

Dpto. de Lógica y Filosofía de la Ciencia
Facultad de Filosofía
Universidad Complutense de Madrid

**Diseño e implementación de un sistema automático
de análisis documental**

Carmen López Rincón

Tesis doctoral dirigida por el Prof. Dr. D. Jaime Sarabia

Madrid, Junio 1995

Índice

ÍNDICE	1
INTRODUCCIÓN	7
DESCRIPCIÓN DE LENDEX	16
INTRODUCCIÓN	16
TIPOS DE ÁREAS	19
<i>Reconocimiento y recuperación: áreas R, áreas RR</i>	<i>19</i>
<i>Definición débil, definición estricta: áreas DD, áreas ED</i>	<i>20</i>
<i>Relación entre unas y otras</i>	<i>21</i>
<i>Un documento ISBD</i>	<i>21</i>
<i>Otros conceptos relacionados</i>	<i>27</i>
DOCUMENTOS, ÁREAS, LEXEMAS	29
<i>La gramática del lexema</i>	<i>32</i>
<i>La gramática común de los documentos</i>	<i>37</i>
LA GRAMÁTICA DE LENDEX	41
G_{LENDEX}	41
COMENTARIOS A G_{LENDEX}	53
UN EJEMPLO DE GRAMÁTICA LENDEX	56
<i>Primera producción de una gramática LENDEX: descripción del documento o clase de documentos a tratar</i>	<i>57</i>
<i>Aserción de las áreas DD del documento con indicación del rasgo asociado</i>	<i>58</i>
<i>Aserción de los fines correspondientes a cada una de las áreas DD, con indicación del tipo de fin: interno, externo_homo o externo_hetero</i>	<i>59</i>
<i>Aserción de las condiciones asociadas a áreas DD y ED, con indicación de su tipo: fuerte o débil</i>	<i>61</i>
<i>Producciones correspondientes a las áreas ED</i>	<i>62</i>
EL PROGRAMA	65
METODOLOGÍA DE PRESENTACIÓN DEL PROGRAMA	68
REPRESENTACIÓN DE LOS DOCUMENTOS	70
<i>Formato de entrada: G_{Input}</i>	<i>70</i>

Un ejemplo de Input.....	73
<i>Formato de salida: G_{Output}</i>	75
Un ejemplo de Output.....	78
ANÁLISIS DE INTEGRIDAD Y CONSISTENCIA.....	81
<i>Errores sintácticos</i>	81
<i>Gramaticalidad de la GDU</i>	82
<i>Restricciones adicionales</i>	87
<i>Procedimientos auxiliares al análisis de integridad</i>	91
GENERACIÓN DE LA BASE DE CONOCIMIENTO PROLOG A PARTIR DE LA GDU.....	93
<i>Generación automática de las gramáticas de áreas DD</i>	95
Definición de una gramática de área DD.....	95
Estructura general de la gramática de áreas DD.....	100
Otras consideraciones de interés relativas a la gramática de áreas DD.....	102
Generación automática de la gramática de áreas DD.....	106
Aplicación a un ejemplo real.....	114
<i>Aserción de las cláusulas para las áreas ED</i>	116
<i>El procedimiento meta_lendex/6: un meta-intérpete de gramáticas</i>	118
Precedentes.....	118
Implementación y desarrollo.....	123
Cláusula de anidamiento de meta-lendex/6.....	123
Cláusula del literal cat_eval/5 a la cabeza de la lista a analizar.....	124
Cláusulas de la conjunción, disyunción, negación, objetivos Prolog y terminales.....	126
Cláusula de las áreas R.....	129
Cláusulas de expresiones específicas LENDEX.....	129
1. Lexemas descritos.....	130
2. Secuencias.....	133
3. Términos cuantificados.....	134
Cláusulas de áreas DD y ED.....	139
Cláusula final.....	143
Procedimientos usados por meta_lendex/6.....	145
ANÁLISIS DE LOS DOCUMENTOS.....	151
<i>Aserción de la cláusula principal de la descripción</i>	151
<i>El procedimiento analisis/4</i>	152
Discusión y justificación.....	154
Explicación.....	158
PROCEDIMIENTOS ESPECÍFICOS DE LENDEX.....	162
<i>El procedimiento hay/6: alternativa a la evaluación Prolog de categorías gramaticales</i>	162

<i>Análisis flexible de una secuencia de lexemas: hay_multiple/5</i>	168
<i>Procedimientos gramaticales para los términos cuantificados</i>	171
Procedimientos relativos a términos cuantificados mediante un operador iterativo	177
1. Cero o uno	178
2. Cero o más	179
3. Uno o más.....	180
Procedimientos relativos a términos cuantificados mediante un cuantor numérico.....	181
1. Desde N.....	182
2. Hasta N.....	183
3. Entre N y M.....	184
4. Exactamente N.....	185
Aplicación sobre un caso concreto	186
<i>Otros procedimientos gramaticales</i>	191
Procedimientos gramaticales de sólo reconocimiento para términos cuantificados mediante un operador iterativo.....	195
Procedimientos gramaticales de sólo reconocimiento para términos cuantificados mediante un cuantor numérico.....	197
REVISIÓN	200
<i>Generación automática de gramáticas para las áreas DD: segunda versión. Analizadores frente a reconocedores</i>	204
Generación de analizadores para las áreas DD.....	206
Nueva versión del programa.....	207
Aplicación a un ejemplo real.....	210
Modificación de meta_lendex/6	211
<i>Un sistema de aserción de lemas</i>	212
<i>Compilación de las expresiones LENDEX. Versión final de S-Lendex</i>	214
Compilación de las producciones de la GDU.....	215
Generación de las gramáticas de áreas DD compilando los fines. Tercera y última versión	221
Aplicación de la compilación a un ejemplo real.....	223
Adaptación de meta_lendex/6	225
Adaptación de analisis/4	229
PROCEDIMIENTOS DE USO GENERAL EMPLEADOS POR S-LENDEX	232
<i>Procedimientos de carácter general</i>	232
<i>Procedimientos previos al análisis de los documentos</i>	234
<i>Procedimientos posteriores al análisis de los documentos</i>	235
APLICACIONES	236

VARIACIONES SOBRE EL ANÁLISIS DE UNA CLASE DE DOCUMENTOS	236
<i>Documento original</i>	237
1ª GDU	238
Resultado.....	238
2ª GDU	240
Resultado.....	240
3ª GDU	241
Resultado.....	241
4ª GDU	242
Resultado.....	242
5ª GDU	243
Resultado.....	243
6ª GDU	244
Resultado.....	244
7ª GDU	245
Resultado.....	245
8ª GDU	246
Resultado.....	246
9ª GDU. última versión.	247
Resultado.....	248
ANÁLISIS DE OTRAS CLASES DE DOCUMENTOS.....	249
<i>Índice del catálogo de discos</i>	249
Documento original.....	249
GDU.....	250
Resultado.....	250
<i>Base de datos documental</i>	252
Documento original.....	252
GDU.....	253
Resultado.....	253
<i>Diccionario</i>	255
Documento original.....	255
GDU.....	256
Resultado.....	256
<i>BOE</i>	258
Documento original.....	258
GDU.....	259
Resultado.....	260

<i>Anuncios de prensa</i>	261
Documento original.....	261
GDU.....	262
Resultado.....	263
<i>Tractatus</i>	264
Documento original.....	264
GDU.....	265
Resultado.....	265
<i>Revistas</i>	267
Documento original.....	267
GDU.....	268
Resultado.....	268
CONCLUSIONES	270
ANEXO 1: UNA GDU	274
ANEXO 2: DOCUMENTOS INPUT	276
BASE DE DATOS DOCUMENTAL.....	276
ÍNDICE.....	278
VILLANCICOS.....	281
DICCIONARIO.....	285
BOE	289
ANUNCIOS DE PRENSA	293
TRACTATUS	295
REVISTAS	298
ANEXO 3: DOCUMENTOS OUTPUT	300
BASE DE DATOS DOCUMENTAL.....	300
ÍNDICE.....	302
VILLANCICOS.....	306
DICCIONARIO.....	310
BOE	315
ANUNCIOS DE PRENSA	318
TRACTATUS	321
REVISTAS	326
LISTAS	329

ABREVIATURAS.....	329
FIGURAS.....	330
TABLAS.....	330
PROCEDIMIENTOS.....	331
BIBLIOGRAFÍA.....	333

Introducción

El objetivo de este trabajo consiste -como su título sugiere- en el desarrollo de un sistema de análisis documental automático: S-Lendex.

Los motivos que llevan a hacer interesante la tarea de analizar automáticamente los documentos son muy variados. Sin entrar en cuestiones de carácter sociológico o histórico, es indudable hoy que una buena gestión de la información en todas sus formas es condición imprescindible para el éxito prácticamente de cualquier empresa. Y esto, la gestión adecuada de la información, pasa necesariamente por un tratamiento automático de la misma, [García-Lucas,87].

El desarrollo de sistemas informáticos tendentes a cubrir esta necesidad se centra fundamentalmente en el problema de la gestión de la información actual: son cada vez más y mejores las herramientas que se enfrentan con solvencia a este reto.

Sin embargo, y por las mismas razones que hoy las hacen absolutamente imprescindibles, el auge de este tipo de sistemas ha desvelado un problema acuciante: mucha de la información necesaria, útil o relevante para quien adopta un sistema de estas características se encuentra en un soporte o formato distintos al exigido por la máquina: pensemos en el papel de las fichas de biblioteca o el formato propio de una base de datos primitiva. Con lo cual acontece la paradoja consistente en que un sistema teórica y prácticamente pensado para resolver de manera automática tareas antes realizadas manualmente impone la exigencia de un trabajo previo -sordo y muy poco reconocido- de trasvase manual de la información original a un formato comprensible por el nuevo sistema.

Este problema, el de la conversión retrospectiva de la información, otorga sentido al desarrollo de S-Lendex; aunque es relevante notar que no de manera exclusiva. Planteado en primera instancia en los centros por excelencia de difusión de la

información, las bibliotecas, hoy puede decirse que es un problema compartido por todos aquéllos que trabajan, de un modo u otro, con información. La proliferación de sistemas diversos con formatos distintos hace que la definición de *standards* de representación de la información se convierta en una tarea de particular importancia. Nos hallamos en una especie de Babel moderna en que la comunicación se hace difícil por la heterogeneidad y diversidad de los medios en que la información se encuentra.

Cualquier herramienta que haga posible el tratamiento automático de la información presente en documentos -bien sea para tareas de conversión y transformación a otros formatos, bien para otro tipo de operaciones diversas: clasificación automática, recuperación flexible, generación de bases léxicas para sistemas de procesamiento del lenguaje natural, resúmenes automáticos, etc- se hace particularmente interesante.

S-Lendex pretende ser un paso más en esta dirección. El objetivo fundamental de este trabajo consiste en analizar automáticamente clases o familias de documentos y conseguir una representación uniforme de los mismos. Los documentos objeto del análisis de S-Lendex contienen fundamentalmente texto estructurado. Este concepto -el de estructura o formato de los documentos- resulta de capital importancia en el desarrollo del sistema: constituye el eje sobre el cual S-Lendex opera.

Efectivamente, planteado el objetivo de tratar información con fines diversos, adquiere relevancia un hecho: ésta, al plasmarse en documentos, tiende a adoptar formas constantes. Las clases de documentos objeto del análisis de S-Lendex se constituyen en tanto que clases en la medida que sus elementos comparten una serie de rasgos uniformes, lo que podría denominarse un aire de familia. Éste consiste en el modo particular en que esa familia de documentos dispone la información. Rasgos físicos como encolumnado, sangría y alineación, así como otros de carácter sintáctico y semántico constituyen lo que aquí denominamos

estructura o formato de los documentos: conocerla es condición *sine qua non* para la adecuada comprensión del documento.

Las regularidades que caracterizan una clase -su estructura-, se encuentran en ocasiones muy bien definidas en forma de normas estrictas que pueden guiar su descripción. Otras veces, sin embargo, no están explicitadas en modo alguno y se requiere un proceso fundamentalmente inductivo para descubrirlas.

En un caso y otro, si los documentos son el resultado de un trabajo de varios años o de una tarea común de agentes distintos, lo que resulta claro es que la descripción de la clase debe ser lo suficientemente flexible como para admitir variaciones en el formato fruto inevitable de ambas circunstancias.

Flexibilidad, así pues, en la descripción concreta de cada una de las clases, y flexibilidad en el lenguaje general de descripción capaz de enfrentarse a familias diversas de documentos.

La primera formulación del propósito de S-Lendex se encuentra en [Sarabia,94]: *“Aunque la versión actual de Laura¹ es capaz de tratar varios formatos de documentos, el campo de aplicación es muy amplio, incluso restringiéndose al ámbito de la literatura de referencia. Un objetivo que parece bastante alcanzable es lograr un instrumento que permita generar los expertos² adecuados de manera fácil y rápida: un sistema que interprete una descripción, de nivel tan alto como posible, de una clase de documentos como un experto para la clase.”*

¹Laura es un sistema de análisis automático de registros bibliográficos desarrollado bajo los auspicios del Vicerrectorado de Investigación de la UCM por el equipo de investigación dirigido por J.Sarabia y al que la autora de este trabajo pertenece: Verba Logica, en el Dpto. de Lógica de la UCM.

²Con “expertos” el autor alude a los sistemas automáticos especializados en un cierto formato.

S-Lendex pretende responder, justamente, al objetivo formulado: generar -a partir de descripciones en un lenguaje de alto nivel- un análisis de los documentos que permita gestionar la información contenida en ellos con fines diversos: un acceso más ágil, una recuperación más eficaz, otra representación, su traducción a un formato distinto, etc.

El contexto en el que este trabajo adquiere sentido es el del tratamiento automático de textos estructurados. El concepto de estructura o formato es -como antes apuntamos- capital en este campo, [Akindele-Belaïd,94]: *“Trabajos recientes en análisis de documentos han mostrado la necesidad de disponer de un modelo estructural de documentos para guiar el reconocimiento y mejorar los resultados”*

Analizar un documento pasa por reconocer su estructura característica y representar la información de la que es medio en un formato distinto del original. El contexto inmediato de S-Lendex es el antes indicado: Laura es el primero de una serie de programas desarrollados por Verba Logica cuyo propósito se enmarca en este ámbito: el del análisis automático de documentos en función de su estructura o formato característicos. Hasta el momento la experiencia se ha centrado en literatura de referencia: fichas bibliográficas en distintos formatos, catálogos y registros de publicaciones seriadas.

S-Lendex pretende ser un paso más en la generalización del proceso de análisis documental en la medida que incorpora un lenguaje de descripción uniforme y la capacidad de generar los analizadores especializados en cada una de las clases. Como muestra del campo posible de aplicación se proponen al final otras familias de documentos ajenas al ámbito bibliográfico: textos legales, diccionarios, noticias de prensa, etc.

Otros trabajos son relevantes a este respecto. Uno de ellos es el desarrollado por el proyecto europeo LIB-MORE para la conversión de catálogos técnicos de la

Biblioteca Real de Bélgica, [Chenevoy-Belaïd,94]. Como indican los autores, el sistema presenta dos versiones: una de demostración en el CRIN y otra industrial a cargo del grupo *Jouve*, en Francia. El trabajo se asemeja al antes citado Laura en la medida que comparte la clase de documentos original: registros bibliográficos y la visión general del problema: *“Se trata de saber representar la estructura física y lógica de manera suficientemente general para describir el conjunto de una clase posible de documentos... La dificultad esencial proviene del número infinito de documentos susceptibles de ser reconocidos por un sistema de este tipo”*

Otro proyecto de interés en este campo es el desarrollado por la Biblioteca Real de Dinamarca, [NORDInfo,92], para la conversión retrospectiva de sus fondos, en la misma línea de investigación que los anteriores. Hasta donde conocemos, las variaciones en el formato que estos dos proyectos son capaces de tratar son menores que las contempladas por Laura.

Importa destacar también el proyecto [Acquilex II,89] de generación de bases de datos léxicas a partir de diccionarios. Aplicado entre otros al bilingüe Vox-Harrap en soporte magnético, en él se desarrolla un análisis de la estructura de las entradas que -con metodología distinta- se asemeja al efectuado por S-Lendex a partir de una GDU.

Asimismo, el problema de la estructuración de los documentos ha sido últimamente objeto de estudio en el contexto de desarrollo de sistemas de OCR: *optical character recognition*. Pese al progreso de este campo en los últimos años, encontramos en bibliografía especializada como [Cochard-Ingold,94] lo siguiente: *“Es forzoso constatar que el avance no es suficientemente rentable para numerosas aplicaciones prácticas. En estos casos, la captación manual resulta, aún hoy, la única alternativa posible”*

Es precisamente esta situación la que ha llevado a proponer sistemas, como ESCROC, en que las diferentes fases del tratamiento: segmentación, OCR,

etiquetado lógico, están más integradas. La idea que subyace es la misma que la apuntada anteriormente: reconocer documentos -en sus distintos niveles- pasa por hacerse cargo, en mayor o menor grado, de su estructura característica.

En una perspectiva más teórica, el problema de la estructuración de los documentos es tratado por [André et al.,88]. El punto de vista de la obra se dirige más al problema de la generación de documentos estructurados que al del reconocimiento de los mismos. Aunque algunas de las herramientas revisadas por los autores están hoy ampliamente superadas por modernos sistemas de edición y procesamiento documentales, hay sugerencias interesantes en lo relativo a la concepción básica de los documentos como conjuntos de objetos de complejidad diversa. En el prefacio de la obra encontramos lo siguiente: *“Un documento puede describirse como una colección de objetos de alto nivel formado de otros objetos primitivos”*

Abundando en esta idea, S-Lendex concibe el documento como la unidad superior de análisis automático de la que dependen unidades de información subordinadas: las áreas, en una relación expresable gramaticalmente y representable como un árbol. De ahí la forma gramatical que adoptan las descripciones LENDEX -el lenguaje de descripción incorporado por el sistema- y la expresión final de cada uno de los documentos como un árbol, a semejanza del modelo adoptado por [Ingold,89] para el reconocimiento automático de textos: *“La estructura lógica de un texto es a menudo representada por un árbol... Este modelo es particularmente útil pues permite ordenar y encadenar jerárquicamente los elementos que expresa”*

En este trabajo partimos de un supuesto: los documentos ya se encuentran en soporte magnético, razón por la cual los problemas de captación de la información por un sistema de *scanner* + *OCR* o de exportación desde una base de datos documental no se abordan.

Todo ello forma parte del diseño y el desarrollo de un programa más general, también real, en el que S-Lendex se integra: el proyecto BiblioTECA³, financiado por la Dirección General XIII de la Comunidad Europea y liderado por Verba Logica.

Asimismo suponemos limpios los documentos de los errores típicos OCR: cambios de letras por números, mayúsculas-minúsculas, etc. Precisamente una de las perspectivas de desarrollo futuro de S-Lendex consiste en la extensión del lenguaje de descripción documental en la dirección de contemplar este tipo de casuística.

La metodología y enfoque general adoptados son los propios del ámbito de la programación lógica, con particular protagonismo de las gramáticas lógicas y la metaprogramación.

Un planteamiento interesante que aúna el objeto de estudio -los textos estructurados- y la metodología -gramáticas lógicas- es el desarrollado por [Tazi-Virbel,85], aunque la perspectiva es distinta por centrarse en el desarrollo de un sistema de edición y generación de documentos estructurados, más que en el problema general de su reconocimiento.

Es relevante asimismo destacar el trabajo desarrollado por [Bastos,94], en la medida que desarrolla -con estrategias semejantes- un análisis de títulos en lengua portuguesa tendente a la recuperación automática de información jurídica.

Vale aquí el punto de vista de [Dahl-St.Dizier,85; p.8] al referirse a las meta-gramáticas lógicas: *“La idea de meta-gramáticas basadas en la lógica surge de proporcionar a un usuario no especializado... un mecanismo de análisis*

³BiblioTECA está formado por un equipo de instituciones públicas y privadas de España, Francia e Italia: Matra Caps Systèmes, la Unidad de Coordinación de Bibliotecas del CSIC, la Biblioteca de Nápoles, la Biblioteca del Instituto Cervantes y el equipo antes mencionado: Verba Logica.

gramatical que le permita expresarse en términos de reglas familiares de rescritura en vez de en un lenguaje de programación aunque sea éste de alto nivel”

Éste es, sin duda, uno de los propósitos de S-Lendex, si bien insistiendo más en las ventajas de un lenguaje de descripción cómodo y simple que permita tratar clases diversas de documentos, que en el carácter no especializado del usuario que ha de utilizarlo. Semejante ventaja nos parece igualmente interesante para usuarios avezados en la programación en la medida que extiende la expresividad habitual de Prolog y permite desarrollar programas declarativamente más claros.

Asumimos el reto, así pues, que la misma autora plantea en la siguiente forma, [Dahl-St.Dizier,85; p.3]: *“Tal proliferación de investigaciones⁴ lleva a la cuestión de si es posible construir un formalismo que combine la eficacia con un alto poder expresivo y oculte al usuario todos los detalles automatizables, proporcionándole así una mecanismo de descripción de los aspectos puramente creativos de la escritura de gramáticas”*

S-Lendex pretende ser un paso más en esta dirección.

Organizamos su presentación en cuatro apartados fundamentales:

- Una descripción general de LENDEX
- La especificación del lenguaje de descripción documental
- La exposición del programa que analiza los documentos
- Aplicaciones

⁴La autora se refiere a los diferentes desarrollos de formalismos metagramaticales tales como *Extraposition Grammars, Gapping Grammars, Modifier Structure Grammars*, etc.

Antes de continuar, los agradecimientos.

A mi director de tesis, Jaime Sarabia, la ocasión brindada hace cuatro años de participar en el grupo de investigación Verba Logica, así como su paciencia y confianza en la consecución de este trabajo.

A mis compañeros del Dpto. de Lógica y Filosofía de la Ciencia -con su director, José Fernández-Prida, a la cabeza-, la comprensión y ayuda mostradas en estos dos últimos años.

A Alejandro Santamaría le debo la recopilación de documentos -constante y solícita- para la prueba de S-Lendex.

Agradezco también el apoyo de Carmen Mataix a lo largo de estos años.

A familia y amigos -como dijo el otro- gracias por estar ahí.

Descripción de LENDEX

Introducción

LENDEX es un lenguaje de descripción de documentos concebido como una herramienta de análisis automático. Entendemos de manera provisional documento como la unidad máxima de información analizable por un sistema artificial de procesamiento de la información.

El concepto de documento no responde a una entidad física sino más bien lógica; no queremos significar el tipo de objeto al que habitualmente nos referimos con la palabra documento, sino el punto de partida de nuestro análisis.

Es posible organizar lógicamente un documento físico en dos o más partes que son analizadas separadamente y que, a nuestros efectos, se constituyen como documentos. Pensemos, por ejemplo, en un catálogo de publicaciones: [Discos,88] o [Villancicos,90]. Cabe al menos distinguir en él dos elementos analizables de manera independiente: el contenido propiamente dicho del catálogo y los índices o tablas de contenido. Nada impide que el resultado del análisis de uno de ellos redunde de alguna manera en el tratamiento del otro, con algún tipo de estrategia de cruce de la información. Sin embargo, y a efectos de su análisis automático por un sistema como S-Lendex, cada uno de ellos es considerado en sí mismo como un documento. Documento cuyo análisis puede brindar a su vez la posibilidad de analizar nuevos documentos. Cada una de las entradas o registros del catálogo, así como cada uno de los asientos del índice son también, en términos de LENDEX, documentos.

El de documento es, por consiguiente, un concepto funcional y no absoluto. Lo que desde una perspectiva actúa como documento puede ser, desde otra, tomado como parte.

De esta manera, describimos los documentos en términos de unidades o segmentos menores de información, a los que llamaremos áreas, cada una de las cuales tiene como característica esencial ser reconocida por una cierta gramática.

Los terminales de las gramáticas de áreas son lexemas, combinación de unidades menores, los caracteres. El concepto de lexema puede asimilarse, *grosso modo*, a la noción intuitiva de palabra.

Entre los lexemas distinguimos dos tipos fundamentales: lexemas fútiles, lexemas plenos. Son lexemas fútiles aquéllos constituidos por caracteres habitualmente denominados ortográficos o de puntuación: coma, punto, puntoycoma, dospuntos, guion, etc.

Son lexemas plenos todos los demás. Entre estos cabe distinguir, en virtud del tipo de los caracteres que los constituyen, dos grandes clases: homogéneos y heterogéneos. Son lexemas homogéneos aquéllos que resultan de la combinación de sólo letras o sólo números. Son heterogéneos el resto. Entre los primeros distinguimos además, cinco tipos distintos: lexemas en mayúscula, en minúscula, con mayúscula, numerales y romanos.

La gramática del documento se establece a partir de la gramática de cada una de sus áreas. En uno y otro caso, la gramática actúa como principio de clasificación: tanto los documentos como sus áreas quedan clasificados respectivamente en el conjunto de los documentos y en el de las áreas por la gramática que tienen asociada.

Un documento se caracteriza así por pertenecer a una clase especial de documentos, a saber, aquella generada por la gramática que satisface. Nada impide

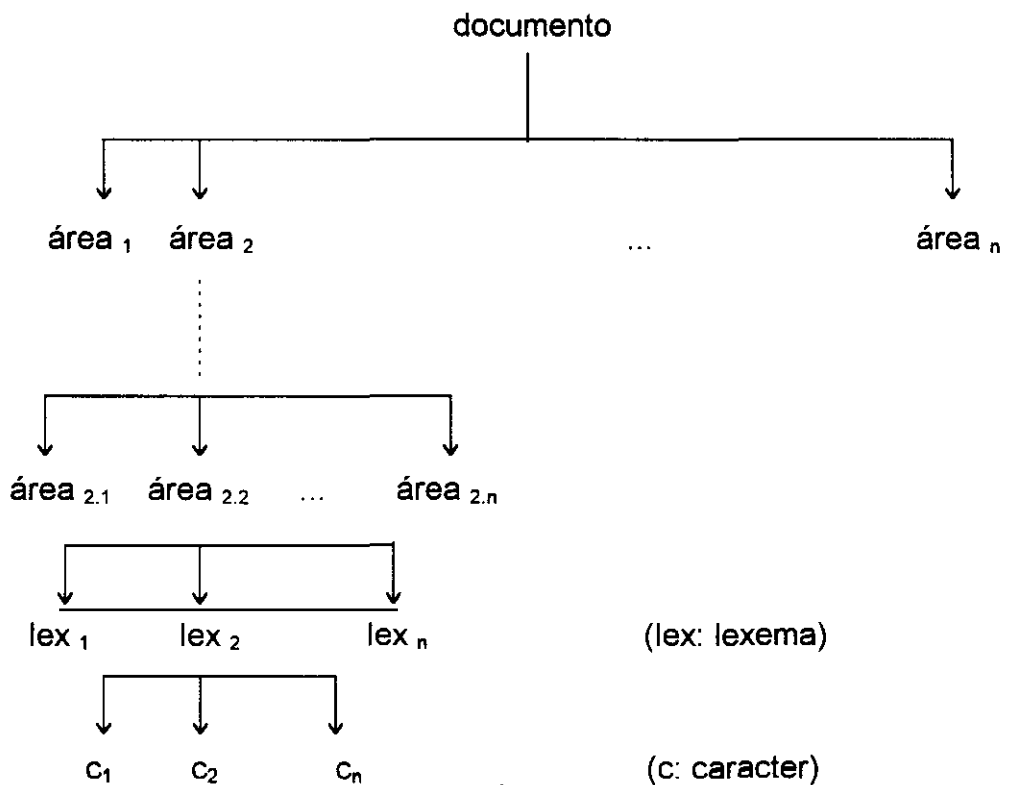
que un documento, descrito de varios modos, pertenezca a familias distintas de documentos.

Todo lo cual puede decirse también de las áreas: cada una de ellas queda emparentada con todas las que su gramática reconoce, bien sean del mismo documento, bien lo sean de documentos distintos, pudiendo pertenecer además a una familia u otra dependiendo de la gramática que se le asocie.

Se pone así de manifiesto algo ya apuntado antes: el carácter funcional de la noción de documento. El documento comparte características comunes con las áreas; viene a ser un área privilegiada o especial, por ser el punto de partida de la descripción.

Si representamos en forma de árbol la jerarquía de entidades planteada, el documento es la raíz o nodo superior del que nacen tantas ramas como áreas, cada una de las cuales puede ser recursivamente analizada en términos de otras áreas que finalmente tienen en los caracteres a las hojas del árbol.

Figura 1: Estructura general de un documento



Tipos de áreas

Vamos a clasificar las áreas atendiendo a un doble criterio. El primero es de tipo práctico: está relacionado con el interés que guía su análisis. El segundo lo es gramatical.

Reconocimiento y recuperación: áreas R, áreas RR

No todas las áreas a las que la gramática del documento se refiere tienen el mismo interés: hay áreas que sólo importa reconocer y otras que además es necesario recuperar. Hablaremos de dos tipos de áreas:

- áreas de sólo reconocimiento; en adelante, áreas R
- áreas de reconocimiento y recuperación; en adelante, áreas RR

Como ya se dijo al principio, LENDEX se concibe como un herramienta de análisis documental automático. El propósito último es, por consiguiente, analizar documentos en términos de unidades progresivamente más simples, las áreas, con vistas a fines posiblemente diversos: otra representación, una recuperación más eficaz, su clasificación automática, la traducción a otro formato de representación, etc. En cualquiera de estos casos, la habilidad de la descripción no se mide sólo por lo que cada una de las gramáticas empleadas sea capaz de reconocer, sino por el valor de vuelta, lo que nosotros llamamos el contenido informativo recuperado.

Desde este punto de vista, áreas R y áreas RR permiten describir los documentos en tanto que secuencias de caracteres o lexemas. Un documento es así el resultado de concatenar, en el orden apropiado, la extensión de todas y cada una de las áreas

-de reconocimiento, y de recuperación y reconocimiento- citadas en la gramática del mismo.

Sólo las segundas, sin embargo, cumplen el cometido originalmente establecido de expresar el documento como un conjunto significativo y ordenado de pares del tipo:

<área, contenido informativo>.

El análisis automático de los documentos debe distinguir adecuadamente la información de lo que no lo es. Por ello, es capital poder expresar en la descripción de los mismos el distinto interés que un área tiene en el resultado final del análisis.

Definición débil, definición estricta: áreas DD, áreas ED

Desde otro punto de vista y atendiendo al tipo de gramática que satisfacen, puede distinguirse dos clases de área:

- aquéllas sólo determinadas en lo que a su final se refiere: son áreas que por carecer de una estructura interna estricta, quedan definidas por algún tipo de marca final: bien sea alguna señal explícita de fin de área, bien sea el comienzo de otra distinta.
- aquéllas con estructura suficiente para ser definidas de manera precisa y unívoca.

Llamaremos área débilmente definida: área DD, a aquélla que caiga bajo la primera clase. Llamaremos área estrictamente definida: área ED, a su contraria. En general, un documento se describe en términos de áreas, que pueden ser áreas DD o áreas ED.

Ciertas áreas DD son analizables de manera flexible en la medida que puede preverse qué subáreas la constituyen, aunque no en qué orden. Por otra parte, es posible asociar a áreas DD y ED condiciones adicionales sólo evaluables una vez determinada su extensión: son condiciones sobre el área en su conjunto y no sobre alguna de sus subsecuencias.

Relación entre unas y otras

En LENDEX, las áreas DD y ED son áreas RR, salvo indicación en contrario. Es en la gramática de las expresiones LENDEX donde se establece el modo de marcar un segmento de texto como de sólo reconocimiento.⁵

Un documento ISBD

Un ejemplo ilustrará bien las clasificaciones hechas. Nos referiremos para ello a la norma *International Standard Book Description*, en adelante ISBD. La ISBD proporciona unas reglas internacionales de catalogación de publicaciones, según las cuales un libro queda descrito en términos de una serie de campos: el encabezamiento, el título, la mención de responsabilidad, la publicación, la descripción física, la serie, etc.

Consideremos un registro bibliográfico -más habitualmente conocido en su versión papel: la ficha de las que hay cientos en una biblioteca y de la que la Figura 2 es un ejemplo-, como un documento en el sentido al que antes hemos aludido: unidad de información analizable en términos de áreas.

⁵Véase la definición de `area_r` en: *La gramática de LENDEX*, en la p.46

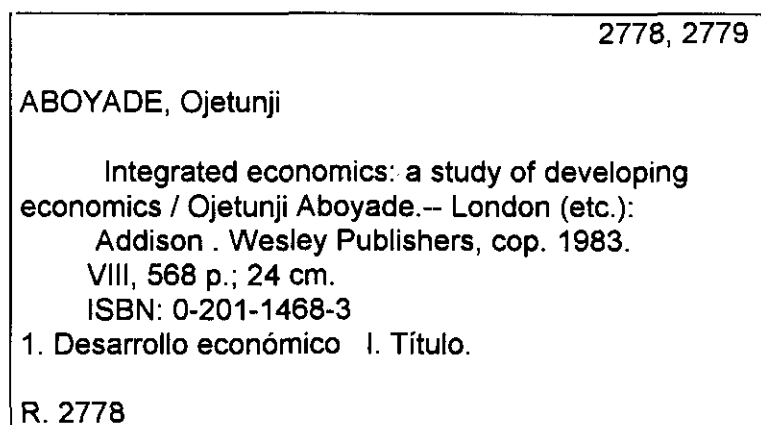
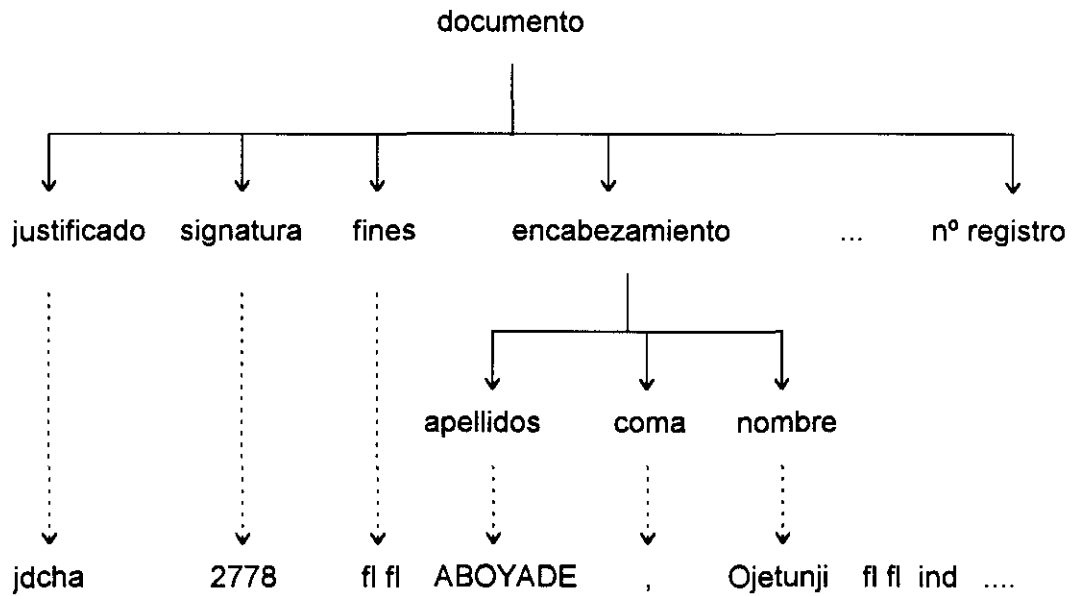


Figura 2: Un documento ISBD

Ilustraremos en primer lugar la distinción establecida entre áreas R y áreas RR. Es útil representar en forma de árbol la información contenida en el registro. Suponemos ya agrupados los caracteres en lexemas con el fin de no complicar en exceso el diagrama. Necesitamos introducir tres lexemas convencionales como son: fin de línea -fl-, justificación a la derecha -jdcha- e indentado -ind-, para poder reflejar adecuadamente la estructura o formato del documento. La primera parte de ésta -suficiente para poner de manifiesto la distinción pretendida- podría representarse así:

Figura 3: Representación en árbol del análisis de un documento en términos de sus áreas R y RR



jdcha: justificación derecha

fl: fin de línea

ind: indentado

Atendiendo al diagrama, vemos cómo el documento se describe en términos de varias áreas, que en primera instancia y por este orden son: justificado, signatura, fines, encabezamiento, ..., hasta llegar a la última que es el n_registro. Desde el punto de vista del análisis del documento todas ellas son relevantes: el documento es el resultado de concatenar los segmentos correspondientes a cada una de las áreas, esto es, su extensión. Sin embargo, con vistas a la expresión de su contenido informativo, podríamos decir que sólo signatura y encabezamiento resultan significativas, declarándolas áreas RR, mientras que justificado y fines serían áreas R.

El resultado final del análisis obviaría, por consiguiente, éstas últimas y expresaría el documento únicamente en términos de las primeras.

De acuerdo con la norma ISBD, el documento se describiría en función de las siguientes áreas RR:

Tabla 1: Primer análisis de un registro ISBD

Áreas	Contenido informativo
signatura	2778, 2779
encabezamiento	ABOYADE, Ojetunji
titulo_responsabilidad	Integrated economics: a study of developing economics / Ojetunji Aboyade
publicacion	London (etc.): Addison . Wesley Publishers, cop. 1983
descripcion_fisica	VIII, 568 p.; 24 cm
isbn	ISBN: 0-201-1468-3
materias_secundarias	1. Desarrollo económico I. Título.
registro	2778

Si por exigencias de los objetivos pretendidos fuera necesario efectuar un análisis más pormenorizado de cada una de las áreas, segmentos que en este nivel forman parte de su contenido informativo, pasarían a formar parte de la clase de áreas R por no ser de interés su recuperación; pensemos, por ejemplo, en los dospuntos que separan título y subtítulo y lugar de publicación de editorial, o la barra que marca en titulo_responsabilidad el comienzo de la mención de autor.

Un análisis más detallado del registro ISBD daría como resultado una descripción del documento en términos de áreas más simples:

Tabla 2: Segundo análisis de un registro ISBD

Áreas	Contenido informativo
signatura	2778, 2779
apellidos	ABOYADE
nombre	Ojetunji
título	Integrated economics
subtítulo	a study of developing economics
responsabilidad	Ojetunji Aboyade
lugar_publicacion	London (etc.)
editorial	Addison . Wesley Publishers
fecha_publicacion	cop. 1983
extension	VIII, 568 p.
dimensiones	24 cm
isbn	ISBN 0-201-1468-3
materias	1. Desarrollo económico
secundarias	I. Título.
registro	2778

Como obviamente puede imaginarse, este proceso no tiene por qué acabar aquí. Depende de los fines perseguidos al analizar este documento llegar a un nivel o a otro en su descripción.

Como igualmente es relativo el modo en que cada una de las áreas se define: el área isbn, por concretar, podría describirse de un modo distinto al aquí planteado: excluyendo de su contenido informativo la mención explícita de ISBN, la secuencia "ISBN: ". Siendo información redundante -ya sabemos que lo contenido en el área es el ISBN-, podríamos considerarla eliminable de la descripción final del documento, que quedaría en lo que a ella respecta así:

Tabla 3: Análisis alternativo del área Isbn

isbn	0-201-1468-3
------	--------------

Para ello, no tendríamos más que declarar en la gramática del documento dicha área como área de sólo reconocimiento, área R, y no área RR.

Otro aspecto en el que el interés determina el modo en que el documento se analiza es el relativo al nexo entre unas áreas y otras. Puede hacerse una descripción plana, del tipo de las propuestas más arriba -en la que el documento se describe como un conjunto ordenado de pares <area, contenido>- u otra en que se conserva la relación de dependencia entre las áreas -supongamos en nuestro ejemplo la existente entre encabezamiento y las de apellidos y nombre- dando lugar así a un análisis con estructura de árbol.

Vamos ahora con la segunda distinción entre áreas: aquella relativa al tipo de gramática asociada. Tomaremos como casos paradigmáticos de las dos clases generales las de titulo_responsabilidad y descripcion_fisica.

De un lado, el área titulo_responsabilidad queda determinada únicamente por una marca explícita de fin de área: .-- (punto, doble guión), ya que no hay una gramática precisa que deba cumplir; prácticamente es texto libre lo que en ella puede aparecer. Su definición depende pues de la aparición de un signo externo: el de fin de área, o del comienzo de otra área distinta, en este caso publicacion. El área titulo_responsabilidad es, por consiguiente, un área débilmente definida, área DD.

Frente a ella, el área descripcion_fisica ofrece características muy distintas. Sí existe una gramática precisa que determina cuándo un segmento de texto cae bajo esa categoría y cuándo no: el conjunto de palabras posibles es mucho más reducido, -romanos, numerales, ciertas abreviaturas-, así como el orden de

aparición: en primer lugar aparece un subsegmento que indica la extensión de la publicación, con indicación de páginas, y a continuación las dimensiones, expresadas en centímetros. La estructura interna del área permite hacer de ella un área estrictamente definida, área ED.

Otros conceptos relacionados

Hemos distinguido, en función de su estructura interna, entre dos tipos de áreas: áreas DD y áreas ED. Para la primera clase, las áreas DD, introducimos ahora otros conceptos relacionados: fin de área y sus tipos, rasgo del área y análisis asociado.

Fin de área: llamamos fin de área a la marca, externa o interna, que determina el final de un área DD. Distinguiremos dos tipos de fin de área: externos e internos. Se dice que un fin de área es **interno** a una cierta área DD cuando forma parte de su extensión. Declararemos un fin de área **externo** cuando suceda lo contrario.

Siguiendo el ejemplo anterior, y suponiendo definida el área publicacion como área DD, podríamos establecer para ella un fin interno: otra área llamada fecha_publicacion o año_publicacion. Ésta constituiría un fin interno al área publicacion porque marca el fin de la misma y debe formar parte de su extensión. Por contra, podríamos definir el subsegmento inicial del área descripcion_fisica como un área denominada extension, que sería, respecto de publicacion, un fin de área externo: determina su final pero pertenece a la extensión del área que la sigue, descripcion_fisica, y no a la propia.

Entre los fines externos distinguimos además los fines homogéneos de los heterogéneos. Se dice que un fin de área es **externo homogéneo** cuando es externo al área aunque de la misma naturaleza: puede formar parte de la extensión de otra área de la misma clase que venga a continuación. Se dice que un fin de área

es **externo heterogéneo** si además de externo hace imposible la aparición de un área de la misma clase que aquella cuyo fin determina.

Volviendo al registro ISBD, supongamos definida el área **signatura** como el resultado de concatenar una o más veces un área llamada **n_signatura**. Esta consiste en un numeral seguido posiblemente de una coma. En este caso, cada uno de los numerales es un fin de área externo homogéneo para el área **n_signatura**: pone fin al área y puede formar parte de otra área **n_signatura**. Un lexema en mayúsculas, indicativo del comienzo de **encabezamiento**, es un fin externo heterogéneo pues marca el fin del área y la imposibilidad de que aparezca a continuación un área de la misma familia.

En general, y para un cierto área DD, cualquiera de las áreas ED que puedan seguirla en la gramática del documento es un fin externo heterogéneo. En la misma medida y como consecuencia directa, un área ED marca el final de la iteración sobre un área DD precedente.

Rasgo del área: llamamos rasgo de un área DD a la propiedad que cumplen todos sus lexemas no fútiles, consistente en ser de alguno de los tipos pertenecientes al conjunto de los rasgos posibles RP:

$$RP = \{en_may, con_may, en_min, numeral, romano\}$$

Se dice que un lexema es del tipo **en_may** si todos sus caracteres son mayúsculas. Un lexema es del tipo **en_min** si todos sus caracteres son minúsculas; **con_may** si comienza con mayúscula y va seguido de minúsculas, **numeral** si está formado sólo por números y **romano** si es un numeral escrito con caracteres romanos.

Tabla 4: Tipos de lexema: ejemplos

Tipos de lexema	Ejemplos del documento anterior ...
en_may	ABOYADE, ISBN, I

en_min	economics, a, study, of, developing, etc, cop, p, cm, económico
con_may	Ojetunji, Integrated, London, Addison, Wesley, Publishers, Desarrollo, Título
numeral	2778, 2779, 1983, 568, 24, 1
romano	VIII, I

Teniendo en cuenta esto, sólo podemos asociar un rasgo a las áreas *signatura* y *registro* -rasgo numeral-, por cuanto todas las demás son combinación de lexemas de distinto tipo. Otra situación sería si definiéramos para cada área una colección de subáreas. En este caso, y por continuar con el ejemplo, podríamos distinguir en el área *encabezamiento* dos subáreas: *apellidos* y *nombre*, la primera de las cuales sí podría asociarse con un rasgo determinado, en concreto *en_may*, porque los lexemas que en ella aparecen -en este caso concreto, sólo uno: ABOYADE- son lexemas en mayúscula.

El distinto tratamiento antes aludido que el área *isbn* pudiera recibir también determinaría la posible atribución de un rasgo global para el área. Si distinguimos en su definición dos partes: *mencion_isbn* y *contenido_isbn*, para quizás despreciar de su contenido informativo la primera -recordemos la distinción entre áreas R y áreas RR-, sí se podría asociar un rasgo característico a cada una de ellas: *en_may* para *mencion_isbn* y *numeral* para el *contenido_isbn*.

Documentos, áreas, lexemas

La teoría informal hasta ahora expuesta presenta el documento como unidad de información susceptible de ser descrita en términos de segmentos menores, llamados áreas. Áreas que se caracterizan fundamentalmente por satisfacer una determinada gramática. Pudiera aceptarse, sin demasiados problemas, que el caso

mínimo de análisis gramatical posible es aquel que define todo el documento como un único área sin estructura: en definitiva, una secuencia de lexemas con una marca -implícita o explícita- de final de documento, es decir, un área DD. Así como admitir que la noción de área EE se equipara en buena medida a la de documento al ser definida en términos de otros subsegmentos o áreas.

En este punto, podemos plantear de manera semiformal el área como una cuádrupla $\langle N, G, S, R \rangle$, donde N es el nombre del área, G es la gramática asociada, S es la secuencia de texto -ocurrencias de lexemas- que satisface G y R es el valor recuperado o contenido informativo del área. La anterior distinción entre áreas R y áreas RR se manifiesta en el cuarto argumento: las áreas R serán aquéllas expresadas en una cuádrupla del tipo $\langle N, G, S, \text{nil} \rangle$, donde nil designa el contenido informativo nulo, la secuencia vacía.

Una familia de áreas queda determinada como el conjunto de cuádruplas:

$$C = \{ \langle N_1, G_1, S_1, R_1 \rangle, \dots, \langle N_n, G_n, S_n, R_n \rangle \}, \text{ cumpliéndose:}$$

$N_1, \dots, N_n = N$, siendo N el nombre del área, y $G_1, \dots, G_n = G$, donde G es la gramática que reconoce las secuencias S_1, \dots, S_n , siendo R_1, \dots, R_n el contenido informativo asociado a cada una de ellas.

Lo que diferencia a un área de otra de la misma familia es, para las áreas RR , el tercer y el cuarto argumentos de la cuádrupla: la secuencia reconocida y el valor recuperado, mientras que para las áreas R coincide también necesariamente el cuarto: es la secuencia vacía o contenido informativo nulo. Nada impide, por otra parte, que dos áreas de la misma familia compartan incluso los mismos lexemas en los argumentos que normalmente las distinguen; lo relevante en este caso es la aparición de los lexemas y no los lexemas mismos.

