto_temp,Ora_temp,Ora_temp1,
                        Resto_temp1,3),
      obten_tipo (Resto_temp1,Tipo_temp1),
      concuerda_persona (Resto_temp1,Ora_temp1,Ora_temp2),
      tipo_verbo (Tipo_temp1,Resto_temp1,[],Verbos,[],
                  Tipo_verbo,Resto_temp2),
      ana_predi (Tipo_verbo,Verbos,Resto_temp2,Prd,0,_,Resto_final),
      coloca (1,5,Ora_temp2,Prd,Ora_sal).


llama_proced (int,Tipo,[Palab|Resto],Ora_ent,Ora_sal,Resto_final):-
      obten (1,1,Palab,Pron_int),
      analiza_objeto_interrogacion (Pron_int,[Palab|Resto],Ora_ent,
                                    Ora_temp,Resto_temp1,Int),
      obten_tipo (Resto_temp1,Tipo_temp1),
      tipo_verbo (Tipo_temp1,Resto_temp1,[],Verbos,[],
                  Tipo_verbo,Resto_temp2),
      ana_predi (Tipo_verbo,Verbos,Resto_temp2,Prd,Int,Sujeto,
                 Resto_final),
      coloca (1,4,Ora_temp,Sujeto,Ora_temp1),
      concuerda_sujeto_predi (Sujeto,Resto_temp1,Ora_temp1,Ora_temp2),
      coloca (1,5,Ora_temp2,Prd,Ora_sal).
```

Argumentos que recibe :

1. Tipo de motivación de la frase.
2. Tipo de la primera palabra de la

frase.
3. Lista de palabras que componen la frase.
4. Lista ora de entrada.

Argumentos que regresa :

5. Lista ora de salida.
6. Parte sobrante de la oración de entrada.

Descripción : Llama a los procedimientos que correspondan, de acuerdo a la motivación de la frase y al tipo de la primera palabra de la misma, para generar su estructura semántica.

- Si la frase es declarativa,

a) llama al predicado llama_ana_sint, que a su vez llama al predicado ana_sint que se encarga de analizar un sintagma nominal, una vez hecho lo cual, coloca la lista sint, resultante de este análisis, en la posición de la lista ora correspondiente al sujeto de la frase,

b) llama al predicado obten_tipo, el cual regresa el tipo de la primera palabra de la lista que recibe como primer argumento.

c) llama, una vez más, a llama_ana_sint, pero esta vez para generar, en caso de que exista, la estructura interna, para el lugar en que se desenvuelve la ación, proceso o estado,

d) obtiene el tipo de la siguiente palabra de la lista por medio de obten_tipo,

e) checa que exista concordancia entre las personas y números de sujeto y verbos de la oración,

f) llama al predicado tipo_verbo, para generar la lista de verbos y obtener de ésta el verbo principal que da el tipo real del predicado (por ejemplo en : "había estado corriendo" , el verbo principal es correr que es un verbo de acción),

g) llama al predicado ana_predi pasándole el resto de la oración y el tipo de verbo que contiene el predicado, junto con otros argumentos. Este predicado se encarga de generar la estructura interna del predicado de la frase.

h) Finalmente coloca en la posición que le corresponde en la lista ora, a la estructura que regresa ana_predi.

De esta manera queda definida la representación interna de una frase declarativa.

En cuanto a las frases interrogativas, se siguen los pasos que enumeramos a continuación :

a) Se obtiene la primera palabra de la lista,

b) se llama al predicado analiza_objeto_interrogación que basado en la primera palabra que obtuvimos en el paso anterior, decide cuál es el caso por el que se está preguntando, y retira de la frase, las palabras interrogativas,

c) se obtiene el tipo de la siguiente palabra de la frase,

d) se llama a tipo_verbo para generar la lista de los verbos de la frase y encontrar el tipo del verbo principal,

e) con este tipo, se llama a ana_predi que analiza el predicado de la frase, regresando la estructura prd correspondiente, así como el sujeto de la misma

f) se coloca la lista prd en el lugar de la lista ora que le corresponde, y,

g) finalmente se checa que exista concordancia entre sujeto y predicado, para terminar colocando el sujeto en la lista ora.

- Predicado ANA_SINT.

Objetivo : De acuerdo al tipo de la primera palabra de la frase, genera la estructura interna, es decir la lista sint, para un sintagma nominal.