La diferencia entre las áreas de la misma familia en lo que al tercer argumento se refiere será tanto más acusada cuanto más flexible sea la gramática G asociada a la

clase. Cobra aquí interés la otra distinción establecida: áreas DD, áreas ED. Las primeras serán, desde este punto de vista, más heterogéneas que las segundas, al imponer condiciones menos estrictas sobre una cierta secuencia a analizar. Cuanto más restrictiva sea la gramática G de un área ED, más se parecerán las secuencias S reconocidas por ella, y más homogénea será la clase correspondiente.

Para representar la gramática común de los documentos, vamos a emplear el formalismo conocido como *Extended Backus-Naur form* -en adelante EBNF-, siguiendo la descripción que se hace en [Gibbins, 88, p.73].

La notación EBNF es una de las formas habituales de expresar la sintaxis de lenguajes de programación generados por una gramática libre de contexto.

En este formalismo, los símbolos no-terminales se mencionan rodeados por los caracteres: “<“ y “>“. Los terminales se representan a sí mismos. El símbolo central de las producciones es “::=” y designa “se describe como” o “consiste en”. Por último, el símbolo reservado para la disyunción es “|”. El modelo general de producción en este formalismo es el siguiente:

$$C ::= C_1, \dots, C_n$$

que ha de interpretarse así: una secuencia de texto S cae bajo la categoría C , si puede expresarse en términos de la concatenación de los subsegmentos S_1, \dots, S_n de S , cada uno de los cuales cae respectivamente bajo las categorías C_1, \dots, C_n . C necesariamente es un no-terminal de la gramática, mientras que C_j ($0 < j < n+1$), puede ser un terminal o un no-terminal.

Hasta aquí, lo que constituye el formalismo conocido como *Backus-Naur form*, BNF. La versión extendida que emplearemos aquí, EBNF, incorpora dos pares de símbolos más: {}, [], para designar:

- {}: cero o más iteraciones de la categoría gramatical rodeada por las llaves

- $[]$: cero o una aparición de la categoría gramatical rodeada por los corchetes

Emplearemos, además, un tercer par: $()$, con el fin de desambiguar expresiones con más de un operador.

El propósito de lo que sigue es expresar gramaticalmente el principio general hasta ahora planteado: los documentos se describen en términos de unidades menores, las áreas, constituidas por lexemas. Una definición previa de lexema parece que se hace, pues, necesaria.

La gramática del lexema

La gramática por la cual el concepto de lexema queda definido es una gramática independiente de contexto: $G_{Lex} = \langle SN_{Lex}, ST_{Lex}, P_{Lex}, S_{Lex} \rangle$, donde:

- SN_{Lex} es el conjunto de símbolos no terminales

$$SN_{Lex} = SN1_{Lex} \cup SN2_{Lex} \cup SN3_{Lex}, \text{ donde:}$$

$$SN1_{Lex} = \{ \text{lexema_pleno, lexema_homogeneo, lexema_heterogeneo,} \\ \text{caracter_futil, caracter_no_futil, letra, mayuscula, minuscula,} \\ \text{romano, interrogacion_i, admiracion_i, porciento} \}$$

$$SN2_{Lex} = \{ \text{lexema_futil, lex_en_may, lex_en_min, lex_con_may,} \\ \text{lex_numeral, lex_romano} \}$$

$$SN3_{Lex} = \{ \text{caracter, numeral, punto, coma, puntoycoma, dospuntos,} \\ \text{guion, guion_bajo, barra_i, barra_d, barra, parentesis_i,} \\ \text{parentesis_d, igual, mayor, menor, interrogacion_d,} \}$$

admiracion_d, mas, llave_i, llave_d, corchete_i, corchete_d,
comita, comillas, ampersand, asterisco, circunflejo, dolar }⁶

- ST_{Lex} es el los símbolos terminales

$ST_{Lex} = ST1_{Lex} \cup ST2_{Lex}$, donde:

$ST1_{Lex} = \{ \grave{c}, \acute{a}, \acute{e}, \acute{i}, \acute{o}, \acute{u}, \grave{a}, \grave{e}, \grave{i}, \grave{o}, \grave{u}, \grave{a}, \grave{e}, \grave{i}, \grave{o}, \grave{u}, \acute{A}, \acute{E}, \acute{I}, \acute{O}, \acute{U}, \grave{A}, \grave{E}, \grave{I}, \grave{O}, \grave{U}, \grave{A}, \grave{E}, \grave{I}, \grave{O}, \grave{U}, \grave{c}, \grave{i}, \% \}$

$ST2_{Lex} = \{ a, b, c, d, e, f, g, h, i, j, l, m, n, \tilde{n}, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, G, H, I, J, L, M, N, \tilde{N}, O, P, Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, , , ; , : , - , / , \ , | , (,) , = , > , < , ? , ! , + , [,] , ' , " , \& , * , ^ , \$ \} \cup \{ ' , \{ , \} \}$ ⁷

Los tres últimos caracteres: ' ', '{', '}' son mencionados sólo con vistas a distinguirlos de los correspondientes metalingüísticos.

- S_{Lex} es el axioma o símbolo inicial:

$S_{Lex} = \langle \text{lexema} \rangle$

- P_{Lex} es el conjunto de las siguientes producciones:

$\langle \text{lexema} \rangle ::= \langle \text{lexema_pleno} \rangle \mid \langle \text{lexema_futil} \rangle$

$\langle \text{lexema_pleno} \rangle ::= \langle \text{lexema_homogeneo} \rangle \mid \langle \text{lexema_heterogeneo} \rangle$

⁶La expresión del conjunto SN_{Lex} como la unión de tres conjuntos $SN1_{Lex}$, $SN2_{Lex}$ y $SN3_{Lex}$ obedece al uso que se hará de los dos últimos en definiciones posteriores.

⁷Lo mismo que lo apuntado en la nota anterior cabe decir para ST_{Lex} respecto de $ST1_{Lex}$ y $ST2_{Lex}$.

<lexema_homogeneo>	::=	<lex_en_may> <lex_en_min> <lex_con_may> <lex_numeral> <lex_romano>
<lexema_heterogeneo>	::=	{<letra> <numeral> } (<letra> <numeral> <numeral> <letra>) {<letra> <numeral> }
<lex_en_may>	::=	<mayuscula> {<mayuscula>}
<lex_en_min>	::=	<minuscula> {<minuscula>}
<lex_con_may>	::=	<mayuscula> {<minuscula>}
<lex_numeral>	::=	<numeral> {<numeral>}
<lex_romano>	::=	<romano> {<romano>}
<lexema_futil>	::=	<caracter_futil>
<caracter>	::=	<caracter_futil> <caracter_no_futil>
<caracter_futil>	::=	<punto> <coma> <puntoycoma> <dospuntos> <guion> <guion_bajo> <barra> <barra_d> <barra_i> <parentesis_d> <parentesis_i> <igual> <mayor> <menor> <interrogacion_d> <interrogacion_i> <admiracion_d> <admiracion_i> <mas> <llave_d> <llave_i> <corchete_d> <corchete_i> <comita> <comillas>

		<ampersand> <asterisco> <circunflejo> <dolar> <porciento>
<caracter_no_futil>	::=	<letra > <numeral> <romano> ⁸
<letra>	::=	<mayuscula> <minuscula>
<mayuscula>	::=	A B C Ç D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z Á É Í Ó Ú À È Ì Ò Ù Ä Ê Ï Ö Ü Â Ê Î Ô Û ° ^a
<minuscula>	::=	a b c ç d e f g h i j k m n ñ o p q r s t u v w x y z á é í ó ú à è ì ò ù ä ë ï ö ü â ê î ô û ° ^{a 9}
<numeral>	::=	0 1 2 3 4 5 6 7 8 9
<romano>	::=	I V X L C D M
<punto>	::=	.
<coma>	::=	,
<puntoycoma>	::=	;

⁸Naturalmente, dada la definición de romano, podría obviarse su mención en la definición de caracter_no_futil: todos los caracteres romanos caen en el conjunto de las mayúsculas y, por consiguiente, pertenecerían también al de los no fútiles. Sin embargo, nos importa ser muy explícitos en la definición tanto de caracter_no_futil como de su complementario, caracter_futil.

⁹La consideración de los caracteres: ° y ° como mayúscula y minúscula obedece al hecho de que es el contexto el que determina para ellos uno u otro rasgo.

<dospuntos>	::=	:
<guion>	::=	-
<guion_bajo>	::=	_
<barra_i>	::=	\
<barra_d>	::=	/
<barra>	::=	
<parentesis_i>	::=	(
<parentesis_d>	::=)
<igual>	::=	=
<mayor>	::=	>
<menor>	::=	<
<interrogacion_i>	::=	¿
<interrogacion_d>	::=	?
<admiracion_i>	::=	¡
<admiracion_d>	::=	!
<mas>	::=	+
<llave_i>	::=	{
<llave_d>	::=	}
<corchete_i>	::=	[
<corchete_d>	::=]
<comita>	::=	'
<comillas>	::=	"
<ampersand>	::=	&
<asterisco>	::=	*
<circunflejo>	::=	^
<dolar>	::=	\$
<porciento>	::=	%

La gramática común de los documentos

Una vez establecida la gramática del lexema, podemos abordar la tarea de definir la gramática común de los documentos, que tendrá como terminales las categorías definidas en el apartado anterior: lexema, lexema fútil, lexema en mayúsculas, etc. . Ésta se expresa como una gramática independiente de contexto:

$G_{Docs} = \langle SN_{Docs}, ST_{Docs}, P_{Docs}, S_{Docs} \rangle$, donde:

- SN_{Docs} es el conjunto de símbolos no terminales

$$SN_{Docs} = \{ \text{documento, areas, area, lexemas, fútiles, lxs_con_rasgo, lxs_en_may, lxs_con_may, lxs_en_min, lxs_numeral, lxs_romano} \} \cup SN_{Lex}^{10}$$

- ST_{Docs} es el los símbolos terminales

$$ST_{Docs} = \{ X \mid X \text{ es un lexema de acuerdo con } G_{Lex} \}$$

- S_{Docs} es el axioma o símbolo inicial:

$$S_{Docs} = \langle \text{documento} \rangle$$

- P_{Docs} es el conjunto de las siguientes producciones:

$$\langle \text{documento} \rangle \quad ::= \quad \langle \text{areas} \rangle$$

$$\langle \text{areas} \rangle \quad ::= \quad \langle \text{area} \rangle \{ \langle \text{area} \rangle \}$$

$$\langle \text{area} \rangle \quad ::= \quad \langle \text{lexemas} \rangle \mid \langle \text{lx_con_rasgo} \rangle$$

¹⁰ SN_{Lex} es un subconjunto de los no terminales de G_{Lex} , SN_{Lex} . Véase: *La gramática del lexema*, en p.32

<lexemas>	::=	<lexema> ¹¹ { <lexema> }
<futiles>	::=	<lexema_futil> { <lexema_futil> }
<lx_s_con_rasgo>	::=	<lx_s_en_may> <lx_s_en_min> <lx_s_con_may> <lx_s_numeral> <lx_s_romano>
<lx_s_en_may>	::=	[<futiles>] <lex_en_may> {<lex_en_may> <lexema_futil>}
<lx_s_en_min>	::=	[<futiles>] <lex_en_min> { <lex_en_min> <lexema_futil> }
<lx_s_con_may>	::=	[<futiles>] <lex_con_may> { <lex_con_may> <lexema_futil>}
<lx_s_numeral>	::=	[<futiles>] <lex_numeral> {<lex_numeral> <lexema_futil>}
<lx_s_romano>	::=	[<futiles>] <lex_romano> {<lex_romano> <lexema_futil>}

¹¹El concepto de *lexema*, así como de los relacionados *lexema_futil*, *lex_en_may*, *lex_en_min*, etc, son los definidos en G_{Lex} . Véase otra vez: *La gramática del lexema*, en p.34

Varios comentarios son pertinentes en este punto. La gramática propuesta G_{Docs} expresa en EBNF el principio general planteado desde el principio: el documento es la entidad superior de la que nos vamos a ocupar, describable en términos de segmentos menores llamados áreas. Estos a su vez se descomponen en unidades más simples, los lexemas, que en última instancia se reescriben como secuencias de caracteres.

Hay, por consiguiente, unas nociones de lexema y de carácter bastante convencionales. Aunque la primera se asimila *grosso modo* a la noción de palabra, existen casos en que divergen: en catalán, por ejemplo, las palabras *col.leccio*, *col.lectiu* quedarían descompuestas en tres lexemas. Palabras inglesas como los ordinales *1st*, *2nd*, *3rd*,..., o francesas como *1re*, *7me*,..., aunque reconocidas en tanto que lexemas plenos, serían clasificadas de una manera un tanto gruesa al caer en la categoría de lexemas heterogéneos: no pertenecen a ninguno de los tipos posibles del conjunto RP de los rasgos¹². Esto no supone una limitación excesiva; siempre sería posible, si necesario, contemplarlos en una segunda instancia con un tratamiento especial.

Lo mismo cabe decir de la definición de lexema romano. La mera combinación de caracteres romanos sabemos que no proporciona un número romano, en el sentido en que habitualmente se habla. Falta, naturalmente, la gramática correspondiente. Cabría decir, utilizando de manera informal los conceptos de corrección y completud, que nuestra caracterización de los lexemas romanos es completa, en el sentido de que cualquier romano es reconocido como tal, pero incorrecta, en la medida que reconoce más de lo que debe. Debido a ello, ciertos lexemas -tales como MIL, CID, VIL y VID- pueden resultar en ocasiones de difícil clasificación,

¹²Véase en la p. 28: $RP = \{en_may, con_may, en_min, numeral, romano\}$

siendo necesario tener en cuenta el contexto para desambiguarlos o desarrollar estrategias especiales para su tratamiento.

La gramática de LENDEX

LENDEX es un lenguaje de descripción de documentos que, como ya se ha indicado anteriormente, se concibe como una herramienta de análisis automático. Nace, por consiguiente, para ser implementado y probado en situaciones reales. Por el momento, propondremos -en un plano estrictamente formal- la gramática del mismo. Más tarde, acometeremos la tarea de darle forma de programa.

Sin perder de vista esto, y teniendo en cuenta el lenguaje de programación a utilizar, Prolog, aparecerán definidos en la especificación de LENDEX conceptos propios del ámbito de la programación lógica: cláusula, término, literal, átomo, etc.

G_{Lendex}

Vamos de nuevo a utilizar la representación EBNF para expresar su gramática. Ésta consiste en la cuádrupla $G_{Lendex} = \langle SN_{Lendex}, ST_{Lendex}, P_{Lendex}, S_{Lendex} \rangle$, donde:

- SN_{Lendex} es el conjunto de símbolos no terminales

$SN_{Lendex} = SN1_{Lendex} \cup SN2_{Lendex} \cup SN3_{Lendex}$, donde:

$SN1_{Lendex} = \{ documento_lendex, produccion_principal, nec_documento, area_lendex, area_dd, area_ed, nec_ed, fin_dd, analisis_dd, nec_dd, condicion, lista_descriptores, descriptores, un_descriptor, descriptor_area, nombre_area_lendex, nombre_area_dd, nombre_area_ed, nombre_reservado, termino_cuantificado, lexema_descrito, lexema_tipo, secuencia, preterminal, area_dcg, operador_iterativo, operador_det, operador_semi, operador_indet,$

cuantor_numerico, cuantor_det, cuantor_semi, cuantor_indet,
 cero_o_uno_det, cero_o_mas_det, uno_o_mas_det, cero_o_uno_semi,
 cero_o_mas_semi, uno_o_mas_semi, cero_o_uno_indet,
 cero_o_mas_indet, uno_o_mas_indet, desde_det, hasta_det,
 exactamente_det, entre_det, desde_semi, hasta_semi, entre_semi,
 desde_indet, hasta_indet, exactamente_indet, entre_indet, area_r,
 rasgo, expresion_lambda, letra_l, mayus_l, minus_l, signo_puntuacion }

$SN2_{Lendex} = \{ \text{procedimiento, clausula_prolog, hecho, regla, objetivo,}$
 $\text{nec, literal_prolog, literal_aridad_0, literal_aridad_mayor, expresion,}$
 $\text{termino, constante, variable, atomo, string, entero, gramatica_dcg,}$
 $\text{produccion, nec_dcg, comentario, corte } \}$

$SN3_{Lendex} = SN3_{Lex}$ ¹³

- ST_{Lendex} es el los símbolos terminales

$ST_{Lendex} = ST2_{Lex}$ ¹⁴

- S_{Lendex} es el axioma o símbolo inicial:

$S_{Lendex} = \langle \text{documento_lendex} \rangle$

- P_{Lendex} es el conjunto formado por las producciones correspondientes a $SN1_{Lendex}$ y $SN2_{Lendex}$, organizadas en los conjuntos $P1_{Lendex}$ y $P2_{Lendex}$ respectivamente:

$P_{Lendex} = P1_{Lendex} \cup P2_{Lendex}$

¹³El conjunto $SN3_{Lendex}$ es el subconjunto $SN3_{Lex}$ de SN_{Lex} . Véase: *La gramática del lexema*, en la p.32

¹⁴Como para los símbolos no terminales, el conjunto de los terminales ST_{Lendex} coincide con el subconjunto $ST2_{Lex}$ de ST_{Lex} . Véase de nuevo: *La gramática del lexema*, en la p.32

$P1_{\text{Lendex}}$ es el conjunto que sigue:

<code><documento_lendex></code>	<code>::= <produccion_principal> <areas_lendex> <condiciones> {<procedimiento> <gramatica_dcg> <comentario>}</code>
<code><areas_lendex></code>	<code>::= (<area_dd> {<area_dd>} <fines> < analisis> {<area_ed>}) ({<area_dd>} <fines> < analisis> <area_ed> {<area_ed>})</code>
<code><fines></code>	<code>::= { <fin_dd> }</code>
<code><condiciones></code>	<code>::= { <condicion> }</code>
<code>< analisis></code>	<code>::= { < analisis_dd> }</code>
<code><produccion_principal></code>	<code>::= documento <nec_documento> <lista_descriptores></code>
<code><nec_documento></code>	<code>::= <igual> <igual> <mayor></code>
<code><lista_descriptores></code>	<code>::= <corchete_i> <descriptores> <corchete_d> <punto></code>
<code><descriptores></code>	<code>::= <un_descriptor> {<coma> <un_descriptor>}</code>
<code><un_descriptor></code>	<code>::= <descriptor_area> <conjuncion_descriptores> <disyuncion_descriptores> <negacion_descriptor></code>

<conjuncion_descriptores>	::= <descriptor_area> <coma> <descriptor_area>
<disyuncion_descriptores>	::= <descriptor_area> <puntoycoma> <descriptor_area>
<negacion_descriptor>	::= not <descriptor_area>
<descriptor_area>	::= <nombre_area_lendex> <termino_cuantificado> <lexema_descrito> <secuencia> <preterminal> <area_r> <area_dcg>
<nombre_area_lendex>	::= <nombre_area_dd> <nombre_area_ed>
<nombre_area_dd>	::= <atomo>
<nombre_area_ed>	::= <atomo>
<nombre_reservado>	::= ign ind inc
<termino_cuantificado>	(<operador_iterativo> <descriptores>) (<cuantor_numerico> <parentesis_j> <entero> <coma> <descriptores> <parentesis_d>
<cuantificador>	::= <operador_iterativo> <cuantor_numerico>
<operador_iterativo>	::= <operador_det> <operador_semi>

	<operador_indet>
<operador_det>	::= <ceros_o_uno_det> <ceros_o_mas_det> <uno_o_mas_det>
<operador_semi>	::= <ceros_o_uno_semi> <ceros_o_mas_semi> <uno_o_mas_semi>
<operador_indet>	::= <ceros_o_uno_indet> <ceros_o_mas_indet> <uno_o_mas_indet>
<cuantor_numerico>	::= <cuantor_det> <cuantor_semi> <cuantor_indet>
<cuantor_det>	::= <desde_det> <hasta_det> <exactamente_det> <entre_det>
<cuantor_semi>	::= <desde_semi> <hasta_semi> <entre_semi>
<cuantor_indet>	::= <desde_indet> <hasta_indet> exactamente_indet <entre_indet>
<ceros_o_uno_det>	::= <mas><barra_d>
<ceros_o_mas_det>	::= <asterisco>
<uno_o_mas_det>	::= <ampersand><barra_d>
<ceros_o_uno_semi>	::= <mas><guion>

<ceros_o_mas_semi>	::= <asterisco><guion>
<uno_o_mas_semi>	::= <ampersand><guion>
<ceros_o_uno_indet>	::= <mas><barra_i>
<ceros_o_mas_indet>	::= <asterisco><barra_i>
<uno_o_mas_indet>	::= <ampersand><barra_i>
<desde_det>	::= desde_d
<hasta_det>	::= hasta_d
<exactamente_det>	::= ex_d
<entre_det>	::= entre_d
<desde_semi>	::= desde_s
<hasta_semi>	::= hasta_s
<entre_semi>	::= entre_s
<desde_indet>	::= desde_i
<hasta_indet>	::= hasta_i
<exactamente_indet>	::= ex_i
<entre_indet>	::= entre_i
<area_r>	::= <interrogacion_d> <descriptores>
<lexema_descrito>	::= <lexema_tipo> (<lexema_tipo> <circunflejo> <expresion>)
<lexema_tipo>	::= lexema lex_en_may lex_con_may lex_en_min numeral romano
<secuencia>	::= <string>
<preterminal>	::= sp espacio fl spfl espaciofl futil

| fsp | futil_sp | punto | punto_sp |
 coma | coma_sp | puntoycoma |
 puntoycoma_sp | dospuntos |
 dospuntos_sp | barra | barra_sp |
 barra_d | barra_dsp | barra_i |
 barra_isp | parentesis_d |
 parentesis_dsp | parentesis_i |
 parentesis_isp | igual | igual_sp |
 interr_d | interr_dsp | interr_i |
 interr_isp | admir_d | admir_dsp |
 admir_i | admir_isp | mas | mas_sp |
 llave_d | llave_dsp | llave_i |
 llave_isp | corchete_d> |
 corchete_dsp | corchete_i |
 corchete_isp | comita | comita_sp |
 comillas | comillas_sp | ampersand |
 ampersand_sp

<area_dd> ::= area_dd <parentesis_j>
 <nombre_area_dd> <coma>
 <rasgo> <parentesis_d> <punto>

<rasgo> ::= en_may | en_min | con_may |
 numeral | romano | true

<fin_dd> ::= fin <parentesis_j>
 (<nombre_area_dd> | <variable>)
 <coma> <descriptores> <coma>
 <tipo_fin> <parentesis_d><punto>

<tipo_fin> ::= interno | externo_hetero |
 externo_homo

< analisis_dd >	::= < nombre_area_dd > < nec_dd > < lista_descriptores >
< nec_dd >	::= < guion > < mayor >
< area_ed >	::= < nombre_area_ed > < nec_ed > < descriptores > < punto >
< nec_ed >	::= < igual > < mayor >
< condicion >	::= condicion < parentesis_i > < nombre_area_lendex > < coma > < expresion_lambda > < parentesis_d > < punto >
< expresion_lambda >	::= < expresion > < circunflejo > < expresion >
< area_dcg >	::= < literal_prolog >
< letra_l >	::= < mayus_l > < minus_l >
< mayus_l > ¹⁵	::= A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z
< minus_l > ¹⁶	::= a b c d e f g h i j k l m

¹⁵Puede observarse que la extensión de mayus_l -por mayúscula en LENDEX- es un subconjunto de la extensión de mayuscula, de G_{lex}. Véase: *La gramática del lexema*, en la p.35

n | ñ | o | p | q | r | s | t | u | v | w | x | y
| z

<signo_puntuacion> ::= <punto> | <coma> |
<puntoycoma> | <dospuntos> |
<guion> | <guion_bajo> | <barra> |
<barra_i> | <barra_d> |
<parentesis_d> | <parentesis_i> |
<igual> | <mayor> | <menor> |
<interrogacion_d> | <corte> | <mas>
| <llave_d> | <llave_i> |
<corchete_d> | <corchete_i> |
<comillas> | <ampersand> |
<asterisco> | <circunflejo> | <dolar>

17

Lo que sigue es el conjunto de reglas correspondiente a la gramática de Prolog. En la medida que una gramática Lendex puede contener -además de lo específicamente propio del lenguaje- cualquier predicado Prolog, nos parece importante precisar cuál es esa gramática. Nos inspiramos para ello en [Abramson-Dahl,89;p.61].

Como se podrá comprobar, las siguientes reglas definen el subconjunto de no terminales $SN2_{Lendex}$ de SN_{Lendex} y constituyen $P2_{Lendex}$:

¹⁶Sirve para minus_! respecto de minuscua de G_{lendex} , lo mismo que lo indicado para mayus_! y mayuscula.

¹⁷La definición de los no terminales en función de los que signo_puntuacion se define se encuentra en G_{Lex} . Puede observarse que constituyen un subconjunto de la extensión de caracter_futil. Véase otra vez: *La gramática del lexema*, en la p.34

<procedimiento>	::= <clausula_prolog> { <clausula_prolog> }
<clausula_prolog>	::= <hecho> <regla> <objetivo>
<hecho>	::= <literal_prolog> <punto>
<regla>	::= <literal_prolog> <nec> <expresion> <punto>
<objetivo>	::= <nec> <expresion> <punto>
<nec>	::= <dospuntos><guion>
<literal_prolog>	::= <literal_aridad_0> <literal_aridad_mayor>
<literal_aridad_0>	::= <atomo>
<literal_aridad_mayor>	::= <atomo> <parentesis_j> <termino> {<coma> <termino>} <parentesis_d>
<expresion>	::= <termino> <expresion_negativa> <expresion_conjuntiva> <expresion_disyuntiva>
<expresion_negativa>	::= not <expresion>
<expresion_conjuntiva>	::= <expresion> <coma> <expresion>
<expresion_disyuntiva>	::= <expresion> <puntoycoma> <expresion>

<termino>	::= <constante> <variable> <literal_aridad_mayor>
<constante>	::= <atomo> <entero> <string>
<atomo>	::= <minus_ > { <letra_ > <numeral> <guion_bajo> }
<entero>	::= <numeral> { <numeral> }
<string>	::= <dolar> {<caracter> ¹⁸ } <dolar>
<variable>	::= (<mayus_ > <guion_bajo>) { <letra_ > <numeral> <guion_bajo> }
<lista> ¹⁹	::= (<corchete_i> <corchete_d>) (<corchete_i> <termino> {<coma> <termino>} <corchete_d>)
<gramatica_dcg>	::= <produccion_dcg> {<produccion_dcg>}
<produccion_dcg>	::= <area_dcg> <nec_dcg> <expresion>

¹⁸El concepto de caracter es el definido en G_{Lex} . : Véase: *La gramática del lexema*, en la p.34

¹⁹La lista en el modo aquí indicado es una variante sintáctica del literal formado por el functor '{' y dos argumentos de los cuales el segundo es una lista. Cae, por consiguiente, en la clase *literal_aridad_mayor*. Si hacemos explícita su definición es por el uso que haremos de ella en la especificación de los procedimientos que presentaremos más adelante

<nec_dcg> ::= <guion><guion><mayor>

<comentario> ::= <inicio_comentario> {<caracter>²⁰}
<fin_comentario>

<inicio_comentario> ::= <barra_d> <asterisco>

<fin_comentario> ::= <asterisco> <barra_d>

<corte> ::= <admiracion_d>

²⁰Nuevamente, el concepto de caracter es el definido en G_{Lex} .

Comentarios a G_{Lendex}

Planteada la gramática de LENDEX, varios son los comentarios obligados.

El primero es el relativo a la distinción entre áreas R y áreas RR. Como ya se expuso, un área consiste en una cuádrupla del tipo $\langle N, G, S, R \rangle$. Las áreas de sólo reconocimiento se identifican -dentro de una misma familia de áreas- por el cuarto argumento: para ellas es el contenido informativo nulo. Dado que una gramática de área puede servir para reconocer secuencias de entrada iguales con contenidos informativos distintos -en un punto del documento interesa recuperar un cierto valor, mientras que, en otro, ese mismo contenido se desprecia-, parecía necesario incorporar en la gramática de LENDEX la posibilidad de que un mismo descriptor de área -o una secuencia de descriptores- fuera calificado de sólo reconocimiento o de reconocimiento y recuperación según el punto en que fuera empleado en la descripción general del documento. De ahí la definición del no terminal `area_r`, p.46. La atribución de uno u otro carácter a un área: R o RR, no es conocimiento que se asocia a su gramática, sino a la instancia concreta de lexemas que constituyen su extensión.

Dos comentarios son precisos en relación con una categoría: la de `termino_cuantificado`, p.44. Uno de ellos relacionado con el operador iterativo `cero_o_mas_det`, p.45, otro relativo a un cuantificador numérico no definido: `exactamente_semi`.

Siendo coherentes con la definición de otros operadores iterativos deterministas, `cero_o_mas_det` debería definirse así:

$$\langle \text{cero_o_mas_det} \rangle \quad ::= \quad \langle \text{asterisco} \rangle \langle \text{barra_d} \rangle$$

Su particular definición excluyendo la barra_d final²¹ obedece al uso que en la categoría Prolog fin_comentario se hace de la misma combinación de signos. La necesidad de contemplar en G_{Lendex} la posibilidad de incluir comentarios como en cualquier programa Prolog hace que evitemos el posible conflicto que pudiera surgir al emplear el operador antes definido y definamos cero_o_mas_det sin la barra_d final.

El segundo comentario de los relacionados con términos cuantificados tiene que ver con una ausencia: la del cuantificador numérico exactamente_semi. Así como de otros cuantificadores existen tres versiones: determinista, semideterminista e indeterminista, del cuantificador exactamente hay sólo dos: determinista e indeterminista²². La razón es estrictamente práctica: como más adelante veremos²³, el cuantificador exactamente_semi sería equivalente a la versión determinista del mismo, lo que hace que carezca de sentido contemplarlo en la especificación formal.

El uso en G_{Lendex} de la definición de carácter de G_{Lex} también merece una explicación. La razón por la que no se usa el alfabeto propio es la siguiente: la gramática de LENDEX, como se ha podido observar, se construye sobre un subconjunto del conjunto de los terminales de $G_{\text{Lex}}, T_{\text{Lex}}$: las expresiones de LENDEX no requieren un juego más amplio de caracteres. Sin embargo, y en la medida en que el lenguaje en última instancia va a referirse a documentos cuya gramática -como ya se vio- contempla un conjunto extenso de terminales, era preciso recoger esa mayor amplitud justamente en la categoría gramatical,

²¹Véase la p.45

²²Véase la p.46 donde se definen.

²³En: *Procedimientos gramaticales para los términos cuantificados*, p.185

secuencia, p.46 , que exige identidad con el texto de entrada. Será más adelante, al atribuir significado a cada una de las categorías definidas, cuando podremos precisar esto suficientemente²⁴.

La definición de *rasgo*, p.47 , contempla todos los elementos del conjunto RP además del rasgo denominado *true*. La atribución de éste a un área representa la ausencia de rasgo alguno. Por otra parte, *lexema_tipo*, p.46 , contiene los cinco tipos correspondientes a los rasgos de RP más el genérico *lexema*, que engloba a todos.

La categoría *nombre_reservado*, p.44 , obedece a la necesidad de restringir la posibilidad de nombrar las áreas de una GDU con tres identificadores característicos de S-Lendex: *ign*, *ind*, *inc*, relativos a los distintos tipos posibles de incidencia que pueden ocurrir en el análisis de una clase de documentos: áreas desconocidas o fallidas, secuencias indeterminadas, áreas que no satisfacen la condición asociada en la GDU.

Un comentario final tiene que ver con el modo de disponer las expresiones LENDEX en una GDU. Conviene tener presente a este respecto el carácter Prolog de las gramáticas escritas de acuerdo con G_{Lendex} ; esto es, la base de conocimiento constituida por el conjunto de hechos y reglas LENDEX. Desde este punto de vista, una GDU debe presentar un aspecto ordenado: los hechos y reglas de la misma clase deben estar agrupados. De ahí la definición de *documento_lendex* y las subordinadas *areas_lendex*, *finis*, *analisis* y *condiciones*. En rigor, la primera producción de G_{Lendex} es un caso concreto de los posibles. También cabría esta otra:

`<documento_lendex> ::= <produccion_principal>`

²⁴El significado de *secuencia* aparece explicitado en la p.133


```
<condiciones>
<areas_lendex>
{ <procedimiento> |
<gramatica_dcg> | <comentario>}
```

o cualquiera que permutara los elementos que aparecen en la descripción. Si hemos dado prioridad a la de arriba es por hacer más comprensible la gramática general y no complicar excesivamente G_{Lendex} en un aspecto secundario.

En este punto parece relevante presentar un ejemplo de gramática LENDEX que ilustre la definición formal.

Un ejemplo de gramática LENDEX

Lo que sigue es, de manera simplificada, un ejemplo de gramática LENDEX. Esta aparece sin comentarios en el Anexo 1. Se trata de una gramática que describe una subclase de documentos en formato ISBD. El registro antes comentado valdría como un miembro de esa clase. En la gramática vamos a distinguir las expresiones específicas del lenguaje. No aparecerán por sí solas expresiones Prolog de carácter general, por cuanto el objetivo pretendido es aclarar en lo posible la definición formal antes planteada.

Si analizamos en detalle la primera producción de G_{Lendex} , se pone de manifiesto que una descripción de documento LENDEX consta necesariamente de:

- una expresión del tipo `produccion_principal`
- otra del tipo `area_dd` o `area_ed`

y opcionalmente, una o varias apariciones de las categorías siguientes:

- area_dd
- area_ed
- fin_dd
- analisis_dd
- condicion
- procedimiento
- gramatica_dcg
- comentario

Las más características del lenguaje LENDEX son las cinco primeras: `area_dd`, `area_ed`, `fin_dd`, `analisis_dd` y `condicion`. En nuestro ejemplo de gramática aparecerán, por consiguiente, sólo ellas. No nos parece relevante prolongarlo más allá de lo necesario con la adición de otras cláusulas Prolog o producciones DCG *standard*.

Si importa notar -como ya se apuntó antes- que todas ellas adoptan la forma de hechos o reglas propios de una base de conocimiento Prolog. Lo que los hace característicos en LENDEX es la semántica asociada que, provisionalmente, sólo presentaremos de manera informal e intuitiva.

Primera producción de una gramática LENDEX: descripción del documento o clase de documentos a tratar

El axioma de la gramática es `documento`. En este caso,

```
documento ==> [* espaciofl, sigs, enc, tit, +/- ed, pub, df, serie, isbn,  
&\ secundaria, regs].
```

el significado pretendido de la producción sería éste: un documento de la clase descrita consiste en un primer segmento de cero o más espacios y fines de línea; a éstos le siguen una signatura, un encabezamiento y un título. A continuación, puede aparecer opcionalmente un área de edición, seguida de la publicación, la descripción física, la serie y el isbn. Finalmente, hay una o más entradas secundarias seguidas de los números de registro.²⁵

Aserción de las áreas DD del documento con indicación del rasgo asociado

```
area_dd(tit, true).  
area_dd(enc, true).  
area_dd(ed, true).  
area_dd(pub, true).  
area_dd(ser, true).  
area_dd(sec, true).
```

La primera se interpretaría así: una de las áreas DD del documento tiene como nombre tit y ningún rasgo especial asociado. Suponiendo que alguna de ellas fuera una secuencia de lexemas con rasgo en_may -secuencia formada por al menos un lexema en mayúsculas y fútiles- pensemos en el encabezamiento, habríamos de expresarlo así:

```
area_dd(enc, en_may).
```

²⁵Obviamos en el ejemplo toda alusión a la distinción entre operadores iterativos deterministas, semideterministas e indeterministas. Lo mismo vale para los cuantores numéricos. Será más adelante cuando esta distinción se trate: *Procedimientos gramaticales para los términos cuantificados*, p.171

Aserción de los fines correspondientes a cada una de las áreas DD, con indicación del tipo de fin: interno, externo_homo o externo_hetero²⁶

La interpretación del primero de ellos, a modo de ejemplo,

fin(ser, parentesis_dsp, externo_hetero).

sería ésta: el área DD ser tiene como fin externo heterogéneo un paréntesis derecho posiblemente rodeado de espacios. Ello significa que la aparición del paréntesis en cuestión determina el final del área así como el de una posible iteración sobre ella. En cambio en el caso que sigue:

fin(enc, fl, interno).

se hace del fin de línea un fin interno al área enc, pretendiendo así significar que forma parte de su extensión.

fin(tit, df, externo_hetero).

Merece destacarse en esa ocasión el hecho de que lo que constituye el fin del área tit no es uno de los preterminales definidos en G_{Lendex} como en las dos anteriores aserciones -parentesis_dsp, fl-, sino otra de las áreas -en concreto la denominada df- que esta gramática introduce más abajo: *Producciones correspondientes a las áreas ED*. Así quedan representados los dos tipos posibles de fin de área DD: una marca explícita de fin de área -casos primero y segundo-, y el comienzo de otra área distinta: último ejemplo.

²⁶Véase: *Otros conceptos relacionados*, en la p.27

Aquí aparece un término novedoso:

fin(tit, desde_d(3, punto), interno).

el cuantor numérico `desde_d` aplicado sobre el preterminal `punto`. Ha de interpretarse como sigue: el área denominado `tit` termina con la aparición de una secuencia de puntos en número igual o mayor que tres. Esta secuencia, por ser un fin interno al área, pertenece a su extensión.

fin(pub, df, externo_hetero).
fin(pub, (numeral^entre(numeral,1900,1995), * futil), interno).

Lo más destacable es en esta ocasión la aparición del no terminal llamado *lexema descrito*: `numeral^entre(numeral,1900,1995)`. El significado pretendido es el que sigue: el área `DD pub` termina con la aparición de un numeral entre 1900 y 1995, seguido opcionalmente de fútiles. Este fin, por ser interno, pertenece a la extensión de la publicación.

La novedad ahora viene dada por la atribución al fin del carácter `externo_homo`:

fin(sec, romano^member(romano, [\$I\$, \$II\$, \$III\$]), externo_homo).

El área `SEC` tiene como fin externo homogéneo un romano que pertenezca a la lista formada por: `I, II, III`. Ello significa que la aparición de uno de los números romanos citados delimita el área `SEC`, pero no supone el fin de una iteración sobre ella. La razón es que cada uno de los romanos marca el comienzo de otra secundaria; el fin es externo pues el romano forma parte del área que viene a continuación, pero homogéneo en la medida que ésta puede ser otra área de la misma familia: una secundaria.

fin(sec, regs, externo_hetero).

Es importante notar lo peculiar de los dos últimos fines:

`fin(X, isbn, externo_hetero).`
`fin(X, fa, interno).`

Utilizando la expresividad de Prolog significamos -por medio del uso de una variable, `X`, en el primero de sus argumentos- que las categorías denominadas `isbn` y `fa` son fines de cualesquiera áreas DD asertadas en la primera sección.

Aserción de las condiciones asociadas a áreas DD y ED, con indicación de su tipo: fuerte o débil

`condicion(enc, Z^(algun Z es Q^lex(Q,en_may)), fuerte).`

La interpretación de la primera de las condiciones que nuestra gramática establece sería ésta: el área `enc` tiene como condición asociada la existencia en su extensión de un lexema en mayúsculas. Esta condición es fuerte: ello significa -dado que `enc` es una área DD- que la aparición de un marcador de fin para el área no es suficiente para darla por concluida: es preciso que lo reconocido hasta entonces como extensión del área cumpla la condición en cuestión: contenga al menos un lexema en mayúsculas. Si esto no sucede, se continúa evaluando el área hasta conseguirlo. Caso de no ser posible, el área `enc` queda desconocida.

`condicion(ed, Z^(algun Z es Q^(Q = $éd$), length(Z,N), N<15), fuerte).`

Importa señalar que la anterior es un ejemplo de condición compleja. Ello es posible porque la definición de `condicion`²⁷ contempla en su segundo argumento la

²⁷Véase: *La gramática de LENDEX*, en la p.48

aparición de una expresión lambda formada por dos no terminales del tipo `expresion`. En este caso concreto, la condición es doble: se exige que el área llamada `ed` contenga el lexema “ed” en su extensión -primer literal- y, adicionalmente, que la longitud global del área no exceda los quince lexemas - literales segundo y tercero-.

Producciones correspondientes a las áreas ED

```
serie =>  parenthesis_isp, ser, parenthesis_dsp.
```

La interpretación de la primera, por ejemplo, sería ésta: el área estrictamente definida `serie` consiste en un paréntesis izquierdo posiblemente rodeado de espacios, un área llamada `ser`, y un paréntesis derecho con espacios opcionales alrededor.

```
isbn    =>  * espaciofl,  
           ? $ISBN$,  
           +/ dospuntos_sp,  
           &/ (numeral, +/ guion_sp).
```

En la segunda producción, encontramos nuevos términos LENDEX. Importa destacar dos: un área de sólo reconocimiento, `area_r`: `? $ISBN$`, y un operador iterativo, `uno_o_mas`: `&/ (numeral, +/ guion_sp)`, aplicado sobre una combinación de literales. Se leería de esta manera: el área ED llamada `isbn` se reescribe como cero o más espacios o fines de línea, la secuencia de sólo recuperación “ISBN”, cero o una aparición del signo de dospuntos rodeado posiblemente de espacios, y uno o más numerales seguidos quizás de un guión con espacios alrededor.

```
sigs    =>  &/ ( +/ (lexema^(lexema= $L$ ; lexema=$D$)),  
              * futil,
```

numeral,
* espaciofl).

En este caso, puede destacarse de nuevo la aparición del no terminal lexema descrito²⁸, siendo esta vez la descripción una disyunción de literales: (lexema=\$L\$; lexema = \$D\$). Cabe destacar también un encadenamiento de operadores iterativos: &/ +/. La lectura posible de la producción sería la siguiente: el área **sig**s consta de una o más apariciones de las secuencias consistentes en el lexema “L” o el lexema “D”, seguido de cero o más fútiles, un numeral y espacios y fines de línea opcionales.

```
regs    =>    $R$, +/- punto_sp, &/ (numeral, * espaciofl).
```

En esta producción es relevante destacar la aparición de una categoría LENDEX también presente en la producción correspondiente al área **isbn**: es la que hemos llamado **secuencia**. Esta consiste en la identificación de una cierta cadena de texto en la entrada a analizar. En este caso, se habría de interpretar así: el área llamada **regs** empieza por la secuencia “R” seguida opcionalmente de un punto rodeado de espacios, y termina con la aparición de uno o más numerales seguidos quizás de espacios y fines de línea.

Las que siguen son producciones que no ofrecen novedad relevante. Aparecen para completar la gramática presentada.

```
fa      =>    desde_d(2, fl) ;  
          (guion, guion) ;  
          (punto_sp, &/ guion_sp) ;  
          (punto_sp, fl, * espacio).  
  
df      =>    +/- romano, +/- coma_sp, * futil,
```

²⁸Para recordar su definición, véase: *La gramática de LENDEX*, en la p.46

pag, +/- puntoycoma_sp, dim, * espaciofl.

dim => * espacio, numeral^entre(numeral,10,40),
+/- coma_sp, \$\$\$, * espacio, cm.

pag => numeral^mayor(numeral,21), espaciofl, pag.

cm => \$cm\$, +/-punto.

pg => \$p\$, +/-punto.

El programa

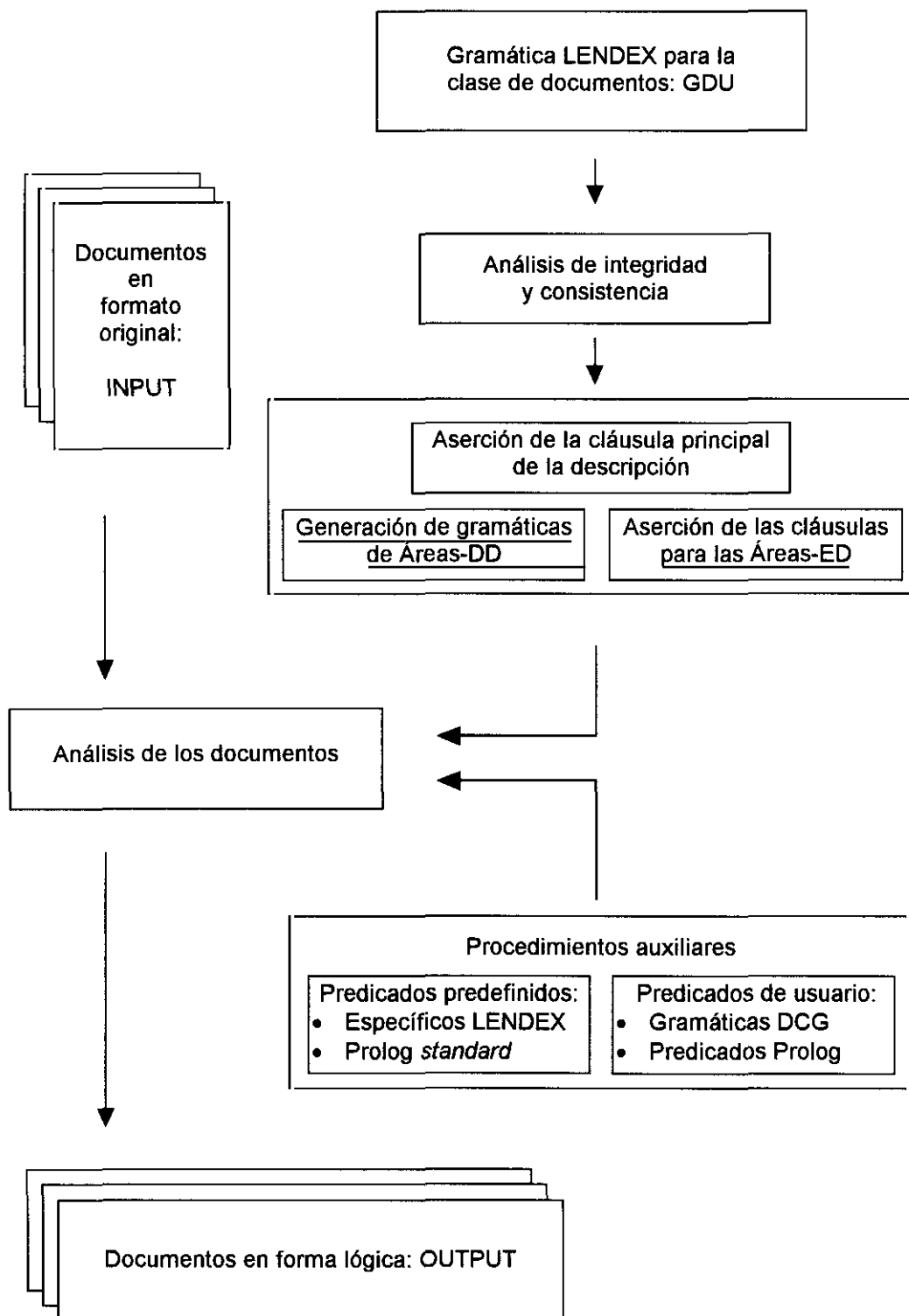
Una vez descrito LENDEX, conocido su propósito general y establecida su gramática, llega el momento de presentar el programa para el que fue concebido: un sistema automático de análisis documental.

S-Lendex, desarrollado en el intérprete Arity Prolog® v.6, integra:

- La descripción LENDEX que un usuario hace de una clase de documentos
- Un sistema general de representación de la información inicial y final
- Un conjunto de procedimientos predefinidos:
 - Prolog *standard*
 - Específicos de LENDEX
- Un conjunto -quizás vacío- de procedimientos definidos por el propio usuario:
 - Predicados Prolog
 - Gramáticas DCG
- Un sistema de generación de analizadores *-parsers-* a partir de los elementos antes enumerados.