```
ana_sint (determ,[Palab|Resto],Sint,Sint_sal,Resto_sal) :-
      obten_atrib (Palab,Atrib,Persona,Numero),
      coloca (1,2,Sint,[Atrib,Persona,Numero],Sint_temp),
      determinante (determ,[Palab|Resto],[],Lista,Resto_sal1),
      coloca (1,3,Sint_temp,Lista,Sint_temp1),
      obten_tipo (Resto_sal1,Tipo),
      adv_o_adj_o_sust (Tipo,Resto_sal1,Sint_temp1,Sint_sal,
                        Resto_sal).

ana_sint (sust,Resto,Sint,Sint_sal,Resto_sal) :-
      sustantivo (sust,Resto,Sint,Sint_sal,Resto_sal).

ana_sint (adjetivo,[Palab|Resto],Sint,Sint_sal,Resto_sal) :-
      adjetivo (det_cali,[Palab|Resto],Sint,Sint_sal,Resto_sal,
                [Palab]).

ana_sint (adverbio,[Palab|Resto],Sint,Sint_sal,Resto_sal) :-
      adverbio_suj (adverbio,[Palab|Resto],Sint,Sint_sal,Resto_sal).

ana_sint (_,Resto,Sint,Sint,Resto).
```

Argumentos que recibe :

1. Tipo de la primera palabra del sintagma.
2. Lista de palabras que componen el sintagma.
3. Lista sint de entrada.

Argumentos que regresa :

4. Lista sint de salida.
5. Parte sobrante de la oración de entrada.

Descripción : Llama a los procedimientos que correspondan para generar la estructura interna del sintagma, de acuerdo al tipo de la primera palabra del mismo, y el tipo de las restantes palabras.

- Si la primera palabra es un determinante :

a) llama al predicado obten_atrib, que obtiene los atributos del determinante en cuanto a género, número y persona.

b) coloca los atributos encontrados, en la segunda posición de la lista sint.

c) llama al predicado determinante, que toma todos los determinantes del sintagma, agrupándolos en una lista , que regresa en la variable "Lista".

d) coloca la lista de determinantes en la posición 3 de la lista sint,

e) obtiene el tipo de la siguiente palabra del sintagma y llama al predicado adj_o_adv_o_sust, pues son éstos los tipos válidos a buscar una vez que fue encontrado un determinante. Este predicado checa el tipo que se le pasa y, de acuerdo al mismo, termina de generar la lista sint.

- Si la primera palabra es un sustantivo :

a) llama al predicado sustantivo que se encarga de analizar el resto del sintagma y de generar la estructura interna para el mismo.

- Si la primera palabra es un adjetivo:

a) llama al predicado adjetivo, que genera una inflexión, y se encarga de analizar el resto del sintagma .

- Si la primera palabra es un adverbio :

a) llama al predicado adverbios_suj que busca algún adjetivo que acompañe al adverbio para generar una inflexión y pasar a analizar el resto del sintagma y a generar la estructura interna para el mismo.

- Si no llega ninguno de estos tipos, el predicado ana_sint, termina su ejecución.

- Predicado ANA_PREDI.

Objetivo : Dependiendo del tipo de verbo que le llegue, llama a los predicados adecuados para reconocer los posibles casos del predicado de la frase.

```
ana_predi(no_existe_verb,_,_,_,_,_,_).

ana_predi (cat_ver_est,Verbos,Resto,Prd_sal,Int,Sujeto,Resto_sal) :-
      define_calif_sujeto (cat_ver_est,Verbos,Resto,Prd,Resto_temp),
      obten_tipo (Resto_temp,Tipo),
      casos (cat_ver_est,Tipo,Resto_temp,Prd,Prd_temp,Resto_sal),
      es_interrogacion (Int,Prd_temp,Prd_sal,Sujeto).

ana_predi (cat_ver_pro,Verbos,Resto,Prd_sal,Int,Sujeto,Resto_sal) :-
      define_calif_sujeto (cat_ver_pro,Verbos,Resto,Prd,Resto_temp),
      obten_tipo (Resto_temp,Tipo),
      casos (cat_ver_pro,Tipo,Resto_temp,Prd,Prd_temp,Resto_sal),
      es_interrogacion (Int,Prd_temp,Prd_sal,Sujeto).