La Figura 4 representa gráficamente el sistema general:

Figura 4: Esquema general de S-Lendex



En ella puede apreciarse cómo el sistema tiene un doble *input*:

- La clase de documentos a analizar
- La gramática LENDEX que el usuario define para describir los documentos

Con respecto a la primera, es necesario prever un sistema de representación de la información que satisfaga tres condiciones imprescindibles: generalidad, fácil acceso y univocidad. Con ese fin, cada documento a analizar por el sistema recibe un número de identificación único que sirve de punto de acceso al mismo y lo identifica dentro de la clase de documentos de manera inequívoca.

Con respecto a la gramática definida por el usuario, en adelante GDU, cabe destacar el análisis de integridad y consistencia al que se ve sometida antes de ser aplicada. En este proceso se efectúa una verificación de la GDU con el fin de detectar posibles problemas de tipo sintáctico o gramatical. Cada una de las expresiones de la descripción debe ser gramaticalmente correcta, esto es, debe ajustarse a las reglas que G_{Lendex} establece. Además son evaluadas ciertas restricciones adicionales como son las relativas a la unicidad -por ejemplo- de las aserciones de áreas DD.

Si la GDU satisface el análisis de integridad al que se ve sometida, se procede a su aplicación sobre los documentos.

Ello supone la elaboración de la base de conocimientos Prolog asociada a las áreas DD y ED así como la generación del analizador para los documentos.

Los apartados en que vamos a estructurar la presentación de S-Lendex son los siguientes:

- Metodología general para la descripción del programa

- Representación de los documentos: formatos inicial y final
- Análisis de integridad y consistencia de la GDU
- Generación de la base de conocimiento Prolog a partir de la GDU
- Procedimientos auxiliares disponibles: Prolog *standard*, específicos de LENDEX
- Generación y funcionamiento del analizador para la clase de documentos

Metodología de presentación del programa

La presentación de los procedimientos de S-Lendex sigue en parte la metodología expuesta en [Deville,90] para el desarrollo de programas. Este autor distingue tres momentos en el desarrollo de programas:

1. la especificación del problema
2. una descripción en LPO independiente del lenguaje de programación
3. la implementación del programa

Atendiendo a [Sterling-Shapiro,86; p.201] donde leemos: *“La experiencia de los autores es que muy raramente una especificación en lógica de primer orden es más breve, más simple o más legible que el programa Prolog correspondiente... En algún sentido, los programas Prolog son especificaciones ejecutables”*, y considerando que gran parte del desarrollo de S-Lendex se desenvuelve en un ámbito -el de la metaprogramación-, no tratado por [Deville,90] obviamos la segunda fase de las previstas por este autor e insistimos en las ventajas de la primera: la especificación.

Asociamos a cada uno de los procedimientos presentados en S-Lendex lo que denominamos *esquema de procedimiento*. Éste consiste en la descripción del mismo atendiendo a cuatro criterios:

- especificación de su nombre y aridad
- determinación del tipo de argumentos de que consta
- direccionalidad
- significado

La determinación del tipo de los argumentos se realiza teniendo en cuenta las categorías definidas en G_{Lendex} : atomo, lista, termino, etc.

Por direccionalidad entendemos lo que en [Deville,90; p.33] se expresa así: *“La direccionalidad de una especificación es la descripción de los posibles usos del procedimiento. Describe la forma de los parámetros antes y después de su ejecución”*

Distinguimos a este respecto las tres formas utilizadas en la especificación de los procedimientos Prolog para su compilación: [Arity Prolog, p.83].

- + el parámetro correspondiente está instanciado
- el parámetro no está instanciado
- ? el parámetro puede estar o no instanciado

La direccionalidad de un procedimiento se expresa en pares del tipo:

(in/A, out/A)

donde A es la aridad del procedimiento y los literales in/A y out/A expresan respectivamente la forma de los parámetros de entrada y salida. Un procedimiento

puede tener varios pares asociados en función de los diversos usos posibles del mismo. La frecuencia -de mayor a menor- es el criterio empleado para ordenarlos.

En lo relativo al significado, adoptamos el uso que [Sterling-Shapiro,86; p.200] hacen del concepto *esquema de la relación*; por éste entienden la explicación que debe acompañar a cada procedimiento en la que se define la relación que éste computa.

Por lo que se refiere a la escritura de los programas, se ha pretendido seguir las recomendaciones habituales en metodología de la programación²⁹: programas generales, declarativamente claros, fácilmente modificables, así como eficaces en lo posible. Si se consiguió o no es algo que no corresponde a la que escribe establecer.

Representación de los documentos

Formato de entrada: G_{input}

LENDEX está concebido para analizar documentos de carácter textual de familias diversas. No hay, en principio, limitación lógica en cuanto al tipo de documento que puede ser objeto de su análisis. Incluyendo en éste naturalmente el análisis trivial que consiste en hacer de todo un documento -supongamos texto libre- un único área.

²⁹[Shapiro, p.192 y ss.], etc.

Sí hay una tónica general: los documentos a analizar se agrupan en familias en función de unos rasgos relativamente estables que constituyen su formato o estructura [Sarabia,94]. Como el autor indica, el grado de estructuración de la información es muy variable. Lo importante, en nuestro caso, es que cada familia de documentos sea caracterizable por ciertas regularidades expresables gramaticalmente.

Pensemos en literatura de referencia como catálogos de publicaciones, índices y registros bibliográficos o documentos que como cierto género de cartas, facturas o informes ofrecen características constantes relativamente bien establecidas.

S-Lendex puede ser útil para ese tipo de documentos.

El primer problema que se plantea es el de hacer accesible al sistema los documentos que se hallan en formato y soporte originales.

En cuanto al soporte, éste puede ser de lo más variado: desde la cartulina de las fichas bibliográficas, pasando por el papel de las cartas o el electrónico de una base de datos documental. En cuanto al formato, cada clase de documentos tiene uno propio: justamente aquél que será el punto de partida de su descripción.

Como ya indicamos en la introducción, aquí no vamos a ocuparnos del modo en que la variedad de soportes se trata. Cuestiones relativas a la captación de los documentos por algún sistema de reconocimiento óptico de caracteres, OCR, o la exportación desde una base de datos documental no van a ser objeto de nuestro análisis.

Nosotros partimos de un sistema general de representación de la información que unifica esa variedad de estados originales y permite tratarlos de manera uniforme. Una representación donde se identifica cada documento y se aplica sobre él la GDU específica que describe su formato.

Con ese fin, definimos una clase de documentos, a los que denominaremos input.

Para ello, suponemos un documento de texto legible por un intérprete Prolog con la siguiente estructura, definida de acuerdo con la gramática G_{Input} :

$G_{\text{Input}} = \langle \text{SN}_{\text{Input}}, \text{ST}_{\text{Input}}, \text{P}_{\text{Input}}, \text{S}_{\text{Input}} \rangle$, donde:

- SN_{Input} es el conjunto de símbolos no terminales

$\text{SN}_{\text{Input}} = \text{SN1}_{\text{Input}} \cup \text{SN2}_{\text{Input}} \cup \text{SN3}_{\text{Input}}$ donde:

$\text{SN1}_{\text{Input}} = \{\text{input}, \text{nserie_documento}, \text{nserie}, \text{documento_origen}\}$

$\text{SN2}_{\text{Input}} = \{\text{parentesis_i}, \text{parentesis_d}, \text{coma}, \text{punto}, \text{f}\}$ ³⁰

$\text{SN3}_{\text{Input}} = \{\text{string}, \text{entero}\}$ ³¹

- ST_{Input} es el los símbolos terminales

$\text{ST}_{\text{Input}} = \text{ST}_{\text{Lex}}$ ³²

- S_{Input} es el axioma o símbolo inicial:

$\text{S}_{\text{Input}} = \langle \text{input} \rangle$

- P_{Input} es el conjunto de las siguientes producciones:

$\langle \text{input} \rangle \quad ::= \quad \langle \text{nserie_documento} \rangle \{$

³⁰Los no terminales de $\text{SN2}_{\text{Input}}$ están definidos en G_{Lex} ∴ $\text{SN2}_{\text{Input}} \subset \text{SN3}_{\text{Lex}}$. Véase: *La gramática del lexema* en la p.32

³¹string y entero están definidos en G_{Lendex} : $\text{SN3}_{\text{Input}} \subset \text{SN2}_{\text{Lendex}}$. Véase: *La gramática de LENDEX* en la p.51

³²El conjunto de símbolos terminales de la clase de documentos input es el definido para G_{Lex} . Véase de nuevo: *La gramática del lexema* en la p.32. Es preciso tener en cuenta que dicha clase representa a los documentos en general; de ahí que su conjunto de terminales deba ser el mismo.

```

                                <nserie_documento }

<nserie_documento> ::= d <parentesis_i <nserie> <coma>
                                <documento_origen> <parentesis_d>
                                <punto> <fl>

<nserie> ::= <entero>
<documento_origen> ::= <string>
    
```

Un ejemplo de Input

Un documento input, en el que encontramos el registro bibliográfico ya conocido de secciones anteriores³³ con otros de su misma familia: registros con formato ISBD, es el que sigue:

d(0,\$	2716	ABADIA, Antonio
Distribución de la renta y composición de la demanda en la economía española / Antonio Abadía.- Madrid : Fundación Empresa Pública; Universidad Compluten-		
se, 1983. (1) h., 22 p.; 30 cm. - (Documento de trabajo; 8301)		
1. Consumo (Economía) - España I. Fundación Em-		
presa Pública, ed. II. Título R. 2716\$).		
d(1,\$	979	ABELLANAS, Pedro
Elementos de Matemáticas ... 11ª ed. Madrid (Pedro Abellanas) (1973) XIV,		
1h., 241 p. grab. 25 cm. 1. Matemáticas I. Título R.979\$).		
d(2,\$	2778	ABOYADE, Ojetunji
Integrated economics: a study of developing economics / Ojetunji Aboyade.- London		
(etc.): Addison . Wesley Publishers, cop. 1983. VIII, 568 p.; 24 cm.		

³³Véase la Figura 2 de la p.22

```
ISBN: 0-201-1468-3 1. Desarrollo económico I. Título. 2583 R.
2778$.

d(3,$ 2583 ABRAHAM-
FROIS, Gilbert Elements de dynamique economique: (fluctuaciones et
croissance) / Gilbert Abraham-Frois. - 4ª ed. - Paris: Dalloz, 1985. 353 p.; 24
cm. - (Mémentos Dalloz) ISBN: 2-247-00462-8 1. Teoría económica I.
Título. R. 2583$).

d(4,$ L ACADEMIA ESPAÑOLA. Madrid
Diccionario de la lengua española. - 20ª ed. - Madrid: Real Academia Española,
1984. 2 V.(XXV,1416 p.) ISBN: 84-4777-7 (obra completa) 1.
Lengua española - Diccionarios 2. Dic- cionarios- Lengua española I.
Título. R. 2589$).
```

Figura 5: Ejemplo de documento Input

En él podemos apreciar cómo a cada documento se le ha asociado un número de serie que lo identifica unívocamente dentro de la clase a la que pertenece. Además, el documento ya tiene una representación accesible para un intérprete de Prolog, en concreto para la implementación que se ha utilizado: Arity Prolog®.

Cada uno de los documentos con su número de serie constituye un literal Prolog de aridad 2. El primer argumento es un entero que actúa de identificador; el segundo es una cadena de texto, un string, que representa al documento.

Observamos que el documento ha pasado por una leve transformación: los fines de línea no aparecen tal y como se encuentran en el documento original³⁴, sino que han sido sustituidos por un carácter especial, en concreto el correspondiente al Ascii 27. Ello obedece al interés por destacar en este contexto la estructura del término Prolog formado. Desde un punto de vista operativo, es indiferente la

³⁴Recordemos de nuevo el ejemplo de la p.22

presencia de los fines reales o de unos caracteres simbólicos equivalentes en su lugar.

Otros ejemplos de documento input, con el formato requerido para constituir una entrada en S-Lendex, se encuentran en el Anexo 2, p.276, de este trabajo. Las clases de documentos representadas son distintas que la anterior. Lo importante es apreciar cómo siendo clases diferentes de documentos, se representan en un formato uniforme de entrada para S-Lendex. Es competencia de la GDU recoger las peculiaridades de formato que cada familia de documentos presenta y conseguir un análisis adecuado de las mismas.

Formato de salida: G_{Output}

Así como es preciso uniformar la representación de los documentos de entrada en LENDEX, se hace inevitable plantear un formato de representación único para el resultado de su análisis.

Teniendo en cuenta que los fines para los que puede utilizarse dicho resultado son variados, la consecución de un formato de representación uniforme, clara y estrictamente definido, se convierte en tarea fundamental. Es el mismo propósito que [Sarabia,94] plantea al hablar de notación uniforme o *interlingua*. La diferencia estriba en el hecho del distinto punto de vista que allí se contempla. Teniendo a la vista el sistema general antes mencionado³⁵, lo que aquí constituye un formato de salida allí es un lenguaje común intermedio que sirve de base para la representación final elegida: el formato conocido como *Standard Generalized Markup Language*, SGML.

³⁵Véase la p.13, donde se habla del proyecto BiblioTECA.

Nuestra representación va a consistir nuevamente en un documento de texto legible por un intérprete Prolog con una estructura precisa: una colección de literales de aridad 2 que representa el resultado del análisis sobre cada uno de los documentos que componen el input.

Para cada uno de los documentos representados en input, existe estrictamente un literal en docout que contiene su número de identificación en el primer argumento y el resultado de efectuar el análisis LENDEX del documento en el segundo.

Con vistas a clarificar semejante representación uniforme, definimos una nueva clase de documentos: la de los denominados output. Su estructura se establece de acuerdo con la gramática G_{Output} :

$G_{Output} = \langle SN_{Output}, ST_{Output}, P_{Output}, S_{Output} \rangle$, donde:

- SN_{Output} es el conjunto de símbolos no terminales

$$SN_{Output} = SN1_{Output} \cup SN2_{Output} \cup SN3_{Output} \text{ donde:}$$

$$SN1_{Output} = \{output, nserie_ analisis, nserie, documento_ fl\}$$

$$SN2_{Output} = SN2_{Input} \cup \{ corchete_ i, corchete_ d \}$$

$$SN3_{Output} = SN3_{Input}^{36}$$

- ST_{Output} es el los símbolos terminales

$$ST_{Output} = ST_{Lex}^{37}$$

³⁶Los no terminales de G_{Output} coinciden, salvo los indicados, con los de G_{Input} . Recuérdese su definición de la p.70

³⁷La razón por la cual $ST_{Output} = ST_{Lex}$ es la misma que la apuntada en el caso de los documentos Input. Véase la nota al pie n° 32, de la p.72

- S_{Output} es el axioma o símbolo inicial:

$S_{Output} = \langle output \rangle$

- P_{Output} es el conjunto de las siguientes producciones:

$\langle output \rangle ::= \langle nserie_ analisis \rangle \{ \langle nserie_ analisis \rangle \}$

$\langle nserie_ analisis \rangle ::= d \langle parentesis_ i \rangle \langle nserie \rangle \langle coma \rangle$
 $\langle documento_ fl \rangle \langle parentesis_ d \rangle$
 $\langle punto \rangle \langle fl \rangle$

$\langle nserie \rangle ::= \langle entero \rangle$

$\langle documento_ fl \rangle ::= \langle areas_ fl \rangle$

$\langle areas_ fl \rangle ::= \langle corchete_ i \rangle \langle area_ fl \rangle \{ \langle coma \rangle$
 $\langle fl_ area \rangle \} \langle corchete_ d \rangle$

$\langle area_ fl \rangle ::= \langle analisis_ area \rangle \mid \langle incidencia_ lx \rangle$

$\langle analisis_ area \rangle ::= \langle nombre_ area_ lendex \rangle$
 $\langle parentesis_ i \rangle \langle contenido_ inf \rangle$
 $\langle parentesis_ d \rangle$

$\langle contenido_ inf \rangle ::= \langle string \rangle \mid \langle areas_ fl \rangle$

$\langle incidencia_ lx \rangle ::= (ign \langle descriptores \rangle) \mid$
 $(ind \langle string \rangle) \mid$
 $(inc \langle area_ fl \rangle)$

Este es el formato de representación del output teniendo en cuenta las incidencias de diverso tipo que durante la aplicación de la GDU sobre los documentos puedan

ocurrir: fallo de una determinada categoría gramatical, indeterminación de un segmento de entrada, imposibilidad para decidir a qué categoría corresponde una cierta secuencia.

A continuación planteamos -a semejanza de lo hecho en el caso del input- un ejemplo que clarifique la definición formal.

Un ejemplo de Output

Vamos a presentar como ejemplo el que sería correspondiente al input de la sección anterior³⁸.

No nos importa en este momento la gramática que dio lugar a estos resultados. Como puede observarse, es una gramática que analiza los registros de acuerdo con la especificación ISBD, distinguiendo en ellos sus áreas características: encabezamiento, título, edición, publicación, etc.

```
d(0,[sigs($2716$),enc([apell($ABADIA$),nombre($Antonio$)]), tit($Distribución
de la renta y composición de la demanda en la economía española / Antonio
Abadía$),pub($Madrid: Fundación Empresa Pública; Universidad Complutense,
1983$), df([ext($1) h., 22 p.$), dim($30 cm.$)]), serie($Documento de trabajo;
8301$), sec($1. Consumo (Economía) - España$), sec($I. Fundación Empresa
Pública, ed.$), sec($II. Título$), regs($R. 2716$)).
```

```
d(1,[sigs($979$), enc([apell($ABELLANAS$),nombre($Pedro$)]), tit($Elementos de
Matemáticas ...$), ed($11ª ed$), pub($Madrid (Pedro Abellanas) (1973$),
df([ext($XIV, 1h., 241 p. grab.$),dim($25 cm.$)]), sec($1. Matemáticas$),sec($I.
Título$),regs($R.979$)).
```

³⁸Véase: *Un ejemplo de Input*, en la p.73

```
d(2,[sigs($2778$),enc([apell($ABOYADE$),nombre($Ojetunji$))], tit($      Integrated
economics: a study of developing economics / Ojetunji Aboyade$),pub($London
(etc.): Addison . Wesley Publishers, cop. 1983$), df([ext($VIII, 568 p.$),dim($24
cm.$)]), isbn($ISBN: 0-201-1468-3$), sec($1. Desarrollo económico$),sec($I.
Título$),regs($R. 2778$)).

d(3,[sigs($2583$), enc([apell($ABRAHAM-FROIS$), nombre($Gilbert$) ]),
tit($Elements de dynamique economique: (fluctuaciones et roissance) / Gilbert
Abraham-Frois$), ed($4ª ed.$), pub($París: Dalloz, 1985$), df([ext($353 p.$),
dim($24 cm$)]), serie($Mémentos Dalloz$), isbn($ISBN: 2-247-00462-8$), sec($1.
Teoría económica$), sec($I. Título$), regs($R. 2583$)).

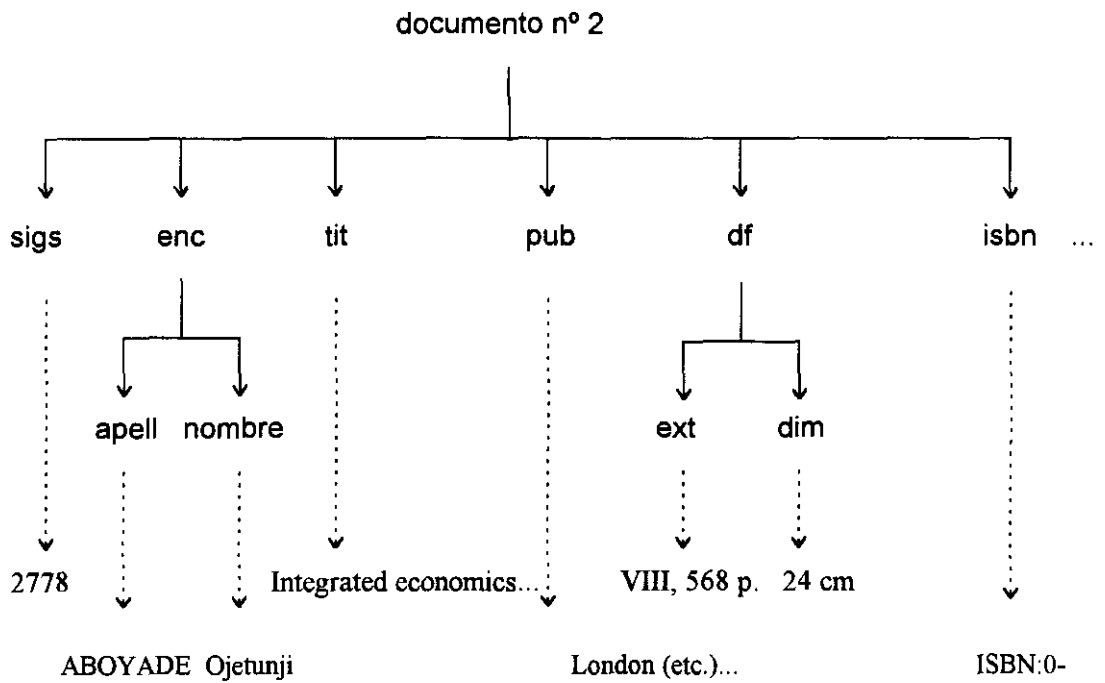
d(4,[sigs($L$),enc($ACADEMIA ESPAÑOLA. Madrid$), tit($Diccionario de la lengua
española$), ed($20ª de$), pub($Madrid: Real Academia Española, 1984$),
df([ext($2V.(XXV,1416 p.)$)]), isbn($ISBN: 84-4777-7 (obra completa)$),sec($1.
Lengua española - Diccionarios$), sec($2. Diccionarios- Lengua española$), sec($I.
Título.$), regs($R. 2589$)).
```

Figura 6: Ejemplo de documento Output

El interés del ejemplo estriba en comprobar la estructura del output y en apreciar la relación que se establece con el input a través del número de identificación. Como ya se indicó antes, cada uno de los documentos representados en el input tiene un correlato en el docout: el resultado de su análisis, el mismo documento en forma lógica.

Ésta se representa muy adecuadamente como un árbol de análisis. Veamos en un caso cómo la representación Prolog recoge dicha estructura:

Figura 7: Representación en forma de árbol del análisis LENDEX de un registro bibliográfico



Es relevante reparar aquí en la relación que esta figura guarda con la

Figura 3 de la p.23.

Allí aparecía también el árbol de análisis correspondiente a la parte inicial del mismo registro bibliográfico. Áreas como justificado y fines en esta ocasión no aparecen. Ello es debido a su carácter de sólo reconocimiento. La representación de los documentos en el output se hace cargo de su contenido informativo, sólo relativo a las áreas RR, quedando fuera de consideración las áreas R.

Análisis de integridad y consistencia

Como indica la Figura 4, p.66, antes de poder aplicar la GDU sobre la clase de documentos, es necesario llevar a cabo un análisis de integridad y consistencia de la descripción que asegure:

- la inexistencia de errores sintácticos básicos
- todas las expresiones de la GDU son gramaticalmente correctas
- se cumplen ciertas restricciones adicionales

Si la GDU se prueba gramaticalmente correcta, el proceso continúa con la aplicación de la misma sobre la clase de documentos. En otro caso, se informa al usuario de los problemas que hacen imposible proseguir.

Errores sintácticos

La captación de errores de carácter sintáctico queda en manos del intérprete de Prolog. Estos errores consisten entre otros en:

- inadecuado uso de signos de puntuación básicos como: punto, coma, puntoycoma, etc.
- pares de signos: corchete_i / corchete_d, parentesis_i / perentesis_d no equilibrados
- incorrecto uso de operadores: problemas de precedencia y asociatividad

Gramaticalidad de la GDU

En este apartado, se pretende detectar si la GDU es reconocible por G_{Lendex} . De las expresiones que pueden aparecer en una GDU:

- produccion_principal
- area_dd, area_ed
- fin_dd
- analisis_dd
- condicion
- procedimiento
- gramatica_dcg
- comentario

nos interesa ahora destacar el común denominador: todas ellas son expresiones Prolog. En la que medida que tales, se ven sometidas al control sintáctico antes aludido del intérprete de Prolog.

Son las expresiones características de LENDEX: documento_lendex, condicion, area_dd, fin/3, area_ed las que exigen un análisis específico que asegure su gramaticalidad.

Este análisis pasa por:

1. Impedir la redefinición de los predicados característicos de LENDEX:
 - 1.1. $(==>)/1$, correspondiente a la descripción del documento
 - 1.2. $area_dd/2$, $fin/3$, relativos a la aserción de las áreas DD
 - 1.3. $(=>)/2$, correspondiente a la aserción de las áreas ED
 - 1.4. $condicion/3$, relativo a las condiciones adicionales sobre las áreas DD y ED
 - 1.5. $(->)/1$, relativo al análisis asociado a las áreas DD
2. Comprobar que los argumentos de las expresiones LENDEX se ajustan asimismo a las especificaciones de G_{Lendex}
 - 2.1. La descripción para la clase de documentos se expresa en una producción del tipo:

documento $==>$ D,

donde D es una lista Prolog

- 2.2. Las áreas DD se expresan en hechos del tipo:

area_dd(A, R),

donde A es el nombre del área: un átomo no perteneciente al conjunto NR de los nombres reservados de área: $NR = \{ign, ind, inc\}$, y R el rasgo asociado. Éste último pertenece a $RP \cup \{true\}$, donde $RP = \{en_may, con_may, en_min, numeral, romano\}$

- 2.3. Cada uno de los fines asociados a un área DD se expresa en un hecho del tipo:

$\text{fin}(A, \text{Fin}, T)$

donde A es el área cuyo fin el hecho establece, Fin es la expresión que marca el fin de A y T pertenece al conjunto $\text{TF} = \{\text{interno}, \text{externo_homo}, \text{externo_hetero}\}$

- 2.4. Cada una de las áreas ED se expresa en una producción del tipo:

$A \Rightarrow B,$

donde A es el nombre del área: un átomo no perteneciente al conjunto NR^{39} , y B la expresión en términos de la cual A se define

- 2.5. Las condiciones adicionales sobre las áreas se expresan mediante hechos del tipo:

$\text{condicion}(A, L, M),$

donde A es el nombre del área, L una expresión lambda y M pertenece al conjunto $\text{MP} = \{\text{fuerte}, \text{debil}\}$

- 2.6. El análisis adicional sobre las áreas DD se expresa mediante reglas del tipo:

$A \rightarrow L,$

donde A es el nombre del área DD y L es una lista de las categorías gramaticales que han de evaluarse una vez delimitada su extensión

Todas estas restricciones cabe plantearlas en una DCG que traduzca en Prolog las especificaciones en EBNF de G_{Lendex} . Esto puede, sin embargo, obviarse debido al

³⁹Véase restricción 2.2 donde NR se define

carácter Prolog de la GDU. En la medida que la descripción de la clase de documentos adopta la forma de un programa Prolog, es posible interrogarle acerca de los predicados que lo constituyen, estableciendo así las restricciones que sean oportunas. Una DCG reconocedora de las gramáticas LENDEX sólo añadiría un control más exhaustivo del tipo de errores que el intérprete Prolog capta por sí mismo, lo que no parece suficiente ventaja.

Por ello, preferimos plantear el análisis de integridad en forma de programa Prolog que actúa como precondition en S-Lendex, asegurando que las expresiones específicas de LENDEX cumplen ciertas restricciones sin las cuales no podría darse el correcto funcionamiento del sistema.

Cada una de estas restricciones se expresan en una cláusula del predicado `problema_integridad/1`. El significado de procedimientos auxiliares como `nombre_reservado/1`, `rasgo_posible/1`, etc, se encuentra en *Procedimientos auxiliares al análisis de integridad*, p.91. El de otros como `lista/1` y `lambda/1`, en *Procedimientos de uso general empleados por S-Lendex*, p.232. Presentamos a continuación el esquema e implementación de `problema_integridad/1`:

Procedimiento	<code>problema_integridad/1</code>
Argumentos	<code>problema_integridad(string)</code>
Direccionalidad	<code>in(-), out(+)</code> <code>in(+), out(+)</code>
Significado	<code>problema_integridad(M)</code> es verdadero si M es el mensaje que informa de un problema detectado en la GDU

`/* Restricciones número 1.1 - 1.5 */`

```
problema_integridad($Imposible redefinir el predicado: '==>'/2$):-
    not es_un_hecho('==>', 2).
```

```
problema_integridad($Imposible redefinir el predicado: area_dd/2$):-
    not es_un_hecho(area_dd, 2).
```

```
problema_integridad($Imposible redefinir el predicado: fin/3$):-  
    not es_un_hecho(fin, 3).
```

```
problema_integridad($Imposible redefinir el predicado: '=>/2$):-  
    not es_un_hecho('=>', 2).
```

```
problema_integridad($Imposible redefinir el predicado condicion/3$):-  
    not es_un_hecho('==>', 2).
```

```
problema_integridad($Imposible redefinir el predicado '->/2$):-  
    not es_un_hecho('->', 2).
```

/ Restricción número 2.1 */*

```
problema_integridad($Producción principal incorrecta$):-  
    D ==> Descripcion,  
    ( D \= documento ; not lista(Descripcion) ).
```

/ Restricción número 2.2 */*

```
problema_integridad(Mensaje):-  
    area_DD(AreaDD, R),  
    (  
        not atom(AreaDD) ;  
        nombre_reservado(AreaDD) ;  
        not rasgo_posible(R)  
    ),  
    pi_aux($Incorrecta definición del área DD$, AreaDD, Mensaje).
```

/ Restricción número 2.3 */*

```
problema_integridad(Mensaje):-  
    fin(A, F, Tipo),  
    not tipo_fin(Tipo),  
    pi_aux($Incorrecta definición del fin$, fin(A,F,Tipo), Mensaje).
```

/* Restricción número 2.4 */

```
problema_integridad(Mensaje):-  
    AreaED => _,  
    ( not atom(AreaED) ; nombre_reservado(AreaED)),  
    pi_aux($Incorrecta definición del área ED$, AreaED, Mensaje).
```

/* Restricción número 2.5 */

```
problema_integridad(Mensaje):-  
    condicion(A, C, M),  
    ( not lambda(C) ; not modo_evaluacion(M) ),  
    pi_aux($Incorrecta definición de la condición$, condicion(A,C,M),  
        Mensaje).
```

/* Restricción número 2.6 */

```
problema_integridad(Mensaje):-  
    (A -> L)  
    not lista(L),  
    pi_aux($Incorrecta definición del análisis$, (A -> L), Mensaje).
```

Restricciones adicionales

Hay restricciones adicionales a las expresadas en G_{Lendex} que deben ser establecidas antes de considerar una GDU apta para el análisis LENDEX. Una de ellas es la relativa a la necesaria relación que debe establecerse entre las expresiones que constituyen la GDU, por ejemplo, un cierto área DD y el fin o fines DD que la acompañan. Esto es, no vale para cada área DD cualquier fin DD. Es preciso que el fin DD lo sea del área para la que actúa como tal fin. Otro tipo de restricción tiene que ver con el número de hechos asociados a las áreas o sus condiciones.

Utilizaremos para expresar estas restricciones el conjunto RA_{Lendex} de los siguientes predicados:

$RA_{Lendex} = \{area_dd/1, area_ed/1, produccion_doc/1, asercion_dd/2, produccion_ed/1, fin_dd/2, condicion/2, analisis_dd/2\}$, con el significado intuitivo que sigue:

$area_dd/1 = \{X \mid X \text{ es un \u00e1rea DD}\}$

$area_ed/1 = \{X \mid X \text{ es un \u00e1rea ED}\}$

$produccion_doc/1 = \{X \mid X \text{ es la produccion principal de la GDU}\}$

$asercion_dd/2 = \{(X,Y) \mid X \text{ es la aserci\u00f3n del \u00e1rea DD } Y\}$

$produccion_ed/2 = \{(X,Y) \mid X \text{ es la producci\u00f3n correspondiente al \u00e1rea ED } Y\}$

$fin_dd/2 = \{(X,Y) \mid X \text{ es la aserci\u00f3n de un fin para el \u00e1rea DD } Y\}$

$condicion/2 = \{(X,Y) \mid X \text{ es la aserci\u00f3n de una condicion adicional sobre el \u00e1rea } Y\}$

$analisis_dd/2 = \{(X,Y) \mid X \text{ es la aserci\u00f3n del an\u00e1lisis asociado al \u00e1rea DD } Y\}$

Las restricciones son las siguientes:

1. $\forall X (area_dd(X) \rightarrow \exists Y fin_dd(Y,X))$
2. $\forall X \forall Y \forall Z ((asercion_dd(X,Z) \wedge asercion_dd(Y,Z)) \rightarrow X=Y)$
3. $\forall X \forall Y \forall Z ((produccion_ed(X,Z) \wedge produccion_ed(Y,Z)) \rightarrow X=Y)$
4. $\forall X \forall Y (fin_dd(X,Y) \rightarrow \neg area_dd(X))$

Las que siguen son restricciones ya establecidas en G_{Lendex} . Su relevancia y la necesidad de presentar el programa que las comprueba hacen que aparezcan aqu\u00ed nuevamente formuladas:

5. $\forall X \forall Y ((\text{descripcion_doc}(X) \wedge \text{descripcion_doc}(Y)) \rightarrow X=Y)$

6. $\forall X \forall Z (\text{fin_dd}(X,Z) \rightarrow \exists Y \text{area_dd}(Y) \wedge Z=Y)$

7. $\forall X \forall Z (\text{condicion}(X,Z) \rightarrow (\exists Y (\text{area_dd}(Y) \vee \text{area_ed}(Y)) \wedge Z=Y))$

8. $\forall X \forall Z (\text{analisis_dd}(X,Z) \rightarrow \exists Y \text{area_dd}(Y) \wedge Z=Y)$

9. $\forall X (\text{area_dd}(X) \rightarrow \neg \text{area_ed}(X))$

Cada una de las restricciones planteadas se puede expresar en forma de cláusula Prolog así:

/ Restricción 1: Cada área DD tiene al menos un fin asociado */*

```
problema_integridad(Mensaje):-  
    area_dd(A, _),  
    not fin(A, _, _),  
    pi_aux($No hay fin para el área: $, A, Mensaje).
```

/ Restricción 2: Sólo puede haber una aserción de cada área DD */*

```
problema_integridad(Mensaje):-  
    findall(A, area_dd(A,_), Lista),  
    duplicados(Lista, D),  
    pi_aux($Hay más de una aserción del área DD : $, D, Mensaje).
```

/ Restricción 3: Sólo puede haber una producción de cada área ED */*

```
problema_integridad(Mensaje):-  
    findall(A, (A => _), Lista),  
    duplicados(Lista, D),  
    pi_aux($Hay más de una aserción del área ED : $, D, Mensaje).
```

/* Restricción 4: Un área DD no puede ser declarada fin de otra */

```
problema_integridad(Mensaje):-
    fin(A, F, _),
    area_dd(F),
    string_termq(StA, A),
    string_termq(StF, F),
    concat([StF, $ no puede ser declarado fin para: $,StA],Mensaje).
```

/* Restricción 5: Sólo puede haber una producción principal */

```
problema_integridad($Hay más de producción principal$):-
    documento ==> D1,
    documento ==> D2,
    D1 \= D2.
```

/* Restricción 6: Un fin DD lo es siempre de al menos una área DD */

```
problema_integridad(Mensaje):-
    fin(A, F,_),
    not area_dd(A, _),
    pi_aux($No hay área asociada al fin: $, F, Mensaje).
```

/* Restricción 7: Toda condición se aplica siempre al menos sobre una área */

```
problema_integridad(Mensaje):-
    condicion(A, C, _),
    not area_dd(A, _),
    not (A => _),
    pi_aux($No hay área asociada a la condición: $, C, Mensaje).
```

/* Restricción 8: Todo análisis DD se aplica al menos sobre un área DD */

```
problema_integridad(Mensaje):-
    (A -> L),
    not area_dd(A, _),
    pi_aux($No hay área asociada al análisis: $, L, Mensaje).
```

/* Restricción 9: Un área DD no puede ser simultáneamente un área ED */

```
problema_integridad(Mensaje):-  
    area_dd(A, _),  
    A => _,  
    pi_aux($Definición ambigua del área: $, A, Mensaje).
```

Definido el procedimiento `problema_integridad/1`, es preciso un procedimiento que evalúe si cada una de las restricciones que el predicado anterior establece se cumple o no para una cierta GDU.

Con este fin definimos `analisis_integridad/1` así:

```
analisis_integridad(Lista):-  
    findall(X, problema_integridad(X), Lista),  
    evaluacion_analisis_integridad(Lista).
```

El procedimiento primitivo `findall/3` sirve perfectamente a este fin. Su aplicación nos brinda una lista con los mensajes asociados a los problemas detectados en la GDU. Si la lista es vacía podremos considerar gramaticalmente aceptable la GDU y proseguir. En otro caso, el procedimiento informará al usuario y detendrá el proceso. Para ello es preciso un procedimiento auxiliar, `evaluacion_analisis_integridad/1`, cuyo éxito depende exactamente del carácter -vacío o no- de la lista resultado del `findall/3`.

```
evaluacion_analisis_integridad([ ]):- !.  
evaluacion_analisis_integridad(Lista):-  
    los Lista son X^outnl(X),  
    !, fail.
```

El significado del procedimiento `son/2` utilizado en la segunda cláusula de `evaluacion_analisis_integridad/1` se encuentra en *Procedimientos de carácter general*, p.232.

Procedimientos auxiliares al análisis de integridad

/ pi_aux(S, T, M) : M es la concatenación de la secuencia de texto S con el resultado de convertir a string el término T */*

/ nombre_reservado(N) : N es un átomo que cae bajo la categoría del mismo nombre definida en G_{Lex} : ign, inc, ind */*

/ rasgo_posible(R) : R es un átomo perteneciente al conjunto RP de los rasgos: RP = {en_may, en_min, con_may, numeral, romano} */*

/ tipo_fin(T) : T es un átomo perteneciente al conjunto TF:
TF = {interno, externo_homo, externo_hetero}*

/ modo_evaluacion(M) : M es un átomo perteneciente al conjunto MP:
MP = {fuerte, debil} */*

/ duplicados(L, D) : D es la lista formada por los elementos duplicados de la lista L */*

Generación de la base de conocimiento Prolog a partir de la GDU

Una vez superada la prueba de integridad, la GDU está en disposición de ser aplicada al análisis de los documentos. Es necesario clarificar cómo.

La GDU escrita por el usuario de acuerdo con las especificaciones de G_{Lendex} consta de varios elementos característicos:

- la producción principal en la que se describe la clase de documentos:

documento \Rightarrow

- los hechos relativos a las áreas DD y sus fines:

area_dd(area_1,...).

fin(area_1,...)

- las producciones correspondientes a áreas ED:

area_ed_1 \Rightarrow ...

- las condiciones asociadas a áreas DD y áreas ED:

condicion(area_1,...)

Éstos son, por sí mismos, una base de conocimiento Prolog *standard*. Constituyen un conjunto de cláusulas escrito de acuerdo con ciertas especificaciones; en concreto, definen ciertos predicados en forma de hechos Prolog. Sin embargo, el programa que como conjunto de cláusulas componen no está listo para el análisis de documentos. No son una gramática comprensible de manera inmediata por un

intérprete de DCGs. Precisan de un programa especializado que interprete cada una de sus afirmaciones y genere un analizador a partir de ellas.

Vamos a distinguir en la exposición el tratamiento dado a la descripción general de la clase de documentos, a las áreas DD y a las ED. La razón es clara. La situación en la que se encuentran las áreas DD en la GDU es considerablemente distinta de la que presentan las áreas ED y el documento en su conjunto.

Para éste último y para las áreas ED existen en la GDU unos hechos característicos, afirmados mediante los predicados $(==>)/2$ y $(=>)/2$ respectivamente, que presentan el aspecto -por su definición infija- de producciones gramaticales *standard*. Están más cerca de ser, por tanto, analizadas por un intérprete de DCGs si salvamos el símbolo de la producción -característico de LENDEX- y lo suponemos sustituido por el propio de las DCGs: $-->$.⁴⁰

Las áreas DD se expresan, sin embargo, de manera considerablemente distinta. No tienen el aspecto de producciones gramaticales sino que se definen mediante hechos Prolog, $area_dd/2$, con uno o más fines asociados, $fin/2$. Ello obedece al carácter impreciso de su definición: por quedar determinadas sólo en lo que a su final se refiere cabe pensar en una forma general de representar el conjunto de producciones que las describen. El objetivo final es efectuar un proceso de metaprogramación gramatical que permita al usuario expresarse en términos declarativos y le evite atender los aspectos más procesuales de la descripción de los documentos. Esto es, dejar que el intérprete LENDEX genere el conjunto de cláusulas que el usuario habría de escribir en cada caso.

Vamos a reproducir este proceso con el fin de clarificar su generación automática.

⁴⁰Más adelante se verá por qué no se escriben producciones DCG *standard* en la GDU para la descripción de la clase de documentos y para las áreas ED

Generación automática de las gramáticas de áreas DD

Dividimos esta sección en los siguientes apartados:

- Definición de una gramática de área DD
- Estructura general de la gramática de áreas DD
- Otras consideraciones de interés relativas a la gramática de áreas DD
- Generación automática de la gramática de un área DD
- Aplicación a un ejemplo real

Definición de una gramática de área DD

El objetivo es presentar una situación concreta en la que se defina una gramática de área DD. A partir de ahí, intentaremos generalizar el proceso pensando en su automatización por S-Lendex.

Recuperamos para ello el documento que nos ha servido de ejemplo anteriormente⁴¹. Vamos a fijarnos en la parte del registro conocida como cuerpo de la descripción una vez superado el análisis de la primera zona comprendida entre la signatura y el comienzo del título.

<p>Integrated economics: a study of developing economics / Ojetunji Aboyade.-- London (etc.): Addison . Wesley Publishers, cop. 1983. VIII, 568 p.; 24 cm. ISBN: 0-201-1468-3</p>

⁴¹Véase: *Un documento ISBD* en la p.22

Supongamos que queremos definir el área `titulo_responsabilidad`. Sabemos que ella acaba con la aparición de una marca explícita de fin de área, típica de la ISBD: `.--` (punto, doble guión).

El siguiente conjunto de producciones DCG la describiría:

```
tit_resp --> fin_area_isbd.  
tit_resp --> [X],  
             tit_resp.
```

La primera de ellas da por concluida la construcción del área `tit_resp` cuando aparece en la secuencia de entrada una subsecuencia que cae bajo la categoría `fin_area_isbd`. La segunda constituye la llamada recursiva que incorpora a la extensión del área los lexemas que no cumplen la condición de fin de la misma. El orden de las cláusulas es, por consiguiente, capital. Lo primero que se comprueba es si el segmento inmediato es un fin para el área en construcción. En caso positivo, el proceso finaliza; en caso negativo, se procede de la misma manera con el siguiente lexema, pasando el anterior a formar parte de la extensión del área.

Además, es conveniente prever qué sucede cuando la secuencia de entrada acaba, esto es, cuando no quedan lexemas por evaluar. En caso de no añadir ninguna producción más, la gramática escrita sólo reconocería un área DD cuando apareciera explícitamente la marca considerada como fin. Una tercera cláusula -que por razones de eficacia pasaría a ser la primera- se hace precisa:

```
tit_resp([ ], [ ]). (*)
```

En ella, observamos cómo el formalismo de las DCGs es abandonado en favor de la expresión en Prolog *standard*. Ello presupone el conocimiento de la

representación interna de las producciones DCG, que en nuestro ejemplo es la siguiente:

```
tit_resp(Entrada,Salida):-  
    fin_area_isbd(Entrada,Salida).  
  
tit_resp([X|Y],Y):-  
    tit_resp(Y).
```

Evitar el abandono de la expresión gramatical supondría añadir una última cláusula del tipo:

```
tit_resp    -->    []. (**)
```

cuya cláusula Prolog equivalente es:

```
tit_resp(L, L).
```

Importa destacar la diferencia entre esta solución y la anteriormente planteada (*). Aquélla se atiene exclusivamente a la situación que queremos describir: la secuencia de entrada es vacía. En ella se da por concluida el área añadiéndose, por consiguiente, una condición adicional de fin para el área `tit_resp`.

En el segundo caso (**) se contempla ésta y otras situaciones. Naturalmente, una de las instancias posibles de la variable `L` que representa la secuencia de entrada es aquella en la que `L` se sustituye por la secuencia vacía. Ello daría lugar a la misma solución -dejando a un lado cuándo se produce- que la proporcionada en antes: listas de entrada y salida iguales a la lista vacía.

Pero ciertamente cualquier estado de la secuencia a analizar se vería también satisfecho con dicha producción. Esto es, para toda lista de entrada, existe una sustitución que devuelve la misma lista como resto sin analizar. Ello supone, consiguientemente, la declaración del área como opcional. Aunque nuestro

propósito se ve cumplido -contemplar el final de la secuencia de entrada-, son excesivas las consecuencias adicionales derivadas de esta solución.

Sólo en un caso las dos soluciones son equivalentes: cuando la definición del área se hace totalmente determinista:

```
(*) tit_resp([ ], [ ]).  
tit_resp --> fin_area_isbd, !.  
tit_resp --> [X],  
            !,  
            tit_resp.  
  
(**) tit_resp --> [ ].
```

La imposibilidad del *backtracking*, debida al corte de las cláusulas principales, hace que la adición de (*) a la cabeza o (**) al final sea indistinta en lo que al significado declarativo se refiere: ambas soluciones aportan el mismo conocimiento. No así en cuanto a su comportamiento operacional. Desde esta perspectiva, también resulta preferible la primera por tener éxito antes. La cláusula final rinde el mismo resultado con la diferencia de que antes que ella se intenta aplicar las cláusulas precedentes sin éxito.

En definitiva, resulta más ajustado a los objetivos pretendidos utilizar la primera de las soluciones propuestas: (*), quedando la definición del área así:

```
tit_resp([ ], [ ]):- !.  
tit_resp --> fin_area_isbd.  
tit_resp --> [X],  
            tit_resp.
```

El corte de la primera cláusula es de los denominados verde: no afecta al significado declarativo del procedimiento y actúa como optimizador al evitar un posible *backtracking* inevitablemente fallido -teniendo en cuenta la definición de la

categoría `fin_area_isbd` expuesta más abajo-, o inútil -suponiendo otra que hiciera opcional la categoría del fin de área-.

La definición de la categoría `fin_area_isbd` será suficiente para terminar de proporcionar una descripción simple del área:

`fin_area_isbd --> [., -, -].`⁴²

Al aplicarse dicha gramática a la secuencia a analizar⁴³:

```
?- tit_resp("Integrated economics: a study of developing□economics /
Ojetunji Aboyade.-- London (etc.):□ Addison . Wesley Publishers,
cop. 1983.□ VIII, 568 p.; 24 cm.□ ISBN: 0-201-1468-3", Resto).
```

el resultado sería el siguiente:

```
Resto = "London (etc.):□ Addison . Wesley Publishers, cop. 1983.□
VIII, 568 p.; 24 cm.□ ISBN: 0-201-1468-3".
yes
```

Esto es, la gramática escrita para el área `titulo_responsabilidad` habría reconocido la secuencia:

"Integrated economics: a study of developing□economics / Ojetunji Aboyade.--",

⁴²Obviamos las marcas propias de los terminales de la gramática *-strings*, (véase la p.51) - para hacer más legible el texto.