ana_predi (cat_ver_acc,Verbos,Resto,Prd_sal,Int,Sujeto,Resto_sal) :-
      define_calif_sujeto (cat_ver_acc,Verbos,Resto,Prd,Resto_temp),
      obten_tipo (Resto_temp,Tipo),
      casos (cat_ver_acc,Tipo,Resto_temp,Prd,Prd_temp,Resto_sal),
      es_interrogacion (Int,Prd_temp,Prd_sal,Sujeto).
```

Argumentos que recibe :

1. Tipo del verbo principal de la frase.
2. Lista de los verbos de la frase.
3. Lista que contiene las restantes palabras de la frase.
5. Número que indica si la frase que se está analizando es una interrogación.

Argumentos que regresa :

4. Lista prd de salida.
6. Sujeto de la frase, si ésta fue interrogativa.

7. Lista que contiene, el resto de la,
oración. Si esta lista no es vacía,
existe un error en la frase.

Descripción : Llama a los predicados adecuados, para generar la lista prd, que representa internamente al predicado de la frase.

Si el verbo es un verbo de estado :

a) Llama al predicado DEFINE_CALIF_SUJETO, con el tipo del verbo. Este predicado, dependiendo del tipo de la palabra que sigue al verbo, define al calificativo de la frase como el verbo de la misma, o bien como un adjetivo o locativo. De acuerdo también a dicho tipo define al sujeto como TEMA o FUENTE.

b) Llama al predicado OBTEN_TIPO , que regresa al tipo de la palabra que sigue al calificativo.

c) Con dicho tipo y el tipo del verbo, llama al predicado CASOS que utiliza al predicado ANA_SINT para definir los casos del predicado.

d) Finalmente, se llama al predicado es_interrogacion, el cual, en caso de que la frase cuyo predicado se está analizando sea una frase interrogativa, toma de los casos definidos para el predicado el que corresponde al sujeto de la frase. (Por ejemplo, si la frase es :"Cómo estaba el robot?", el predicado CASOS, asignó al sintagma "el robot" como TEMA. Y por su parte, es_interrogación lo toma como SUJETO y vacía el caso de TEMA).

Si el verbo es un verbo de acción :

a) Llama al predicado DEFINE_CALIF_SUJETO, con el tipo del verbo. Este predicado, define al verbo o a los verbos, como calificativo de la frase. Así mismo define al sujeto de la misma como TEMA o AGENTE dependiendo de la voz en que esté dada la frase.

b) Llama al predicado OBTEN_TIPO , que regresa al tipo de la palabra que sigue al calificativo.

c) Con dicho tipo y el tipo del verbo, llama al predicado CASOS que utiliza al predicado ANA_SINT para definir los casos del predicado.

Si el verbo es un verbo de proceso :

a) Llama al predicado DEFINE_CALIF_SUJETO, con el tipo del verbo. Este predicado define al verbo o a los verbos, como calificativo de la frase. Así mismo define al sujeto de la misma como TEMA o FUENTE dependiendo de la voz en que esté dada la frase.

b) Llama al predicado OBTEN_TIPO , que regresa al tipo de la palabra que sigue al calificativo.

c) Con dicho tipo y el tipo del verbo, llama al predicado CASOS que utiliza al predicado ANA_SINT para definir los casos del predicado.

Finalmente, en cualquiera de los casos, se llama al predicado es_interrogacion, el cual, en caso de que la frase cuyo predicado se está analizando sea una frase interrogativa, toma de los casos definidos para el predicado el que corresponde al sujeto de la frase. (Por ejemplo, si la frase es :"Cómo estaba el robot?", el predicado CASOS, asignó al sintagma "el robot" como TEMA. Y por su parte, es_interrogación lo toma como SUJETO y vacía el caso de TEMA).

7.3.2 Codificación del CONTESTADOR.

- Predicado CONTESTADOR.

Objetivo : Encontrar la respuesta a una pregunta hecha por el usuario y guardar temporalmente la estructura de dicha respuesta en la memoria.

```
contestador(Respuesta) :-
      call(ora(Int)),
      cambia_estruct(Int,Int_temp),
      busca_orac(Int_temp,Int_temp1),
      encuentra_respuesta(Int_temp1,Respuesta),
      genera_respuesta(Respuesta).
```

Argumentos que regresa :

1. Lista Respuesta que es vacía si no existió una respuesta.

Descripción : Toma de la memoria, la estructura ora que representa a la frase interrogativa, y llama al predicado cambia_estruct, el cual hace algunos cambios (mencionados en el capítulo VI), en dicha estructura. El contestador, llama después al predicado busca_orac, que se encarga de buscar alguna frase que contenga la respuesta a la pregunta. Llama después al predicado encuentra_respuesta para que encuentre en la frase obtenida mediante el predicado anterior, la respuesta a dicha pregunta. Finalmente llama al predicado genera_respuesta que se encarga de agregar temporalmente la respuesta a la memoria.

7.3.3 Codificación del GENERADOR.

- Predicado GENERADOR.

Objetivo : Generar a partir de una estructura ora, una frase en español.

```
generador :-
      call(resp(Respuesta)),
      analiza_resp(Respuesta,Frase),
```

```
acentua(Frase,Frase_temp),
escribe_frase(Frase_temp),
abolish(resp/1).
```

Descripción : Toma de la memoria, la estructura resp que representa a la respuesta de una frase interrogativa, y llama al predicado analiza_resp. Este predicado llama a los predicados genera_sint y genera_prd, para analizar el sujeto y el predicado de la frase y generar con ellos la frase en español. El predicado generador llama a acentua para que genere en dicha frase, acentos y ñ´s, para finalmente llamar al predicado escribe_frase que despliega la frase. Finalmente quita de la memoria a la estructura que representa a la respuesta, pues ésta no volverá a ser utilizada.

Estos han sido algunos ejemplos del uso de predicados de Prolog para codificar los procedimientos que hemos definido.

VIII. Interfaz con el usuario

8.1 Interfaz con el usuario.

El medio ambiente de Andrew, fue desarrollado con una serie de predicados definidos por Arity Prolog para el manejo de ventanas y en particular de diálogos, que permiten la interacción con el usuario.

Andrew, utiliza un menú principal. En la figura 8.1, vemos este menú.

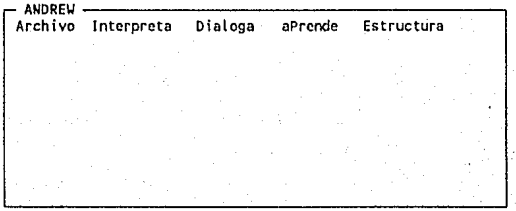


Fig. 8.1.
Menú principal de Andrew.

El menú del sistema tiene una serie de opciones que a continuación describiremos.

1) ARCHIVO.

Como sabemos, la función principal de Andrew, es la de leer una o más frase, y contestar a las preguntas que el usuario le hace sobre dichas frases. Ahora bien, el sistema presenta dos maneras de introducir las frase. Una de ellas es por medio de un diálogo interactivo en el que el usuario puede dar las frases del texto, al mismo tiempo que va

haciendo preguntas, a las que Andrew va contestando. La otra manera de introducir un texto, es por medio de la lectura de un archivo en el que el usuario previamente grabó un texto. Precisamente para el manejo de esta segunda alternativa, el sistema presenta la opción de ARCHIVO.

Esta opción, reúne a aquellas funciones que Andrew permite realizar con un archivo, para lo cual presenta un submenú con las opciones que aparecen en la figura 8.2 :

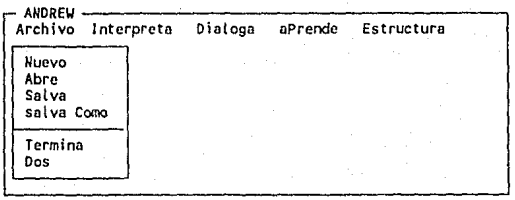


Fig. 8.2.
Submenú ARCHIVO.

a) Nuevo .- Al elegir esta opción, el sistema presenta una pantalla limpia de edición en la que el usuario podrá capturar el texto que deseé introducir. La manera en que se realiza la edición , es muy parecida a la que el mismo Arity Prolog, utiliza cuando se edita un programa. Sus principales características son las que a continuación enumeramos :

i) El movimiento en pantalla se realiza por medio de las teclas de movimiento del cursor y las de "PageUp","PageDown", "Home" y "End".

ii) La tecla de Insert , controla el modo Inserción o Sobreescritura.

iii) Para marcado de palabras, lineas, o bloques de texto, se tiliza la tecla Shift junto con las de movimiento en pantalla.

iv) El copiado de bloques se realiza por medio de la tecla F2, una vez que se ha marcado un bloque y posicionando después el cursor en el lugar al que se desea

copiar el bloque, para finalmente oprimir las teclas Shift-Insert.

v) El borrado de palabras se realiza por medio de la tecla Del, y/o la de backspace. Por otra parte, el borrado de bloques, se realiza presionando la tecla Del una vez que el bloque ha sido marcado .

Es importante hacer notar que cada frase del texto, debe ser dada en una sola linea del mismo.

El usuario puede concluír la edición de un texto por medio de la tecla ESC. En ese momento, el archivo aún no ha sido grabado en disco, para ello se deberá usar la opción Salva o salva Como.

b) Abre .- Por medio de esta opción, el sistema permite elegir alguno de los archivos ya existentes. Para ello presenta una pantalla en la que muestra los archivos que se encuentran en el directorio actual, y permite elegir de entre esa lista a aquél que se quiere editar, ya sea para consulta o bien para hacer, sobre él, algún cambio.

Una vez que se eligió un archivo, se presenta una pantalla en la que el mismo aparece desplegado para ser editado.

c) Salva .- El usuario puede utilizar esta opción, cuando desea grabar en disco un archivo con el mismo nombre que dicho archivo ya tenía. La opción salva es principalmente usada cuando se quiere grabar en disco un archivo que ya existía y que fue abierto y modificado por medio de la opción Abre.

d) salva Como .- Esta opción es utilizada cuando se acaba de editar un archivo por medio de la opción nuevo, por lo que se asume que se quiere dar un nombre al archivo y

grabarlo a disco.

e) Termina .- El usuario utiliza esta opción cuando quiere dar por terminada la sesión con Andrew.

f) Dos .- Esta opción permite salir temporalmente al sistema operativo DOS. Para regresar al sistema, basta con teclear la palabra exit.