⁴³Pasamos por alto en este punto las tareas previas al análisis de la secuencia: sustitución de los fines de línea por caracteres simples, lexematización, etc, así como su representación exacta. Nos importa tener por el momento una visión general del proceso.

dejando un resto sin analizar a partir de la marca establecida como fin del área.

En el caso de que el área descrita tuviera otro fin posible, pensemos por ejemplo en la secuencia formada por tres espacios seguidos de un fin de línea, habría que añadir antes de la llamada recursiva una nueva producción al conjunto de cláusulas anteriormente escrito:

```
tit_resp --> espacio, espacio, espacio, fl.
```

o alternativamente enriquecer con una disyunción la cláusula del fin:

```
tit_resp --> fin_area_isbd
              ;
              (espacio, espacio, espacio, fl).
```

En general, y para cada uno de los fines asociados a un área DD, es preciso añadir una producción que dé por concluida la construcción del área cuando el segmento inicial de la secuencia a analizar caiga bajo la extensión del fin.

Estructura general de la gramática de áreas DD

La forma general, por tanto, de la gramática de un área DD sería la siguiente:

```
area_dd_X([ ], [ ]):- !.

area_dd_X --> fin_X_1.
area_dd_X --> fin_X_2.
...
area_dd_X --> fin_X_n.

area_dd_X --> [X],
              area_dd_X.
```

O lo que es equivalente:

```
area_dd_X([ ], [ ]):- !.
```

```

area_dd_X    -->  fin_X_1
                ;
                fin_X_2
                ;
                ...
                ;
                fin_X_n.

area_dd_X    -->  [X],
                area_dd_X
    
```

Encontramos una primera cláusula expresada en Prolog *standard* que contempla el fin de la secuencia de entrada: hace de esta circunstancia una condición adicional de fin para el área. Podríamos naturalmente evitar esta opción. No habría más que enriquecer G_{Lendex} en la dirección de asociar a cada una de las áreas el conocimiento apropiado: un tercer argumento añadido a los actuales `area_dd/2` expresaría si el final de la lista a analizar constituye o no una condición de fin para el área. En función de él, la primera cláusula ofrecería dos versiones:

1. El fin de la secuencia de entrada constituye un fin para el área:

```
area_dd_X([ ], [ ]):- !.
```

2. El fin de la secuencia de entrada no determina un fin para el área:

```
area_dd_X([ ], _):-
    !, fail.
```

La opción actual es la primera. Ello debido al interés de no complicar más de lo necesario la descripción de los documentos. La consecuencia inmediata es que la restricción relativa a la necesidad de que cada área DD tenga un fin asociado⁴⁴ queda en suspenso en la medida que S-Lendex ya incorpora una condición de fin para ellas.

⁴⁴Véase: *Análisis de integridad y consistencia*, en la p.89

Hay a continuación tantas producciones como fines asociados tiene el área DD en la GDU -o una sola con una disyunción equivalente en su parte derecha-.

Por último una regla recursiva -constructora del área DD- cierra el grupo de cláusulas que la definen.

Otras consideraciones de interés relativas a la gramática de áreas DD

Hasta ahora no nos ha preocupado distinguir el tratamiento dado a los fines de diversa índole: internos, externos, homogéneos, heterogéneos. Todos los fines considerados, en la medida que pertenecientes a la extensión del área cuyo fin establecen, serían fines internos.

Tampoco hemos aludido al carácter de reconocimiento o reconocimiento y recuperación del área definida. En primera instancia y sin más explicación parece que el esquema general previsto lo es para áreas R. No hemos visto aún cómo recuperar la extensión reconocida por el área DD ni menos aún su contenido informativo, condiciones necesarias para considerar un área RR.

Vamos a intentar dilucidar en lo posible estos temas.

En caso de querer definir un fin externo al área de nuestro ejemplo, podríamos utilizar un mecanismo incorporado por el intérprete Arity Prolog para las DCGs consistente en dejar como resto una cierta secuencia que pasa a formar parte del resto original como subsegmento inicial.

La forma general del tipo de producción que hace uso de este mecanismo es la que sigue:

area_X, Secuencia --> definicion_area_X.

donde **Secuencia** tiene que ser una lista. Con ello se significa que una vez evaluada el **area_X** de acuerdo con la definición que el cuerpo de la producción indica, ha de anteponerse al resto que la evaluación haya dejado la secuencia representada por **Secuencia**.

En nuestro ejemplo, y por concretar. Consideremos la gramática definida para el área **tit_resp**:

```
tit_resp([ ], [ ]):- !.  
  
tit_resp -->    fin_area_isbd.  
tit_resp -->    espacio, espacio, espacio, fl.  
  
tit_resp -->    [X],  
                tit_resp.
```

Supongamos que queremos definir el primero de los fines como externo al área. Una ligera modificación de la segunda producción servirá a nuestro objetivo:

```
tit_resp, [., -, -] -->    fin_area_isbd.
```

Al aplicarse la nueva gramática a la secuencia a analizar:

```
?- tit_resp("Integrated economics: a study of developing economics /  
Ojetunji Aboyade.-- London (etc.): Addison . Wesley Publishers,  
cop. 1983. VIII, 568 p.; 24 cm. ISBN: 0-201-1468-3", Resto).
```

el resultado sería el siguiente:

```
Resto = "-- London (etc.): Addison . Wesley Publishers, cop.  
1983. VIII, 568 p.; 24 cm. ISBN: 0-201-1468-3".  
  
yes
```

Vemos cómo la secuencia reconocida por la categoría **fin_area_isbd** aparece como subsegmento anterior del resto general. Ese es el efecto de la modificación

realizada. El fin que antes caía bajo la extensión del área `tit_resp` se convierte de esta manera en externo al quedar como segmento inicial del resto una vez evaluada la categoría.

Otra posibilidad -independiente del intérprete de Arity Prolog- consiste en definir el mismo mecanismo explícitamente. Una categoría pseudogramatical o de control: `apr_side/3`, [Sarabia,93], sirve a estos fines.

```
apr_side(A, I, O):-
    ifthenelse(
        (A = [ ] ; A = [ _| _]),
        append(A, I, O),
        append([A], I, O)
    ).
```

El carácter externo de la categoría `fin_area_isbd`, antes expresado así:

```
tit_resp, [., -, -] --> fin_area_isbd.
```

sería establecido con `apr_side/3` así:

```
tit_resp --> fin_area_isbd, apr_side([., -, -]).
```

La ventaja de este procedimiento consiste en la mayor generalidad -puede añadirse al resto cualquier término en cualquier punto de la producción, no sólo una lista una vez terminada su evaluación- y la transparencia. Importa, eso sí, destacar el carácter de control -como el corte, `!/0`, por poner un ejemplo- que la categoría tiene.

De la misma manera que hemos dejado como resto una secuencia homogénea con las listas de entrada y salida -lexemas-, nada impide que dejemos una secuencia heterogénea, por ejemplo, una lista formada por un literal que contenga la misma información. Supongamos esta nueva producción:

```
tit_resp, [fa(--)] --> fin_area_isbd.
```

o, utilizando `apr_side/3`:

```
tit_resp --> fin_area_isbd, apr_side(fa(--)).
```

el resultado de interrogar de la misma manera al sistema sería el siguiente⁴⁵:

Resto = "fa(--)	London (etc.):	Addison . Wesley Publishers, cop.
1983.	VIII, 568 p.; 24 cm.	ISBN: 0-201-1468-3".
yes		

Veremos más adelante cómo esta posibilidad será utilizada con distintos fines. Sólo sugerir en este punto la ventaja de guardar resultados parciales ya computados en la optimización del sistema.

Por el momento, lo que parece claro es que disponemos de un esquema general para representar la gramática de las áreas DD. Y con el esquema general el camino expedito para pensar en su generación automática.

Un punto de decisiva importancia queda por aclarar: cómo invocar a las categorías involucradas en los fines si éstas son áreas LENDEX y cómo calcular la extensión de los fines externos para dejarlos como segmento inicial del resto. Ambas cuestiones -comentadas en detalle más adelante- se ven solucionadas con la definición de un procedimiento, `meta_lendex/6`, que más tarde presentaremos.

Éste también dará respuesta al problema sin resolver planteado al principio de esta sección relativo a cómo distinguir el carácter R o RR de las áreas DD.

⁴⁵Insistimos en el carácter representativo de la salida: nos importa fundamentalmente aclarar las posibilidades del mecanismo expuesto.

Generación automática de la gramática de áreas DD

Una GDU contiene entre otros hechos los relativos a las áreas DD y sus fines. Pensemos que nuestro ejemplo podría verse expresado de la siguiente manera:

```
area_dd(tit_resp, true).  
  
fin(tit_resp, fin_area_isbd, interno).  
fin(tit_resp, (espacio, espacio, fl), interno).
```

Pues bien, se trata de generar automáticamente a partir de estos hechos Prolog una gramática para el área de `tit_resp`.

Ésta debe ajustarse a la forma general prevista⁴⁶. Los predicados Prolog *standard* de la familia de `assert/1` sirven a nuestro objetivo al añadir cláusulas a la base de conocimiento. Por otra parte, `expand_term/2` permite traducir a Prolog *standard* una producción DCG. Disponemos de todo lo necesario para programar la generación automática de una gramática de área DD.

Supongamos en general un área a la que llamaremos `Area` con el rasgo asociado `Rasgo` y un fin del tipo `TipoFin` al que designaremos `Fin`. Ambos expresados en el par de hechos Prolog que sigue:

```
area_dd(Area, Rasgo).  
fin(Area, Fin, TipoFin).
```

El procedimiento principal que genera automáticamente la gramática asociada al área llamada `Area` es `aserta_gramatica/2`:

Procedimiento	<code>aserta_gramatica/2</code>
Argumentos	<code>aserta_gramatica(nombre_area_lendex, rasgo).</code>

⁴⁶Véase la p.100 donde ésta se indica.

Direccionalidad	in(+,+), out(+,+)
Significado	aserta_gramatica(Area,Rasgo) genera la gramática del área DD Area con el rasgo asociado Rasgo y la añade a la base de conocimiento

```

aserta_gramatica(Area, Rasgo):-
    abolish(Area/2),
    aserta_lvacia(Area),
    aserta_fines(Area),
    aserta_recurcion(Area, Rasgo),
    !.

```

Como puede verse, este procedimiento llama a tres procedimientos auxiliares: `aserta_lvacia/1`, `aserta_fines/1`, `aserta_recurcion/2`, que se corresponden con los tres tipos fundamentales de cláusulas presentes en el esquema general de la gramática de una área DD⁴⁷. Hay además una llamada al primitivo `abolish/1` para asegurar que el procedimiento a definir es nuevo en la base de conocimiento.

Definimos a continuación el procedimiento que añade a la base de conocimiento la cláusula correspondiente al vaciado de la lista de entrada: `aserta_lvacia/1`.

Procedimiento	<code>aserta_lvacia/1</code>
Argumentos	<code>aserta_lvacia(nombre_area_lendex)</code> .
Direccionalidad	in(+), out(+)
Significado	<code>aserta_lvacia(Area)</code> añade a la base de conocimiento la primera cláusula de la gramática del área DD Area que establece el final de la secuencia a analizar como una condición adicional de fin para Area.

```

aserta_lvacia(Area):-
    Term =.. [Area, [ ], [ ]],
    asserta((Term:- !)).

```

⁴⁷Véase: *Estructura general de la gramática de áreas DD* en la p.100

El siguiente procedimiento añade las cláusulas correspondientes a los fines de área asociados a Area.

Procedimiento	aserta_fines/1
Argumentos	aserta_fines(nombre_area).
Direccionalidad	in(+), out(+)
Significado	aserta_fines(Area) añade a la base de conocimiento las cláusulas correspondientes a los fines asociados al área DD Area a continuación de la cláusula del fin de la secuencia de entrada y en el orden que indica la GDU

```

aserta_fines(Area):-
    ifthenelse(
        condicion(Area, _, fuerte),
        Corte = { true },
        Corte = !
    ),
    e_mientras(
        fin(Area, Fin, TipoFin),
        aserta_un_fin(Area, Fin, TipoFin, Corte)
    ).
    
```

Un comentario por cada uno de los procedimientos llamados por `aserta_fines/1`. El primero relativo al carácter determinista o no de la definición de las cláusulas correspondientes a los fines. Vemos como `ifthenelse/3` interroga acerca de la existencia de alguna condición adicional sobre el área de carácter fuerte.

Si esto sucede, las cláusulas de los fines definidas a continuación no serán deterministas -salvo las de los heterogéneos-, con el fin de que pueda producirse un *backtracking* que reconstruya el área en caso de que ésta no satisfaga la condición adicional impuesta. Si no hay condición adicional alguna o ésta es *debil*, se pasa como parámetro un corte con el fin de hacer determinista la producción correspondiente y ganar en eficacia.

El segundo comentario es el relativo al procedimiento `e_mientras/2`⁴⁸. En éste se produce la llamada al procedimiento subordinado `aserta_un_fin/4`, encargado de añadir una cláusula por cada uno de los fines asociados al área. Este procedimiento debe hacerse cargo del tipo de fin indicado en su tercer argumento: `interno`, `externo_homo`, `externo_hetero`, para lo cual consta de tres cláusulas distintas.

Procedimiento	<code>aserta_un_fin/4</code>
Argumentos	<code>aserta_un_fin(nombre_area_lendex, expresion, tipo_fin, corte).</code>
Direccionalidad	<code>in(+,+,+,+), out(+,+,+,+)</code>
Significado	<code>aserta_un_fin(Area, Fin, TF, Corte)</code> añade en último lugar a la base de conocimiento la cláusula correspondiente al fin de área <code>Fin</code> , de tipo <code>TF</code> , del área <code>DD Area</code> , de manera más o menos determinista según indique el parámetro <code>Corte</code> .

```
aserta_un_fin(Area, Fin, interno, Corte):-
    expand_term((Area --> meta_lendex(Fin, sin, _, _), Corte),
                Clausula),
    assertz(Clausula),
    !.
```

```
aserta_un_fin(Area, Fin, externo_homo, Corte):-
    expand_term((Area --> meta_lendex(Fin, con, Ext, CI),
                apr_side(cat_eval(Fin, homo, Area, Ext, CI)),
                Corte), Clausula),
    assertz(Clausula),
    !.
```

```
aserta_un_fin(Area, Fin, externo_hetero, _):-
    expand_term((Area --> meta_lendex(Fin, con, Ext, CI),
                apr_side(cat_eval(Fin, hetero, Area, Ext, CI)),
                !), Clausula),
    assertz(Clausula),
    !.
```

⁴⁸`e_mientras/2` es uno de los procedimientos de uso general en S-Lendex. Su significado se encuentra en *Procedimientos de carácter general*, p.232

De nuevo se imponen varios comentarios. El primero se refiere a algo ya apuntado antes: el carácter determinista de las cláusulas de los fines.

Vemos cómo el cuarto argumento, *Corte*, sólo es tenido en cuenta en las dos primeras cláusulas, las correspondientes a los fines internos y externos homogéneos. En estos casos, la presencia de alguna condición adicional a evaluar durante la construcción del área hace que se permita el *backtracking* si aquélla no se ve satisfecha. Esto no sucede con los fines externos heterogéneos, incondicionalmente expresados en una cláusula determinista. Ello obedece al carácter peculiar de estos fines. Como se recordará⁴⁹, un fin externo heterogéneo marca el final de un cierto área y la imposibilidad de que aparezca a continuación un área de la misma familia. Esto hace que pueda ser tratado de manera distinta a los fines internos y externos homogéneos al representar un grado más fuerte de final de área.

En cualquier caso, la opción de hacer determinista la cláusula de los fines externos heterogéneos es perfectamente revisable, pudiendo ser sustituida -si la experiencia demuestra su necesidad- por esta otra:

```
aserta_un_fin(Area, Fin, externo_hetero, Corte):-
    expand_term((Area --> meta_lendex(Fin, con, Ext, CI),
                apr_side(cat_eval(Fin, hetero, Area, Ext, CI)),
                Corte), Clausula),
    assertz(Clausula),
    !.
```

El segundo comentario tiene que ver con el uso del procedimiento antes mencionado: *apr_side/3*⁵⁰ en las cláusulas correspondientes a los fines externos.

⁴⁹Véase: *Otros conceptos relacionados* en la p.27

⁵⁰Recuérdese su definición de la p.104

Éste se utiliza para dejar como segmento inicial del resto un literal `cat_eval/5`, con el siguiente esquema:

Literal	<code>cat_eval(Fin, TF, A, Ext, Ci)</code>
Argumentos	<code>cat_eval(expresion, tipo_fin, nombre_area, lexemas, expresion)</code>
Significado	El fin de tipo TF del área A nombrado con la variable Fin, ha sido evaluado y tiene como datos asociados la extensión designada con Ext y el contenido informativo Ci.

Como entonces sugeríamos, dicho mecanismo puede ser de enorme utilidad en la optimización del sistema por la conservación de resultados parciales, ya computados, que efectúa. Éste es uno de los objetivos que pretende cumplir aquí.

Vemos cómo el resultado de aplicar el procedimiento `meta_lendex/6` es apuntado en forma de literal, `cat_eval/5`, que contiene entre sus argumentos -cuarto y quinto- los devueltos por él. Puede entenderse que el uso en esta ocasión de `apr_side/3` obedece a un principio elemental de economía de recursos consistente en no calcular más veces de las necesarias ciertos objetivos posiblemente costosos.

Sin embargo, este objetivo no es el único.

Aquí desempeña además una decisiva función al transmitir el conocimiento del tipo de fin que acaba de ser evaluado y el área a la que puso fin. Teniendo en cuenta que los fines externos heterogéneos imponen condiciones sobre las categorías que aparecen a continuación -impiden que sean de la misma familia del área cuyo fin establecen-, es necesario un mecanismo que permita decidir si una determinada categoría es posible o no. Y para esto es preciso disponer de este conocimiento en el momento de evaluarla.

El último comentario tiene que ver con el procedimiento `meta_lendex/6`, llamado en las tres cláusulas. Un sólo apunte aquí: `meta_lendex/6` es el procedimiento que calcula para una cierta categoría gramatical su extensión y contenido informativo.

De esta manera permite resolver el problema apuntado en la p.105 relativo a cómo invocar categorías gramaticales LENDEX no reconocibles sin más por un intérprete *standard* de DCGs y cómo calcular la extensión de un cierto fin externo a un área DD que haga posible su definición.

Más tarde estudiaremos en profundidad el significado e implementación de `meta_lendex/6`.⁵¹

El último procedimiento llamado por `aserta_gramatica/1` es `aserta_recursion/2`

Procedimiento	<code>aserta_recursion/2</code>
Argumentos	<code>aserta_recursion(nombre_area, rasgo).</code>
Direccionalidad	<code>in(+,+), out(+,+)</code>
Significado	<code>aserta_recursion(Area, Rasgo)</code> añade en último lugar a la base de conocimiento la cláusula correspondiente a la llamada recursiva de la gramática del área DD <code>Area</code> con el rasgo asociado <code>Rasgo</code>

```
aserta_recursion(Area, Rasgo):-
    expand_term((Area -->
                [X], { satisface_rasgo(Rasgo,X) },
                Area), Clausula),
    aserta(Clausula),
    !.
```

El procedimiento auxiliar `satisface_rasgo/2` impone la condición adicional sobre las áreas DD expresada en el segundo argumento de los hechos `area_dd/2`. Comprueba si el lexema que constituye la cabeza de la lista de entrada -al que se nombra con la variable `X`- satisface el rasgo del área declarado en la GDU: variable `Rasgo`. En caso positivo, el lexema `X` pasa a formar parte de la extensión del área. En caso negativo, la construcción del área se detiene.

⁵¹Véase *El procedimiento meta_lendex/6: un meta-intérpete de gramáticas*, p.118

Procedimiento	satisface_rasgo/2
Argumentos	satisface_rasgo(rasgo, string).
Direccionalidad	in(+,+), out(+,+)
Significado	satisface_rasgo(R, T) es verdadero si T es un lexema que satisface el rasgo R o un literal del tipo oculto(L), cuyo primer argumento L es una lista de lexemas con dicho rasgo. Esto es, caen bajo la clase de lexemas que él nombra: en_may, con_may, en_min, numeral, romano, true.

```
satisface_rasgo(true, _):- !.
```

```
satisface_rasgo(Rasgo, oculto(L):-
    !,
    los L son X^satisface_rasgo(Rasgo, X).
```

```
satisface_rasgo(_, Lex):-
    futil(Lex), !.
```

```
satisface_rasgo(Rasgo, Lex):-
    lex(Lex, Rasgo).
```

La definición de `satisface_rasgo/2` expresa la idea expuesta en la presentación de LENDEX en torno a las condiciones que hacen que un área satisfaga un cierto rasgo⁵²: todos sus lexemas no fútiles son de la clase formada por él.

La primera cláusula establece que todo lexema satisface el rasgo `true`, en la medida que éste representa la ausencia de rasgo alguno. La tercera permite incluir, independientemente del rasgo asociado, a los lexemas fútiles en la extensión de un área. La cuarta, por último, es la que comprueba si efectivamente un lexema cae bajo la clase que el rasgo define. El significado de `futil/1` y `lex/2` se encuentra más adelante en *Procedimientos de carácter general*, p.232.

⁵²Recordemos la extensión del conjunto RP en *Otros conceptos relacionados*, p.28

El sentido de la segunda cláusula, cuyo último argumento es el literal `oculto/1`, se clarificará con la presentación del procedimiento `meta_lendex/6`⁵³. Por el momento sólo destacar la aplicación del procedimiento general `son/2`, útil para comprobar que una cierta condición es satisfecha por todos los elementos de una lista. Su significado se encuentra detallado en el mismo apartado que los procedimientos `futil/1` y `lex/2`, antes citados.

Aplicación a un ejemplo real

Presentado el programa que genera automáticamente la gramática asociada a un área DD, es relevante mostrar el resultado de aplicarlo sobre un ejemplo real. Consideremos la GDU del Anexo 1 y veamos la gramática generada para las áreas DD enc y tit:

```
/* Gramática asociada al área DD enc */
enc([ ], [ ]) :-
    !.
enc(A, B) :-
    meta_lendex(fl, sin, C, A, B).

enc(A, B) :-
    meta_lendex(fa, sin, C, A, B).

enc(A, [cat_eval(isbn, B, enc, hetero) | C]) :-
    meta_lendex(isbn, con, B, A, C),
    !.
enc([A|B], C) :-
    satisface_rasgo(enc, en_may, A, D, E),
    enc(B, C).

/* Gramática asociada al área DD tit */

tit([ ], [ ]) :-
```

⁵³Véase la explicación de la cláusula de `meta_lendex/6`, p.125, donde se introduce el literal `oculto/1`

```
!.
tit(A, [cat_eval(df, B, tit, hetero)| C]) :-
    meta_lendex(df, con, B, A, C),
    !.
tit(A, B) :-
    meta_lendex(desde_d(3, punto), sin, C, A, B),
    !.
tit(A, B) :-
    meta_lendex(fa, sin, C, A, B),
    !.
tit(A, [cat_eval(isbn, B, tit, hetero)| C]) :-
    meta_lendex(isbn, con, B, A, C),
    !.
tit([A|B], C) :-
    satisface_rasgo(tit, true, A, D, E),
    tit(B, C).
```

Aserción de las cláusulas para las áreas ED

Como ya indicamos en la introducción a la presentación de S-Lendex, el tratamiento dado a las áreas DD es considerablemente distinto del recibido por las áreas ED. Esta diferencia se pone de manifiesto en primera instancia en la propia gramática del lenguaje, G_{Lendex} , donde se especifica el distinto modo de expresar unas y otras.

Mientras las áreas DD lo hacen en forma de hechos Prolog de aridad 2, las áreas ED se escriben en producciones gramaticales cuyo aspecto puede considerarse relativamente *standard*. Ello obedece al carácter infijo del predicado que las expresa: $\Rightarrow/2$, que separa una cabeza de la producción -el área a describir-, del cuerpo de la misma: aquéllo en términos de lo cual el área se describe.

Desde el punto de vista Prolog, ambas tienen la misma entidad. El conjunto de cláusulas correspondientes a las áreas DD define el procedimiento `area_dd/2`, variable de unas GDU a otras. Lo mismo cabe decir de las áreas ED respecto del predicado $\Rightarrow/2$.

Nada impediría que las segundas se expresaran así:

`area_ed(Area, Descripcion).`

en lugar de la forma elegida:

`Area => Descripcion.`

El interés de expresar diferenciadamente áreas DD y ED se sigue del distinto grado de elaboración que tiene la gramática asociada a unas y otras en la GDU. Esto es, la gramática de áreas ED se encuentra en un estado más parecido al que tendría si se expresara en forma de DCG. La gramática de las áreas DD, por el contrario, no

existe más que en la forma que los hechos `area_dd/2` y `fin/3` expresan. Precisan, así pues, del proceso automático de generación de gramáticas descrito en el apartado anterior.

Cabe plantear por qué no elegir una DCG *standard* para la expresión de las gramáticas de áreas ED. La razón es sencilla. Igual que las DCGs constituyen un edulcorante sintáctico respecto de Prolog, puede decirse que las producciones LENDEX de áreas ED lo son también respecto de las DCGs. En ellas aparecen términos específicos del lenguaje no comprensibles directamente por un intérprete Prolog, tales como los lexemas descritos, las secuencias, términos cuantificados, etc⁵⁴. Se trata, como ya se dijo antes, de facilitar en lo posible la expresión de la estructura de los documentos en un lenguaje de alto nivel.

Por tanto, es necesario un programa especializado que se ocupe de traducir adecuadamente esas expresiones y convertirlas en objetivos Prolog verificables.

Y es aquí donde caben distintas posibilidades en torno al momento de efectuar dicha traducción. La primera que vamos a exponer es la más directa y consiste en sostener las expresiones LENDEX hasta el final, de manera que éstas son traducidas siempre que se necesite. Es decir, lo que en términos de la jerga habitual de desarrollo de software recibe el nombre de intérprete. Otra, que expondremos más adelante, consiste en efectuar un proceso inicial de compilación de las gramáticas con el fin de no repetir operaciones posiblemente costosas. La alternativa es la misma discutida por [Pereira,90] y [Sarabia,92]. Una y otra soluciones ofrecen ventajas e inconvenientes que en cada caso discutiremos. Adelantando un poco la solución final adoptada, cabe decir que una estrategia mixta resulta satisfactoria en la medida que intenta aprovechar lo más positivo de

⁵⁴Véase: *La gramática de LENDEX* en las pp. 44, 46

ambas: la sencillez en la programación y depuración de los intérpretes y la eficacia de los compiladores.

En esta primera versión, el proceso de aserción de las cláusulas para las áreas ED pasa por reconsultar la GDU e incorporarla a la base de conocimiento general de S-Lendex, de manera que las producciones de áreas ED sean transparentes e interrogables en todo momento por el sistema.

Una vez hecho esto, un procedimiento general ya mencionado cuando explicamos la generación automática de las gramáticas de áreas DD, `meta_lendex/6`, interpreta las expresiones presentes en las producciones de áreas ED igual que lo hace con los fines de área DD. Este procedimiento asocia una extensión y un contenido informativo preciso a cada una de las áreas ED definida en la GDU así como distingue entre áreas R y RR.

Parece que se impone la necesidad de definirlo y discutirlo en detalle.

El procedimiento `meta_lendex/6`: un meta-intérpete de gramáticas

Precedentes

En [Sterling-Shapiro,86; p.303] encontramos la siguiente definición de meta-intérpete: *“Un meta-intérpete de un lenguaje es un intérpete del lenguaje escrito en el propio lenguaje”*.

Para los autores, un intérpete -se entiende, de programas lógicos- es un tipo especial de programa que partiendo de una interrogación Q -del inglés *query*- y un cierto programa P, devuelve Sí o No dependiendo de que Q sea o no deducible de P, [Sterling-Shapiro,86; p.12]

El ejemplo más simple de meta-intérprete que puede escribirse en Prolog es, siguiendo a los mismos autores:

$$\text{“ solve}(A) \leftarrow A. \text{”}$$

Una variante, levemente más compleja, de este mismo procedimiento es el denominado intérprete *vanilla*, [Hill-Lloyd,89; p.24]:

$$\begin{aligned} \text{“ solve}(\text{empty}) &\leftarrow \\ \text{solve}(x \ \& \ y) &\leftarrow \text{solve}(x) \wedge \text{solve}(y) \\ \text{solve}(x) &\leftarrow \text{clause}(x, y) \wedge \text{solve}(y) \text{”} \end{aligned}$$

Otros ejemplos más sofisticados de meta-intérprete los encontramos en [Bruffaerts-Henin,89; pp.183-184]: `prove_success/3` y `prove_failure/3`. Ambos procedimientos construyen el árbol de prueba de un cierto objetivo en caso de éxito y fallo, respectivamente. De manera semejante a como, en [Yalçinalp-Sterling,89; pp.191-203], el meta-intérprete `solve_goal/2` se ve progresivamente ampliado hasta conseguir unos resultados semejantes.

Pues bien, teniendo todo esto en cuenta, cabe decir que `meta_lendex/6` es -en una primera aproximación- un meta-intérprete escrito en Prolog de gramáticas LENDIX. A partir de una gramática de este tipo `meta_lendex/6` traduce las interrogaciones sobre una cierta secuencia de entrada a preguntas Prolog, y devuelve **SÍ** o **NO** dependiendo de que exista un segmento inicial de la secuencia a evaluar que satisfaga la categoría sobre la que se interroga; esto es, la pregunta resultante de la traducción sea o no deducible del programa asociado a la gramática.

Para ello `meta_lendex/6` ha de traducir tanto las expresiones gramaticales específicas de LENDIX como las *standard* DCG.

Por otra parte, `meta_lendex/6` asocia a cada categoría gramatical una extensión y un contenido informativo permitiendo que otro programa que expondremos más adelante, `analisis/4`, satisfaga el objetivo original de LENDEX: analizar documentos automáticamente a partir de una descripción general de la clase a la que pertenecen.

El punto de partida de `meta_lendex/6` -su precedente más inmediato- se encuentra en [Sarabia,93]. Allí encontramos un genuino meta-intérprete de DCGs, `meta/3`, que reproducimos:

```
/* Primer grupo de cláusulas de meta/3: expresiones DCG */

meta(_, E, _):-
    var(E),
    !, fail.

meta((A, B), E, S) :-
    !,
    meta(A, E, S1),
    meta(B, S1, S).

meta((A ; B), E, S) :-
    !,
    ( meta(A,E,S) ; meta(B,E,S) ).

meta(not A, E, S) :-
    !,
    not meta(A, E,S).

meta({A}, L, L):- !, call(A).

meta([ ], L, L):- !.

meta([X| B], L, R):-
    !,
    append([X|B], R, L).

/* Segundo grupo de cláusulas de meta/3: expresiones Prolog */

meta(fail, _, _):- !, fail.

meta(true, L, L):- !, true.
```

```
meta(A = B, L, L):-  
    !,  
    call(A = B).
```

```
meta(A == B, L, L):-  
    !,  
    call(A == B).
```

/ Cláusula que compone y ejecuta un objetivo Prolog a partir de una expresión DCG */*

```
meta(A, C, F):-  
    !,  
    A =.. [Fu| L],  
    append(L, [C, F], L1),  
    NewA =.. [Fu| L1],  
    !,  
    call(NewA).
```

En este meta-intérprete -cuyo autor indica que más bien podría llamarse `dcg_call/1`⁵⁵- encontramos traducción de las expresiones típicas de una DCG - primer grupo de cláusulas- más alguna otra habitual de Prolog -segundo grupo-.

No está tratado el corte: en el original aparece la cláusula correspondiente comentada:

```
/* meta(!, L, L):-!. */
```

por la misma razón que [Sterling-Shapiro,88; p.307] arguyen al presentar el meta-intérprete antes citado⁵⁶: *“Simular el comportamiento del corte correctamente es un problema. La solución ingenua es considerarlo un predicado de sistema. Esto es, añadir la siguiente cláusula:*

⁵⁵En la medida que es el equivalente para DCGs del `call/1` en Prolog.

⁵⁶Véase la definición de `solve/1` de la p.122

solve(!) ← !.

Esta cláusula no tiene el efecto requerido... El alcance del corte es excesivamente local."

Naturalmente, *meta_lendex/6* tampoco se sustrae a esta dificultad. Podríamos seguir un camino semejante al de [Cavaliere et al.,89] y desarrollar un intérprete que se ocupara de gestionar el corte mediante la consideración del contexto en que un objetivo se prueba -como hacen los autores en el procedimiento *solve/3*-. La idea original que subyace a S-Lendex, consistente en analizar documentos a partir de descripciones en un lenguaje de alto nivel, hace que prefiramos evitar el uso del corte a quien define una GDU y no suponer así un conocimiento preciso del mecanismo de prueba de Prolog y de su significado. Esta es la razón por la que muchos de los procedimientos definidos más adelante⁵⁷ ya previstos en *G_{Lendex}*⁵⁸ tienen tres versiones: determinista, semideterminista e indeterminista.

El meta-intérprete que a continuación presentamos sigue el modelo del procedimiento *meta/3*. Otra opción en el desarrollo del meta-intérprete habría sido la de escribir dos procedimientos distintos en lugar de uno, tal y como se indica en [Sarabia,92]. Allí el autor propone una versión anterior del mismo procedimiento: *meta/3*, en la que distingue una fase de traducción de la expresión DCG a su equivalente Prolog, -a través de un procedimiento al que denomina *tr/4*- y su posterior ejecución con el predefinido *call/1*:

meta(A) --> tr(A,C), {call(C)}.

⁵⁷*Procedimientos gramaticales para los términos cuantificados*, p.171

⁵⁸Véase: *GLendex* en la p.44

Es un camino similar al que encontramos en [Sterling-Shapiro,88; p.259] cuando presenta el procedimiento `translate/2` que traduce una regla gramatical DCG a cláusula Prolog, semejante a su vez al desarrollado por [Barklund,89; pp.390-391] con `dcg_expansion/2`.

Presentamos ahora `meta_lendex/6`, meta-intérprete de gramáticas LENDEX.

Implementación y desarrollo

El procedimiento `meta_lendex/6` tiene el siguiente esquema:

Procedimiento	<code>meta_lendex/6</code>
Argumentos	<code>meta_lendex(expresion, atomo, lista, termino, lista, lista)</code>
Direccionalidad	<code>in(+,+,-,+,-)</code> , <code>out(+,+,+,+,+)</code> <code>in(+,+,+,+,+)</code> , <code>out(+,+,+,+,+)</code>
Significado	<code>meta_lendex(Cat, M, Ext, Ci, E, S)</code> invoca a la categoría gramatical <code>Cat</code> y calcula cuando <code>M</code> vale 'si' su extensión <code>Ext</code> y contenido informativo <code>Ci</code> , siendo <code>E</code> la lista de lexemas para la cual se evalúa <code>Cat</code> y <code>S</code> el resto que queda de su evaluación.

Cláusula de anidamiento de meta-lendex/6

La primera cláusula del procedimiento trata el fenómeno de anidamiento de llamadas a `meta_lendex/6`. Dado que ciertos procedimientos meta-gramaticales⁵⁹ invocan a este procedimiento en su definición, es necesaria una cláusula en `meta_lendex/6` que los interprete adecuadamente. Su sentido es el de asimilar los

⁵⁹Como son los definidos en *Procedimientos gramaticales para los términos cuantificados*, p.171

valores recuperados, Ext y Ci, por la primera y la segunda llamadas a meta_lendex/6:

```
meta_lendex(meta_lendex(A, M, Ext, Ci), M, Ext, Ci, E, S):-  
    !,  
    meta_lendex(A, M, Ext, Ci, E, S).
```

Cláusula del literal cat_eval/5 a la cabeza de la lista a analizar

Las cuatro cláusulas que siguen constituyen el grupo de cláusulas cuya lista de entrada -quinto argumento- tiene como cabeza el literal cat_eval/5⁶⁰, resto complejo dejado por las gramáticas de áreas DD.

```
meta_lendex(Cat, _, Ext, Ci, [cat_eval(Cat, _, _, Ext, Ci)| E], E):-  
    !.
```

```
meta_lendex(Cat, _, _, _, [cat_eval(Otra, hetero, Cat, _, _)| _, _):-  
    Cat \= Otra  
    !,  
    fail.
```

```
meta_lendex(Cat, M, Ext, Ci, [cat_eval(Otra, homo, Cat, Sec,_)| E], S):-  
    Cat \= Otra,  
    !,  
    meta_lendex(Cat, M, Ext, Ci, [oculto(Sec)| E], S).
```

```
meta_lendex(Cat, M, Ext, Ci, [cat_eval(Otra, _, Previa, Sec,_)| E], S):-  
    Cat \= Otra,  
    Cat \= Previa,  
    !,  
    append(Sec, E, E1),  
    meta_lendex(Cat, M, Ext, Ci, E1, S).
```

La primera de las cuatro cláusulas presentadas cumple uno de los objetivos pretendidos al incorporar el literal cat_eval/5 al resto de la secuencia a analizar: la

⁶⁰Su esquema se encuentra en la p.111

optimización del sistema. Cuando la categoría a evaluar, *Cat*, es la misma que la evaluada inmediatamente antes -primer argumento de *cat_eval/5*- entonces la extensión, *Ext*, y el contenido informativo, *Ci* asociados a ésta pasan a ser los de la primera -*Cat*-, evitándose así un nuevo cálculo que rendiría el mismo resultado.

La segunda trata el resto dejado por los fines externos heterogéneos: impide, después de un fin de este tipo, la aparición de una categoría equivalente a aquella cuyo fin estableció. Si la categoría a evaluar -*Cat*- es la misma que evaluó la categoría inmediatamente antes invocada -*Otra*-, *meta_lendex/6* provoca un fallo no reintentable. El corte de la primera cláusula hace innecesaria la comprobación $Cat \neq Otra$ de la segunda; por su escaso coste y mayor claridad la hacemos explícita.

La tercera cláusula contempla el resto dejado por los fines externos homogéneos cuando la relación entre las categorías gramaticales involucradas, *Cat* y *Otra* es la misma que la prevista en la cláusula anterior. En este caso, se efectúa una llamada recursiva sobre *Cat* y se deja la extensión *Sec* asociada a la categoría antes evaluada, *Otra*, “camuflada” en la lista de entrada *E* mediante el literal *oculto/1*. Lo pretendido es evitar el bucle que se produciría si la misma categoría que dejó un resto evaluado fuera llamada a continuación con idéntica lista de entrada. Es preciso tener en cuenta que los fines externos homogéneos sí permiten -a diferencia de los heterogéneos- la aparición de una categoría de la misma familia que aquella de la que son fin. Cobra ahora sentido la cláusula del procedimiento *satisface_rasgo/2* correspondiente al literal *oculto/1*, p.113. Dado que la evaluación de *meta-lendex/6* puede dejar a su paso literales de este tipo, es preciso que el procedimiento que comprueba si un área satisface el rasgo asociado en su definición lo tenga en cuenta y pruebe que la secuencia oculta -todos sus lexemas- son del rasgo en cuestión.

La última cláusula representa la condición final para la aparición de literales del tipo *cat_eval/5*. En ella se ignora el resto dejado por el fin inmediatamente antes

evaluado, por no cumplirse ninguna de las situaciones previstas en las cláusulas anteriores. De ahí que se incorpore la extensión **Sec** a la lista de entrada **E** como segmento inicial y se llame recursivamente a `meta_lendex/6` sobre **Cat**.

Cláusulas de la conjunción, disyunción, negación, objetivos Prolog y terminales

Las seis cláusulas que siguen se corresponden con las que componen el primer grupo del procedimiento `meta/3` expuesto en la p.120. Como puede verse, el tratamiento dado al primer argumento -la categoría gramatical a evaluar- y a los dos últimos -listas de entrada y salida- es exactamente el mismo. La novedad radica en los tres argumentos característicos de S-Lendex -segundo, tercero y cuarto- que indican respectivamente si se requiere (**M/si**) o no (**M/no**) calcular la extensión **Ext** y el contenido informativo **Ci** asociados a la categoría a evaluar.

```
/* Cláusula de la conjunción */
```

```
meta_lendex((A, B), M, Ext, Ci, E, S):-  
    !,  
    meta_lendex(A, M, ExtA, CiA, E, S1),  
    meta_lendex(B, M, ExtB, CiB, S1, S),  
    flatten([ExtA| ExtB], Ext),  
    flatten([CiA| CiB], Ci).
```

```
/* Cláusula de la disyunción */
```

```
meta_lendex((A ; B), M, Ext, Ci, E, S):- !,  
    (  
        meta_lendex(A, M, Ext, Ci, E, S)  
        ;  
        meta_lendex(B, M, Ext, Ci, E, S)  
    ).
```

```
/* Cláusula de la negación */
```

```
meta_lendex(not A, M, Ext, Ci, E, S) :-  
    !,  
    not meta_lendex(A, M, Ext, Ci, E, S).
```

```
/* Cláusula de los objetivos Prolog llamados desde la gramática */
```

```
meta_lendex({A}, _, [], [], L, L):-
    !,
    call(A).
```

```
/* Cláusulas de los terminales: lista de lexemas */
```

```
meta_lendex([], _, [], [], L, L) :- !.
```

```
meta_lendex([X|Y], _, [X|Y], [X|Y], E, S):-
    !,
    append([X|Y], S, E).
```

Es relevante destacar cómo las tres primeras cláusulas reproducen las condiciones de verdad de la conjunción, la disyunción y la negación. La operación auxiliar de la primera, `flatten/2`, cumple la misión de allanar las listas devueltas por la doble llamada recursiva de `meta_lendex/6` sobre cada uno de los términos involucrados en la conjunción.

La cuarta cláusula prevé la llamada de objetivos Prolog desde una gramática `LENDEX`, para lo cual se sigue la misma convención que en las DCGs al usar las llaves rodeando el objetivo a cumplir. Naturalmente no hay extensión ni contenido informativo que asociar, por lo que el valor del segundo argumento se ignora y los del tercero y cuarto son la lista vacía.

Las dos últimas de este grupo son las correspondientes a los terminales, referidos en el modo habitual de las DCGs, como listas. La operación `append/3` comprueba si la lista indicada es un segmento inicial de la de entrada `E`.

Ciertamente estas dos cláusulas son unificables en una:

```
meta_lendex(L, _, L, L, E, S):-
    !,
```


append(L, S, E).

pero la eficacia que supone aprovechar el mecanismo interno de Prolog -la unificación- evitando que intenten satisfacer esta cláusula términos que no son listas, (cualquier término se unificaría en primera instancia con la variable L), hace que sea preferible la primera versión al explicitar los dos casos posibles de lista: la vacía -primera cláusula- y la no vacía -segunda-.

Cabría plantear una tercera posibilidad:

```
meta_lendex([], _, [], [], L, L) :- !.  
meta_lendex([X|Y], [X|Y], [X|Y], [X|E], S):-  
    !,  
    meta_lendex(Y, E, S).
```

La primera de las cláusulas no presenta diferencia alguna con la arriba propuesta. La novedad viene dada por la segunda de ellas. Esta solución tiene una ventaja y un inconveniente respecto de la anterior. La ventaja consiste en aprovechar un poco más el mecanismo de unificación de Prolog al pedir identidad de la cabeza de la lista a analizar -primer argumento- con la cabeza de la lista de entrada -quinto argumento-, lo que en general agiliza el proceso. El inconveniente estriba en la llamada recursiva que se produce por cada uno de los terminales de la lista, posiblemente muy larga. Ello tiene dos consecuencias de distinto carácter: un árbol de prueba de mayor profundidad -que se manifiesta en un coste adicional de memoria- y una depuración sin duda más lenta que la asociada al anterior par de cláusulas. Todo lo cual hace que éstas sigan siendo las preferidas para tratar el caso de llamadas a terminales en gramáticas LENDEX.

Es necesario notar también cómo la extensión y el contenido informativo de una lista L de terminales son iguales a la propia lista.

Cláusula de las áreas R

La cláusula que sigue se ocupa de distinguir las áreas R de las RR. Como G_{Lendex} estableció⁶¹, para indicar que un área es de sólo reconocimiento se utiliza el operador ? prefijo:

```
meta_lendex(? Cat, M, [inane(Ext)], [ ], E, S):-  
    !,  
    meta_lendex(Cat, M, Ext, _, E, S).
```

`meta_lendex/6`, al encontrarse con una expresión del tipo: ? Cat, hace una llamada recursiva sobre Cat asociándole un contenido informativo nulo, [], y una extensión compleja, `inane(Ext)`. El sentido de ésta última es conservar la información de la extensión de Cat al mismo tiempo que se indica su carácter irrelevante. Es lo pretendido al hacer de sólo reconocimiento un cierto área.

Cláusulas de expresiones específicas LENDEX

Las tres cláusulas que presentamos a continuación interpretan expresiones específicas de LENDEX: los lexemas descritos, las secuencias y los términos cuantificados. Todas ellas están definidas en G_{Lendex} en las páginas 46, 46, 44, respectivamente.

```
/* Cláusula para los lexemas descritos */  
  
meta_lendex(LxTerm, M, Ext, Ci, E, S):-  
    lexema_descrito(LxTerm, O),  
    !,  
    sustitucion_vlex(O, lexema/X, O2),  
    meta_lendex([X], {O2}), M, Ext, Ci, E, S).
```

⁶¹Véase la definición de la categoría `area_r` en la p.46

/ Cláusula para las secuencias */*

```
meta_lendex(Secuencia, _, SL, SL, L, R):-
    string(Secuencia),
    !,
    l_lexemas(Secuencia, SL),
    append(SL, R, L).
```

/ Cláusula de los términos cuantificados */*

```
meta_lendex(CntfTerm, _, Exts, Cis, E, S):-
    termino_cntf(CntfTerm, Area, Fu, Args),
    !,
    meta_lendex_cntf(Fu, Args, Area, Exts, Cis, E, S).
```

Las tres utilizan procedimientos no definidos que exigen explicación. Vamos en primer lugar con los llamados por la cláusula correspondiente a los lexemas descritos.

1. Lexemas descritos

El primero de ellos es el procedimiento `lexema_descrito/2`. Su esquema es el que sigue:

Procedimiento	<code>lexema_descrito/2</code>
Argumentos	<code>lexema_descrito(expresion, expresion)</code>
Direccionalidad	<code>in(+,-), out(+,+)</code> <code>in(+,+), out(+,+)</code>
Significado	<code>lexema_descrito(LD, O)</code> es verdadero si LD es un lexema descrito de acuerdo con la definición de G_{Lendex} y O es el objetivo asociado al lexema en términos de la expresión "lexema" y el procedimiento <code>lex/2</code>

```
lexema_descrito(lexema^O, O):- !.
```

```
lexema_descrito(X^O1, (lex(lexema, Rasgo), O2)):-
```

```

ix_terminal(X, Rasgo),
sustitucion_vlex(O1, X/lexema, O2),
!.

```

```

lexema_descrito(lexema, true):- !.

```

```

lexema_descrito(X, O):-
    ix_terminal(X, _),
    !,
    lexema_descrito(X^true, O).