2) INTERPRETA.

Esta opción permite llevar a cabo la interpretación de un archivo creado o modificado por medio de algunas de las opciones del submenú ARCHIVO que describimos en el inciso anterior.

Para ello , el sistema presenta una pantalla en la que aparece una lista de los archivos existentes, permitiendo que el usuario elija alguno de ellos. Posteriormente llama al procedimiento INTERPRETA que a su vez se encarga de llamar a los procedimientos adecuados para convertir cada una de las frases que componen el texto en Español, a la representación interna (semántica) que Andrew utiliza y que ya hemos descrito anteriormente.

Una vez concluído este proceso, el usuario podrá utilizar la opción Dialoga del menú principal para hacer a Andrew algunas preguntas sobre el archivo interpretado.

3) DIALOGA.

Las dos opciones que hemos visto hasta este momento son las encargadas del

manejo de archivos. Sin embargo, como comentamos en el inciso 1, existe otra manera de dar una serie de frases a Andrew. Dicha manera representa un método interactivo en el que el usuario da la información al sistema y al mismo tiempo éste la procesa para poder dar en ese momento la respuesta a las preguntas que el usuario le haga al respecto.

Para llevar a cabo este proceso de diálogo, el sistema presenta una pantalla en la que aquellas palabras que aparecen con un color mas brillante son las emitidas por Andrew, mientras que las de el usuario se manejan en un color blanco mate.

Al principio, esta pantalla aparece en blanco, excepto por un mensaje de bienvenida y el prompt ">" que indica que el sistema está esperando a que el usuario comience a dar frases o bien a hacer preguntas sobre la información preexistente ya sea por un diálogo anterior o bien por un archivo que fue interpretado.

En caso de dar una frase declarativa, Andrew informará que comprende la frase por medio de la palabra "ok". En caso de existir algún problema avisará de que problema se trata y en el particular caso de que existan palabras que no tiene definidas en su vocabulario dará la opción de definirlas.

Para terminar la sesión de diálogo basta con teclear la palabra "adios" en el prompt de Andrew.

En el apéndice A, daremos ejemplos de diálogo con el sistema.

4) APRENDE.

Por medio de esta opción del menú principal le enseñamos nuevas palabras a Andrew. Este módulo es también utilizado por el sistema, si durante el diálogo aparece algún concepto desconocido.

Para dar una explicación del funcionamiento de este módulo, vale la pena recordar la manera en que está organizado el vocabulario. Como ya habíamos mencionado, cada palabra del mismo , está dada por el predicado "pal" que contiene cuatro argumentos : la raiz de la palabra, el padre de la misma y una serie de atributos.

Dada esta representación, el usuario no solo debe enseñar a Andrew la nueva palabra, sino de alguna manera debe informar quien es el padre y algunos atributos de la misma.

Para facilitar esta tarea, el sistema no se limita a preguntar estos datos, sino que por el contrario, presenta una serie de pantallas que describiremos en el Apéndice B.

Lo primero que Andrew hace es preguntar la nueva palabra. Una vez que el usuario la da, checa que no exista ya la raiz de dicha palabra en su diccionario. De existir, manda un mensaje y pide alguna otra palabra.

En caso de que efectivamente la palabra no esté ya definida, presenta una pantalla en la que muestra las categorías de palabras que tiene definidas.

Estas categorías corresponden a los tipos sintácticos de las palabras , es decir, sustantivo, determinante, etc.

De entre esta lista el usuario podrá escoger una categoría, simplemente moviéndose con las teclas de movimiento de cursor y tecleando "return" en la categoría adecuada. En la figura 3.8., vemos un ejemplo de la pantalla en que el usuario define la categoría de una palabra.

Una vez elegida la categoría, el sistema toma a todas las palabras que descienden directamente de la misma y una vez mas las muestra al usuario para que elija a la que de alguna manera englobe a la palabra que quiere definir.

Demos un ejemplo, si quisiéramos definir la palabra "mesa", la categoría que eligiríamos sería la de sustantivo.

Después el sistema nos daría las opciones : animado, inanimado, ninguno.

Como estamos definiendo una "mesa", eligiríamos obviamente la palabra "inanimado".

El sistema continúa presentando este tipo de pantallas hasta que el usuario después de revisar una lista de palabras se da cuenta de que la que quiere definir no desciende ya de ninguna de las palabras que se encuentran en dicha lista. En ese momento la elección del usuario deberá ser la de "ninguno", por lo que Andrew, pasa a preguntar datos particulares de la palabra.