```

Las cláusulas tercera y cuarta contemplan la aparición en una GDU de expresiones del tipo: *lexema*, *numeral*, *romano*, *lex_en_may*, *lex_con_may*, *lex_en_min*, tal y como prevé G_{Lexdex} al definir la categoría *lexema_tipo*⁶². El procedimiento trata éstas de manera uniforme con las contempladas por las dos primeras cláusulas al asociarlas una condición adicional *true* que satisfacen trivialmente y efectuar una llamada recursiva sobre la expresión compuesta. La definición del procedimiento *ix_terminal/2* viene exigida precisamente por la necesidad de especificar qué rasgo tiene asociado cada uno de los *lexemas* tipo que no son el genérico *lexema*.

Procedimiento	<i>ix_terminal /2</i>
Argumentos	<i>ix_terminal (lexema_tipo, rasgo)</i>
Direccionalidad	<i>in(+,-), out(+,+)</i> <i>in(?,?), out(+,+)</i>
Significado	<i>ix_terminal (L, R)</i> es verdadero si L es uno de los tipos posibles de <i>lexema</i> y R su rasgo asociado.

```

ix_terminal(numeral, numeral).
ix_terminal(romano, romano).
ix_terminal(lex_en_may, en_may).
ix_terminal(lex_con_may, con_may).
ix_terminal(lex_en_min, en_min).

```

⁶²En *La gramática de LENDEX*, p.46

Las dos primeras cláusulas de `lexema_descrito/2` contemplan la aparición de expresiones formadas por lexemas tipo con condiciones adicionales que satisfacer. En este caso se uniformiza la salida en términos de una expresión en la que sólo se usa `lexema` al sustituir por éste todas las apariciones de los lexemas tipo mediante `sustitución_vlex/3`, y anteponer a la condición adicional impuesta por el usuario la correspondiente a la satisfacción del rasgo asociado mediante `lex/2`⁶³.

Un ejemplo clarificará los dos tipos posibles de transformación que `lexema_descrito/2` lleva a cabo. Supongamos las expresiones:

- 1.1 `numeral^mayor(numeral, 20)`
- 2.1 `lex_en_may`
- 3.1 `lex_con_may^(length(lex_con_may, N), N < 5)`

Sus imágenes según `lexema_descrito/2` serán:

- 1.2 `(lex(lexema, numeral), mayor(lexema, 20))`
- 2.2 `(lex(lexema, en_may), true)`
- 3.2 `(lex(lexema, con_may), length(lexema, N), N < 5)`

Una vez construidas estas expresiones, `meta_lendex/6` efectúa la sustitución `lexema/X`⁶⁴ y compone los términos que siguen:

- 1.3 `[X], { (lex(X, numeral), mayor(X, 20)) }`

⁶³El significado del procedimiento `lex/2` se encuentra en *Procedimientos de carácter general*, en p.232

⁶⁴Cláusula correspondiente a los lexemas descritos, p.129

2.3 [X], { (lex(X, en_may), true) }

3.3 [X], { (lex(X, con_may), length(X, N), N < 5) }

evaluables por un intérprete *standard* de DCGs y, naturalmente, por las cláusulas correspondientes de `meta_lendex/6`⁶⁵.

2. Secuencias

La siguiente cláusula de `meta_lendex/6` que comentamos es la correspondiente a la aparición del término denominado en G_{Lendex} **secuencia**. Éste constituye una edulcoración sintáctica específica de LENDIX en la medida que permite aludir a los terminales de la gramática de una manera más cómoda que mediante los corchetes indicativos de lista. Pensemos en las dos expresiones siguientes:

`$¿ Es éste un terminal citado mediante el término secuencia ?$`

`[$¿$, $$, Es, $$, $éste$, $$, un, $$, $terminal$, $$, $citado$,
$mediante$, $$, el, $$, $término$, $$, $secuencia$, $$, $?$]`

Las dos son equivalentes desde el punto de vista gramatical. Obviamente, la primera es más cómoda al evitar al usuario conocer el modo en que el programa lexematiza y permitirle referirse a la secuencia que quiere mencionar tal y como aparece en el documento. Como contrapartida, ofrece el inconveniente de ser un poco más costosa desde un punto de vista procesual al obligar a S-Lendex a

⁶⁵Cláusulas de la conjunción: p.126, de los objetivos Prolog: p.127, y de los terminales: p.127

interpretarla para expresarla en términos de la segunda y poder tratarla de manera equivalente.

El procedimiento que efectúa esta transformación es `l_lexemas/2`. Este procedimiento genera una lista de lexemas a partir de una secuencia de texto. Es, naturalmente, el mismo que se ocupa de hacer del documento de entrada una lista de terminales -lexemas- analizable por la GDU⁶⁶.

3. Términos cuantificados

Después de la cláusula de la categoría **secuencia**, aparece la correspondiente a los términos cuantificados⁶⁷. Ejemplos de estos son las siguientes expresiones:

- `* punto`
- `+$R.$, numeral, futil_sp)`
- `&\(nombre_area1; (nombre_area2, *\ futil_sp))`
- `desde_d(10, espacio)`
- `hasta_i(3, guion_sp)`

En ellas aparecen operadores iterativos -tres primeros ejemplos- o cuantores numéricos -las dos últimas-. `meta_lendex/6` trata uno y otro caso de manera uniforme aplicando dos procedimientos auxiliares: `termino_cntf/4` y `meta_lendex_cntf/7`.

El primero de ellos, `termino_cntf/4`, identifica las expresiones que caen bajo la categoría `termino_cuantificado` y tiene el siguiente esquema:

⁶⁶Problema abordado más adelante, en la p.234

⁶⁷Definidos en `G_Lendex` en la p.44

Procedimiento	termino_cntf/4
Argumentos	termino_cntf(expresion, expresion, atomo, entero)
Direccionalidad	in(+, -, -, -), out(+, +, +, +) in(+, ?, ?, ?), out(+, +, +, +)
Significado	termino_cntf(Term, Exp, C, Args) es verdadero si Term es un término cuantificado aplicado sobre una expresión Exp formado por el operador iterativo o cuantor numérico C y los argumentos asociados Args.

```

termino_cntf(Term, Exp, C, Args):-
    functor(Term, C, Ari),
    cuantificador(C, Ari),
    arg(Ari, Term, Exp),
    !,
    ifthenelse(Ari=1, Args=[ ], (arg(1, Term, Arg), Args = [Arg])).
    
```

termino_cntf/4 establece si el functor principal C de la expresión a evaluar Term es uno de los cuantificadores previstos por G_{Lendex}^{68} y contemplados por el procedimiento auxiliar cuantificador/2.

Procedimiento	cuantificador/2
Argumentos	cuantificador(atomo, entero)
Direccionalidad	in(+,-), out(+,+) in(?,?), out(+,+)
Significado	cuantificador (C, A) es verdadero si C es uno de los operadores iterativos o cuantores numéricos definidos en G_{Lendex} y A es su aridad.

```

cuantificador(*, 1):- !.
cuantificador(*\, 1):- !.
cuantificador(*-, 1):- !.

cuantificador(&/, 1):- !.
cuantificador(&\, 1):- !.
cuantificador(&-, 1):- !.
    
```

⁶⁸Véase la p.44 donde están definidos.


```
cuantificador(+/, 1):-!.
cuantificador(+\, 1):-!.
cuantificador(+-, 1):-!.

cuantificador(desde_d, 2):-!.
cuantificador(desde_i, 2):-!.
cuantificador(desde_s, 2):-!.

cuantificador(hasta_d, 2):-!.
cuantificador(hasta_i, 2):-!.
cuantificador(hasta_s, 2):-!.

cuantificador(ex_d, 2).
cuantificador(ex_i, 2).

cuantificador(entre_d, 3):-!.
cuantificador(entre_i, 3):-!.
cuantificador(entre_s, 3):-!.
```

Una vez establecido **C** como uno de los cuantificadores LENDEX se determina sobre qué expresión **Exp** se aplica y los argumentos **Args** que la acompañan

Veamos el análisis que `termino_cntf/4` efectúa sobre dos de los ejemplos ya presentados en la p.134:

- 1.1 * punto
- 2.1 desde_d(10, espacio)

La aplicación de `termino_cntf/4` sobre cada uno de ellos dará el siguiente resultado:

- 2.1 termino_cntf(* punto, Exp, C, Args).
 - Exp = punto
 - C = *
 - Args = []
- 2.2 termino_cntf(desde_d(10, espacio), Exp, C, Args).

Exp = espacio

C = desde_d

Args = [10]

De esta manera se analizan los términos cuantificados, quedando listos para pasar como argumento al siguiente procedimiento `meta_lendex_cntf/7`.

Procedimiento	<code>meta_lendex_cntf/7</code>
Argumentos	<code>meta_lendex_cntf(cuantificador, lista, expresion, lista, termino, lista, lista)</code>
Direccionalidad	<code>in(+,+,+, -, -,+, -), out(+,+,+,+,+,+,+)</code> <code>in(+,+,+,?, ?,+, ?), out(+,+,+,+,+,+,+)</code>
Significado	<code>meta_lendex_cntf(Op, Args, E, Ext, Ci, I, O)</code> es verdadero si <code>Ext</code> es la extensión y <code>Ci</code> el contenido informativo asociado al término cuantificado que forman el functor <code>Op</code> , los argumentos indicados en <code>Args</code> y la expresión <code>E</code> ; aplicado todo ello sobre la lista de entrada <code>I</code> con un resto resultado del análisis <code>O</code> .

```
meta_lendex_cntf(Op, Args, E, Ext, Ci, I, O):-
    [!
    append(
        Args1,
        [(Ext1, C1)^meta_lendex(E, con, Ext1, C1), Lista, I, O],
        Args2
    ),
    Exp =.. [Op|Args2]
    !],
    call(Exp),
    ext_ci_aux(Lista, Ext, Ci),
```

Encontramos tres apartados bien diferenciados en el procedimiento:

1. la composición de la expresión `Exp` que va a ser evaluada: aplicación de `append/3` y el predefinido univ: `=../2`.
2. la llamada mediante `call/1` de dicha expresión

3. la generación a partir de la lista resultante *Lista*, producto del paso anterior, de la extensión *Ext* y el contenido informativo *Ci* mediante la llamada al procedimiento auxiliar *ext_ci_aux/3*.

Puede decirse que el paso decisivo es el primero. En él se compone la expresión que será evaluada después y que dará lugar a la extensión y el contenido informativo del término cuantificado analizado. Los ejemplos anteriores son nuevamente de interés para ilustrar el proceso. Se trata de analizar el tipo de expresión que *meta_lendex_cntf/7* compone y luego ejecuta:

3.1 $*((Ext, Ci)^{meta_lendex(punto, con, Ext, Ci)}, Lista, I, O)$

3.2 $desde_d(10, (Ext, Ci)^{meta_lendex(espacio, con, Ext, Ci)}, Lista, I, O)$

Estas expresiones constituyen la versión Prolog de procedimientos gramaticales específicos de LENDEX. Más tarde los estudiaremos en profundidad⁶⁹. Por el momento sólo resaltar lo que de común tienen: la presencia de las listas de entrada y salida como últimos argumentos: *I*, *O*, así como la expresión lambda y el argumento *Lista* que recoge los valores resultantes de la evaluación de la categoría: pares del tipo (*Ext*, *Ci*) -prefijo de la expresión lambda- donde *Ext* y *Ci* designan respectivamente la extensión y el contenido informativo de la categoría gramatical *E* -en este caso, punto y espacio-, evaluada por *meta_lendex/6* tantas veces como el cuantificador indica: cero o más en el primero, desde diez en el segundo.

La llamada mediante *call/1* de la expresión resultante -apartado segundo de *meta_lendex_cntf/7*- instancia la variable *Lista*; ésta es posteriormente

⁶⁹*Procedimientos gramaticales para los términos cuantificados*, p.171

transformada -último apartado del procedimiento- en otras dos: las que representan la extensión del término cuantificado a evaluar y su contenido informativo.

Cláusulas de áreas DD y ED

Las dos cláusulas que siguen contemplan la aparición de áreas ED y DD. Son las encargadas de construir un nodo en el árbol de análisis generado por `meta_lendex/6` en el argumento correspondiente al contenido informativo: cuarto argumento, así como de comprobar que las condiciones impuestas sobre las áreas en la GDU -mediante el hecho `condicion/3`- se cumplen.

```
meta_lendex(Area, M, Ext2, Ci2, E, S):-  
    (Area => Body),  
    !,  
    Ci =.. [Area, CiB],  
    meta_lendex(Body, M, Ext, CiBody, E, S),  
    flatten([CiBody], CiB),  
    satisface_condicion(Area, Ext, Ext2, Ci, Ci2).
```

```
meta_lendex(Area, M, Ext3, Ci3, E, S):-  
    area_dd(Area, _),  
    !,  
    Ci =.. [Area, Ext],  
    dcg_term(Area, A2, E, S),  
    call(A2),  
    segmento_anterior(E, S, Ext),  
    satisface_condicion(Area, Ext, Ext2, Ci, Ci2),  
    meta_lendex_dd(Area, Ext2, Ci2, Ext3, Ci3).
```

Ambas cláusulas comparten en primera instancia dos operaciones:

- la comprobación de que la categoría a evaluar es una de las áreas declaradas en la GDU, área ED en la primera cláusula, área DD en la segunda.
- la construcción del contenido informativo: Ci

La comprobación se efectúa interrogando a la base de conocimiento por los hechos característicos de una y otra área: una producción del tipo `Area => Body` para las áreas ED y un hecho del tipo `area_dd/2` para las DD.

El primitivo `univ, =./2`, construye el contenido informativo `Ci` al componer un literal de aridad 1 cuyo functor es -tanto para las áreas DD como para las ED- el nombre del área. En este punto, la diferencia estriba en el único argumento que el literal construido tiene: la extensión `Ext` para las áreas DD, el contenido informativo del cuerpo de la producción, `CiB`, para las ED.

Esta distinción marca una diferencia relevante entre áreas DD y ED: las áreas DD constituyen -si no hay análisis asociado mediante el hecho (->)/2- nodos terminales en el árbol de análisis final del documento; todas sus ramas terminan en hojas: los lexemas que forman su extensión. Por el contrario, las áreas ED pueden ramificarse a su vez en áreas de manera recursiva haciendo de este modo crecer el árbol en profundidad.

La complejidad de éste depende del número de áreas ED definidas en la GDU cuya descripción dependa a su vez de áreas ED.

Las dos cláusulas comparten además otra operación -última para las áreas ED, penúltima para las DD- :

- la comprobación de que el área evaluada, su extensión, satisface la condición impuesta en la GDU.

Esta comprobación se realiza mediante el procedimiento `satisface_condicion/5`, cuyo esquema e implementación se encuentran en el apartado siguiente,

Procedimientos usados por meta_lendex/6, p.145.

El análisis de estas dos cláusulas de *meta_lendex/6* pone de manifiesto la especificidad del tratamiento exigido por áreas DD y ED. Estas últimas precisan de una llamada recursiva de *meta_lendex/6* sobre el cuerpo de la producción **Body**, por la razón antes indicada: las áreas ED pueden tener como descripción todos los elementos que describen al documento, incluidas otras áreas ED. De ahí el cálculo de la extensión **Ext** y el contenido informativo **CiB** asociados a **Body** y su incorporación a la extensión y el contenido informativo del área: la propia **Ext** y el término **Ci** construido mediante el predefinido **univ** sobre el nombre del área y **CiB**.

Las áreas DD no requieren de una llamada recursiva de *meta_lendex/6*. Por estar ya asertadas en la base de conocimiento⁷⁰, el tratamiento por el que pasan es similar al recibido por las categorías gramaticales definidas por el usuario y las predefinidas **LENDEX**⁷¹:

- Construcción mediante *dcg_term/4* de un objetivo gramaticalmente evaluable formado por el nombre del área y dos argumentos adicionales: la lista de entrada **E** y la de salida **S**.
- Llamada mediante *call/1* de dicho objetivo.

Una vez instanciada la lista **S** de salida, *meta_lendex/6* calcula la extensión **Ext** del área DD mediante el procedimiento *segmento_anterior/3* a partir de la secuencia de entrada **E** y la de salida **S**.

⁷⁰Recordemos la fase de generación de las gramáticas de áreas DD, p.106

⁷¹Véase la cláusula siguiente y última de *meta_lendex/6* en la p.143

La última operación efectuada sobre las áreas DD consiste en el análisis flexible - caso de existir en la GDU la producción correspondiente- mediante el procedimiento `meta_lendex_dd/5`.

Procedimiento	<code>meta_lendex_dd/5</code>
Argumentos	<code>meta_lendex_dd(nombre_area_lendex, lista, termino, lista, termino)</code>
Direccionalidad	<code>in(+,+,+, -, -), out(+,+,+,+,+)</code> <code>in(+,+,+,?,?), out(+,+,+,+,+)</code>
Significado	<code>meta_lendex_dd(A, E1, Ci1, E2, Ci2)</code> es verdadero si E2 es la extensión y Ci2 el contenido informativo asociado al área DD A una vez efectuado -si exigido- el análisis de ésta mediante <code>hay_multiple/4</code> .

```

meta_lendex_dd(Area, Ext1, Ci1, Ext2, Ci2):-
    A -> L1,
    !,
    (S, ((E,C)^meta_lendex(S,con,E,C)))^true transforma_f L1 en L2,
    hay_multiple(L2, Pares, Ext1, _),
    ext_ci_aux(Pares, EE, CC),
    ifthenelse(
        CC \= [],
        (argrep(Ci1, 1, CC, Ci2), Ext2 = EE),
        (Ci2 = Ci1, Ext2 = Ext)
    ), !.

meta_lendex_dd(_, E, Ci, E, Ci).
    
```

Las operaciones clave del procedimiento son dos:

- Construcción a partir de la lista L1 -segundo argumento de los hechos LINDEX (->)/2 - de la lista de expresiones lambda L2
- Evaluación de L2 sobre la extensión previamente delimitada del área mediante el procedimiento `hay_multiple/4`

Dada la relación que este procedimiento guarda con otros expuestos más adelante⁷², dejamos para entonces su presentación⁷³.

Cláusula final

La última cláusula de `meta_lendex/6` contempla, finalmente, la aparición en la GDU de categorías gramaticales *standard*: definidas por el usuario o predefinidas LINDEX.

```
meta_lendex(A, M, L, L, E, S):-
    dcg_term(A, A2, E, S),
    !,
    call(A2),
    ifthenelse(M = con, segmento_anterior(E, S, L), L = []).
```

El procedimiento `dcg_term/4`, presente también en la cláusula anterior, cumple la misma finalidad: construir un objetivo Prolog evaluable a partir de un término gramatical mediante la incorporación de dos argumentos adicionales, correspondientes a las listas de entrada y salida.

Procedimiento	<code>dcg_term/4</code>
Argumentos	<code>dcg_term(termino, termino, lista, variable)</code>
Direccionalidad	<code>in(+, -, +, -)</code> , <code>out(+, +, +, -)</code> <code>in(+, +, +, -)</code> , <code>out(+, +, +, -)</code>
Significado	<code>dcg_term(T1, T2, E, S)</code> es verdadero si T2 es el término de aridad N+2 formado a partir del término T1 de aridad N por adición de los argumentos E y S.

⁷²Véase: *El procedimiento hay/6: alternativa a la evaluación Prolog de categorías gramaticales*, p.162

⁷³En: *Análisis flexible de una secuencia de lexemas: hay_multiple/5*, p.168


```
dcg_term(T1, T2, E, S):-  
    T1 =.. [Fu| L1],  
    append(L1, [E, S], L2),  
    T2 =.. [Fu| L2],  
    !.
```

Supongamos, por ilustrar su definición, dos categorías gramaticales: una predefinida LENDEx y otra definida por el usuario:

- 1.1 punto_sp
- 2.1 categoria_u(X)

El resultado de aplicar sobre ellas `dcg_term/4` es el siguiente:

- 2.1 `dcg_term(punto_sp, T, Entrada, Salida)`
 `T = punto_sp(Entrada, Salida)`
- 2.2 `dcg_term(categoria_u(X), T, Entrada, Salida)`
 `T = categoria_u(X, Entrada, Salida)`

Queda sólo por comentar la última línea de la cláusula, protagonizada por el primitivo `ifthenelse/3`. En ella adquiere valor el segundo argumento de `meta_lendex/6`, aquél que indica la necesidad o no de calcular la extensión del área expresada en el primer argumento. Dado que este cálculo es costoso⁷⁴ se pretende así evitar hacerlo siempre que no sea estrictamente necesario. Esto es lo que sucede, por ejemplo, en el caso de la llamada a `meta_lendex/6` presente en la cláusula de los fines internos a un área DD generada automáticamente por S-Lendex⁷⁵.

⁷⁴Recordemos la función del procedimiento `segmento_anterior/3`, encargado de dicho cálculo, en la p.141.

⁷⁵Véase la primera cláusula de `aserta_un_fin/4` en la p.109

Procedimientos usados por meta_lendex/6

1. sustitucion_vlex/3

Procedimiento	sustitucion_vlex/3
Argumentos	sustitucion_vlex/3
Direccionalidad	in(+, +, -), out(+, +, +) in(+, +, +), out(+, +, +)
Significado	sustitucion_vlex(E1, X/Y, E2) es verdadero si E2 es el resultado de sustituir en E1 X por Y.

```
sustitucion_vlex(X, Z/W, W):-
    X == Z,
    !.
```

```
sustitucion_vlex(X, _, X):-
    var(X),
    !.
```

```
sustitucion_vlex(T1, X, T2):-
    T1 =.. [Fu|Args1],
    vlex(Args1, X, Args2),
    T2 =.. [Fu|Args2],
    !.
```

```
vlex([ ], _, [ ]):- !.
```

```
vlex([X|Y], Z/W, [X1|Y1]):-
    sustitucion_vlex(X, Z/W, X1),
    !,
    vlex(Y, Z/W, Y1).
```

```
vlex([T|Y1], Z/W, [T|Y2]):-
    !,
    vlex(Y1, Z/W, Y2).
```

Este procedimiento tiene un interés que trasciende su uso en meta_lendex/6. De hecho, en S-Lendex se emplea también en otros contextos: véase a este respecto la definición de meta_dis/5, p.174. Su principal virtud es conservar libres las

variables del término sobre el que se efectúa la sustitución, razón por la cual es útil en metaprogramación.

2. *ext_ci_aux/3*

Procedimiento	<code>ext_ci_aux /3</code>
Argumentos	<code>ext_ci_aux/3</code>
Direccionalidad	<code>in(+, -, -), out(+, +, +)</code> <code>in(+, ?, ?), out(+, +, +)</code>
Significado	<code>ext_ci_aux(P, E, Ci)</code> es verdadero si E es la extensión y C el contenido informativo resultado de transformar la lista de pares P <Ei,Ci> en dos listas independientes

```

ext_ci_aux(Pares, Ext, Ci):-
    ((S,_)S)^true transforma Pares en Exts,
    flatten(Exts, Ext),
    ((_,S)S)^true transforma Pares en Ci,
    !.
    
```

2. *satisface_condicion/5*

Procedimiento	<code>satisface_condicion/5</code>
Argumentos	<code>satisface_condicion(nombre_area, expresion, expresion, expresion, expresion)</code>
Direccionalidad	<code>in(+, +, ?, +, ?), out(+, +, ?, +, ?)</code>
Significado	<code>satisface_condicion(A, E1, E2, Ci1, Ci2)</code> es verdadero si E2 es la extensión del área A y Ci2 su contenido informativo una vez comprobado que el área A -con E1 y Ci1 como extensión y contenido informativo provisionales- satisface o no la condición -caso de existir- impuesta en la GDU.

```

satisface_condicion(Area, Sec, SecA, Ci, Ci):-
    condicion(Area, Lambda, fuerte),
    segmento_complejo(syo, Sec, SecA),
    !,
    Lambda = SecA^G,
    call(G),
    !.
    
```

```
satisface_condicion(Area, Sec, SecA, Ci, Ci2):-
    condicion(Area, Lambda, debil),
    segmento_complejo(syo, Sec, SecA),
    Lambda = SecA^G,
    ifthenelse(
        call(G),
        Ci2 = Ci,
        (Ci2 =.. [inc,[Ci]]
    ), !.

satisface_condicion(_, Sec, SecA, Ci, Ci):-
    segmento_complejo(syo, Sec, SecA),
    !.
```

Importa señalar la necesidad del procedimiento auxiliar `segmento_complejo/3` para evaluar la condición. Es preciso recordar que la extensión del área, representada en una lista, puede tener entre sus elementos literales del tipo `inane/1`, `oculto/1`, `cat_eval/5` derivados de la aplicación de sucesivas cláusulas de `meta_lendex/6`. La evaluación de la condición exige, por tanto, una tarea previa que elimine o transforme -según el caso- esos literales y haga de la extensión una secuencia de lexemas evaluable por `satisface_condicion/5`.

Nótese el diferente trato que reciben las condiciones fuertes y débiles. Mientras las primeras constituyen un requisito *sine qua non* para el éxito del procedimiento -primera cláusula-, las segundas sólo comprueban si la condición es o no satisfecha y construyen -en caso negativo- el literal `inc/1`. Ésta constituye una de las tres incidencias posibles contempladas por S-Lendex⁷⁶.

⁷⁶Véase la definición de incidencia_lx en: *Formato de salida: GOutput*, p.77

3. segmento_complejo/3

Procedimiento	segmento_complejo/3
Argumentos	segmento_complejo(atomo, lista, lista)
Direccionalidad	in(+,+,?), out(+,+,+)
Significado	<p>segmento_complejo(Modo, SecIn, SecOut) transforma una lista compleja, SecIn, de strings y literales en otra, SecOut, de acuerdo con lo que indique Modo: syo/soei.</p> <p>syo: strings, ocultos y otros literales</p> <p>soei: strings, ocultos, inanes y otros literales</p>

```
segmento_complejo(Modo, SecIn, SecOut):-
    (S1, S2)^segm_aux(Modo, S1, S2) transforma SecIn en Sec,
    flatten(Sec, SecOut),
    !.
```

```
segm_aux(Modo, oculto(Lista), L2):-
    (S,S1)^segm_aux(Modo, S, S1) transforma Lista en L2,
    !.
```

```
segm_aux(syo, inane(_), _):-
    !, fail.
```

```
segm_aux(soei, inane(Lista), L2):-
    (S,S1)^segm_aux(soei, S, S1) transforma Lista en L2,
    !.
```

```
segm_aux(Modo, cat_eval(_, _, _, L1, _), L2):-
    segmento_complejo(Modo, L1, L2),
    !.
```

```
segm_aux(_, St, St):-
    string(St),
    !.
```

```
segm_aux(Modo, T1, T2):-
    T1 =.. [Area, L1],
    T2 =.. [Area, L2],
    segmento_complejo(Modo, L1, L2),
    !.
```

La clave del procedimiento está en el recorrido que *transforma/3*⁷⁷ hace por la secuencia de entrada *SecIn*, encontrando -según indique *Modo*- la imagen que el auxiliar *segm_aux/3* establece para generar la secuencia de salida *SecOut*.

4. *segmento_anterior/3*

Procedimiento	<i>segmento_anterior/3</i>
Argumentos	<i>segmento_anterior</i> (lista, lista, lista)
Direccionalidad	in(+, +, ?), out(+, +, +)
Significado	<i>segmento_anterior</i> (L1, L2, SA) es verdadero si SA es un segmento anterior de L1 que coincide con la extensión expresada por L2 en forma de lexemas o literales complejos del tipo <i>oculto/1</i> y <i>cat_eval/5</i> .

```

segmento_anterior(L1, [cat_eval(_, _, _, Sec, _)| L2], SA):-
    !,
    segmento_complejo(soei, Sec, SecA),
    append(SecA, L2, SecB),
    resta(L1, SecB, SA),
    !.

segmento_anterior(L1, L2, SA):-
    resta(L1, L2, SA),
    !.
    
```

También en este caso el procedimiento *segmento_anterior/3* debe hacerse cargo de la posible presencia en la lista de salida de literales complejos resultado de la evaluación de las áreas DD. De ahí la llamada a *segmento_complejo/3*. Además, otro procedimiento auxiliar sobre listas, *resta/3*, se hace necesario:

⁷⁷El significado de *transforma/2* se indica en *Procedimientos de carácter general*, p.232

4. *resta/3*

Procedimiento	<i>resta/3</i>
Argumentos	<i>resta(lista, lista, lista)</i>
Direccionalidad	<i>in(+,+,-), out(+,+,+)</i> <i>in(+,+,+), out(+,+,+)</i>
Significado	<i>resta(L1, L2, L3)</i> es verdadero si <i>L2</i> es un segmento final de la lista <i>L1</i> y <i>L3</i> es la lista que resulta de eliminar <i>L2</i> de <i>L1</i>

resta([], _, []) :- !.

resta(X, X, []) :- !.

resta([X|Y], Y, [X]) :- !.

resta([X|Z], Y, [X|W]) :-
 !,
 resta(Z, Y, W).

La razón de implementar *resta/3* es la eficacia. Un procedimiento bien conocido como *append/3* rinde el mismo resultado si instanciamos su segundo y tercer argumentos y dejamos libre el primero. La diferencia es el coste que esto entraña frente a la estrategia de *resta/3*, que explota en lo posible la unificación del resto de la lista *L1* con *L2*.

Análisis de los documentos

Una vez expuestos los procesos de generación automática de las gramáticas de áreas DD y aserción de las cláusulas correspondientes a las ED, conviene revisar de nuevo el gráfico de la p.66 en el que se presentaba un esquema general de S-Lendex. Nuestra atención se dirige ahora al apartado correspondiente al análisis de los documentos.

El objetivo de esta sección es el de exponer cómo se produce efectivamente dicho análisis, dada una GDU que los describe.

Aserción de la cláusula principal de la descripción

De la misma manera que en el caso de la aserción de las cláusulas de áreas ED, la aserción de la cláusula principal de la descripción pasa por un proceso inicial de reconsulta de la GDU que incorpora su producción principal -expresada en un hecho del tipo (\Rightarrow)/2- a la base de conocimiento de S-Lendex.

Dado que la producción principal de la GDU, del tipo:

documento \Rightarrow Descripción

puede estar formada por expresiones específicas LENDEX, no interpretables por un intérprete *standard* de DCGs, se hace necesario -igual que ocurría con las producciones de áreas ED- un programa especializado que las traduzca.

La discusión relativa al momento de hacer dicha traducción⁷⁸ es igualmente válida en este punto. La alternativa entre el desarrollo de un intérprete o de un compilador se plantea también ahora y su solución, naturalmente, está en relación con la adoptada entonces.

Una primera versión, la que estamos desarrollando, es la más directa: interpretar las expresiones LENDEX siempre que sea necesario. En el caso de la producción principal, esto pasa por incorporarla a la base de conocimiento tal y como se expresa en la GDU, sin tratamiento previo alguno. Una vez hecho esto, un programa especializado, *analisis/4*, se encarga de efectuar a partir de ella el análisis de los documentos.

El procedimiento analisis/4

El procedimiento *analisis/4* utiliza el meta-intérprete definido en la sección anterior, *meta_lendex/6*⁷⁹, para efectuar el análisis de la clase de documentos a partir de la descripción expresada en la producción principal de la GDU.

Cabe plantear entonces qué añade a *meta_lendex/6* que justifique su existencia; es decir, qué incorpora además de un intérprete de las expresiones LENDEX. Anticipándonos al desarrollo pormenorizado efectuado más adelante, cabe decir en pocas palabras que *analisis/4* representa una alternativa al mecanismo *standard* de evaluación Prolog de gramáticas DCG. Hace más tolerable al error, en el primer nivel de la descripción -producción principal de una GDU-, el análisis LENDEX de una clase de documentos.

⁷⁸Reordémosla en: *El procedimiento meta_lendex/6: un meta-intérpete de gramáticas*, p.118

⁷⁹Véase: *El procedimiento meta_lendex/6: un meta-intérpete de gramáticas*, p.118

Veamos su esquema e implementación.

Procedimiento	analisis/4
Argumentos	analisis(lista, lista, lexemas, lista)
Direccionalidad	in(+,?,+,?), out(+,+,+,+)
Significado	analisis(D, A, E, S) es verdadero si A es el resultado de analizar la lista de lexemas E mediante la descripción D y S es un segmento final de E no analizado.

```

analisis([], C, E, []):-
    segmento_complejo(soei, E, S),
    ifthenelse(
        algun S es X^(not futil(X)),
        C = ind(S),
        C = []
    ), !.

```

```

analisis([C| Rdesc], [X| Rci], E, O2):-
    meta_lendex(C, con, _, X, E, O),
    analisis(Rdesc, Rci, O, O2).

```

```

analisis(D1, D2, [], []):-
    (S, ign(S))^true transforma D1 en D2,
    !.

```

```

analisis([Ign, C | Rdesc], [ign(Ign), ind(A), Ci | Rci], E, O2):-
    hay_d(X^meta_lendex(C, con, _, X), Ci, Antes, E, O),
    segmento_complejo(soei, Antes, A),
    !,
    analisis(Rdesc, Rci, O, O2).

```

```

analisis([C | Rdesc], [ind(A), Ci | Rci], E, O2):-
    hay_d(X^meta_lendex(C, con, _, X), Ci, Antes, E, O),
    segmento_complejo(soei, Antes, A),
    !,
    analisis(Rdesc, Rci, O, O2).

```

```

analisis([Cat| Rdesc], [ign(Cat)| Rci], E, O):-
    !,
    analisis(Rdesc, Rci, E, O).

```

Junto al procedimiento *meta_lendex/6*, ya aludido, vemos también en la tercera y cuarta cláusulas de *analisis/4* un procedimiento conocido de la sección anterior: *segmento_complejo/3*.

La novedad se encuentra precisamente en la llamada que esas dos cláusulas efectúan a un procedimiento desconocido hasta el momento: *hay_d/5*. Por virtud de ésta tiene lugar la aportación fundamental que *analisis/4* hace al análisis de los documentos.

Aportación que tiene que ver, naturalmente, con la representación que la descripción de la clase de documentos recibe en la GDU. A diferencia de las áreas ED, expresadas en producciones *standard* cuyo operador es $(\Rightarrow)/2$, la producción principal de la GDU utiliza otro operador: $(\Rightarrow)/2$ y, lo que es más importante, una lista en su segundo argumento, aquél que expresa la descripción para la clase. Veremos ahora cómo esta representación permite a *analisis/4*, en relación con *hay_d/5*, efectuar el análisis de los documentos.

Discusión y justificación

El procedimiento de prueba de Prolog es *top-down*, en profundidad, de izquierda a derecha y con posibilidad de *backtracking* [Pereira-Shieber,87; p.65]. En la medida que las DCGs son una variante sintáctica de Prolog, tienen incorporado el mismo mecanismo prueba. No nos interesa en este momento discutir las implicaciones de carácter general que esto tiene. Sí nos importa destacar alguna de las consecuencias derivadas en el concreto ámbito del análisis gramatical efectuado por las DCGs.

Supongamos la siguiente secuencia de entrada a analizar:

“3 h., 475 p. ; 25 cm”

Esta puede ser la secuencia correspondiente al área de descripción física de un registro ISBD, del tipo del que ha servido de ejemplo en la p.22

Una DCG simple puede reconocerla⁸⁰:

descripcion_fisica	-->	hojas, signo, paginas, signo, dimensiones.
hojas	-->	[X], {numeral(X)}, [h.].
hojas	-->	[].
paginas	-->	[X], {numeral(X)}, [p.].
paginas	-->	[].
dimensiones	-->	[X], {numeral(X)}, [cm].
dimensiones	-->	[].
signo	-->	[.].
signo	-->	[;].
signo	-->	[].

Esta gramática -que declara opcionales todas las categorías involucradas- contempla un gran número de casos posibles de descripción física. En concreto, todos los que son de la forma de la secuencia de ejemplo o subsecuencias:

“14 h., 189 p. , 21 cm” ,

“65 p. ; 22 cm” ,

“3 h., 475 p.”

“5 h. ; 20 cm”

Sin embargo, hay otro buen número de casos que no se dejan tratar por esta gramática. Pensemos en los siguientes ejemplos:

1.1 “3 h., 475 p. ; 25 cm + 1 disco compacto”

⁸⁰Evitamos en ella los signos de puntuación reales por hacerla más legible.

- 2.1 "3 h., 475 p. : il ; 25 cm"
- 3.1 "3 h. en col., 475 p. ; 25 cm"
- 4.1 "VIII, 3 h., 475 p. ; 25 cm"

Vemos cómo en cada uno de los casos hay una única diferencia con el ejemplo anterior: una secuencia que aparece en éstos y no aparecía en el primero:

- 1.1' la secuencia relativa al material anejo: "+ 1 disco compacto"
- 2.1' la mención de ilustraciones: ": il"
- 3.1' la descripción adicional de las hojas: "en col."
- 4.1' la indicación de páginas numeradas en romanos: "VIII"

La diferencia entre ellos estriba en el punto de aparición de la secuencia desconocida: progresivamente antes conforme avanzan los ejemplos. La aplicación de la gramática antes propuesta sobre estas secuencias rinde los siguientes resultados:

2.1:

?- descripcion_fisica("3 h., 475 p. ; 25 cm + 1 disco compacto", Resto). Resto = "+ 1 disco compacto" yes

2.2:

?- descripcion_fisica("3 h., 475 p. : il ; 25 cm", Resto). Resto = ": il ; 25 cm" yes

2.3:

?- descripcion_fisica("3 h. en col., 475 p. ; 25 cm", Resto). Resto = "en col., 475 p. ; 25 cm" yes

2.4:

?- descripcion_fisica("VIII, 3 h., 475 p. ; 25 cm", Resto).

Resto = "VIII, 3 h., 475 p. ; 25 cm"

yes

El resultado de la aplicación de la gramática en los distintos casos es bastante elocuente: vemos cómo la gramática reconoce menos de la secuencia de entrada en la medida que el segmento desconocido aparece antes. El caso límite es el último; en él comprobamos cómo la aparición de un segmento no contemplado por la gramática al comienzo de la secuencia de entrada hace que la gramática fracase en su reconocimiento, pese a que una buena porción de la secuencia es perfectamente reconocible por ella. En el primer ejemplo, sin embargo, la aparición al final de la secuencia desconocida hace que la gramática sea operativa para la secuencia precedente.

Este distinto comportamiento obedece al mecanismo de prueba de Prolog antes aludido: en la medida que éste acontece de izquierda a derecha, un error se convierte en progresivamente más relevante cuanto más pronto aparece.

El objetivo pretendido al emplear el procedimiento *hay_d/5* en *analisis/4* consiste justamente en aliviar en lo posible esta dificultad: evitar que un fallo en la evaluación de una categoría al comienzo del análisis eche por tierra el resto de la descripción. Más tarde lo explicaremos en detalle⁸¹. Lo importante ahora es aclarar su función: saltar en la secuencia de entrada siempre que aparezca en la descripción una categoría desconocida o mal definida y hacer así posible que el análisis prosiga.

⁸¹En el apartado *El procedimiento hay/6: alternativa a la evaluación Prolog de categorías gramaticales*, p.162

De momento, hemos de clarificar el procedimiento `analisis/4`, generador del contenido informativo de los documentos a partir de una cierta GDU.

Explicación

Las cláusulas primera y tercera de `analisis/4` responden al planteamiento doblemente recursivo del procedimiento.

En la medida que éste consiste en una recursión sobre la lista de entrada -los lexemas del cuarto argumento- y otra sobre la que constituye la descripción de los documentos -segundo argumento de la producción principal de la GDU: `(==>)/2` y primer argumento de `analisis/4`- , se precisan dos cláusulas que establezcan la condición de parada de la recursión: el fin de la descripción, primera cláusula y el fin de la secuencia de entrada, tercera.

```
/* 1ª y 3ª cláusulas de analisis/4: fin de la recursión */
```

```
analisis([ ], C, E, [ ]):-  
    segmento_complejo(soei, E, S),  
    ifthenelse(  
        algun S es X^(not futil(X)),  
        C = ind(S),  
        C = [ ]  
    ),  
    !.  
  
analisis(D1, D2, [ ], [ ]):-  
    (C, ign(C))^true transforma D1 en D2,  
    !.
```

La primera de ellas responde al fin de la lista que expresa la descripción de la clase de documentos. En este caso, el resto de lexemas sin analizar, tercer argumento: `E`, queda indeterminado: es por esto transformado en el literal `ind(S)`, donde `S` es el resultado de aplicar sobre `E` el procedimiento `segmento_complejo/3` con el fin

de eliminar cualquier aparición de literales del tipo inane/1, oculto/1 o cat_eval/5, derivados de la aplicación de meta_lendex/6.

En la segunda, una vez agotada la lista de lexemas a analizar, todas las categorías gramaticales restantes de la descripción permanecen ignotas. De ahí la aplicación de transforma/2 sobre D1 construyendo para cada C el literal ign(C) en D2.

La segunda cláusula de analisis/4 constituye la llamada principal a meta_lendex/6:

```
/* 2ª cláusula de analisis/4 */  
analisis([C| Rdesc], [X| Rci], E, O2):-  
    meta_lendex(C, con, _, X, E, O),  
    analisis(Rdesc, Rci, O, O2).
```

Consiste simplemente en la atribución para la categoría gramatical C -cabeza de la descripción: primer argumento- de una imagen en el segundo coincidente con su contenido informativo. Una recursión de análisis/4 sobre el resto de la descripción Rdesc y sobre el resto de lexemas O dejado por la evaluación de meta_lendex/6 sobre la categoría C finalizan la cláusula.

Las dos que siguen incorporan la llamada al procedimiento hay_d/5 antes comentado:

```
/* 4ª y 5ª cláusulas de analisis/4 */  
analisis([Ign, C | Rdesc], [ign(Ign), ind(A), Ci | Rci], E, O2):-  
    hay_d(X^meta_lendex(C, con, _, X), Ci, Antes, E, O),  
    segmento_complejo(soei, Antes, A),  
    !,  
    analisis(Rdesc, Rci, O, O2).  
  
analisis([C | Rdesc], [ind(A), Ci | Rci], E, O2):-  
    hay_d(X^meta_lendex(C, con, _, X), Ci, Antes, E, O),
```

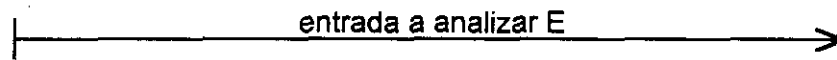

segmento_complejo(soei, Antes, A),
!,
 analisis(Rdesc, Rci, O, O2).

La primera de ellas responde al fallo de la cláusula inmediatamente anterior, tercera de analisis/4. En la medida que insatisfecha ésta, se declara ignota la categoría fallida, Ign, y se intenta evaluar la siguiente de la descripción: C mediante el procedimiento hay_d/5. Si se encuentra una subsecuencia de E que satisfaga C se declara indeterminado: ind(A) el segmento anterior, Antes, resultado de la evaluación de hay_d/5, una vez aplicado sobre él el procedimiento segmento_complejo/5 ⁸²; se incorpora el contenido informativo Ci de la categoría C al segundo argumento y se prosigue: última línea de la cláusula, llamada recursiva de analisis/4.

La hipótesis manejada en este caso es la de categoría desconocida o incorrectamente definida. Podría plantearse el siguiente esquema como representación de la situación prevista:

⁸²La razón es la misma que la comentada con ocasión de su uso en la segunda cláusula

Supuesta una categoría fallida Ign , y la siguiente en la descripción C ,



Ign , C ...

¿ hay una secuencia en E que satisfaga C ?

Si: aquella cuyo contenido informativo es C_i

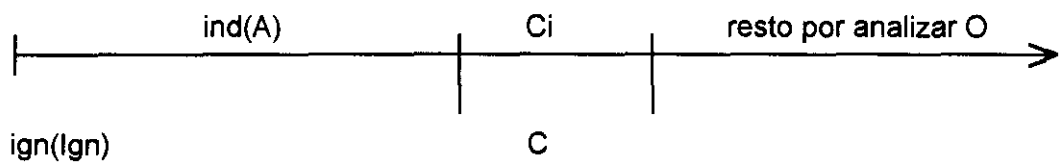


Figura 8: Significado del salto dado por `hay_d/5` en `analisi/4`

La quinta cláusula de `analisi/4` contempla un posible exceso de segmentación: no satisfecha la categoría para el segmento de entrada -tercera cláusula- ni encontrada la inmediatamente siguiente en la descripción -cuarta-, se explora la posibilidad de que la categoría a evaluar esté más adelante en la secuencia de entrada. En caso positivo, se incorpora al segundo argumento el segmento anterior indeterminado `ind(A)` después de la necesaria transformación de `Antes` mediante `segmento_complejo/3`.

La última cláusula de `analisi/4` declara `ign(Cat)` la categoría resistente a las cláusulas anteriores y prosigue.

```
/* Última cláusula de analisis/4 */
analisi( [Cat| Rdesc], [ign(Cat)| Rci], E, O):-
    !,
    analisis(Rdesc, Rci, E, O).
```

Procedimientos específicos de LENDEX

Una vez explicada la generación de la base de conocimientos Prolog para el análisis de los documentos, así como el procedimiento principal que efectúa éste, se plantea la necesidad de presentar los procedimientos auxiliares específicos de LENDEX que organizamos de la siguiente manera:

- El procedimiento `hay/6`: alternativa a la evaluación Prolog de categorías gramaticales
- Análisis flexible de una secuencia de lexemas: `hay_multiple/5`
- Procedimientos gramaticales empleados por S-Lendex para el tratamiento de términos cuantificados
- Otros procedimientos gramaticales
- Procedimientos generales empleados por S-Lendex

Importa señalar que los procedimientos a continuación presentados pertenecen a la base de conocimiento general de S-Lendex. Por ello, están a disposición del usuario que define una GDU para describir una clase de documentos.

El procedimiento `hay/6`: alternativa a la evaluación Prolog de categorías gramaticales

El procedimiento `hay_d/5`, usado en `analisis/4`, pertenece a una familia que tiene su origen en el procedimiento `gap/4`, [Sarabia,93]. Naturalmente emparentado con `meta/3`, antes expuesto⁸³, el propósito general de `gap/4` es resolver un problema

⁸³Véase la p.118 donde se indican los precedentes de `meta_lendex/6`

bien conocido en el ámbito del desarrollo de gramáticas lógicas: el de los constituyentes no inmediatos. Toda la tradición de las *Extraposition Grammars*, *Discontinuous Grammars*, etc⁸⁴, responde precisamente al intento de solucionar el problema que plantea la evaluación de categorías gramaticales que no se encuentran al comienzo de la lista de terminales a analizar sino en un punto más remoto. En definitiva, son un paso más en la línea de mejorar el mecanismo de prueba de las DCGs y su capacidad expresiva.

El procedimiento aquí empleado, *hay_d/5*, es la aplicación determinista de uno más general, *hay/6*, cuyo esquema e implementación son los que siguen:

Procedimiento	<i>hay/6</i>
Argumentos	<i>hay(atomo, lambda, termino, lista, lista, lista)</i>
Direccionalidad	<i>in(+,+, -, -,+,?)</i> , <i>out(+,+,+,+,+,+)</i> <i>in(+,+,?,?,+,?)</i> , <i>out(+,+,+,+,+,+)</i>
Significado	<i>hay(M, L, Q, Antes, E, Despues)</i> es verdadero si L es una expresion lambda del tipo X^AY , donde X es un término e Y una categoría gramatical que satisface -en el modo indicado por M- la secuencia E, dejando un resto anterior Antes y uno posterior Despues. Q es el resultado de instanciar X tras la evaluación de Y.

```
hay( _, _, _, _, [ ], [ ] ):-
    !, fail.
```

```
hay(Dis, X^Y, Que, [ ]) --> meta_dis(Dis, X^Y, Que),
    { ifthen(Dis = det, !) }.
```

```
hay(Dis, Lambda, X, [A| B], [A| D], E):-
    ifthen(Dis = det, !),
    hay(Dis, Lambda, X, B, D, E).
```

⁸⁴Véanse [Pereira-Shieber,87], [Abramson-Dahl,89], [Dahl,94], etc.

Este procedimiento, al igual que los presentados más adelante en *Procedimientos gramaticales para los términos cuantificados*, p.171, llama al procedimiento *meta_dis/5*⁸⁵, encargado de discriminar entre la evaluación determinista, semideterminista e indeterminista de una cierta categoría gramatical.

La versión aquí empleada, *hay_d/5*, es la que sigue:

$hay_d(X^AY, Q, A) \rightarrow hay(det, X^AY, Q, A).$

Su aplicación sobre alguno de los ejemplos expuesto al justificar su uso por *análisis/4* terminará de aclarar su significado. Supongamos la secuencia de entrada 1.1 y otra gramática:

descripcion_fisica_2 --> *paginas, [,], dimensiones.*
paginas --> *[X], {numeral(X)}, [p].*
dimensiones --> *[X], {numeral(X)}, [cm].*

Comprobamos que esta gramática contempla menos casos que la anterior y es más restrictiva: no declara opcionales las categorías involucradas. Veamos el resultado de aplicar esta nueva gramática sobre la secuencia:

"3 h., 475 p. ; 25 cm + 1 disco compacto"

El resultado es el que sigue:

?- <i>descripcion_fisica_2</i> ("3 h., 475 p. ; 25 cm + 1 disco compacto", Resto). no
--

⁸⁵Véase su definición de la p.174

Como cabía esperar, la gramática falla. La indefinición de la categoría *hojas* hace que la gramática no pueda reconocer una secuencia de texto que comienza así, al igual que sucedía en el ejemplo 4.1 antes visto. La distinta respuesta obtenida, *si/no*, depende del carácter opcional que la primera gramática tenía, ausente de ésta.

Apliquemos esta misma gramática a la secuencia pero con la mediación del procedimiento *hay_d/5*.

```
?- hay_d(X^descripcion_fisica_2, Que, Antes, "3 h., 475 p. ; 25 cm + 1
disco compacto", Resto).
```

```
Que = X
Antes = "3 h."
Resto = "1 disco compacto"
yes
```

Observamos un resultado considerablemente distinto. Desde el punto de vista del reconocimiento, la gramática ha tenido todo el éxito posible: ha encontrado en la secuencia de entrada un segmento que satisface la categoría evaluada *descripcion_fisica_2* dejando dos restos, anterior *Antes* y posterior *Resto* sin analizar. La variable *Que* permanece variable en la medida que la expresión lambda que debía instanciarla no lo ha hecho.

Veamos un caso en que esta variable sí toma un valor:

```
?- hay_d(X^meta_lindex(descripcion_fisica_2, con, X, _), Que, Antes, "3
h., 475 p. ; 25 cm + 1 disco compacto", Resto).
```

```
Que = "475 p. ; 25 cm"
Antes = "3 h."
Resto = "1 disco compacto"
yes
```

Aquí hacemos que la variable **Que** -que denota lo que el predicado **hay_d/5** dice que hay en la secuencia de entrada- se instancie con la extensión asociada a la categoría **descripcion_fisica_2**, dando el resultado esperado.

Podríamos también suponer que **descripcion_fisica_2** es una de las áreas definidas en una GDU⁸⁶ e interrogar de esta otra manera:

```
?- hay_d(X^meta_lendex(descripcion_fisica_2, con, _, X), Que, Antes, "3
h., 475 p. ; 25 cm + 1 disco compacto", Resto).

Que = descripcion_fisica_2([paginas("475 p."), signo(" ; "), dimensiones("25
cm")])
Antes = "3 h."
Resto = "1 disco compacto"

yes
```

El resultado sería naturalmente distinto. Al hacer que la variable **Que** se instancie con el contenido informativo de **descripcion_fisica_2**, obtendríamos el árbol de análisis correspondiente a esta categoría.

Este es precisamente el uso que hacemos del procedimiento **hay_d/5** en *análisis/4*⁸⁷. El objetivo es evitar que el fallo en la evaluación de una determinada categoría gramatical al comienzo de la descripción haga fracasar necesariamente ésta.

Puede plantearse por qué no usar la versión no determinista o semideterminista de la misma idea. Esto es, los procedimientos **hay_i/5** y **hay_s/5**, respectivamente:

$$\text{hay}_i(X^Y, Q, A) \quad \text{-->} \quad \text{hay}(\text{indet}, X^Y, Q, A).$$

⁸⁶Para lo cual sólo habría que sustituir el símbolo típico de las DCGs: **-->** por el específico **LENDEX: =>**

⁸⁷Véase la p.152 donde está definido.

hay_s(X^AY , Q, A) --> hay(semi, X^AY , Q, A).

El primero de ellos, hay_i/5, posibilita el *backtracking* tanto en lo que a la recursión como a la evaluación de la categoría se refiere. En hay_s/5 -versión semideterminista- se anula sólo el valor posiblemente indeterminista de la categoría evaluada, tolerándose, sin embargo, el *backtracking* sobre la recursión.

La razón por la que se ha optado por la versión determinista, hay_d/5, es de índole práctico y tiene que ver también con el carácter prioritariamente determinista del procedimiento principal analisis/4. Dado que las GDUs con las que el programa puede trabajar no están dadas de antemano, así como tampoco lo están los tipos de documento, nos parecía preferible garantizar unos resultados -quizás no los óptimos- en un gran número de casos que permitir un *backtracking* que fuera saludable en algunos pero mortífero en otros. Es preciso tener en cuenta que una excesiva longitud de los documentos de entrada o una gramática demasiado compleja son factores de riesgo no desdeñables en el diseño de una herramienta como S-Lendex que se propone tratar de manera general clases de documentos.

No descartamos, sin embargo, explorar en el futuro las implicaciones que una implementación menos determinista pueda tener, o idear algún tipo de estrategia general que permita establecer al usuario el grado de *backtracking* permisible. De hecho, éste es el planteamiento seguido en el caso de ciertos procedimientos como son los correspondientes a los términos cuantificados -estudiados más adelante⁸⁸- con tres versiones distintas: determinista, indeterminista, semideterminista.

Otro planteamiento posible y quizás más ambicioso es el de incorporar un mecanismo en S-Lendex que le permita autoexplorar las condiciones de trabajo -

⁸⁸En *Procedimientos gramaticales para los términos cuantificados*, p.171

clase de documentos a analizar, complejidad de la gramática- y establecer en función de ellas la estrategia más apropiada.

Análisis flexible de una secuencia de lexemas: hay_multiple/5

Al exponer meta_lendex/6 nos referimos a un procedimiento, hay_multiple/4, encargado de efectuar el análisis asociado a las áreas DD mediante hechos LENDEX del tipo Area -> Lista⁸⁹. Ahora que conocemos el significado de hay/6 estamos en disposición de presentarlo.

Estos son su esquema e implementación:

Procedimiento	hay_multiple/4
Argumentos	hay_multiple(lista, lista, lista, lista)
Direccionalidad	in(+, -, +, -), out(+,+,+,+) in(+,?,+,?), out(+,+,+,+,+)
Significado	hay_multiple(L1, L2, E, R) es verdadero si L1 es una lista de expresiones lambda del tipo X^Y, E es una lista de terminales sobre la que se evalúan sucesivamente las categorías Y de L1, y L2 contiene los valores resultantes de instanciar los términos X de L1 si al menos un segmento de E satisface Y, o bien, los literales formados por el functor ign y un único argumento coincidente con la expresión X^Y correspondiente. R contiene la concatenación de los segmentos de E que no han satisfecho ninguna de las categorías expresadas en L1

hay_multiple([], [], E, E):- !

hay_multiple(_, _, [], []):- !.

hay_multiple([L1|RL], [Q1|RQ], Entrada, R):-
hay_d(L1, Q1, Antes, Entrada, R1),

⁸⁹Recordemos la p.143 , donde se encuentra esta mención

```
append(Antes, R1, NR),  
!,  
hay_multiple(RL, RQ, NR, R).
```

```
hay_multiple([L1|RL], [ign(L1)|Que], Entrada, R):-  
    hay_multiple(RL, Que, Entrada, R).
```

Las dos primeras cláusulas de `hay_multiple/4` responden a las dos condiciones posibles de fin del procedimiento: el final de la secuencia a analizar -tercer argumento-, y el final de la lista que contiene las categorías lambda evaluables sobre ésta -primer argumento-.

El sentido del procedimiento es lograr un mecanismo general de aplicación de `hay_d/5` sobre una cierta secuencia cuya estructura no está lo suficientemente definida como para admitir la evaluación ordinaria de categorías gramaticales. Sí cabe, sin embargo, prever en ella ciertos subsegmentos relativamente bien definidos y analizables con independencia del punto de la secuencia en que aparecen.

Es relevante a este respecto referirnos a uno de los ejemplos desarrollados en el apartado *Aplicaciones*: el de los anuncios de prensa, p.261, problema por otra parte clásico en el ámbito del procesamiento automático del lenguaje natural, [Courant-Robin,85]. Tomemos un par de ejemplos para fijar aquí las ideas:

```
$CANILLEJAS, San Hilario, cuarto,  
exterior, 120 metros. cuatro dormito-  
rios, dos baños, agua y calefacción  
central, ascensores, pisazo.  
23.000.000. T. 91/4305412.$
```

```
$LISTA, dos dormitorios, 8.500.000.  
tercera planta, soleado.  
T. 91/5753388.$
```

Estos consisten en anuncios de venta de pisos. Sabemos que el texto que en ellos aparece no cumple una gramática estricta: observamos -por concretar- dos sucesos distintos:

1. La posición relativa de las indicaciones de piso y dormitorios no es la misma en un anuncio y otro: en el primero el número de dormitorios aparece después de la mención de planta, mientras que en el segundo aparece antes
2. Las indicaciones de precio y teléfono son en un caso consecutivas y en el otro no

En la medida que cada anunciante es libre de ordenar la información casi absolutamente, una estrategia *standard* de evaluación de categorías gramaticales está condenada al fracaso o bien a una explosión de casos contemplados. El procedimiento *hay_multiple/4*, en la medida que generaliza el uso de *hay_d/5* sobre la secuencia a analizar, permite buscar en ella de manera flexible los subsegmentos que satisfacen las categorías indicadas, independientemente de su lugar de aparición.

Al evaluar cada una de las categorías mediante *hay_d/5*, se dan dos posibilidades, contempladas respectivamente en las cláusulas tercera y cuarta del procedimiento:

1. Existe un segmento que la satisface; en tal caso se apunta el valor recuperado *Q1*, se renueva la lista a analizar concatenando los restos anterior y posterior - llamada a *append/3* -, y se prosigue: recursión de *hay_multiple/4* sobre la nueva lista a analizar *NR*
2. No existe subsegmento que satisfaga la categoría: se apunta el suceso -categoría ignota *ign(L1)* - y se continúa con una recursión sobre la misma lista de entrada

Gracias a este modo no *standard* de evaluación gramatical se consigue minimizar la relevancia del orden de aparición de los distintos segmentos significativos del texto, encontrándose todas las subsecuencias posibles que satisfacen alguna de las

categorías indicadas en el primer argumento; en los ejemplos anteriores: planta, dormitorios, precio y teléfono⁹⁰.

Dos comentarios finales. El primero de ellos es relativo al uso de `hay_d/5` en vez de sus equivalentes semideterminista e indeterminista `hay_s/5` y `hay_i/5`. La razón es la misma que la apuntada en la p.167 al explicar su uso en el procedimiento `analisis/4`: parece más conveniente asegurar el éxito del análisis en el mayor número posible de casos que permitir un grado excesivo de *backtracking* que lo ponga en peligro en situaciones problemáticas: secuencias de análisis muy largas, gramáticas complejas.

El segundo comentario se relaciona con el orden de la lista que establece el análisis asociado a un área DD: hechos del tipo `A -> L`. En la medida que `L` se procesa comenzando por el principio, es relevante al definir la GDU disponer las categorías atendiendo al mayor grado de seguridad que éstas nos ofrezcan. Esto es, las de éxito más claro es conveniente que aparezcan antes que las demás en la medida que así se optimiza su evaluación: la lista a analizar se reduce y son menos los casos a contemplar para las categorías peor establecidas. En caso de semejanza en la confianza de éxito de todas las categorías involucradas, el criterio para su disposición puede ser el orden final de salida: éste coincidirá con el impuesto por el usuario en la producción correspondiente de la GDU.

Procedimientos gramaticales para los términos cuantificados

La familia de procedimientos presentada en este apartado completa la explicación del meta-intérprete de gramáticas LENDEX `meta_lendex/6`. Directamente

⁹⁰Véase la GDU que analiza esta clase en la p.261 y el Anexo 3, p.318, con el resultado de su aplicación

relacionados con el procedimiento *meta_lendex_crtf/7*⁹¹, estos procedimientos hacen posible la evaluación de categorías gramaticales consistentes en lo que en *G_{Lendex}*, p.44, denominamos términos cuantificados, formados por un operador iterativo o un cuantor numérico aplicado sobre un cierto descriptor gramatical.

Todos ellos comparten el siguiente rasgo: expresan la idea consistente en que dada una secuencia a analizar, la categoría o descriptor gramatical sobre la que se aplican puede repetirse tantas veces como indica el cuantificador: uno o más, cero o una, entre dos y tres, menos de cuatro, etc.

Definimos para cada uno de ellos un procedimiento general que contempla tres aplicaciones posibles del mismo: determinista, indeterminista, semideterminista, cada una de las cuales tiene a su vez asociado un procedimiento específico que lo computa.

Importa señalar el parentesco de estos procedimientos con lo que presentaremos más adelante⁹² así como su diferencia. Los incluidos en esta sección son -como ya se ha dicho- los empleados por el meta-intérprete para evaluar los términos cuantificados. Requieren para ello recuperar la extensión y el contenido informativo asociados a la categoría sobre la que se aplican. Los que expondremos más adelante se limitan a reconocerla tantas veces como el cuantificador indica.

Es común a ambos la relación de cada uno de los procedimientos generales con sus aplicaciones específicas: determinista, semideterminista e indeterminista.

⁹¹Véase la p.138 donde son aludidos

⁹²En *Otros procedimientos gramaticales*, p. 191

El esquema general es el expuesto a continuación:

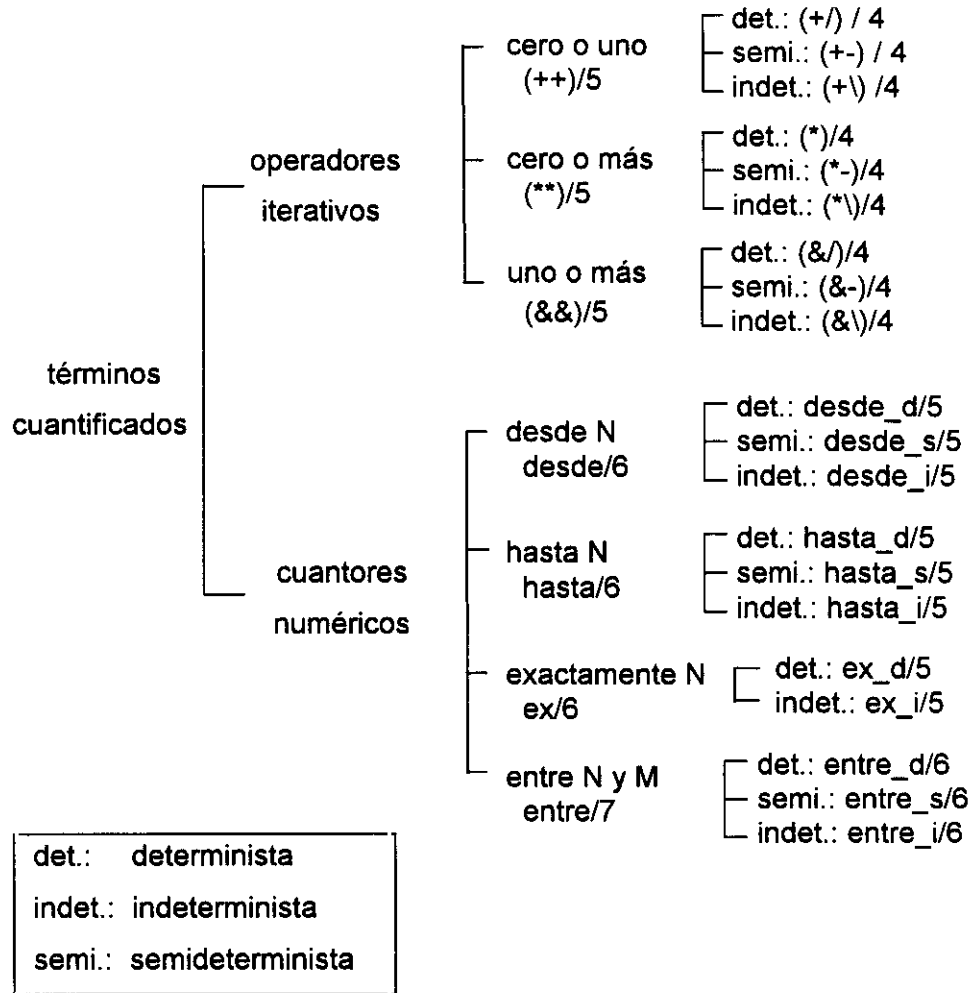


Figura 9: Cuantificadores LENDIX de reconocimiento y recuperación

Los deterministas admiten una única solución para la categoría sobre la que se aplican. Los semideterministas permiten el *backtracking* sobre la cuantificación pero no sobre la categoría: el carácter indeterminista que ésta pueda tener es anulado por el procedimiento específico LENDEX: @/1. Finalmente, los procedimientos indeterministas no limitan en modo alguno el *backtracking* permitiendo explorar soluciones alternativas tanto relativas a la cuantificación como a la categoría.

Los procedimientos generales (++)/5, (**)/5, (&)/5, desde/6, hasta/6, ex/6, entre/7 comparten un procedimiento auxiliar meta_dis/5 cuyo esquema e implementación indicamos:

Procedimiento	meta_dis / 5
Argumentos	meta_dis(atomo, lambda, expresion, lista, lista)
Direccionalidad	in(+,+, -,+, -), out(+,+,+,+,+) in(+,+,?,+,?), out(+,+,+,+,+)
Significado	meta_dis(Dis, L, Z, E, S) es verdadero si L es una expresión lambda del tipo X^Y y Z es el resultado de evaluar Y mediante meta_lendex/6 tras la sustitución en L de X por Z. La evaluación de Y sobre la lista de entrada E se efectúa en el modo que Dis indica dejando un resto S. det: determinista semi: semideterminista indet: indeterminista

meta_dis(semi, X^Y, Z) --> {sustitucion_vlex(X^Y, X/Z, Z^Cat)},
@ meta_lendex(Cat, con, _, _).

meta_dis(Dis, X^Y, Z) --> {
Dis \= semi,
sustitucion_vlex(X^Y, X/Z, Z^Cat)
},
meta_lendex(Cat, con, _, _).

Este procedimiento emplea uno conocido de secciones anteriores: `sustitucion_vlex/3`⁹³ que cumple aquí la misión de conservar no instanciadas las variables involucradas en la expresión lambda y permitir así la correcta evaluación de la categoría gramatical a la que ésta se refiere.

Por otra parte, el procedimiento discrimina entre la aplicación semideterminista de los cuantificadores -primera cláusula- y cualquier otra -segunda-. En el primer caso, se hace uso de un procedimiento específico `LENDEX, @/1`, que asegura la evaluación determinista de la categoría gramatical `C` sobre la que se aplica. Este procedimiento es equivalente al primitivo de Arity Prolog denominado `snips`: `[! !]/1` que anula el carácter posiblemente indeterminista de la categoría `C` permitiendo una única solución para `C`. El interés de la definición de `@/1` consiste en la independencia que otorga a `S-Lendex` de su implementación en Arity.

El procedimiento `@/1` está inspirado en el propuesto por [Covington, p.300] `once/1`, útil para objetivos Prolog *standard* y como el propio autor indica equivalente a los `snips`.

`once(G):- G, !.`

`@/1` representa la versión gramatical del mismo procedimiento. Su esquema y primera implementación son los siguientes:

Procedimiento	<code>@/3</code>
Argumentos	<code>'@'</code> (atomo, expresion, expresion, lista, lista)
Direccionalidad	<code>in(+,+,?)</code> , <code>out(+,+,+)</code>
Significado	<code>'@'</code> (Cat, E, S) es verdadero si la categoría gramatical <code>Cat</code> satisface la lista de entrada <code>E</code> dejando un resto <code>S</code> .

⁹³Véase la p.145 donde está definido


```
'@(C, E, S):-  
    dcg_term(C, CE, E, S),  
    call(CE),  
    !.
```

El procedimiento `@/3` construye mediante el ya conocido `dcg_term/4`⁹⁴ un objetivo Prolog evaluable a partir de la categoría gramatical `C` y las listas de entrada `E` y salida `S`. Evalúa a continuación dicho objetivo y finaliza aplicando un corte, impidiendo así cualquier intento de *backtracking* sobre `C`.

Si queremos extender el uso de `@/3` a cualquier expresión gramatical LINDEX, la implementación debe ser esta otra:

```
@ C    -->    meta_lindex(C, sin, _, _),  
            !.
```

Organizamos la explicación de los cuantificadores LINDEX en los siguientes apartados:

- Procedimientos relativos a términos cuantificados mediante un operador iterativo
- Procedimientos relativos a términos cuantificados mediante un cuantor numérico
- Aplicación sobre un ejemplo concreto

⁹⁴Véase la p.143, donde está definido.

Procedimientos relativos a términos cuantificados mediante un operador iterativo

El origen de estos procedimientos se encuentra en el tratamiento que expresiones DCG del mismo tipo reciben en [Sarabia,92 y 93] por medio del procedimiento `meta/3` antes aludido⁹⁵. Dos son aquí las novedades:

- el tratamiento de expresiones específicas LENDEX mediante el procedimiento `meta_lendex/6`
- la distinción ante ya apuntada entre cuantificadores deterministas, indeterministas y semideterministas

Siguiendo al mismo autor, establecemos tres tipos de operadores:

1. cero o uno
2. cero o más
3. uno o más

Les asociamos respectivamente los procedimientos `'++'/5`, `(**)/5` y `(&&)/5` que dan lugar a los específicos `(+)/4`, `(+ -)/4`, `(+ \)/4`, `(*)/4`, `(* -)/4`, `(* \)/4`, `(&)/4`, `(& -)/4`, `(& \)/4` representados en el esquema anterior, p.173

⁹⁵En p.120, al hablar de los precedentes de `meta_lendex/6`

1. Cero o uno

Procedimiento	(++) / 5
Argumentos	'++'(atomo, lambda, lista, lista, lista)
Direccionalidad	in(+, +, -, +, -), out(+, +, +, +, +) in(+, +, ?, +, ?), out(+, +, +, +, +)
Significado	'++'(M, L, Lista, E, S) es verdadero si L es una expresión lambda del tipo X^{Cat} y Lista contiene el resultado de instanciar X con la evaluación de Cat por meta_dis/5 cero o una veces en el modo indicado por M sobre la lista de entrada E y dejando un resto S. Valores de M: det: determinista semi: semideterminista indet: indeterminista

'++'(Dis, X^Y , [Z]) --> meta_dis(Dis, X^Y , Z),
{ ifthen(Dis = det, !)}.

'++'(_, _, []) --> [].

1. Aplicación determinista de (++)/5:

'+'(X^Y , L) --> '++'(det, X^Y , L).

2. Aplicación semideterminista de (++)/5:

'+'(X^Y , L) --> '++'(semi, X^Y , L).

3. Aplicación indeterminista de (++)/5:

'+'(X^Y , L) --> '++'(indet, X^Y , L).

2. Cero o más

Procedimiento	(**) / 5
Argumentos	***(atomo, lambda, lista, lista, lista)
Direccionalidad	in(+, +, -, +, -), out(+, +, +, +, +) in(+, +, ?, +, ?), out(+, +, +, +, +)
Significado	*** (M, L, Lista, E, S) es verdadero si L es una expresión lambda del tipo X^Cat y Lista contiene el resultado de instanciar X con la evaluación de Cat por meta_dis/5 cero o más veces en el modo indicado por M sobre la lista de entrada E y dejando un resto S. Valores de M: det, semi, indet.

***(_ , _ , [], [], []):- !.

*** (Dis, X^Y, [Z | Rz]) --> meta_dis(Dis, X^Y, Z),
{ ifthen(Dis = det, !) },
*** (Dis, X^Y, Rz).

***(_ , _ , []) --> [].

1. Aplicación determinista de (**)/5:

** (X^Y, L) --> *** (det, X^Y, L).

2. Aplicación semideterminista de (**)/5:

*- (X^Y, L) --> *** (semi, X^Y, L).

3. Aplicación indeterminista de (**)/5:

*\ (X^Y, L) --> *** (indet, X^Y, L).

Procedimientos relativos a términos cuantificados mediante un cuantor numérico

Recuperamos de nuevo el esquema de la p.173. Presentamos a continuación los procedimientos asociados al segundo grupo de cuantificadores LENDEX: aquéllos formados por un cuantificador numérico.

Distinguimos cuatro generales:

1. desde N
2. hasta N
3. entre N y M
4. exactamente N

Definimos para cada uno de ellos, como en el caso de los operadores iterativos, un procedimiento general: *desde/6*, *hasta/6*, *ex/6*, *entre/7* con tres aplicaciones posibles: determinista, semideterminista, indeterminista.

1. Desde N

Procedimiento	desde / 6
Argumentos	desde(atomo, entero, lambda, lista, lista, lista)
Direccionalidad	in(+,+,+, -,+, -), out(+,+,+,+,+) in(+,+,+,?,+,?), out(+,+,+,+,+)
Significado	desde(M, N, L, Lista, E, S) es verdadero si L es una expresión lambda del tipo X^Cat y Lista contiene el resultado de instanciar X con la evaluación de Cat por meta_dis/5 tantas veces como sea posible con un mínimo de N en el modo indicado por M sobre la lista de entrada E y dejando un resto S. Valores para M: det, semi, indet

```

desde(_, N, _, [], [], []):-          ifthenelse(N <= 0, !, (!, fail)).

desde(Dis, N, X^Y, [ Z | Rz ]) -->    meta_dis(Dis, X^Y, Z),
                                       { ifthen(Dis = det, !), dec(N, N1) },
                                       desde(Dis, N1, X^Y, Rz).

desde(_, N, _, []) -->                {N <= 0}.
    
```

1. Aplicación determinista de desde/6:

```
desde_d(N, X^Y, L) --> desde(det, N, X^Y, L).
```

2. Aplicación semideterminista de desde/6:

```
desde_s(N, X^Y, L) --> desde(semi, X^Y, L).
```

3. Aplicación indeterminista de desde/6:

```
desde_i(N, X^Y, L) --> desde(indet, X^Y, L).
```

2. Hasta N

Procedimiento	hasta/ 6
Argumentos	hasta(atomo, entero, lambda, lista, lista, lista)
Direccionalidad	in(+,+,+, -,+, -), out(+,+,+,+,+,+) in(+,+,+,?,+,?), out(+,+,+,+,+,+)
Significado	hasta(M, N, L, Lista, E, S) es verdadero si L es una expresión lambda del tipo X^Cat y Lista contiene el resultado de instanciar X con la evaluación de Cat por meta_dis/5 tantas veces como sea posible con un máximo de N en el modo indicado por M sobre la lista de entrada E y dejando un resto S. Valores para M: det, semi, indet

```

hasta(_, N, _, [], [], []):-      N >= 0, !.

hasta(_, 0, _, [])                -->   !.

hasta(Dis, N, X^Y, [ Z | Rz ]) -->   meta_dis(Dis, X^Y, Z),
                                     { ifthen(Dis = det, !), dec(N, N1) },
                                     hasta(Dis, N1, X^Y, Rz).

hasta(_, _, _, [])                -->   [].

```

1. Aplicación determinista de hasta/6:

```

hasta_d(N, X^Y, L)                -->   hasta(det, N, X^Y, L).

```

2. Aplicación semideterminista de hasta/6:

```

hasta_s(N, X^Y, L)                -->   hasta(semi, X^Y, L).

```

3. Aplicación indeterminista de hasta/6:

```

hasta_i(N, X^Y, L)                -->   hasta(indet, X^Y, L).

```


3. Entre N y M

Procedimiento	entre / 7
Argumentos	entre(entero, entero, atomo, lambda, lista, lista, lista)
Direccionalidad	in(+,+,+,+, -,+, -), out(+,+,+,+,+,+,+) in(+,+,+,+, ?,+, ?), out(+,+,+,+,+,+,+)
Significado	entre(M, N1, N2, L, Lista, E, S) es verdadero si L es una expresión lambda del tipo X^Cat y Lista contiene el resultado de instanciar X con la evaluación de Cat por meta_dis/5 el máximo número de veces entre N1 y N2 en el modo indicado por M sobre la lista de entrada E y dejando un resto S. Valores para M: det, semi, indet.

```

entre(_, N1, N2, _, [], [], []):- ifthenelse(
                                (N1 =< 0, N2 >= 0),
                                !,
                                (!, fail)
                                ).

entre(_, N, 0, _, []) --> N =< 0, !.

entre(Dis, N, M, X^Y, [ Z | Rz ]) --> meta_dis(Dis, X^Y, Z),
{
  ifthen(Dis=det, !),
  dec(N,N1), dec(M,M1)
},
entre(Dis, N1, M1, X^Y, Rz).

entre(_, N, _, _, []) --> {N =< 0}.

```

1. Aplicación determinista de entre/7:

```
entre_d(N1, N2, X^Y, L) --> entre(det, N1, N2, X^Y, L).
```

2. Aplicación semideterminista de entre/7:

```
entre_s(N1, N2, X^Y, L) --> entre(semi, N1, N2, X^Y, L).
```

3. Aplicación indeterminista de `entre/7`:

`entre_i(N1, N2, X^Y, L) --> entre(indet, N1, N2, X^Y, L).`

4. Exactamente N

Procedimiento	<code>ex / 6</code>
Argumentos	<code>ex(atomo, entero, lambda, lista, lista, lista)</code>
Direccionalidad	<code>in(+,+,+, -,+, -), out(+,+,+,+,+,+)</code> <code>in(+,+,+,?,+,?), out(+,+,+,+,+,+)</code>
Significado	<code>ex(M, N, L, Lista, E, S)</code> es verdadero si L es una expresión lambda del tipo <code>X^Cat</code> y Lista contiene el resultado de instanciar X con la evaluación de Cat por <code>meta_dis/5</code> exactamente N veces en el modo indicado por M sobre la lista de entrada E y dejando un resto S. Valores para M: <code>det, semi, indet</code>

```

ex(_, N, _, [], [], []):-      ifthenelse(N = 0, !, (!,fail)).
ex(_, 0, _, [])                --> [].
ex(Dis, N, X^Y, [ Z | Rz ])    --> meta_dis(Dis, X^Y, Z),
                                { ifthen(Dis = det, !), dec(N, N1) },
                                ex(Dis, N1, X^Y, Rz).
    
```

1. Aplicación determinista de `ex/6`:

`ex_d(N, X^Y, L) --> ex(det, N, X^Y, L).`

2. Aplicación semideterminista de `ex/6`. Aunque podemos darla:

`ex_s(N, X^Y, L) --> ex(semi, X^Y, L).`

es preciso considerar, sin embargo, que $ex_s/5$ resulta en todo equivalente a $ex_d/5$. Ello debido a dos datos fundamentales:

- i. Los cuantificadores semideterministas añaden a los deterministas la posibilidad de reintentar la cuantificación
- ii. El cuantificador $ex/6$ debe impedir por su propia definición el *backtracking*; no hay más soluciones -en cuanto al número de secuencias que satisfacen la categoría sobre la que se aplica- que la primera: justamente la indicada por el número que acompaña en cada caso a $ex/6$.

3. Aplicación indeterminista de $ex/6$:

$$ex_i(N, X^AY, L) \quad \rightarrow \quad ex(indet, X^AY, L).$$

Aplicación sobre un caso concreto

El interés de lo que sigue es mostrar mediante un ejemplo el distinto significado que tienen las variantes determinista, semideterminista e indeterminista de los procedimientos presentados.

Supongamos dos categorías gramaticales distintas: una determinista, $area_d/3$, otra indeterminista, $area_i/3$, definidas mediante las siguientes DCGs:

```
/* DCG determinista */
area_d([ ], [ ], [ ]):-!.
area_d([ ]) --> [ ], !.
area_d([ ]) --> [ ], !.
```

```

area_d([X|Y]) --> [X],
                  area_d(Y).

/* DCG indeterminista */
area_i([ ], [ ], [ ]):- !.

area_i([ ]) --> [ ].

area_i([ ]) --> [ ].

area_i([X|Y]) --> [X],
                  area_i(Y).

```

Desde el punto de vista LENDEX, ambas áreas serían áreas DD en la medida que conocemos de ellas únicamente cómo acaban: con un punto, una coma o el fin de la secuencia a analizar. Supongamos además el segmento de texto siguiente:

“Secuencia con punto, coma y espacios. Fin”

y veamos cómo es analizado por las distintas versiones de los cuantificadores aplicados sobre `area_d/3` y `area_i/3`.

Tomemos uno de los cuantificadores: `(**)/5` y estudiemos su comportamiento.

Apliquemos en primer lugar el procedimiento `(*)/4` sobre `area_d/3`:

```

?- **'(X^area_d(X), L, "Secuencia con punto, coma y espacios. Fin", R).

L = ["Secuencia con punto", " coma y espacios", " Fin"]
R = ""

yes

```

El resultado de evaluar `(*)/4` sobre `area_d/3` es una única solución: una lista `L` formada por tres secuencias del tipo `area_d` y un resto `R` vacío. Por ser éste un cuantificador determinista, S-Lendex encuentra una solución que satisface la DCG sobre la secuencia a analizar e impide toda posibilidad de *backtracking* sobre sí

mismo y sobre la categoría. Esto último hace que su aplicación sobre la categoría indeterminista, `area_i/3`, rinda exactamente el mismo resultado.

Veamos qué sucede si en lugar del cuantificador determinista aplicamos el semideterminista:

```
?- '*(X^area_d(X), L, "Secuencia con punto, coma y espacios. Fin", R).
L = ["Secuencia con punto"," coma y espacios"," Fin"]
R = "" ;

L = ["Secuencia con punto"," coma y espacios"]
R = " Fin" ;

L = ["Secuencia con punto"]
R = " coma y espacios. Fin" ;

L = []
R = "Secuencia con punto, coma y espacios. Fin"

yes
```

Las soluciones en este caso son cuatro. La primera de ellas coincide con la primera y única del ejemplo anterior. Las tres siguientes resultan del *backtracking* efectuado sobre la cuantificación. Comprobamos efectivamente que las soluciones alternativas propuestas reducen progresivamente la lista resultante de la primera evaluación. Esta reducción llega al límite con la última de las soluciones en la que la lista `L` es vacía y el resto `R` coincide exactamente con la secuencia de entrada.

¿Qué sucede si en lugar de aplicar el cuantificador semideterminista `(*)/4` sobre `area_d/3` lo hacemos sobre `area_i/3` ? La misma consideración antes planteada para el determinista `(*)/4` vale aquí también. En la medida que uno y otro anulan el

carácter indeterminista de $area_i/3$, el resultado de aplicar sobre esta categoría el cuantificador semideterminista es el mismo que el generado sobre $area_d/3$.⁹⁶

Vayamos con la última de las versiones del cuantificador: el indeterminista $(\wedge)/4$. Consideremos primero su aplicación sobre el área determinista $area_d$. Dos son los datos a considerar:

1. El cuantificador indeterminista añade al semideterminista la posibilidad de reintentar la categoría gramatical sobre la que se aplica
2. La categoría gramatical $area_d$ no es reintentable.

La consecuencia inmediata de estos dos hechos es la que la evaluación sobre $area_d/3$ del cuantificador indeterminista $(\wedge)/4$ coincide con la del semideterminista $(\ast-)/4$.

No sucede así, naturalmente, si la categoría evaluada es indeterminista. Veamos qué sucede al aplicar $(\wedge)/4$ sobre $area_i/3$:

<p>?- $\wedge(X^{area_i(X)}, L, \text{"Secuencia con punto, coma y espacios. Fin"}, R)$.</p> <p>L = ["Secuencia con punto", " coma y espacios", " Fin"] R = ""</p> <p>L=["Secuencia con punto", " coma y espacios"] R = " Fin"</p> <p>L = ["Secuencia con punto", " coma y espacios. Fin"] R = ""</p> <p>L = ["Secuencia con punto"] R = " coma y espacios. Fin"</p> <p>L = ["Secuencia con punto, coma y espacios", " Fin"]. R = ""</p>
--

⁹⁶Recordemos en este punto la función desempeñada por el procedimiento $@/3$ en la p.175

```
L = ["Secuencia con punto, coma y espacios"]
R = " Fin"

L = ["Secuencia con punto, coma y espacios. Fin"]
R = ""

L = []
R = "Secuencia con punto, coma y espacios. Fin"

yes
```

Las soluciones generadas son ocho. Encontramos entre ellas las cuatro ya conocidas de la aplicación semideterminista: 1ª, 2ª, 4ª y 8ª además de cuatro inéditas hasta el momento: aquéllas que resultan de reintentar la categoría gramatical `area_i` tantas veces como es posible: 3ª, 5ª, 6ª y 7ª.

Podemos extraer varias consecuencias a la vista de estos resultados:

- Los cuantificadores deterministas, semideterministas e indeterministas son significativamente distintos.
- La aplicación de los deterministas y los semideterministas sobre una categoría gramatical C1 anula el carácter posiblemente indeterminista de ésta, resultando equivalente a la aplicación sobre una categoría C2 cuya única diferencia con C1 fuera dicho carácter.

Los cuantificadores semideterminista e indeterministas son equivalentes si la categoría sobre la que se aplican es determinista.

Otros procedimientos gramaticales

Los siguientes procedimientos tienen mucho que ver con los presentados en la sección anterior, *Procedimientos gramaticales para los términos cuantificados*, p.171, como ya entonces apuntamos.

Consisten en una versión más simple de los mismos, en la que no se precisa recuperar valor alguno: responden a la distinción bien conocida entre áreas R y RR. Son usados por S-Lendex en procedimientos auxiliares de carácter general que más tarde presentaremos⁹⁷.

Desde el punto de vista del usuario que se dispone a definir una GDU, su empleo en una DCG *standard* se asemeja en todo al modo de escribir los términos cuantificados LENDEX salvo en lo relativo a las expresiones específicas del lenguaje: lexemas descritos, secuencias, etc. Un ejemplo podría ser el siguiente:

```
categoria_dcg--> desde(3, punto_sp), * espacio.
```

La diferencia viene dada por la traducción efectuada por S-Lendex al encontrar esas expresiones en los hechos y producciones LENDEX característicos, así como en la extensión y contenido informativo automáticamente recuperados.

En la medida que versiones más simples, podría plantearse su definición en términos de las anteriores o incluso negarles el sentido. Esto último se descarta si atendemos a la ventaja que desde el punto de vista de la expresividad obtenemos

⁹⁷Como `l_lexemas/2`, `aseo/2`, etc, en *Procedimientos de uso general empleados por S-Lendex*, p.232.

con su definición: es preferible para al usuario no tener que usar una expresión lambda y declarar anónimo un argumento si ello no es estrictamente necesario:

`categoria_dcg --> desde(3, ^punto_sp, _), '*'(^espacio, _).`

La definición en términos de los anteriores no nos parece tampoco una buena solución: la recuperación de la extensión y el contenido informativo se efectúa por mediación de un procedimiento, `sustitucion_vlex/3`, cuya aplicación en este contexto no está justificada. Dado que en todo caso la definición se impone -por la razón antes argüida-, un principio elemental de economía de recursos exige que ésta sea lo más ajustada a las necesidades concretas que se pretenden cubrir.

Organizamos su presentación, como en el caso anterior, en dos grandes grupos:

1. Procedimientos gramaticales de sólo reconocimiento para términos cuantificados mediante un operador iterativo
2. Procedimientos gramaticales de sólo reconocimiento para términos cuantificados mediante un cuantor numérico

y asociamos a cada uno de los cuantificadores un procedimiento general que da lugar a tres específicos: las aplicaciones determinista, semideterminista e indeterminista de los mismos.

Para ello requerimos el procedimiento equivalente a `meta_dis/5` en este contexto: `meta_dis/4`:

Procedimiento	<code>meta_dis / 4</code>
Argumentos	<code>meta_dis(atomo, expresion, lista, lista)</code>
Direccionalidad	<code>in(+,+,+,?)</code> , <code>out(+,+,+,+)</code>
Significado	<code>meta_dis(Dis, C, E, S)</code> es verdadero si la categoría C satisface la lista de entrada E dejando un resto S mediante su evaluación por <code>meta_lendex/6</code> en el modo indicado por Dis: det, semi, indet.

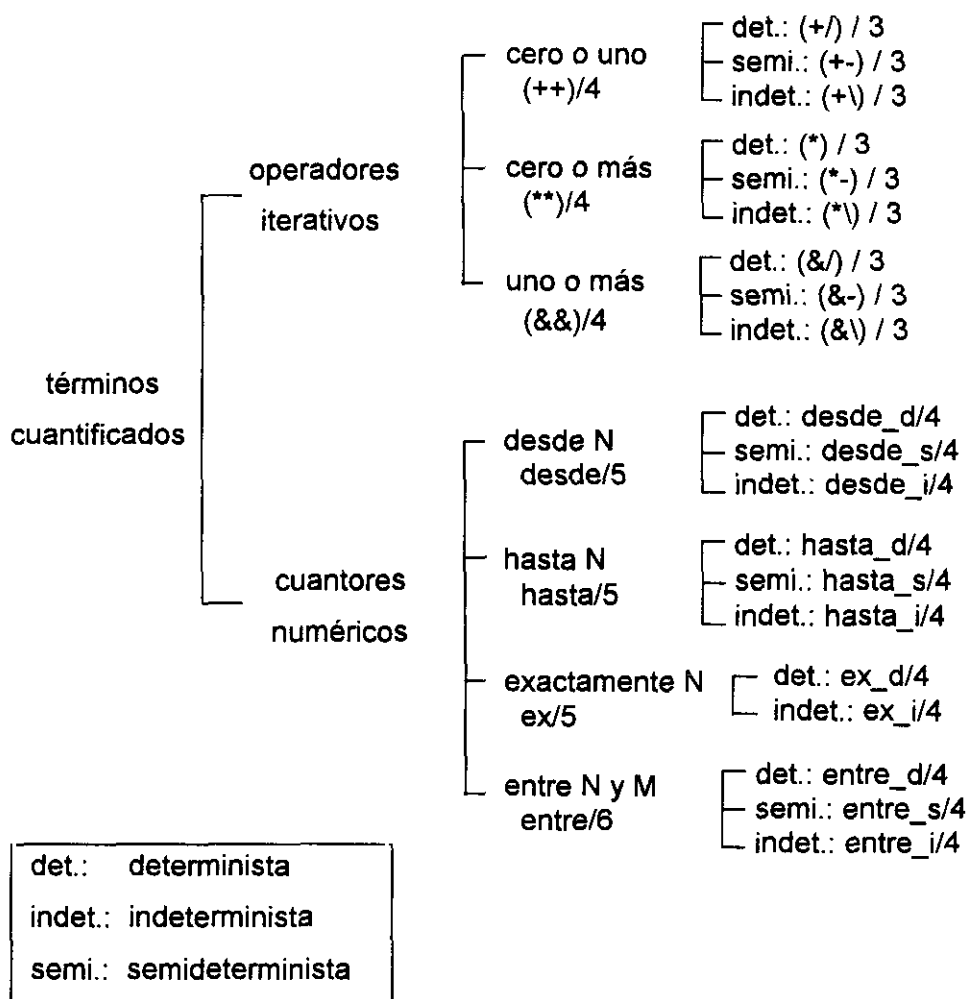
```
meta_dis(semi, Cat) --> @ meta_lendex(Cat, sin, _, _).  
meta_dis(Dis, Cat)   --> { Dis \= semi },  
                        meta_lendex(Cat, sin, _, _).
```

Este procedimiento discrimina como `meta_dis/5` entre la aplicación semideterminista de los cuantificadores -primera cláusula- y cualquier otra -segunda- y se diferencia de aquél en la ausencia del uso de `sustitucion_vlex/3`, aquí innecesario por el carácter de sólo reconocimiento de la cuantificación.

Además observamos diferente el segundo argumento de `meta_lendex/6`, precisamente aquél que indica la necesidad o no de calcular la extensión y contenido informativo asociados a la categoría gramatical `Cat` evaluada.

Podemos representar en este nuevo esquema la clasificación relativa a estos cuantificadores, de manera semejante a como lo hicimos en la p.173:

Figura 10: Cuantificadores LENDEX de sólo reconocimiento



Procedimientos gramaticales de sólo reconocimiento para términos cuantificados mediante un operador iterativo

Procedimiento	(++) / 4
Argumentos	'++'(atomo, expresion, lista, lista)
Direccionalidad	in(+,+,+,?), out(+,+,+,+)
Significado	'++'(M, C, E, S) es verdadero si la categoría gramatical C satisface cero o una veces la secuencia de entrada E dejando un resto S mediante su evaluación por meta_dis/4 en el modo indicado por M: set, semi, indet

```
'++'(Dis, Cat) --> meta_dis(Dis, Cat),
                    { ifthen(Dis = det, ! ) }.
```

```
'++'(_ , _) --> [].
```

Aplicaciones de (++) / 4:

```
+ / Cat --> '++'(det, Cat).
```

```
+ - Cat --> '++'(semi, Cat).
```

```
+ \ Cat --> '++'(indet, Cat).
```

Procedimiento	(**) / 4
Argumentos	'**'(atomo, expresion, lista, lista)
Direccionalidad	in(+,+,+,?), out(+,+,+,+)
Significado	'**'(M, C, E, S) es verdadero si la categoría gramatical C satisface cero o más veces la secuencia de entrada E dejando un resto S mediante su evaluación por meta_dis/4 en el modo indicado por M: det, semi, indet

```
'**'(_ , _ , [ ], []) :- !.
```

```
'**'(Dis, Cat) --> meta_dis(Dis, Cat),
                    { ifthen(Dis = det, ! ) },
                    '**'(Dis, Cat).
```

```
'**'(_ , [ ]) --> [].
```

Aplicaciones de (**)/4:

- * Cat --> '''(det, Cat).
- *- Cat --> '''(semi, Cat).
- *\ Cat --> '''(indet, Cat).

Procedimiento	(&&) / 4
Argumentos	'&&'(atomo, expresion, lista, lista)
Direccionalidad	in(+,+,+,?), out(+,+,+,+)
Significado	'&&'(M, C, E, S) es verdadero si la categoría gramatical C satisface una o más veces la secuencia de entrada E dejando un resto S mediante su evaluación por meta_dis/4 en el modo indicado por M: det, semi, indet

- '&&'(Dis, Cat) --> meta_dis(Dis, Cat),
 { ifthen(Dis = det, !) },
 '''(Dis, Cat).

Aplicaciones de (&&)/4:

- &/ Cat --> '&&'(det, Cat).
- &- Cat --> '&&'(semi, Cat).
- &\ Cat --> '&&'(indet, Cat).