Los datos que pregunta dependen de la categoría a la que pertenece la palabra y de la palabra misma. En el caso de ser un verbo, por ejemplo, pide el tiempo, el infinitivo, el tipo y la persona.

De esta manera el usuario puede definir nuevas palabras sin tener que adivinar que es lo que debe dar como tipo o categoría.

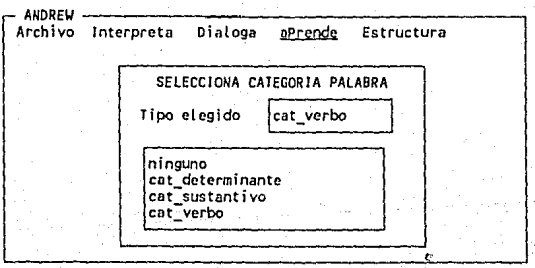


Fig. 8.3.
Ejemplo de pantalla de la opción APRENDE.

5) ESTRUCTURA.

Esta última opción muestra, en impresora o en pantalla, la estructura semántica que Andrew genera para una frase dada. Para ello pide al usuario le de una frase en español y después de llamar al ANALIZADOR, presenta la estructura que éste genera para dicha frase.

De esta manera el usuario podrá ver la manera en que nuestro sistema almacena sus conocimientos. Un ejemplo de esta estructura se muestra en el Apéndice C.

CONCLUSIONES.

Como mencionamos en la introducción, el objetivo central de este trabajo fue el desarrollo de un analizador de frases utilizando el modelo de gramticas de casos. Este analizador fue desarrollado y probado con un sistema creado para hacer preguntas y respuestas sobre las frases dadas.

Como pudimos apreciar en los ejemplos que mostramos, las frases que se reconocen son más o menos complejas y la estructura semántica creada por el sistema funciona de acuerdo a la definición de las gramáticas de casos, la cual resulta ser eficiente en el sentido que el sistema puede de alguna manera comprender el papel semántico que juega cada elemento de la frase y con base en dicho conocimiento, contestar a las preguntas del usuario. Sin embargo, en esta primera etapa, el sistema presenta algunas limitaciones que hemos ido señalando en el transcurso de su presentación y que a continuación resumimos.

Una de las principales limitaciones del sistema, es que no maneja las frases dentro de un contexto y que no se aboca al problema del reconocimiento de pronombres. El reconocimiento de pronombres permitiría la interpretación de todo un tipo de frases que por el momento no se manejan y permitiría también responder a preguntas complicadas, ya no sobre frases aisladas, sino sobre personajes o conceptos definidos en dichas frases pero considerándolas como componentes de una historia.

En cuanto al manejo del vocabulario, nuestro sistema utiliza una estructura de arbol sintáctico-semántico pero como ya mencionamos, el nivel semántico, aunque esta definido, no se explota realmente. Una posible extension del sistema, es justamente el introducir el manejo del nivel semántico del arbol. Esto ayudaria al sistema en el problema del razonamiento deductivo, utilizando las propiedades asociadas a cada nodo por medio de la herencia y la informacion dada por las frases dentro del diálogo.

Esta mejora es una de las mas importantes y con este fín se uilizo Prolog como lenguaje de programación, ya que nos permite contar con una maquina de inferencia para el razonamiento.

En el capítulo III, mencionamos algunas de las limitaciones de los casos utilizados para expresar ciertas relaciones. Por ejemplo, en la frase "El robot corre feliz", el sistema asocia el hecho de "ser feliz" al "robot", independientemente de la acción de "correr", aunque en realidad el "ser feliz" pueda deberse precisamente a dicha acción. Este tipo de limitaciones podrían afinarse con un estudio más cuidadoso sobre el conjunto de casos utilizado y su manejo.

Por último, mencionamos que entre los casos que manejamos se encuentran los casos de TIEMPO y LUGAR. Nuestro sistema utiliza estos casos únicamente para contestar a preguntas directas sobre el "cuándo" o "dónde", se realizó alguna acción. Sin embargo, esta información podría ser utilizada para manejar el tiempo y el lugar dentro de un contexto, lo cual nos permitiría por ejemplo, saber si algún hechò tomó lugar antes o después de otro. El uso de la temporalidad en el sistema es otra de las posibilidades a manejar en algún desarrollo futuro.

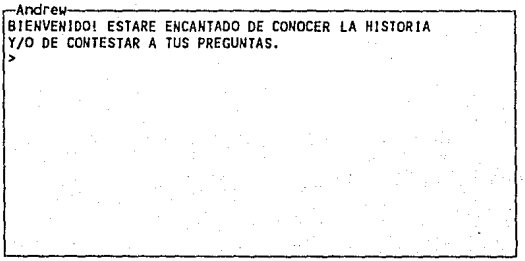
Como puede apreciarse, hay todavia mucho trabajo por hacer en el sistema, para poder explotar realmente el modelo de gramticas de casos en el reconocimiento y la comprensión del lenguaje natural. Por esta razon, nuestro trabajo debe situarse como la primera piedra para el desarrollo de un buen n#mero de nuevos proyectos de investigacion alrededor del uso de gramaticas de casos como modelo de representacion de informacion en un dialogo.

APENDICE A.

- Ejemplos de diálogo.

En este apéndice mostraremos ejemplos de diálogos entre Andrew y el usuario, y el tipo de respuesta que presenta.