Procedimientos gramaticales de sólo reconocimiento para términos cuantificados mediante un cuantor numérico

Procedimiento	desde / 5
Argumentos	desde(atomo,entero,expresion, lista, lista)
Direccionalidad	in(+,+,+,+,?), out(+,+,+,+,+)
Significado	desde(Dis, N, Cat, E, S) es verdadero si la categoría gramatical C satisface la lista de entrada E mediante su evaluación tantas veces como sea posible con un mínimo de N por meta_dis/4 en el modo indicado por Dis y deja un resto S. Valores de Dis: det, semi, indet.

```

desde(_, N, _, [ ], [ ]):-      ifthenelse(N =< 0, !, (!, fail)).

desde(Dis, N, Cat)  -->  meta_dis(Dis, Cat),
                        { ifthen(Dis = det, !), dec(N, N1) },
                        desde(Dis, N1, Cat).

desde(_, N, _ )    -->  {N =< 0}.
    
```

Aplicaciones de desde/5:

```

desde_d(N, Cat)    -->  desde(det, N, Cat).

desde_s(N, Cat)    -->  desde(semi, N, Cat).

desde_i(N, Cat)    -->  desde(indet, N, Cat).
    
```

Procedimiento	hasta/ 5
Argumentos	hasta(atomo, entero, expresion, lista, lista)
Direccionalidad	in(+,+,?,+,?), out(+,+,+,+,+)
Significado	hasta(Dis, N, Cat, E, S) es verdadero si la categoría gramatical C satisface la lista de entrada E mediante su evaluación tantas veces como sea posible con un máximo de N por meta_dis/4 en el modo indicado por Dis y deja un resto S. Valores de Dis: det, semi, indet.

```

hasta(_, N, _, [], []):- N >= 0, !.
hasta(_, 0, _) --> !.
hasta(Dis, N, Cat) --> meta_dis(Dis, Cat),
                        { ifthen(Dis = det, !), dec(N, N1) },
                        hasta(Dis, N1, Cat).
hasta(_, _, _) --> [].
    
```

Aplicaciones de hasta/5:

```

hasta_d(N, Cat) --> hasta(det, N, Cat).
hasta_s(N, Cat) --> hasta(semi, N, Cat).
hasta_i(N, Cat) --> hasta(indet, N, Cat).
    
```

Procedimiento	entre / 6
Argumentos	entre(entero, entero, atomo, expresion, lista, lista)
Direccionalidad	in(+,+,+,+,+?), out(+,+,+,+,+)
Significado	entre(Dis, N1, N2, C, E, S) es verdadero si la categoría gramatical C satisface la lista de entrada E mediante su evaluación el mayor número posible de veces entre N1 y N2 por meta_dis/4 en el modo indicado por Dis y deja un resto S. Valores de Dis: det, semi, indet

```

entre(_, N1, N2, _, [], []):- ifthenelse(
                                (N1 =< 0, N2 >= 0),
                                !,
                                (!, fail)
                                ).
entre(_, N, 0, _) --> N =< 0, !.
entre(Dis, N, M, Cat) --> meta_dis(Dis, Cat),
                            {
                              ifthen(Dis = det, !),
                              dec(N, N1), dec(M, M1)
                            },
                            entre(Dis, N1, M1, Cat).
    
```

entre(_, N, _, _) --> { N =< 0 }.

Aplicaciones de entre/6:

entre_d(N1, N2, Cat) --> entre(det, N1, N2, Cat).

entre_s(N1, N2, Cat) --> entre(semi, N1, N2, Cat).

entre_i(N1, N2, Cat) --> entre(indet, N1, N2, Cat).

Procedimiento	ex/ 5
Argumentos	ex(entero, atomo, expresion, lista, lista)
Direccionalidad	in(+,+,+,+,?), out(+,+,+,+)
Significado	ex(Dis, N, Cat, E, S) es verdadero si la categoría gramatical C satisface la lista de entrada E mediante su evaluación exactamente N veces por meta_dis/4 en el modo indicado por Dis y deja un resto S. Valores de Dis: det, semi, indet.

ex(_, N, _, [], []):- ifthenelse(N = 0, !, (!,fail)).

ex(_, 0, _) --> [].

ex(Dis, N, Cat) --> meta_dis(Dis, Cat),
{ ifthen(Dis = det, !), dec(N, N1) },
ex(Dis, N1, Cat).

Aplicaciones de ex/5:

ex_d(N, Cat) --> ex(det, N, Cat).

ex_s(N, Cat) --> ex(semi, Cat).⁹⁸

ex_i(N, Cat) --> ex(indet, Cat).

⁹⁸La misma consideración consideración hecha en el caso de ex/6 vale también en este contexto: p.185

Revisión

La exposición completa del núcleo de S-Lendex nos permite revisar algunas de las opciones tomadas. El desarrollo efectuado hasta ahora en la generación de la base de conocimiento necesaria para su funcionamiento nos enfrenta con un problema ya adelantado en la p.116: el trabajo específico de S-Lendex consistente en traducir una GDU expresada en LENDEX a código Prolog ejecutable se ha desarrollado del modo más directo posible: las expresiones específicas LENDEX se sostienen hasta el final y se traducen siempre que resulta necesario.

Esto hace que exista un buen número de operaciones que pueden considerarse redundantes. Supongamos, por concretar, la GDU del Anexo 1 y tomemos de ella la definición del área ED df correspondiente a la descripción física en un registro ISBD:

```
df      =>    +/ romano, +/ coma_sp, * futil,
           pag, +/ puntoycoma_sp, dim, * espaciofl.

dim     =>    * espacio, numeral^entre(numeral,10,40),
           +/ coma_sp, $5$, * espacio, cm.

pag     =>    numeral^mayor(numeral,21), espaciofl, pag.

cm      =>    $cm$, +/punto.
pg      =>    $p$, +/punto.
```

Dado que df está declarada fin de las áreas DD tit y pub -áreas de título y publicación- mediante los hechos:

```
fin(tit, df, externo_hetero).
fin(pub, df, externo_hetero).
```

ahora sabemos que S-Lendex deberá evaluar df en cada uno de los pasos de la recursión necesaria para la construcción de dichas áreas. Recordemos a este respecto la gramática generada automáticamente por el sistema para las áreas DD: p.114.

En la medida que fin de tit y pub, la evaluación de df se hace inevitable; es discutible, sin embargo, cuál es el modo más apropiado de efectuarla. La versión presentada hasta el momento exige a la evaluación de df una primera operación: la interpretación de las expresiones LENDEX específicas; esto es, prácticamente todas: operadores iterativos como $+/$ y $*$, lexemas descritos como los presentes en dim y pag, y secuencias como $\$5\$$, $\$cm\$$, $\$p\$$.

La traza de S-Lendex en este punto resulta bastante elocuente. Supongamos la secuencia de título correspondiente al registro conocido de otras secciones⁹⁹:

Integrated economics: a study of developing economics / Ojetunji Aboyade.-- London (etc.): Addison . Wesley Publishers, cop. 1983. VIII, 568 p.; 24 cm.
--

y supongámosla convertida en lexemas¹⁰⁰:

[Integrated, ,economics,;, ,a, ,study, ,of, ,developing,□,economics, ,/,
,Ojetunji, ,Aboyade,.,-,-, ,London, ,(,etc,.,),;□, , , , , , , , , , ,Addison, , ,
,Wesley, ,Publishers,.,, ,cop,.,, ,1983,.,□, , , , , , , , , , ,VIII,.,, ,568, ,p.,.,, ,24,
,cm,.]

Recordemos además la gramática de área DD generada por S-Lendex para el título a la vista de la GDU que comentamos, p.114, en *Aplicación a un ejemplo real*.

⁹⁹Véase la Figura 2 de la p.22

¹⁰⁰Evitamos de nuevo las marcas características de strings, $\$ \$$, con el fin de facilitar la lectura

Reproduzcamos aquí únicamente dos cláusulas: la relativa al fin determinado por df y la última, suficientes para ilustrar el funcionamiento de S-Lendex:

```

tit(A, [cat_eval(df, B, tit, hetero)| C]) :-
    meta_lendex(df, con, B, A, C),
    !.
tit([A|B], C) :-
    satisface_rasgo(true, A),
    tit(B, C).

```

Veamos cómo se aplican a la secuencia a analizar:

```

?- tit_resp([Integrated, ,economics,:, ,a, ,study, ,of,
,developing,□,economics, ,/, ,Ojetunji ...], R)

?- meta_lendex(df, con, [Integrated, ,economics,:, ,a, ,study, ,of,
,developing,□,economics, ,/, ,Ojetunji ...], R)
...
?- meta_lendex((+/ romano, +/ coma_sp, * futil, pag, +/
puntoycoma_sp, dim, * espaciofl), con, [Integrated, ,economics,:, ,a,
,study, ,of, ,developing,□,economics, ,/, ,Ojetunji ...], R)
...
?- meta_lendex(+/ romano, con, [Integrated, ,economics,:, ,a,
,study, ,of, ,developing,□,economics, ,/, ,Ojetunji ...], R)
...
(1)  ?- termino_cntf(+/ romano, Cat, Op, Args)
      exit : termino_cntf(+/ romano, romano, +/, [ ])

      ?- meta_lendex_cntf(+/, [ ], romano, Exts, Cis, [Integrated,
,economics,:, ,a, ,study, ,of, ,developing,□,economics, ,/,
,Ojetunji ...], R)
      ...
      fail

      ...
      fail

      ...
      fail

?- satisface_rasgo(true, Integrated)
exit : satisface_rasgo(true, Integrated)

?- tit_resp([ ,economics,:, ,a, ,study, ,of, ,developing,□,economics, ,/,
,Ojetunji ...], R)
...
...

```

```
...
(2)  ?- termino_cntf(+/ romano, Cat, Op, Args)
      exit : termino_cntf(+/ romano, romano, +/, [ ])

      ?- meta_lendex_cntf(+/, [ ], romano, Exts, Cis, [
        ,economics,;, ,a, ,study, ,of, ,developing,□,economics, ./,
        ,Ojeturji ...], R)
      ...
      fail
...
...
```

Observamos que (1) y (2) son exactamente la misma operación: por cada uno de los lexemas que incorpora a la extensión del área, S-Lendex debe interpretar la expresión `+/ romano` mediante el procedimiento `termino_cntf/4`. Esto es general: cada lexema exige que los términos cuantificados, los lexemas descritos y las secuencias que aparecen en las cláusulas de los fines sean interpretados como si de la primera vez se tratara. Todo ello para descartar en la mayor parte de los casos el lexema como un fin para el título y proseguir.

Parece que se impone hacer a S-Lendex menos “desmemoriado”. La alternativa que se presenta es la siguiente:

- Mantener la estrategia actual de interpretación de las expresiones específicas del lenguaje incorporando algún sistema de aserción de lemas.
- Compilar las expresiones del lenguaje de manera que sea una sola vez la que se traducen.

Por otra parte, y ya dispuestos a revisar lo desarrollado hasta el momento, cabe plantear una modificación relativa a la generación automática de las áreas DD. Por ser de carácter menos general comenzaremos por esta última cuestión y continuaremos más adelante con la relativa a la alternativa planteada.

Organizamos lo que sigue en estos tres apartados:

1. Generación automática de gramáticas para las áreas DD: segunda versión. Analizadores frente a reconocedores.
2. Un sistema de aserción de lemas
3. Compilación de las expresiones de LENDEX. Versión final de S-Lendex

Generación automática de gramáticas para las áreas DD: segunda versión. Analizadores frente a reconocedores.

Forma parte del acervo común de la programación lógica aplicada al tratamiento del lenguaje natural la distinción entre reconocedores y analizadores. En [Covington,94; p.51] encontramos lo siguiente: *“Un analizador debe hacer algo más que decir si una oración es o no aceptable; debe también informar acerca de su estructura”*

Analizadores y reconocedores representan la implementación de dos tipos generales de gramáticas lógicas:

1. Aquéllas que se limitan a establecer cuándo una secuencia cae bajo una categoría dada: gramáticas reconocedoras
2. Aquéllas otras que además de reconocer construyen mediante el recurso de los argumentos propio de las DCGs un contenido asociado a la secuencia reconocida

S-Lendex desarrolla automáticamente un proceso de generación de gramáticas para las áreas DD. Es importante reparar en el carácter de reconocimiento que las gramáticas generadas tienen.

Una DCG como la discutida en la introducción a esta sección¹⁰¹ es, por sí misma, capaz de reconocer una secuencia como perteneciente a la clase de las reconocidas por ella. Esto se ha combinado además con un meta-intérprete de gramáticas LENDEX, *meta-lendex/6*¹⁰², que hace posible el cálculo de la extensión y contenido asociados.

Recordemos la cláusula del meta-intérprete correspondiente al tratamiento de las áreas DD:

```
meta_lendex(Area, M, Ext3, Ci3, E, S):-
    area_dd(Area, _),
    !,
    Ci =.. [Area, Ext],
    dcg_term(Area, A2, E, S),
    call(A2),
    segmento_anterior(E, S, Ext),
    satisface_condicion(Area, Ext, Ext2, Ci, Ci2),
    meta_lendex_dd(Area, Ext2, Ci2, Ext3, Ci3).
```

Observamos cómo el cálculo de la extensión del área Ext -una vez evaluada mediante *call/1*- es debido al procedimiento *segmento_anterior/3*. El contenido informativo, expresado en el término Ci, se construye también en este punto.

Cabe plantearse si es posible evitar el cálculo, en ocasiones costoso, que *segmento_anterior/3* efectúa. La solución está precisamente en generar para las áreas DD una gramática no sólo reconocedora sino capaz de establecer por sí misma su extensión.

Veamos cómo.

¹⁰¹Véase la gramática generada para las áreas DD enc y tit en la p.114

¹⁰²Recordemos *El procedimiento meta_lendex/6: un meta-intérpete de gramáticas*, p. 118

Generación de analizadores para las áreas DD

Supongamos el mismo ejemplo que empleamos en la exposición del proceso de generación automática de gramáticas para las áreas DD, p.95:

```
tit_resp([ ], [ ]):- !.  
  
tit_resp --> fin_area_isbd.  
tit_resp --> espacio, espacio, espacio, fl.  
  
tit_resp --> [X],  
            tit_resp.
```

Esta DCG es capaz de reconocer las secuencias que caen bajo el área `tit_resp` pero no recupera valor alguno. Transformémosla en otra que sí lo haga:

```
tit_resp([ ], [ ], [ ]):- !.  
  
tit_resp([.-]) --> fin_area_isbd.  
tit_resp([ ]) --> espacio, espacio, espacio, fl.  
  
tit_resp([X|Y] --> [X],  
                tit_resp(Y).
```

El recurso de los argumentos resulta apropiado en este contexto. La única diferencia con la anterior estriba precisamente en el argumento añadido que permite recuperar la extensión reconocida por la gramática. No nos importa en este punto discutir el distinto modo de definir los fines internos, externos heterogéneos y homogéneos¹⁰³. Lo relevante es explorar la posibilidad de generar automáticamente una gramática para las áreas DD capaz de recuperar la extensión reconocida.

¹⁰³La gramática propuesta trata efectivamente de manera distinta el primer fin y el segundo: mientras hace interno al primero -recuperando en la lista su extensión-, el segundo sería externo al asociarle la lista vacía.

Igual que establecíamos una forma general de la gramática de un área DD¹⁰⁴, podemos plantear aquí cuál sería esa forma general para las gramáticas analizadoras.

Haciendo uso del recurso de los argumentos, la forma general sería la siguiente:

```
area_dd_X([ ], [ ], [ ]):- !.  
  
area_dd_X(F1)      -->  fin_X_1.  
area_dd_X(F2)      -->  fin_X_2.  
...  
area_dd_X(F3)      -->  fin_X_n.  
  
area_dd_X([A|B])   -->  [A],  
                        area_dd_X(B).
```

donde F_i es la extensión asociada al área `area_dd_X` cuando encuentra el fin `fin_X_i`.

Respecto de la primera cláusula, lo apuntado entonces en relación con la consideración o no del fin de la secuencia de entrada como un fin para el área es igualmente vigente ahora. Lo novedoso aquí es el proceso de construcción de la extensión del área, en paralelo con su delimitación.

Veamos cómo es el programa que hace esto posible.

Nueva versión del programa

El procedimiento principal `aserta_gramatica/2` es muy parecido al propuesto en la versión anterior¹⁰⁵. Siguen siendo válidos su esquema y explicación. La modificación es muy leve y relativa a su primera línea:

¹⁰⁴Véase: *Estructura general de la gramática de áreas DD*, p.100


```
aserta_gramatica(Area,Rasgo):-
    abolish(Area/3),
    aserta_lvacia(Area),
    aserta_fines(Area),
    aserta_recurcion(Area, Rasgo),
    !.
```

Efectivamente, la única diferencia entre uno y otro consiste en lo afectado por el primitivo `abolish/1`: entonces se borraba de la base de conocimiento toda referencia al procedimiento `Area/2`, mientras que aquí lo eliminado es `Area/3`. Ello es, naturalmente, consecuencia directa del argumento añadido.

La llamada a los tres procedimientos auxiliares: `aserta_lvacia/1`, `aserta_fines/1`, `aserta_recurcion/2`, sigue siendo vigente así como su esquema. Cada uno de ellos se ve modificado en el modo que ahora indicamos.

```
aserta_lvacia(Area):-
    Term =.. [Area, [], [], []],
    aserta((Term:- !)).
```

El primero, `aserta_lvacia/1`, corre una suerte parecida al general: la modificación es muy leve y sólo relativa al argumento extra: primera lista vacía, correspondiente al valor de la extensión del área cuando la secuencia a analizar se acaba.

El segundo, `aserta_fines/1`, no sufre modificación alguna¹⁰⁶. No sucede así con el procedimiento subordinado `aserta_un_fin/4`, que queda como sigue:

¹⁰⁵Recordemos su definición de la p.107

¹⁰⁶Véase la p.108 donde se define

```
aserta_un_fin(Area, Fin, interno, Corte):-
    AreaArg =.. [Area, Ext],
    expand_term((AreaArg --> meta_lendex(Fin, con, Ext, _), Corte),
                Clausula),
    assertz(Clausula),
    !.

aserta_un_fin(Area, Fin, externo_homo, Corte):-
    AreaArg =.. [Area, [ ]],
    expand_term((AreaArg -->
                meta_lendex(Fin, con, Ext, Ci),
                apr_side(cat_eval(Fin, homo, Area, Ext, Ci )),
                Corte), Clausula),
    assertz(Clausula),
    !.

aserta_un_fin(Area, Fin, externo_hetero, _):-
    AreaArg =.. [Area, [ ]],
    expand_term((AreaArg -->
                meta_lendex(Fin, con, Ext, Ci),
                apr_side(cat_eval(Fin, hetero, Area, Ext, Ci)),
                !), Clausula),
    assertz(Clausula),
    !.
```

Los comentarios de entonces relativos al carácter más o menos determinista de las cláusulas de los fines¹⁰⁷ siguen valiendo ahora. Así como los relacionados con el tratamiento recibido por los fines externos mediante el literal `cat_eval/5` y el procedimiento `apr_side/3`. La optimización derivada de guardar resultados parciales y el conocimiento transmitido a la evaluación de la categoría siguiente - útil para distinguir los fines externos heterogéneos de los homogéneos- se reproduce sin variación alguna. Estamos, sin embargo, en una mejor disposición para entender el alcance de la aplicación en las cláusulas de los fines del procedimiento `meta_lendex/6` y el sentido de lo planteado al principio de esta sección.

¹⁰⁷Véanse las p.109 y siguientes

El último procedimiento, `aserta_recurcion/2`, queda así:

```
aserta_recurcion(Area, Rasgo):-
    AreaXY =.. [Area, [X|Y] ],
    AreaY =.. [Area, Y],
    expand_term((AreaXY -->
                [X], { satisface_rasgo(Rasgo, X) },
                AreaY
                ), Prolog),
    aserta(Prolog),
    !.
```

Aplicación a un ejemplo real

De manera equivalente a como hicimos en la p.114, resulta interesante mostrar el resultado de aplicar el programa de generación automática de analizadores a la GDU del Anexo 1. Este es el producto para las áreas DD enc y tit:

```
/* Gramática asociada al área DD enc */

enc([ ], [ ], [ ]) :-
    !.

enc(A, B, C) :-
    meta_lendex(fl, con, A, D, B, C).

enc(A, B, C) :-
    meta_lendex(fa, con, A, D, B, C).

enc([ ], A, B) :-
    meta_lendex(isbn, con, C, D, A, E),
    apr_side(cat_eval(isbn, hetero, enc, C, D), E, B),
    !.

enc([A|B], [A|C], D) :-
    satisface_rasgo(enc, en_may, A, E, F),
    enc(B, C, D).
```

```

/* Gramática asociada al área DD tit */

tit([], [], []) :-
    !.

tit([], A, B) :-
    meta_lendex(df, con, C, D, A, E),
    apr_side(cat_eval(df, hetero, tit, C, D), E, B),
    !.

tit(A, B, C) :-
    meta_lendex(desde_d(3, punto), con, A, D, B, C),
    !.

tit(A, B, C) :-
    meta_lendex(fa, con, A, D, B, C),
    !.

tit([], A, B) :-
    meta_lendex(isbn, con, C, D, A, E),
    apr_side(cat_eval(isbn, hetero, tit, C, D), E, B),
    !.

tit([A|B], [A|C], D) :-
    satisface_rasgo(tit, true, A, E, F),
    tit(B, C, D).

```

Modificación de meta_lendex/6

Generada la gramática que reconoce y recupera la extensión de un área DD, hemos de adaptar la cláusula correspondiente de `meta_lendex/6` a la nueva situación:

```

meta_lendex(Area, M, Ext3, Ci3, E, S):-
    area_dd(Area, _),
    !,
    Ci =.. [Area, Ext],
    dcg_term(Ci, A2, E, S),
    call(A2),
    satisface_condicion(Area, Ext, Ext2, Ci, Ci2),
    meta_lendex_dd(Area, Ext2, Ci2, Ext3, Ci3).

```

Observamos respecto de la versión anterior, p.139, la ausencia del cálculo de la extensión efectuado en la versión anterior por el procedimiento `segmento_anterior/3`: en ésta se ha tornado innecesario. Por lo demás, la

cláusula sólo presenta una modificación: el primer argumento de `dcg_term/4`. Donde antes se pasaba el nombre del área, `Area`, ahora es un literal de aridad 1, precisamente el término `Ci` que representa su contenido informativo y cuyo único argumento es la extensión. La evaluación de `A2` a cargo de `call/1` instancia adecuadamente la variable `Ext` correspondiente a la extensión.

Un sistema de aserción de lemas

La primera opción contemplada para optimizar el funcionamiento de S-Lendex en lo relativo a la interpretación de las expresiones LENDEX específicas consiste en el desarrollo de un sistema de aserción de lemas.

Una solución de este tipo cae bajo lo que en [Sterling-Shapiro,86] se denomina "memo-functions": *"Las funciones-memo guardan el resultado de subcomputaciones para usarlo más tarde en una computación... El prototipo de función-memo es lemma(Goal)... Su implementación es la que sigue:*

$$\text{lemma}(P) \leftarrow P, \text{asserta}((P \leftarrow !)).$$

Con una función de este tipo `meta_lendex/6` se vería modificado del siguiente modo: todas y cada una de las llamadas a procedimientos que interpretan expresiones LENDEX lo serían mediante `lemma/1`. Veámoslo en la cláusula correspondiente a los términos cuantificados:

```
/* Cláusula de los términos cuantificados */  
meta_lendex(CntfTerm, _, Exts, Cis, E, S):-  
    lemma(termino_cntf(CntfTerm, Area, Fu, Args)),  
    !,  
    meta_lendex_cntf(Fu, Args, Area, Exts, Cis, E, S).
```

En este caso parece que `lemma/1` está indicado: `termino_cntf/4` es un procedimiento cuyo significado no se ve afectado por el corte que `lemma/1`

introduce. Otra situación bien distinta sería con un procedimiento no determinista: en tal caso se anularían sus soluciones alternativas. Esta es una de las limitaciones que la implementación de `lemma/1` presenta.

Hay otras razones, sin embargo, que nos han hecho descartar una solución de este tipo. La primera es su menor utilidad en las otras cláusulas de `meta_lendex/6` relativas a expresiones LENDIX: la de los lexemas descritos y la de las secuencias. Si recordamos su implementación de la p.129, observamos que los objetivos que interpretan dichas expresiones son complejos. Para las secuencias, por ejemplo, hay primero una comprobación de que la expresión es una cadena de texto, un *string*, y a continuación la lexematización mediante `l_lexemas/2`. Esto hace que el procedimiento `lemma/1` no sea directamente aplicable a todo el objetivo:

```
((string(S), l_lexemas(S, L))
```

dado que su aplicación prevista es sobre objetivos simples. Esto podría subsanarse con una definición distinta de `lemma/1`, una definición alternativa de los objetivos LENDIX o una aplicación parcial sobre el segundo de los objetivos: el realmente costoso:

```
meta_lendex(Secuencia, _, SL, SL, L, R):-  
    string(Secuencia),  
    !,  
    lemma(l_lexemas(Secuencia, SL)),  
    append(SL, R, L).
```

La razón final que nos decanta hacia una compilación de las expresiones LENDIX es la mayor economía que supone frente a esta solución. Aunque con la aserción de lemas se evita calcular más de una vez la interpretación de secuencias, términos cuantificados y lexemas descritos, la llamada a los predicados correspondientes se efectúa de hecho tantas veces como en la versión anterior, cosa que no sucede si

las expresiones se traducen de una vez por todas y se procede luego a su evaluación por una versión adaptada de `meta_lendex/6`.

Esta es la solución que a continuación presentamos.

Compilación de las expresiones LENDEX. Versión final de S-Lendex

La segunda opción entre las posibles para evitar que S-Lendex repita una y otra vez operaciones en ocasiones costosas relativas a la interpretación de expresiones características de LENDEX es desarrollar una compilación previa de éstas.

Esto exige dos procesos:

- Traducir las producciones de la GDU: principal y de áreas ED
- Generar las gramáticas de áreas DD de acuerdo con la traducción de los fines correspondientes

Además, es preciso adaptar el meta-intérprete `meta_lendex/6` a la nueva situación así como el procedimiento principal de análisis de los documentos `analisis/4`.

Organizamos lo que sigue de acuerdo con estos apartados:

1. Compilación de las producciones de la GDU
2. Generación de las gramáticas de áreas DD compilando los fines. Tercera y última versión
3. Aplicación a un ejemplo real
4. Adaptación de `meta_lendex/6`
5. Adaptación de `analisis/4`

Compilación de las producciones de la GDU

Distinguimos en este proceso dos subordinados:

1. Compilación de la producción principal: (==>)/2
2. Compilación de las producciones de áreas ED: (=>)/2

El procedimiento que lleva a cabo la compilación de las producciones de la GDU es compilación_prods/0:

Procedimiento	compilacion_prods/0
Significado	compilacion_prods/0 traduce cada una de las producciones de la GDU y las susituye en la base de conocimiento de S-Lendex por su resultado

```
compilacion_prods:-  
    compilacion_principal,  
    compilacion_areas_ed.
```

Este procedimiento llama a dos subordinados: compilacion_principal/0 y compilacion_areas_ed/0 que efectúan respectivamente los procesos indicados más arriba.

El primero de ellos se implementa así:

```
compilacion_principal:-  
    abolish(precompilado/2),  
    (documento ==> D1),  
    (S1, S2)^(  
        compilacion_lx_dcg(S1, S2),  
        assertz(precompilado(S1, S2))  
    )  
    transforma_f D1 en D2,  
    retract(documento ==> D1),  
    assertz(documento ==> D2).
```

La primera operación efectuada por este procedimiento es borrar de la base de conocimiento del sistema toda referencia al predicado precompilado/2, asertado

dinámicamente por el sistema. Éste cumple la misión de conservar el vínculo entre cada uno de los elementos de la descripción -expresada en la lista D1- y su transformado en D2. Resulta de capital importancia, como más tarde veremos¹⁰⁸, en el correcto funcionamiento de analisis/4.

Una vez efectuada la traducción de la descripción D1 en D2 mediante la aplicación del procedimiento `transforma_f/2`¹⁰⁹ sobre los procedimientos `compilacion_lx_dcg/2` y `assertz/1`, se sustituye en la base de conocimiento de S-Lendex la producción principal original: `(documento ==> D1)`, por la generada en la compilación: `(documento ==> D2)`.

Antes de explicar el procedimiento `compilacion_lx_dcg/2`, núcleo de la traducción, presentamos el segundo procedimiento subordinado, `compilacion_areas_ed/0`, que también lo usa:

```
compilacion_areas_ed:-
    abolish(pre_cmpl/2),
    e_mientras(
        ((Area => B1), not pre_cmpl(ed, Area, B1) ),
        (
            compilacion_lx_dcg(B1, B2),
            retract((Area => B1)),
            assertz((Area => B2)),
            assertz(pre_cmpl(Area, B1))
        )
    ),
    abolish(pre_cmpl/2).
```

Este procedimiento aserta dinámicamente como el anterior un predicado, `pre_cmpl/2`, que tiene un alcance estrictamente local. Mientras que en el caso de

¹⁰⁸En la adaptación de analisis/4 a la versión compilada, p. 229

¹⁰⁹El significado del procedimiento `transforma_f/2` se encuentra en la p.232

compilacion_principal/0 la aserción de los hechos precompilado/2 es utilizada más tarde en el análisis de los documentos, el interés de pre_cmpl/2 se circunscribe a la ejecución del procedimiento donde es asertado. De ahí que vuelva a producirse la llamada a abolish/1 para liberar a S-Lendex de conocimiento innecesario.

La función cumplida por pre_cmpl/2 es la de guardar memoria de la producciones que han sido ya traducidas y evitar el bucle que caso de no hacerlo produciría la aplicación de e_mientras/2.

Presentamos ahora el esquema e implementación del procedimiento compilacion_lx_dcg/2, núcleo -como antes indicamos- de la precompilación.

Procedimiento	compilacion_lx_dcg/2
Argumentos	comiplacion_lx_dcg(expresion, expresion)
Direccionalidad	in(+,-), out(+,+) in(+,+), out(+,+)
Significado	compilacion_lx_dcg(T1, T2) es verdadero si T2 es el resultado de traducir las expresiones LENDEX de T1 a expresiones DCG en T2

/* 1ª cláusula: áreas R */

```
compilacion_lx_dcg(? A, ? A1):-
    compilacion_lx_dcg(A, A1),
    !.
```

/* 2ª, 3ª y 4ª cláusulas: conjunción, disyunción y negación */

```
compilacion_lx_dcg((A, B), (A1, B1)):-
    !,
    compilacion_lx_dcg(A, A1),
    compilacion_lx_dcg(B, B1),
    !.
```

```
compilacion_lx_dcg((A ; B), (A1 ; B1)):-  
    !,  
    compilacion_lx_dcg(A, A1),  
    compilacion_lx_dcg(B, B1),  
    !.
```

```
compilacion_lx_dcg(not A, not A1):-  
    !,  
    compilacion_lx_dcg(A, A1),  
    !.
```

/* 5ª, 6ª y 7ª cláusulas: lexemas descritos, secuencias y términos cuantificados */

```
compilacion_lx_dcg(LxTerm, ([X], {O2}) ):-  
    lexema_descrito(LxTerm, O),  
    !,  
    sustitucion_vlex(O, lexema/X, O2),  
    !.
```

```
compilacion_lx_dcg(Secuencia, SL):-  
    string(Secuencia),  
    !,  
    l_lexemas(Secuencia, SL),  
    !.
```

```
compilacion_lx_dcg(CntfTerm,cntf(DCGTerm,Pares)):-  
    termino_cntf(CntfTerm, Area, Fu, Args),  
    !,  
    compilacion_cntf(Fu, Args, Area, DCGTerm, Pares),  
    !.
```

/* 8ª y 9ª cláusulas: áreas DD y ED */

```
compilacion_lx_dcg(A, area_dd(A, Var, A2)):-  
    area_dd(A, _),  
    !,  
    A2=.. [A, Var].
```

```
compilacion_lx_dcg(A, area_ed(A)):-  
    (A => _),  
    !.
```

/* 10ª cláusula: otras expresiones */

```
compilacion_lx_dcg(A, A):- !.
```

Encontramos paralelismos entre este procedimiento y `meta_lendex/6` que no son en modo alguno casuales: una cláusula para las áreas R -primera en un caso y otro- un grupo de cláusulas para la conjunción, la disyunción y la negación, otro para las expresiones LENDEX específicas: lexemas descritos, secuencias y términos cuantificados y un par de cláusulas para las áreas DD y las ED. En la medida que uno y otro traducen expresiones características de LENDEX, los casos que han de contemplar son necesariamente los mismos. Las cláusulas adicionales presentes en `meta_lendex/6` y ausentes del compilador obedecen al distinto carácter que un procedimiento y otro tienen: mientras `compilacion_lx_dcg/2` se limita a traducir las expresiones LENDEX a expresiones DCG *standard*, `meta_lendex/6` traduce esas mismas expresiones a objetivos Prolog y los evalúa. Por ello, las cláusulas de `meta_lendex/6` directamente relacionadas con dicha evaluación -recordemos el grupo de las que contemplan la aparición del literal `cat_eval/5`, p.124- desaparecen en `compilacion_lx_dcg/2`.

Una idea fundamental, ya comentada, subyace a este procedimiento: traducir las expresiones específicas de LENDEX una sola vez. Subsidiariamente, `compilacion_lx_dcg/2` efectúa un proceso adicional de identificación de las expresiones traducidas que permite a `meta_lendex/6` optimizar su posterior evaluación.

Este es el sentido de los literales `cntrf/2`, `area_dd/3` y `area_ed/1` en las cláusulas séptima, octava y novena: expresiones no identificables de manera inmediata adoptan un aspecto unificado que permite tratarlas de manera uniforme. Veremos más adelante su utilidad al presentar la nueva versión de `meta_lendex/6`, p.225.

El resto de las expresiones contempladas por `compilacion_lx_dcg/2`: cláusulas primera hasta séptima, son directamente evaluables por cláusulas especiales de `meta_lendex/6`. De ahí que no se incorpore en su imagen literal alguno.

En general, todas las cláusulas de `compilacion_lx_dcg/2` resultan familiares. Los procedimientos presentes en ella: `lexema_descrito/2`, `termino_cntf/4`, `sustitucion_vlex/3`, `l_lexemas/2`, son conocidos de antes: aparecen también usados en la primera versión de `meta_lendex/6`, p.118. Uno solo, `compilacion_cntf/5`, en la cláusula séptima, resulta novedoso. Veamos su esquema e implementación:

Procedimiento	<code>compilacion_cntf/5</code>
Argumentos	<code>compilacion_cntf(atomo, lista, expresion, expresion, lista)</code>
Direccionalidad	<code>in(+,+,+,?,?)</code> , <code>out(+,+,+,+,+)</code>
Significado	<code>compilacion_cntf(Op, Args, C, Term, P)</code> es verdadero si <code>Term</code> es una expresión DCG evaluable formada por el functor <code>Op</code> y los argumentos adicionales <code>Args</code> sobre la categoría <code>C</code> , y <code>P</code> es la lista de pares (E, CI) -E: extensión, CI: contenido informativo de cada una de las instancias de <code>C</code> - que aparece en <code>Term</code>

```

compilacion_cntf(Op, OtrosArgs, Cat, DCGTerm, Pares):-
    compilacion_lx_dcg(Cat, C1),
    append(OtrosArgs,
           [(X, Y)^meta_lendex(C1, con, X, Y), Pares],
           Args),
    DCGTerm =.. [Op| Args],
    !.

```

El procedimiento `compilacion_cntf/5` construye una expresión DCG evaluable `DCGTerm` del tipo de las previstas en *Procedimientos gramaticales para los términos cuantificados*, p.171. La clave del procedimiento está en la aparición del último argumento, `Pares`, en dicha expresión. Ello hace posible que la evaluación posterior de `DCGTerm` a cargo de `meta_lendex/6` instancie adecuadamente `Pares`, esto es, queden calculados la extensión y contenido informativo asociados a cada una de las secuencias que caen bajo la categoría `Cat` de acuerdo con la evaluación de `meta_lendex/6`.

Expuestos los procesos de compilación de la producción principal del documento y de las producciones de áreas ED, se impone presentar la adaptación del procedimiento que genera automáticamente las gramáticas de áreas DD.

Generación de las gramáticas de áreas DD compilando los fines. Tercera y última versión

El proceso de generación automática de gramáticas para las áreas DD se ve también afectado por el nuevo diseño. Es preciso incorporar en las cláusulas de los fines la traducción según `compilacion_lx_dcg/2` de todas y cada una de las expresiones que constituyen un fin para un área DD.

Efectuamos la adaptación de `aserta_geamatica/2` a partir de la última versión presentada: aquella que genera analizadores para las áreas DD y no reconocedores. De este modo aunamos una ventaja y otra.

El esquema del procedimiento `aserta_gramatica/2` y el de los procedimientos subordinados: `aserta_lvacia/1`, `aserta_fines/1` y `aserta_recursion/1`, es el mismo que el presentado en las páginas 208, 208 y 210, respectivamente. La implementación que se ve modificada con el nuevo planteamiento es la relativa a la aserción de los fines. Ésta queda así:

```
aserta_fines(Area):-
    ifthenelse(
        condicion(Area, _, fuerte),
        Corte = { true },
        Corte = !
    ),
    compilacion_lx_dcg(Area, AreaComp),
    e_mientras(
        fin(Area, Fin, TipoFin),
        (
            cmplcn_lx_dcg(Fin, FinComp),
            aserta_un_fin(Area, AreaComp, FinComp, TipoFin, Corte)
        )
    ).
```

Observamos que el procedimiento `e_mientras/2` efectúa la compilación de los fines asociados a un área DD antes de pasarlos como argumento, `FinComp`, al procedimiento que se ocupa de su aserción, `aserta_un_fin/5`. Este tiene un argumento extra respecto de la versión anterior: el que contiene el resultado de compilar el área DD, `AreaComp`, necesario para completar la información del literal `cat_eval/5` en las cláusulas de los fines externos. La nueva versión del procedimiento `aserta_un_fin/5`, cuya única modificación es la indicada, queda así:

```
aserta_un_fin(Area, _, CFin, interno, Corte):-
    AreaArg =.. [Area, L],
    expand_term((AreaArg --> meta_lendex(CFin, con, L, _),
                Corte), Clausula),
    assertz(Clausula),
    !.
```

```
aserta_un_fin(Area, AC, CFin, externo_hetero, _):-
    AreaArg =.. [Area, [ ]],
    expand_term((AreaArg -->
                meta_lendex(CFin, con, Ext, Ci),
                apr_side(cat_eval(CFin, hetero, AC, Ext, Ci)),
                !), Clausula),
    assertz(Clausula),
    !.
```

```
aserta_un_fin(Area, AC, CFin, externo_homo, Corte):-
    AreaArg =.. [Area, [ ]],
    expand_term((AreaArg -->
                meta_lendex(CFin, con, Ext, Ci),
                apr_side(cat_eval(CFin, homo, AC, Ext, Ci)),
                Corte), Clausula),
    assertz(Clausula),
    !.
```

Aplicación de la compilación a un ejemplo real

Resulta ilustrativo mostrar el resultado de aplicar a un ejemplo real los dos procesos expuestos:

- Compilación de las producciones de la GDU
- Generación de las gramáticas de áreas DD previa compilación de los fines

Utilizaremos de nuevo la gramática del Anexo 1, p.274. Presentamos sólo una parte de los resultados por no alargar innecesariamente el ejemplo:

```
/* Compilación de la producción principal */
```

```
documento ==> [  
  cntf('*'/'((A, B)^meta_lendex(espaciofl, con, A, B), C), C),  
  area_ed(sigs),  
  area_dd(enc, D, enc(D)),  
  area_dd(tit, E, tit(E)),  
  cntf('+/'((F, G)^meta_lendex(area_dd(ed,H,ed(H)), con, F,G),l),l),  
  area_dd(pub, J, pub(J)),  
  area_ed(df),  
  area_ed(serie),  
  area_ed(isbn),  
  cntf('&'/'((K, L)^meta_lendex(secundaria, con, K, L), M), M),  
  area_ed(regs)  
].
```

```
/* Compilación de dos producciones de áreas ED */
```

```
serie => parentesis_isp , area_dd(ser, A, ser(A)) , parentesis_dsp.
```

```
isbn => cntf('*'/'((A, B)^meta_lendex(espaciofl, con, A, B), C), C) ,  
? [ $ISBN$ ],  
cntf('+/'((D, E)^meta_lendex(dospuntos_sp, con, D, E), F), F) ,  
cntf('&'/'((G, H)^meta_lendex(  
  ([ I ] , {lex(l,numeral), true}),  
  cntf('+/'((J, K)^meta_lendex(guion_sp,con, J, K), L), L)  
), con, G, H), M), M).
```



```

/* Compilación de dos áreas DD */

enc([], [], []) :-
    !.

enc(A, B, C) :-
    meta_lendex(fl, con, A, D, B, C).

enc([], A, B) :-
    meta_lendex(area_ed(isbn), con, C, D, A, E),
    apr_side(cat_eval(area_ed(isbn), hetero,
        area_dd(enc, F, enc(F)), C, D), E, B),
    !.

enc(A, B, C) :-
    meta_lendex(area_ed(fa), con, A, D, B, C).

enc([A|B], [A|C], D) :-
    satisface_rasgo(enc, en_may, A, E, F),
    enc(B,C,D).

tit([], [], []) :-
    !.

tit([], A, B) :-
    meta_lendex(area_ed(df), con, C, D, A, E),
    apr_side(cat_eval(area_ed(df), hetero,
        area_dd(tit, F, tit(F)), C, D), E, B),
    !.

tit(A, B, C) :-
    meta_lendex(cntf(desde_d(3, (D,E)^
        meta_lendex(punto, con, D, E), F), F), con, A, G, B, C),
    !.

tit([], A, B) :-
    meta_lendex(area_ed(isbn), con, C, D, A, E),
    apr_side(cat_eval(area_ed(isbn), hetero,
        area_dd(tit, F, tit(F)), C, D), E, B),
    !.

tit(A, B, C) :-
    meta_lendex(area_ed(fa), con, A, D, B, C),
    !.

tit([A|B], [A|C], D) :-
    satisface_rasgo(tit, true, A, E, F),
    tit(B, C, D).

```

Importa quizás destacar la transformación que el procedimiento `compilacion_lx_dcg/2` ha llevado a cabo. La siguiente tabla de equivalencias, con algunos de los ejemplos más significativos, puede ser a estos efectos clarificadora:

Expresión	Compilación
* espaciofl	<code>cntf('*((A, B)^meta_lendex(espaciofl, con, A, B), C), C)</code>
sigs	<code>area_ed(sigs)</code>
enc	<code>area_dd(enc, D, enc(D)),</code>
+ / ed	<code>cntf('+/'((F, G)^meta_lendex(area_dd(ed,H,ed(H)), con, F,G),I),I),</code>
? \$ISBN\$? [\$ISBN\$]
numeral	<code>([I] , {lex(I, numeral), true}),</code>

Tabla 5: Compilación de una GDU

Expuesta la compilación de la GDU, es el momento de presentar la adaptación de `meta_lendex/6` a la nueva situación.

Adaptación de meta_lendex/6

Es preciso aclarar por qué se hace necesario adaptar `meta_lendex/6` a la nueva situación. El dato fundamental es que una parte importante de la tarea antes desarrollada por el meta-intérprete consistía en la traducción de las expresiones LENDIX características de la GDU. Dado que en el nuevo planteamiento esta traducción se efectúa a priori por `compilacion_lx_dcg/2`, es preciso liberar a `meta_lendex/6` de hacerla.

Esta es la nueva implementación de `meta_lendex/6`, última versión del programa. Señalamos con **negrita** las cláusulas que han sido modificadas respecto de la

versión anterior, p.123. Aunque son más las que permanecen iguales, preferimos reproducir a totalidad del procedimiento con el fin de hacer más fácil su lectura:

```
/* Cláusula de anidamiento de meta_lendex/6 */
```

```
meta_lendex(meta_lendex(A, M, Ext, Ci), M, Ext, Ci, E, S):-
    !,
    meta_lendex(A, M, Ext, Ci, E, S).
```

```
/* Categorías pre-evaluadas mediante el literal cat_eval/5 */
```

```
meta_lendex(Cat, _, Ext, Ci, [cat_eval(Cat, _, _, Ext, Ci)| E], E):- !.
```

```
meta_lendex(Cat, _, _, _, [cat_eval(Otra, hetero, Cat, _, _)| _], _):-
    Cat \= Otra,
    !,
    fail.
```

```
meta_lendex(Cat, Modo, D, T, [cat_eval(Otra, homo, Cat, Sec, _)| E], S):-
    Cat \= Otra,
    !,
    meta_lendex(Cat, Modo, D, T,[oculto(Sec)| E], S).
```

```
meta_lendex(Cat, Modo, D, T, [cat_eval(Otra, _, Previa, Sec, _)| E], S):-
    Cat \= Otra,
    Cat \= Previa,
    !,
    append(Sec, E, E1),
    meta_lendex(Cat, Modo, D, T, E1, S).
```

```
/* Cláusulas de la conjunción, disyunción y negación */
```

```
meta_lendex((A, B), M, Ext, Ci, E, S):-
    !,
    meta_lendex(A, M, ExtA, CiA, E, S1),
    meta_lendex(B, M, ExtB, CiB, S1, S),
    flatten([ExtA| ExtB], Ext),
    flatten([CiA| CiB], Ci).
```

```
meta_lendex((A ; B), M, Ext, Ci, E, S):- !,
    (
    meta_lendex(A, M, Ext, Ci, E, S)
    ;
    meta_lendex(B, M, Ext, Ci, E, S)
    ).
```

```
meta_lendex(not A, M, L, C, E, S) :-  
    !,  
    not meta_lendex(A, M, L, C, E, S).
```

/* Cláusula de los objetivos Prolog */

```
meta_lendex({A}, _, [ ], [ ], L, L) :-  
    !,  
    call(A).
```

/* Cláusulas de los terminales */

```
meta_lendex([ ], _, [ ], [ ], L, L):-  
    !.
```

```
meta_lendex([X|Y], _, [X|Y], [X|Y], E, S):-  
    !,  
    append([X|Y], E, S).
```

/* Cláusula de las áreas R */

```
meta_lendex(? Cat, M, [inane(Ext)], [ ], E, S):-  
    !,  
    meta_lendex(Cat, M, Ext, _, E, S).
```

/* Cláusula de los términos cuantificados */

```
meta_lendex(cntf(Cntf,Pares),_, Ext, Cis, E, S):-  
    !,  
    dcg_term(Cntf, NG, E, S),  
    call(NG),  
    ext_ci_aux(Pares, Ext, Cis).
```

/* Cláusulas de áreas DD y ED */

```
meta_lendex(area_dd(A, Ext, Ci), _, Ext3, Ci3, E, S):-  
    !,  
    dcg_term(Ci, A2, E, S),  
    call(A2),  
    satisface_condicion(A, Ext, Ext2, Ci, Ci2),  
    meta_lendex_dd(A, Ext2, Ci2, Ext3, Ci3).
```

```

meta_lendex(area_ed(Area), Modo, Ext2, Ci2, E, S):-
    !,
    (Area => Body),
    Ci =.. [Area, CiBody],
    meta_lendex(Body, Modo, Ext, T, E, S),
    flatten([T], CiBody),
    satisface_condicion(A, Ext, Ext2, Ci, Ci2).