La siguiente es la primera pantalla que Andrew presenta al elegir, el usuario la opción de Diálogo en el menú.



En esta pantalla Andrew invita al usuario a introducir una serie de frases, o bien a realizar preguntas sobre las frases dadas en algún archivo o al momento del diálogo. Andrew utiliza el símbolo ">", para indicar que espera alguna acción de parte del usuario.

Veamos un ejemplo del tipo de diálogo sostenido entre Andrew y un usuario.

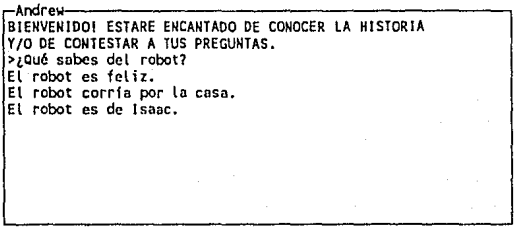
```
Andrew
¡BIENVENIDO! ESTARE ENCANTADO DE CONOCER LA HISTORIA
Y/O DE CONTESTAR A TUS PREGUNTAS.
>El feliz robot corría por la casa.
ok
>El robot es de Isaac.
ok
>Los niños que estaban en la cocina comieron las frutas.
ok
>¿Quién corría por la casa?
El robot que es feliz.
>¿En dónde corría el robot?
Por la casa.
>¿Qué niños comieron las frutas?
Los niños que estaban en la cocina.
>¿De quién es el robot?
De Isaac.
```

Para cada frase declarativa dada, Andrew utiliza el ANALIZADOR y genera su estructura interna. Si la frase no presenta palabras desconocidas o errores de sintáxis, Andrew indica que todo está bien por medio de la palabra "ok". En caso de que exista un error de sintáxis, Andrew lo informa al usuario y espera que éste introduzca una frase correcta.

En el siguiente apéndice daremos un ejemplo del proceso que Andrew sigue si en la frase existen conceptos desconocidos.

En el caso de las frases interrogativos, Andrew llama a los operadores que permiten contestar. Si la respuesta a una pregunta existe, Andrew la presenta al usuario, de lo contrario contesta con un "no se", y espera una nueva frase.

El usuario puede también pedir que Andrew le hable sobre algún concepto en particular. Veamos un ejemplo:



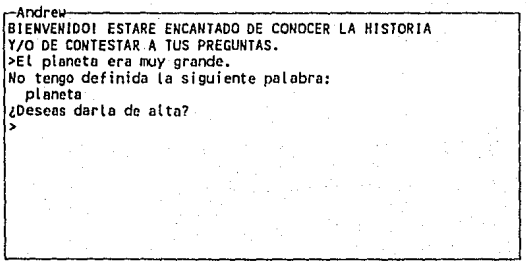
En este ejemplo Andrew muestra lo que sabe del "robot", de acuerdo a las frases dadas anteriormente por el usuario.

APENDICE B.

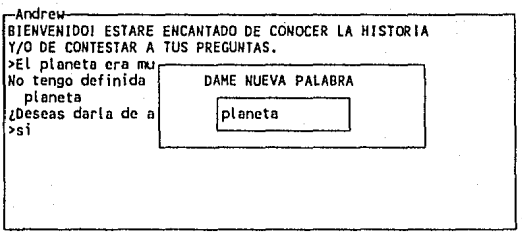
- **Ejemplo de aprendizaje de conceptos.**

En este apéndice mostraremos un ejemplo del aprendizaje de un concepto durante el diálogo.

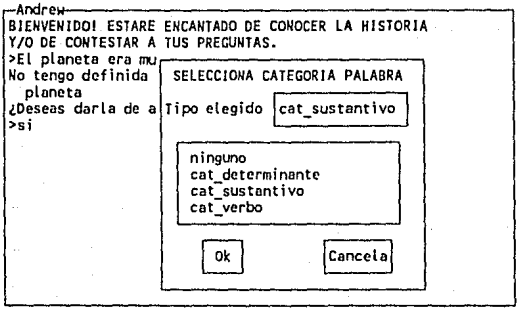
Supongamos que durante el diálogo, en una frase existe una palabra desconocida. Andrew reacciona de la siguiente manera :



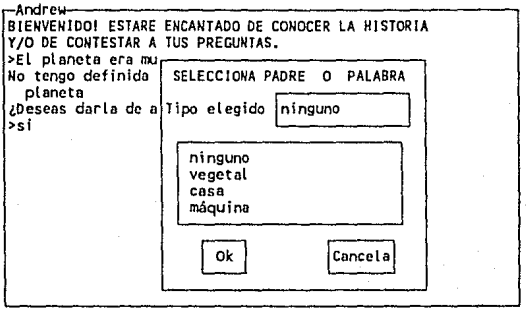
Como podemos ver, al detectar un concepto que no conoce, Andrew lo informa al usuario y le pregunta si desea dar de alta dicho concepto en el vocabulario. Una vez hecho ésto, Andrew espera una respuesta afirmativa o negativa. Veamos lo que ocurre si el usuario contesta que "si".



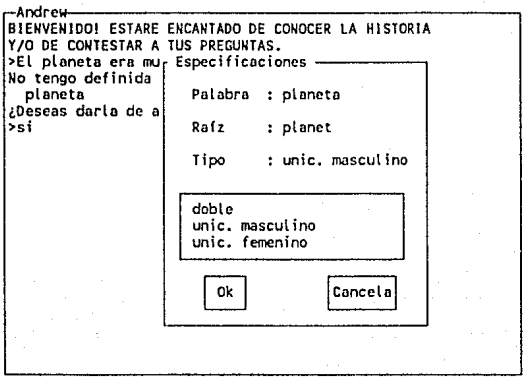
En esta pantalla Andrew pide la palabra a definir. Una vez dada esta palabra , presenta la siguiente pantalla :



Similares a ésta, Andrew presenta una serie de pantallas en las que permite definir la categorías del árbol del vocabulario a las que el concepto "planeta" pertenece. Veamos la última de estas pantallas en la que dicho concepto ya no pertenece a ninguna de las categorías presentadas, por lo que el usuario elige la opción "ninguno".



Una vez que el usuario llegó al nodo del árbol en que el concepto debe ser colocado, Andrew pasa a preguntar atributos particulares a dicho concepto. A continuación podemos ver la pantalla en que se piden las especificaciones particulares del concepto :



Una vez dada las especificaciones del concepto, Andrew envía un mensaje en el que informa que ya conoce la palabra y continúa con el diálogo.

Si más de una palabra de la frase resultaron desconcidas, Andrew sigue este proceso para cada una de ellas.

La pantalla de especificaciones es diferente para cada categoría gramatical. Si el concepto es por ejemplo un verbo, las especificaciones piden información sobre el tipo, tiempo y persona del verbo.

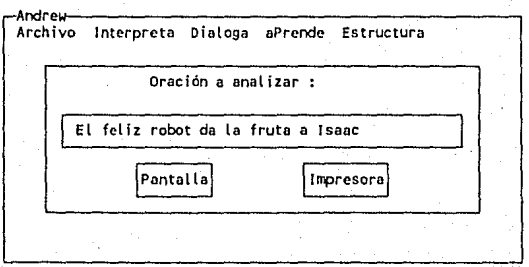
La opción **Aprende** del menú principal de Andrew permite utilizar este proceso para definir nuevos conceptos, en el momento que el usuario lo requiera, independientemente del diálogo.

APENDICE C.

- Ejemplo de despliegue de estructura semántica.

En este apéndice mostraremos un ejemplo del despliegue de la estructura semántica asociada a una frase. Este proceso se realiza por medio de la opción **Estructura** del menú principal.

Cuando el usuario elige esta opción, el sistema presenta una pantalla en la que pide la oración cuya estructura se desea conocer. Esta pantalla es como la siguiente :



Una vez dada la frase Andrew utiliza al ANALIZADOR para generar la estructura interna de la misma y la presenta en impresora o pantalla, según haya elegido el usuario.

La estructura generada en este caso es la siguiente :

"El feliz robot da la fruta a Isaac"

```
TIPO FRASE    : decl

SUJETO        :
      CONCEPTO      : [[robot,inv]]
      ATRIBUTOS     : [unm,3,s]
      DETERM.       : [el]

      INFLEXION     :
            TIPO FRASE    : decl
            SUJETO        :
                  CONCEPTO      : [[robot,inv]]
                  ATRIBUTOS     : [unm,3,s]
                  DETERM.       : [el]

            PREDICADO     :
                  CALIF(S)      : [[feliz,pres]]
                  FUENTE        : sujeto

PREDICADO     :
      CALIF(S)      : [[dar,pres]]
      AGENTE        : sujeto
      DESTINO       :
            CONCEPTO      : [[isaac,pro]]
            ATRIBUTOS     : [-,6]

      TEMA          :
            CONCEPTO      : [[frut,dob]]
            ATRIBUTOS     : [unf,s,3]
            DETERM.       : [la]
```

BIBLIOGRAFIA

[1] C.J. FILLMORE, 1968
"The case of case", in Bach and Harms eds. universals in linguistics theory, Chicago : Holt, Rinchart and Winston Inc. U.S.A.

[2] M. CELCE - MURCIA, 1972
"English comparatives", Ph. D. thesis, Los Angeles UCLA. Department of Linguistics. U.S.A.

[3] C. LOYO, 1976
Doctorat thèse : "Une experience informatique de compréhension du langage naturel". L'Universite Pierre et Marie Curie. France.

[4] G. SABAH, 1978
Doctorat thèse : " Contribution a la compréhension effective d'un récit". L'Universite Pierre et Marie Curie. France.

[5] T. WINOGRAD
" Language as a cognitive process ", Addison-Wesley Publishing Company, U.S.A.

[6] JAMES ALLEN, 1987
" Natural Language Understanding", The Benjamin Kummings Publishing Company Inc., U.S.A.

[7] G. GAZDAR, C. MELLISH, 1989
" Natural Language Processing in Prolog", Addison - Wesley Publishing Company, U.S.A.

[8] E. CHARNIAK, D. McDERMOTT, 1987
"Introduction to Artificial Intelligence", Addison - Wesley Publishing Company, U.S.A.

[9] E. RICH, 1976
"Artificial Intelligence", McGraw-Hill, U.S.A.

[10] L. STERLING, E. SHAPIRO
"The art of Prolog", The MIT Press, U.S.A.

[11] ARITY CORPORATION, 1986
" Building Arity Prolog Applications ", Arity Corporation, U.S.A.