```

/ Cláusula final: otras categorías gramaticales */*

```

meta_lendex(A, M, L, L, E, S):-
    dcg_term(A, A2, E, S),
    !,
    call(A2),
    ifthenelse(M = con, segmento_anterior(C, E, S), S = []).

```

Importa destacar un hecho: la modificación del procedimiento no es sólo relativa a las tres cláusulas que aparecen en **negrita**; hay además un cambio respecto de la versión anterior consistente en la desaparición de dos cláusulas: las correspondientes a las secuencias y los lexemas descritos. El sentido de éstas era únicamente traducir a objetivos Prolog expresiones específicas de LENDEX, razón por la cual han desaparecido con el desarrollo del compilador. El resultado de su traducción está ahora contemplado en las cláusulas de los terminales, de los objetivos Prolog llamados desde una gramática y en la de la conjunción. Recordemos en este punto la traducción según `compilacion_lx_dcg/2` de las secuencias y los lexemas descritos, p.218, de la que damos ahora un ejemplo:

Expresión	Compilación
\$Ejemplo de secuencia\$	[\$Ejemplo\$, \$ \$, \$de\$, \$ \$, \$secuencia\$]
numeral^mayor(numeral,5)	[X]^(lex(X, numeral), mayor(X, 5))

Tabla 6: Traducción de las secuencias y los lexemas descritos

Hay otro aspecto digno de comentario: la mejora del procedimiento en lo que a eficacia se refiere. Nótese cómo las cláusulas de esta versión imponen restricciones

más fuertes que las de la anterior al emplear de forma generalizada el recurso de la unificación. La evaluación a cargo de este procedimiento se agiliza en la medida que es el propio intérprete de Prolog quien preselecciona la cláusula apropiada a cada expresión. Éste es el sentido de incorporar los literales `cntf/2`, `area_dd/3` y `area_ed/1` en el procedimiento `compilacion_lx_dcg/2`.

Una última nota respecto de la cláusula de los cuantificadores: en el literal `cntf(Cntf, Pares)`, el segundo argumento llega sin instanciar. Es la evaluación del objetivo `Cntf`, en el que aparece `Pares`, la que hace que ésta adquiera un valor permitiendo generar la extensión `Exts` y el contenido informativo `Cis` asociados al término cuantificado cuya compilación dio lugar al literal `cntf/2`.

Adaptación de analisis/4

El procedimiento principal de análisis de documentos también se ve levemente modificado por el nuevo planteamiento. Ello reponde a la necesidad de informar de las incidencias sucedidas en el transcurso del mismo -áreas desconocidas, secuencias indeterminadas- en los términos en los que se expresó el usuario en la GDU.

Esta es la razón por la que en el procedimiento `compilacion_principal/O`¹¹⁰, se asertan dinámicamente los hechos `precompilado/2` que establecen la equivalencia entre las expresiones usadas por el usuario en la descripción de la clase de documentos y la traducción hecha por el compilador.

¹¹⁰Recordemos su definición de la p.215

El resultado de la aplicación de la compilación al ejemplo antes desarrollado¹¹¹ en lo que se refiere al predicado `precompilado/2` es el siguiente:

```
precompilado(* espaciofl, cntf('*'((A,B)^meta_lendex(espaciofl, con, A, B),
C), C)).
```

```
precompilado(sigs, area_ed(sigs)).
precompilado(enc, area_dd(enc, A, enc(A))).
precompilado(tit, area_dd(tit, A, tit(A))).
precompilado(+/ ed, cntf('+'((A,B)^meta_lendex(area_dd(ed, C, ed(C)),
con, A, B), D), D)).
```

```
precompilado(pub, area_dd(pub, A, pub(A))).
precompilado(df, area_ed(df)).
precompilado(serie, area_ed(serie)).
precompilado(isbn, area_ed(isbn)).
precompilado(&\ secundaria, cntf('&'((A,B)^meta_lendex(secundaria,
con, A, B), C),C)).
```

```
precompilado(regs, area_ed(regs)).
```

De esta manera adquiere sentido la nueva versión del procedimiento `analisis/4` que interroga a la base de conocimiento por el equivalente compilado de las categorías gramaticales que permanecen ignotas, cláusulas 3^a, 4^a y 6^a:

```
analisis([ ], C, E, [ ]):-
    segmento_complejo(soei, E, S),
    ifthenelse(
        algun S es X^(not futil(X)),
        C = ind(S),
        C = [ ]
    ),
    !.
```

```
analisis([C] Rdesc, [X] Rci, E, O2):-
    meta_lendex(C, con, _, X, E, O),
    analisis(Rdesc, Rci, O, O2).
```

¹¹¹Véase: *Aplicación de la compilación a un ejemplo real*, en la p.223

```
analisis(D1, D2, [ ], [ ]):-  
    (S, ign(S1))^precompilado (S1, S) transforma D1 en D2,  
    !.
```

```
analisis([IgnO, C | Rdesc], [ ign(Ign), ind(A), Ci | Rci], E, O2):-  
    hay_d(X^meta_lendex(C, con, _, X), Ci, Antes, E, O),  
    segmento_complejo(soei, Antes, A),  
    precompilado(Ign, IgnO),  
    !,  
    analisis(Rdesc, Rci, O, O2).
```

```
analisis([C | Rdesc], [ ind(A), Ci | Rci], E, O2):-  
    hay_d(X^meta_lendex(C, con, _, X), Ci, Antes, E, O),  
    segmento_complejo(soei, Antes, A),  
    !,  
    analisis(Rdesc, Rci, O, O2).
```

```
analisis( [Cat| Rdesc], [ign(C1)| Rci], E, O):-  
    !,  
    precompilado(C1, Cat),  
    analisis(Rdesc, Rci, E, O).
```

Son únicamente las cláusulas destacadas las que han sufrido modificaciones. En todas ellas se interroga a S-Lendex por los hechos asertados dinámicamente por `compilacion_principal/0` con el fin de que los literales del tipo `ign/1` se refieran a las categorías gramaticales escritas por el usuario y no a sus equivalentes compiladas.

Procedimientos de uso general empleados por S-Lendex

Distinguimos los siguientes apartados:

- Procedimientos de carácter general
- Procedimientos previos al análisis de los documentos
- Procedimientos finales de transformación del resultado del análisis

Procedimientos de carácter general

Entre los procedimientos de uso general utilizados por S-Lendex los hay de dos tipos: unos pertenecen al acervo común de la programación lógica, tales como *append/3*, *member/2* y *flatten/2*; otros pertenecen al *folklore* particular del grupo Verba Logica y se encuentran en [Sarabia,93] y [VL,94]. Presentamos aquí su significado:

/ los L son X^R : los miembros de la lista L satisfacen la condición R */*

/ algun L es X^R : al menos un miembro de la lista L satisface la condición R */*

/ (X,Y)^R transforma_f L1 en L2 : L2 es la lista formada por las imágenes de L1 según la regla de transformación R. Cada elemento de L1 tiene una y sólo una imagen en L2 */*

/ (X,Y)^R transforma L1 en L2 : L2 es la lista formada por las imágenes de L1 según la regla de transformación expresada por R. Cada elemento de L1 puede tener cero, una o más imágenes en L2 */*

/ A mientras B : mientras se cumple la condición expresada por B, se evalúa A */*

/ e_mientras(A, B) : mientras se cumple la condición expresada por A, se evalúa B : si se cumple B, se continúa evaluando A ; en otro caso, e_mientras/2 falla */*

/ while(A, B) : mientras se cumple la condición expresada por A, se evalúa B de manera determinista */*

/ futil(X) : el lexema X es un fútil, de acuerdo con la definición de G_{Lex} */*

/ lex(X, R) : X es un lexema del rasgo indicado por R: en_may, en_min, con_may, numeral, romano */*

/ mayor(X, Y) : el entero correspondiente al numeral X es mayor o igual que el entero Y */*

/ menor(X, Y) : el entero correspondiente al numeral X es menor o igual que el entero Y */*

/ entre(X, Y, Z) : el entero correspondiente al numeral X está entre los enteros Y y Z o es uno de ellos */*

/ normaliza(X, Y) : Y es la transformación del lexema X consistente en bajar a minúsculas los caracteres que estuvieran en mayúsculas y eliminar tildes */*

/ prefijo(M, X, Y) : el lexema X es un prefijo del lexema Y del tipo indicado por M: estricto/light */*

/ sufijo(M, X, Y) : el lexema X es un sufijo del lexema Y del tipo indicado por M: estricto/light */*

/ infijo(M, X, Y) : el lexema X es un infijo del lexema Y del tipo indicado por M: estricto/light */*

`/* lista(L) : L es una lista */`

`/* es_un_hecho(N, A) : existe una cláusula en la base de conocimiento
cuya cabeza es el literal de nombre N y aridad A y cuyo cuerpo es true */`

Por cada uno de los preterminales de G_{Lendex} , p.46, definimos un procedimiento primitivo LENDEX cuyo significado es inmediato. Aquéllos que se refieren a lexemas fútiles se explican por sí mismos: punto, coma, espacio, etc. Los acabados en el sufijo “_sp”: punto_sp, coma_sp, etc, se refieren a los mismos lexemas fútiles rodeados antes y después por un número máximo de cuatro espacios.

Procedimientos previos al análisis de los documentos

Dado que los terminales de las gramáticas LENDEX son lexemas, es necesario un procedimiento previo al análisis de los documentos que transforme éstos en una lista de lexemas. Este problema, clásico en el ámbito del procesamiento automático del lenguaje natural, es el conocido como *tokenizing* o lexematización. Son muchos los modelos e implementaciones disponibles: [Covington,94; pp.318-320], [Abramson-Dahl, 89; pp.64-65], [McCord,87; pp.333-334.], ...

S-Lendex usa la de [Sarabia,93] por ajustarse muy adecuadamente a la noción de lexema definida en G_{Lex} ¹¹². Un procedimiento asociado a éste, guiones/2, del mismo autor, es el consistente en la eliminación de los guiones superfluos debidos a la partición de palabras al final de línea.

¹¹² Recordémosla en la p.32

Este el significado de ambos procedimientos:

/ l_lexemas(S, L) : L es la lista de los lexemas de la secuencia S de acuerdo con la definición de G_{Lex} */*

/ guiones(L1, L2): L2 es la lista de lexemas que resulta de eliminar de la lista L1 los guiones superfluos y unir los lexemas correspondientes */*

Procedimientos posteriores al análisis de los documentos

El procedimiento *analisis/4* recibe una lista de lexemas como *input* y devuelve el contenido informativo del documento como *output*. Este primer resultado del análisis expresa la extensión y contenido informativo de las áreas del documento en términos de listas de lexemas. Se hace necesario un procedimiento que efectúe la operación inversa a la lexematización; esto es, genere a partir de las listas de lexemas secuencias únicas: el primitivo de Arity Prolog *concat/2* sirve a estos efectos.

Adicionalmente y de manera previa a esta operación, sometemos a las listas de lexemas a un proceso de limpieza de fútiles a cabeza y cola. Denominamos al procedimiento que lo efectúa *aseo/2*.

Aplicaciones

El objetivo de este capítulo es el de mostrar la flexibilidad de S-Lendex en el tratamiento de distintas clase de documentos. Organizamos lo que sigue en dos apartados:

- Variaciones sobre el análisis de una clase de documentos
- Análisis de otras clases de documentos

En las GDUs obviamos la indicación de las bases de datos léxicas necesarias para el análisis de las distintas clases de documentos con el fin de centrar la atención en el conjunto de hechos y reglas característico de LENDEX.

Variaciones sobre el análisis de una clase de documentos

El ejemplo que vamos a desarrollar trata de poner de manifiesto algunas de las posibilidades expresivas de LENDEX.

Partimos de un documento original perteneciente a [Villancicos,90]. Éste consiste en un catálogo de villancicos y oratorios de los siglos XVIII y XIX recopilados por la Biblioteca Nacional en el año 1990. Esta clase de documentos es la elegida como ejemplo de aplicación en el proyecto BiblioTECA¹¹³.

Aquí proponemos distintas GDUs que la describen con el fin de apreciar la funcionalidad de S-Lendex.

¹¹³Véase la p.13 donde éste es aludido

Documento original

Un ejemplo de esta clase es el siguiente:

263

SALAZAR, Diego José. Letras de los villancicos, que se cantaron en la... Iglesia Metropolitana y Patriarcal... de Sevilla, en los solemnes Maytines del Nacimiento de Nuestro Señor Jesu-Christo... de 1706 / compuestos por D. Diego Joseph de Salazar, Racionero y Maestro de Capilla...—Impresso en Sevilla: por Juan Francisco de Blas, 1706.—[8] p.; 4°
Sign.: A*-4.—Datos de pie de imp. tomados del colofón.—Texto a dos col.

Contiene:

1. Villancico Primero. De Kalenda: "Para libertar su Reyno..." [Int., Estr. y Coplas]
2. Villancico II: "Zagalejas al Niño adoremos..." [Estr. y Coplas]
3. Villancico III: "Con varias mercaderias..." [Int., Estr. y Coplas]
4. Villancico IV: "Pajarillo canoro..." [Estr. y Coplas]
5. Villancico V. Negro. "A ziolo Flazico ezcucheme?..." [Estr. y Coplas]
6. Villancico VI: "Cytaras de cristal..." [Estr. y Coplas]
7. Villancico VII: "Por divertir à el Dios Niño..." [Int., Estr. y Coplas]
8. Villancico VIII: "Niño mio, que al yelo..." [Estr. y Coplas]
9. Villancico IX: "A Què tocan?..." [Estr. y Coplas]

Aguilar: Imp. sev., 88.
· VE/ 531-37

114

....

¹¹⁴Otros documentos de esta clase se encuentran en el Anexo 2, p.281

1ª GDU

La primera gramática que presentamos distingue en el documento cuatro grandes áreas. Nótese la definición del área DD villa: tiene como fin asociado un área ED, v, en cuya definición aparece ella misma.

```

documento ==> [registro, cuerpo, contenido, resto].

area_dd(villa, true).
area_dd(resto, true).
area_dd(cuerpo, true).

fin(villa, v, externo_homo).
fin(villa, (fl,fl), interno).
fin(cuerpo, cont, externo_hetero).
registro => * futil, numeral, fl.
contenido => cont, villancicos.
cont => $Contiene$, +/- dospuntos_sp.
villancicos => &/ v.
v => ? fl, num, villa.
num => numeral, hasta_d(2,espacio), +/- punto_sp.

```

Resultado

registro(\$263\$)

cuerpo(\$SALAZAR, Diego José. Letras de los villancicos, que se cantaron en la... Iglesia Metropolitana y Patriarcal... de Sevilla, en los solemnes Maytines del Nacimiento de Nuestro Señor Jesu-Christo... de 1706 / compuestos por D. Diego Ioseph de Salaçar, Racionero y Maestro de Capilla...-- Impreso en Sevilla: por Juan Francisco de Blas, 1706.--[8] p.; 4º Sign.: A*-4.--Datos de pie de imp. tomados del colofón.--Texto a dos col\$),

contenido([

cont(\$Contiene:\$),

villancicos([

v([

num(\$1\$),

villa(\$Villancico Primero. De Kalenda: "Para libertar su Reyno..." [Int., Estr. y Coplas]\$))),

```

v([
  num($2$),
  villa($Villancico II: "Zagalejas al Niño adoremos..." [Estr. y
    Coplas]$)),
v([
  num($3$),
  villa($Villancico III: "Con varias mercaderias..." [Int., Estr. y
    Coplas]$)),
v([
  num($4$),
  villa($Villancico IV: "Pajarillo canoro..." [Estr. y Coplas]$))),
v([
  num($5$),
  villa($Villancico V. Negro. "A ziolo Flazico ezcucheme?..." [Estr.
    y Coplas]$)),
v([
  num($6$),
  villa($Villancico VI: "Cytaras de cristal..." [Estr. y Coplas]$)),
v([
  num($7$),
  villa($Villancico VII: "Por divertir à el Dios Niño..." [Int., Estr. y
    Coplas]$)),
v([
  num($8$),
  villa($Villancico VIII: "Niño mio, que al yelo..." [Estr. y
    Coplas]$)),
v([
  num($9$),
  villa($Villancico IX: "A Què tocan?..." [Estr. y Coplas]$))
])

```

resto(\$Aguilar: Imp. sev., 88. · VE/ 531-37\$).

... 115

¹¹⁵El resultado del análisis sobre otros documentos de esta clase, en la p. 306

2ª GDU

Reducimos un nivel el árbol de análisis asociado al área contenido con la eliminación de las áreas cont y villancicos.

```

documento ==> [registro, cuerpo, contenido, resto].

area_dd(villa, true).
area_dd(resto, true).
area_dd(cuerpo, true).

fin(villa, v, externo_homo).
fin(villa, (fl,fl), interno).
fin(cuerpo, $Contiene$, externo_hetero).

registro => * futil, numeral, fl.
contenido => $Contiene$, +/ dospuntos_sp, &/ v.
contenido -> cont, villancicos.
cont -> $Contiene$, +/ dospuntos_sp.
villancicos -> &/ v.

v => ? fl, num, villa.
num => numeral, hasta_d(2,espacio), +/ punto_sp.
    
```

Resultado

```

registro($263$)

cuerpo($SALAZAR, Diego José. Letras de los villancicos, que ... $)

contenido([
  $Contiene$, $:$,
  v([
    num($1$),
    villa($Villancico Primero. De Kalenda: "Para libertar su Reyno..." [Int.,
      Estr. y Coplas$])),
  v([
    num($2$),
    villa($Villancico II: "Zagalejas al Niño adoremos..." [Estr. y Coplas$])),
  v([
    num($3$),
    villa($Villancico III: "Con varias mercaderias..." [Int., Estr. y Coplas$])),
  ...
])

resto($Aguilar: Imp. sev., 88. · VE/ 531-37$).
    
```

3ª GDU

Declaramos área R la introducción al área contenido:

```

documento ==> [registro, cuerpo, contenido, resto].

area_dd(villa, true).
area_dd(resto, true).
area_dd(cuerpo, true).

fin(villa, v, externo_homo).
fin(villa, (fl,fl), interno).
fin(cuerpo, $Contiene$, externo_hetero).

registro => */ futil, numeral, fl.
contenido => ? ($Contiene$, +/ dospuntos_sp), &/ v.

v => ? fl, num, villa.
num => numeral, hasta_d(2,espacio), +/ punto_sp.
    
```

Sería equivalente a conservar el área cont de la 1ª GDU y definir contenido así:

```

contenido => ? cont, &/ v.
    
```

Resultado

```

registro($263$)

cuerpo($SALAZAR, Diego José. Letras de los villancicos, que ... $)

contenido([
  v([
    num($1$),
    villa($Villancico Primero. De Kalenda: "Para libertar su Reyno..." [Int.,
      Estr. y Coplas]$))),
  v([
    num($2$),
    villa($Villancico II: "Zagalejas al Niño adoremos..." [Estr. y Coplas]$))),
  v([
    num($3$),
    villa($Villancico III: "Con varias mercaderias..." [Int., Estr. y Coplas]$))),
  ...
])

resto($Aguilar: Imp. sev., 88. - VE/ 531-37$).
    
```

4ª GDU

Convertimos cada uno de los villancicos del documento - área ED v - en áreas del mismo nivel que registro, cuerpo y resto, mediante su incorporación a la producción principal de la GDU:

```

documento ==> [registro, cuerpo, ? ($Contiene$, +/ dospuntos_sp), &/v,
resto].

area_dd(villa, true).
area_dd(resto, true).
area_dd(cuerpo, true).

fin(villa, v, externo_homo).
fin(villa, (fl,fl), interno).
fin(cuerpo, $Contiene$, externo_hetero).

registro => * futil, numeral, fl.
contenido => $Contiene$, +/ dospuntos_sp, &/ v.

v => ? fl, num, villa.
num => numeral, hasta_d(2,espacio), +/ punto_sp.
    
```

Resultado

```

registro($263$)

cuerpo($SALAZAR, Diego José. Letras de los villancicos, qve ... $)

v([
  num($1$),
  villa($Villancico Primero. De Kalenda: "Para libertar su Reyno..." [Int., Estr. y
  Coplas]$)),
v([
  num($2$),
  villa($Villancico II: "Zagalejas al Niño adoremos..." [Estr. y Coplas]$)),
v([
  num($3$),
  villa($Villancico III: "Con varias mercaderias..." [Int., Estr. y Coplas]$)),
...
])

resto($Aguilar: Imp. sev., 88. · VE/ 531-37$).
    
```

5ª GDU

Recuperamos el área `cont` de la 1ª GDU -hace más legible la producción principal- y establecemos una condición adicional sobre el área `DD villa` de carácter debil: debe aparecer en su extensión el lexema "Niño". Importa reparar en el área `inc` que esto genera cuando la condición no se cumple.

```

documento ==> [registro, cuerpo, ? cont, &/ v, resto].

area_dd(villa, true).
area_dd(resto, true).
area_dd(cuerpo, true).

fin(villa, v, externo_homo).
fin(villa, (fl,fl), interno).
fin(cuerpo, cont, externo_hetero).

condicion(villa, X^member($Niño$,X), final).

registro => * futil, numeral, fl.
cont => $Contiene$, ? (+/ dospuntos_sp).
v => ? fl, num, villa.
num => numeral, hasta_d(2,espacio), +/ punto_sp.

```

Resultado

registro(\$263\$)

cuerpo(\$SALAZAR, Diego José. Letras de los villancicos, que ... \$)

v([
 num(\$1\$),
 inc([villa(\$Villancico Primero. De Kalenda: "Para libertar su Reyno..." [Int.,
 Estr. y Coplas]\$)]))],

v([
 num(\$2\$),
 villa(\$Villancico II: "Zagalejas al Niño adoremos..." [Estr. y Coplas]\$)]),

v([
 num(\$3\$),
 inc([villa(\$Villancico III: "Con varias mercaderias..." [Int., Estr. y Coplas]\$)]))],

...
)

resto(\$Aguilar: Imp. sev., 88. · VE/ 531-37\$).

6ª GDU

Eliminamos información redundante de los villancicos: definimos una nueva área ED: n_romano, y la declaramos área R.

```

documento ==> [registro, cuerpo, ? cont, &/ v, resto].

area_dd(villa, true).
area_dd(resto, true).
area_dd(cuerpo, true).

fin(villa, v, externo_homo).
fin(villa, (fl,fl), interno).
fin(cuerpo, cont, externo_hetero).

registro => * futil, numeral, fl.
cont => $Contiene$, ? (+/ dospuntos_sp).
v => ? fl, num, ? n_romano, villa.
num => numeral, hasta_d(2,espacio), +/- punto_sp.

n_romano => $Villancico$, * espacio,
            ( romano ; ordinal ),
            +/- dospuntos_sp.

ordinal => lexema^orden(lexema).
    
```

Resultado

```

registro($263$)

cuerpo($SALAZAR, Diego José. Letras de los villancicos, qve ... $)

v([
  num($1$),
  villa($De Kalenda: "Para libertar su Reyno..." [Int., Estr. y Coplas]$)]),
v([
  num($2$),
  villa($"Zagalejas al Niño adoremos..." [Estr. y Coplas]$)],
v([
  num($3$),
  villa($"Con varias mercaderias..." [Int., Estr. y Coplas]$)],
...
])

resto($Aguilar: Imp. sev., 88. · VE/ 531-37$).
    
```

7ª GDU

Una alternativa a la definición anterior del área `n_romano`. Hacemos de ella un área `DD`, con la ventaja de la mayor flexibilidad que ofrece -su definición acepta más secuencias que la anterior: nótese la inclusión de la cadena "De Kalenda"- y el inconveniente de depender más de la puntuación concreta que marca su final: los dos puntos.

```

documento ==> [registro, cuerpo, ? cont, &/ v, resto].

area_dd(villa, true).
area_dd(resto, true).
area_dd(cuerpo, true).
area_dd(n_romano, true).

fin(villa, v, externo_homo).
fin(villa, (fl,fl), interno).
fin(cuerpo, cont, externo_hetero).
fin(n_romano, $:$, interno).

registro => * futil, numeral, fl.
cont => $Contiene$, ? (+/ dospuntos_sp).
v => ? fl, num, ? n_romano, villa.
num => numeral, hasta_d(2,espacio), +/ punto_sp.

```

Resultado

```

registro($263$)

cuerpo($SALAZAR, Diego José. Letras de los villancicos, qve ... $)

v([
  num($1$),
  villa($"Para libertar su Reyno..." [Int., Estr. y Coplas]$)),
v([
  num($2$),
  villa($"Zagalejas al Niño adoremos..." [Estr. y Coplas]$)),
v([
  num($3$),
  villa($"Con varias mercaderias..." [Int., Estr. y Coplas]$)),
...
])

resto($Aguilar: Imp. sev., 88. · VE/ 531-37$).

```

8ª GDU

Distinguiamos en el área villa el título del villancico -que seguimos denominando villa- de las partes de que consta: area DD partes.

```

documento ==> [registro, cuerpo, ? cont, &/ v, resto].

area_dd(villa, true).
area_dd(resto, true).
area_dd(cuerpo, true).
area_dd(n_romano, true).
area_dd(partes, true).

fin(villa, $$, externo_hetero).
fin(partes, v, externo_homo).
fin(partes, (fl,fl), interno).
fin(cuerpo, cont, externo_hetero).
fin(n_romano, $$, interno).

registro => * futil, numeral, fl.
cont => $Contiene$, ? (+/ dospuntos_sp).
v => ? fl, num, ? n_romano, villa, partes.
num => numeral, hasta_d(2,espacio), +/ punto_sp.

```

Resultado

```

registro($263$)

cuerpo($SALAZAR, Diego José. Letras de los villancicos, que ... $)

v([
  num($1$),
  villa($"Para libertar su Reyno..."$),
  partes($[Int., Estr. y Coplas]$)),
v([
  num($2$),
  villa($"Zagalejas al Niño adoremos..."$),
  partes($[Estr. y Coplas]$)),
v([
  num($3$),
  villa($"Con varias mercaderias..."$),
  partes($[Int., Estr. y Coplas]$)),
...
])

resto($Aguilar: Imp. sev., 88. · VE/ 531-37$).

```

9ª GDU. última versión.

Analizamos el cuerpo del registro en las distintas áreas principales de que consta.

```
documento ==> [registro, +/- autor, titulo, resp, pie_imprenta, d_fisica,
                signatura, notas, ? cont, &/ v, resto].
```

```
area_dd(villa, true).
```

```
area_dd(cuerpo, true).
```

```
area_dd(resto, true).
```

```
area_dd(n_romano, true).
```

```
area_dd(partes, true).
```

```
area_dd(apell, en_may).
```

```
area_dd(nombre, true).
```

```
area_dd(titulo, true).
```

```
area_dd(resp, true).
```

```
area_dd(pie_imprenta, true).
```

```
area_dd(d_fisica, true).
```

```
area_dd(signatura, true).
```

```
area_dd(notas, true).
```

```
fin(villa, $$, externo_hetero).
```

```
fin(partes, v, externo_homo).
```

```
fin(partes, (fl,fl), interno).
```

```
fin(cuerpo, cont, externo_hetero).
```

```
fin(n_romano, $$, interno).
```

```
fin(apell, coma_sp, externo_hetero).
```

```
fin(nombre, punto, interno).
```

```
fin(titulo, $$, interno).
```

```
fin(resp, $-$, interno).
```

```
fin(pie_imprenta, $-$, interno).
```

```
fin(d_fisica, (fl, desde(2, espacio), interno).
```

```
fin(signatura, $-$, interno).
```

```
fin(notas, cont, externo_hetero).
```

```
registro => * futil, numeral, fl.
```

```
cont => $Contiene$, ? (+/ dospuntos_sp).
```

```
v => ? fl, num, ? n_romano, villa, partes.
```

```
num => numeral, hasta_d(2, espacio), +/- punto_sp.
```

```
autor => apell, ? coma_sp, nombre.
```


Resultado

```
registro($263$)
autor([
  apell($SALAZAR$),
  nombre($Diego José$)
]),
titulo($Letras de los villancicos, que se cantaron en la... Iglesia Metropolitana y
  Patriarcal... de Sevilla, en los solemnes Maytines del Nacimiento de
  Nuestro Señor Jesu-Christo... de 1706$),
resp($compuestos por D. Diego Ioseph de Salazar, Racionero y Maestro de
  Capilla$),
pie_imprensa($Impresso en Sevilla: por Juan Francisco de Blas, 1706$),
d_fisica($[8] p.; 4° $),
signatura(Sign.: A*-4$),
notas($Datos de pie de imp. tomados del colofón.--Texto a dos col$),
v([
  num($1$),
  villa($"Para libertar su Reyno..."$),
  partes($[Int., Estr. y Coplas]$)),
v([
  num($2$),
  villa($"Zagalejas al Niño adoremos..."$),
  partes($[Estr. y Coplas]$)),
v([
  num($3$),
  villa($"Con varias mercaderias..."$),
  partes($[Int., Estr. y Coplas]$)),
  ...
])
resto($Aguilar: Imp. sev., 88. · VE/ 531-37$).
```

Análisis de otras clases de documentos

Los ejemplos aquí desarrollados corresponden a los docinput del Anexo 2. En el Anexo 3 se encuentra el resultado global de su análisis. Los documentos originales de este apartado son los primeros de cada clase.

Interesa reparar en la presencia de clases de documentos de muy diverso tipo. La intención es poner de manifiesto la generalidad de S-Lendex y su capacidad de adaptación a intereses muy distintos.

Indice del catálogo de discos

El siguiente ejemplo procede del índice de intérpretes de [Discos,88; p.883]. Éste es un catálogo que reúne los fondos musicales de la Biblioteca Nacional de discos de 78 r.p.m. Los números de registro que acompañan a cada una de las entradas representan los asientos bibliográficos a los que ellas se refieren.

Documento original

Ejemplos de esta clase son los siguientes:

\$MACCORMAC, Pat: 4, 1.868.\$
\$MACDONALD, Jeannette: 518, 806.\$
\$MACINTIRE, Lani: 5.497.\$
\$MACIVER, Allan: 3.829, 4.128\$

... 116
...

¹¹⁶Véase la p.278, donde hay más documentos de esta clase

GNU

```

documento ==> [autor, regs].

area_dd(dir, true).
area_dd(nombre, true).
area_dd(apell, en_may).

fin(X, $$, interno).
fin(apell, $$, externo_hetero).

condicion(nombre, [H_]^lex(H, con_may), fuerte).

autor => inv ; dir.

inv => apell, ? coma_sp, nombre.

regs => &/ reg.

reg => ? (+/ dospuntos_sp),
      ((numeral^entre(numeral,0,7), ? (+/ punto_sp), numeral)
      ;
      numeral),
      ? (* futil_sp).
    
```

Resultado

```

autor([
  inv([
    apell($MACCORMAC$),
    nombre($Pat$)
  ]),
])

regs([
  reg($4$),
  reg($1868$)
])

-----

autor([
  inv([
    apell($MACDONALD$),
    nombre($Jeannette$)
  ]),
])
    
```

```
regs([
  reg($518$),
  reg($806$)
])
```

```
autor([
  inv([
    apell($MACINTIRE$),
    nombre($Lani$)
  ])
])
```

```
regs([
  reg($5497$)
])
```

```
autor([
  inv([
    apell($MACIVER$),
    nombre($Allan$)
  ])
])
```

```
regs([
  reg($3829$),
  reg($4128$)
])
```

... 117
...

¹¹⁷En la p.302 está el resto de documentos output

Base de datos documental

El que sigue es un ejemplo de documento input procedente de una exportación desde base de datos documental: Knosys. Cada uno de los documentos representados es un registro bibliográfico consituido por una secuencia de campos que *grosso modo* coinciden con los del formato ISBD. En ellos puede observarse los rótulos correspondientes a cada uno de los campos de la base y su contenido.

Documento original

\$
Signatura=4747
CDU:
Encabezamiento=BLASCO GASCO, Francisco de P.
Título_Uniforme=
Título_Mención=Objeto de hipoteca y ejecución hipotecaria : determinación legal... el objeto hipotecado : doctrina y jurisprudencia / Francisco de P. Blasco Gasco ; prólogo Vicente Montés Penedés
Edición:
Publicación=Valencia : Edit. General de Derecho, D.L. 1994
Colación:208 p.; 24 cm.
Serie=
Notas=
ISBN/ISSN=84-605-0043-8
NIPO=
Materias=1. Hipotecas - Espana
Secundarias=I. Título
Registro/s=4747
Anotaciones=
\$

... 118
...

¹¹⁸Véase la p.276, donde hay más documentos de esta clase

GDU

```
documento ==> [? (* espaciofl), &/ campo].

area_dd(nombre, true).
area_dd(valor, true).

fin(nombre, ($:$ ; $=$), externo_hetero).
fin(valor, fl, externo_hetero).

campo => ? (+/ fl), nombre, ? (+/ futil), valor.
```

Resultado

```
campo([
  nombre($Signatura$),
  valor($4747$)]),

campo([
  nombre($CDU$),
  valor($$)]),

campo([
  nombre($Encabezamiento$),
  valor($BLASCO GASCO, Francisco de P.$)]),

campo([
  nombre($Título_Uniforme$),
  valor($$)]),

campo([
  nombre($Título_Mención$),
  valor($Objeto de hipoteca y ejecución hipotecaria : determinación legal... el
  objeto hipotecado : doctrina y jurisprudencia / Francisco de P. Blasco
  Gasco ; prólogo Vicente Montés Penedés$)]),

campo([
  nombre($Edición$),
  valor($$)]),

campo([
  nombre($Publicación$),
  valor($Valencia : Edit. General de Derecho, D.L. 1994$)]),

campo([
  nombre($Colación$),
  valor($208 p.; 24 cm$)]),

campo([
  nombre($Serie$),
  valor($$)]),
```

```
campo([
  nombre($Notas$),
  valor($$)],

campo([
  nombre($ISBN/ISSN$),
  valor($84-605-0043-8$)],

campo([
  nombre($NIPO$),
  valor($$)],

campo([
  nombre($Materias$),
  valor($1. Hipotecas - Espana$)],

campo([
  nombre($Secundarias$),
  valor($1. Título$)],

campo([
  nombre($Registro/s$),
  valor($4747$)],

campo([
  nombre($Anotaciones$),
  valor($$)],

campo([
  nombre($$),
  valor($$)].
```

... 119

¹¹⁹El resto de documentos output se encuentra en la p. 300

Diccionario

Los que siguen son documentos procedentes de [DRAE,70]. Se trata en esta ocasión de poner de manifiesto la capacidad de S-Lendex para enfrentarse a clases de documentos no necesariamente relacionadas con el ámbito bibliográfico.

Documento original

Ejemplos de esta clase son los siguientes:

\$florales. (Del lat. *florales ludi*, juegos florales.) adj. pl. Aplícase a las fiestas o juegos que celebraban los gentiles en honor de la diosa Flora. A su imitación se han instituido después en Provenza y en otras partes. || 2. V. juegos florales.\$

\$florar. intr. Dar flor. Dícese de los árboles y las plantas, singularmente de los que se cultivan para cosechar sus frutos.\$

\$flordelisado, da. adj. Blas. V. cruz flordelisada.\$

\$flordelisar. tr. Blas. Adornar con flores de lis una cosa.\$

... 120

¹²⁰En la p.285 hay más documentos ed esta clase

GNU

```

documento ==> [palabra, +/- origen, ? (* spfl), &/ acepcion].

area_dd(palabra, true).
area_dd(origen, true).
area_dd(sigdo,true).

fin(palabra, $$, interno).
fin(origen, $$, interno).

fin(sigdo, num, externo_hetero).
fin(sigdo, usase, externo_hetero).

condicion(origen, X^member($($,X), fuerte).

acepcion =>    +/- num, ? (* spfl),
               +/- aux, ? (* spfl),
               (vease ; sigdo),
               +/- usase.

num => $||$, * spfl, numeral, * sp, +/- $$$.

aux =>    &/ (    lexema^abr(lexema, _), +/- $$$, * spfl,
               +/- (($y $ ; $o $ ; $de $), lexema, +/- $$) ).

vease => ? ($V.$), sigdo.
usase => $ú.$, +/- spfl, &/ (lexema^abr(lexema,_) , +/- punto_sp).

```

Resultado

palabra(\$florales\$),

origen(\$(Del lat. florales ludi, juegos florales.)\$),

acepcion([

aux(\$adj. pl.\$),

sigdo(\$Aplicase a las fiestas o juegos que celebraban los gentiles en honor de la diosa Flora. A su imitación se han instituido después en Provenza y en otras partes\$)),

acepcion([

num(\$2\$),

vease([

sigdo(\$juegos florales\$)))]

]).

palabra(\$florar\$)

```
acepcion([
  aux($intr.$),
  sigdo($Dar flor. Dicese de los árboles y las plantas, singularmente de los que
    se cultivan para cosechar sus frutos$)
])
```

palabra(\$flordelisado, da\$),

```
acepcion([
  aux($adj. Blas.$),
  vease([
    sigdo($cruz flordelisada$))
  ])
])
```

palabra(\$flordelisar\$),

```
acepcion([
  aux($tr. Blas.$),
  sigdo($Adornar con flores de lis una cosa$))
])
```

... 121

¹²¹Cfr. la p.310 para ver más entradas analizadas

BOE

Los siguientes son resoluciones del BOE nº125, fecha 26-Mayo-95, p.15387, correspondientes al apartado de Universidades. El análisis propuesto permitiría generar una base de datos de manera prácticamente inmediata.

Documento original

Documentos de esta clase son los siguientes:

\$Resolución de 5 de mayo de 1995, de la Universidad de Vigo, por la que se nombra Catedrática de Escuela Universitaria del área de conocimiento de "Filología Francesa", del departamento de Filología Francesa, a doña Terencia Inés Silva Rojas\$

\$Resolución de 20 de abril de 1995, de la Universidad de Málaga, por la que se nombra Profesora de la misma a doña Eva Millán Valideperas\$

\$Resolución de 20 de abril de 1995, de la Universidad de Málaga, por la que se nombra Profesor de la misma a don Carlos Rossi Jiménez\$

...¹²²
...

¹²²Más documentos de esta clase se encuentran en la p.289

GDU

```

documento ==> [? res, fecha, ? i, uni, ? i, info, ? i].

area_dd(i, true).

fin( i, cargo, externo_hetero).
fin( i, entidad, externo_hetero).
fin( i, "$", externo_hetero).
fin( i, ($don$ ; $doña$), externo_hetero).

info => (categoria, ? i, +/- area, ? i, nombre)
        ;
        (nombre, ? i, categoria, ? i, +/- area).

res => ? ( * espacio, $Resolución de $).

fecha => numeral^entre(numeral,1,31), $ de $,
        lexema, $ de $,
        numeral^menor(numeral, 1996).

uni => $Universidad$, &\( lexema^(lex(lexema, con_may) ; sctg(lexema))).

categoria => cargo, ? i, +/- entidad.

cargo => $Profesora titular$ ; $Profesor titular$ ; lexema^cargo(lexema).

entidad => $Universidad$ ; ($Escuela$, futil, $Universitaria$).

area => "$", &/(\(lexema^(lexema \= "$")), "$".

nombre => ? ($don$ ; $doña$),
        desde_d(3, lexema^(not cargo(lexema),
                          (lex(lexema, con_may) ; sctg(lexema)
                          )
        ).

```

Resultado

```
fecha($5 de mayo de 1995$),  
uni($Universidad de Vigo$),  
info(  
  categoria(  
    cargo($Catedrática$),  
    entidad($Escuela Universitaria$)  
  )  
  area($"Filología Francesa"$),  
  nombre($Terencia Inés Silva Rojas$))  
).
```

```
-----  
fecha($20 de abril de 1995$),  
uni($Universidad de Málaga$),  
info(  
  categoria(  
    cargo($Profesora$)  
  )  
  nombre($Eva Millán Valideperas$))  
).
```

```
-----  
fecha($20 de abril de 1995$),  
uni($Universidad de Málaga$),  
info(  
  categoria(  
    cargo($Profesor$)  
  )  
  nombre($Carlos Rossi Jiménez$))  
).
```

```
... 123
```

¹²³Otros documentos output están en la p.315

Anuncios de prensa

La clase de documentos que a continuación se propone corresponde a la sección de venta de pisos del periódico El Mundo de fecha 7-Junio-95. Es interesante notar el análisis efectuado del área DD descripción.

Documento original

Ejemplos de esta clase son los siguientes:

\$CANILLEJAS, San Hilario, cuarto, exterior, 120 metros. cuatro dormitorios, dos baños, agua y calefacción central, ascensores, pisazo. 23.000.000. T. 91/4305412.\$

\$ARTURO SORIA, principio, cuatro dormitorios, dos baños, jardines, piscina. T. 91/5165921.\$

\$CENTRIQUISIMO, metro Lavapies, tres dormitorios, todo exterior, 1.500.000. Grandísimas facilidades. T. 5763030.\$

... 124

¹²⁴Cfr. la p.293 para ver más documentos de esta clase

GDU

```
documento ==> [(reclamo ; zona), descripcion,  
               ? (* futil), +/- precio, +/- comentario, +/- telefono].  
  
area_dd(zona, true).  
area_dd(descripcion, true).  
area_dd(comentario, true).  
  
fin(zona, ($, $ ; $, $), interno).  
fin(X, $T.$, externo_hetero).  
fin(X, (numeral, $, $), externo_hetero).  
  
condicion(zona, [X|_] ^ lex(X, en_may), fuerte).  
  
reclamo => lexema ^ reclamo(lexema).  
  
telefono => ?($T.$, * spfl), +/- prefijo, ?(* spfl), tfno.  
  
prefijo => numeral ^ mayor(numeral, 90), ? barra_dsp.  
  
tfno => numeral ^ string_length(numeral, 7).  
  
precio => numeral ^ (string_length(numeral, N), (N=1; N=2)), $, $,  
           numeral ^ string_length(numeral, 3), $, $,  
           numeral ^ string_length(numeral, 3).  
  
dorms => lexema ^ num(lexema), ? pref_aux($dormit$).  
  
planta => lexema ^ (orden(lexema) ; piso(lexema)).  
  
baños => lexema ^ num(lexema, _), ? pref_aux($baño$).  
  
metros => numeral, ? (* spfl, lexema ^ ext(lexema)).  
  
estado => lexema ^ estado(lexema).  
  
situacion => ($exterior$ ; $interior$).  
  
descripcion -> [planta, dorms, metros, estado, baños, situacion].
```

Resultado

zona(\$CANILLEJAS\$),

descripcion([
 planta(\$cuarto\$),
 dorms(\$cuarto\$),
 metros(\$120\$),
 baños(\$dos\$),
 situacion(\$exterior\$)
]),

precio(\$23.000.000\$),

telefono([
 prefijo(\$91\$),
 tfno(\$4305412\$)])

zona(\$ARTURO SORIA\$),

descripcion([
 dorms(\$cuatro\$),
 baños(\$dos\$)
]),

telefono([
 prefijo(\$91\$),
 tfno(\$5165921\$)])

reclamo(\$CENTRIQUISIMOS\$),

descripcion([
 dorms(\$tres\$),
 situacion(\$exterior\$)
]),

precio(\$1.500.000\$),

comentario(\$Grandísimas facilidades\$),

telefono([
 tfno(\$5763030\$)])

....¹²⁵

¹²⁵Más documentos analizados en la p. 318

Tractatus

Los documentos ahora presentados pertenecen al *Tractatus*, [Wittgenstein,69]. El interés del análisis estriba en posibilitar un acceso flexible a cada uno de los textos de la obra mediante raíces léxicas clave, así como en desarrollar algún tipo de clasificación automática de los mismos atendiendo a idéntico criterio.

Documento original

Ejemplos de esta clase son los siguientes:

\$2.025 Sie ist Form und Inhalt.\$

\$2.0251 Raum, Zeit und Farbe (Färbigkeit) sind Formen der
Gegenstände.\$

\$2.026 Nur wenn es Gegenstände gibt, kann es eine feste Form
der Welt geben.\$

\$2.027 Das Feste, das Bestehende und der Gegenstand sind
Eins.\$

...¹²⁶

¹²⁶Más documentos del *Tractatus* se encuentran en la p.295

GDU

```
documento ==> [dictum].

area_dd(texto, true).
area_dd(numero, numeral).

fin(numero, lex_con_may, externo_hetero).

dictum => numero, r_side(R), texto, apr_side(R), claves.

claves => * (c ; ? lexema).

c => lexema^clave(lexema).

clave(X):-
    normaliza(X, X1),
    tr(W),
    infijo(estricto, W, X1).
```

Resultado

```
dictum([
    numero($2.025$),
    texto($Sie ist Form und Inhalt$),
    claves([
        c($Form$),
        c($Inhalt$)
    ])
]).
```

```
dictum([
    numero($2.0251$),
    texto($Raum, Zeit und Farbe (Färbigkeit) sind Formen der Gegenstände$),
    claves([
        c($Raum$),
        c($Zeit$),
        c($Farbe$),
        c($Färbigkeit$),
        c($Formen$),
        c($Gegenstände$)
    ])
]).
```

```
dictum([
  numero($2.026$),
  texto($Nur wenn es Gegenstände gibt, kann es eine feste Form der Welt
  geben$),
  claves([
    c($Gegenstände$),
    c($feste$),
    c($Form$),
    c($Welt$)
  ])
]).
```

```
dictum([
  numero($2.027$),
  texto($Das Feste, das Bestehende und der Gegenstand sind Eins$),
  claves([
    c($Feste$),
    c($Bestehende$),
    c($Gegenstand$)
  ])
]).
```

.....¹²⁷

¹²⁷Otros documentos de esta clase, en la p.321

Revistas

La clase a continuación analizada corresponde a la obra [Indice,91] publicada por el CSIC. Se trata en este caso de un índice de revistas de investigación en Ciencias Sociales a las que se asocia el ISSN, *International Standard Series Number*, la institución que publica y su dirección.

Documento original

Ejemplos de esta clase son los siguientes:

\$ABACO 0213-6252 CENTRO DE ESTUDIOS ECONOMICOS
Y SOCIALES SAN BERNARDO, 40, 5º, D. 33201 GIJON (ASTURIAS)\$

\$ADICCIONES. REVISTA DE SOCIDRO-GALCOHOL 0214-
4840 SOCIODROGALCOHOL RAMBLA, 15, 2º Y 3º 07303 PALMA DE
MALLORCA\$

\$ADUANAS. REVISTA DE COMERCIO IN-TERNACIONAL Y ESTUDIOS
FISCALES 0400-5732 CONS. DE ESTUDIOS ADUANEROS Y CO-
MERCIO EXTERIOR EVARISTA SAN MIGUEL, 10 28008 MADRID\$

... 128

¹²⁸Otros documentos de esta clase, en la p.298

GNU

```
documento ==> [? (* espaciofl), titulo, issn, editor, donde].
```

```
area_dd(titulo, true).  
area_dd(issn, numeral).  
area_dd(editor, true).  
area_dd(donde, true).  
area_dd(direccion, true).
```

```
fin(X, (fl,fl), interno).  
fin(titulo, numeral^string_length(numeral,4), externo_hetero).  
fin(direccion, c_postal, externo_hetero).
```

```
c_postal => numeral^string_length(numeral,5).  
ciudad => &/ lexema.
```

```
donde -> [direccion, c_postal, ciudad].
```

Resultado

```
titulo($ABACOS$),
```

```
issn($0213-6252$),
```

```
editor($CENTRO DE ESTUDIOS ECONOMICOS Y SOCIALES$),
```

```
donde([  
  direccion($SAN BERNARDO, 40, 5°, D.$),  
  c_postal($33201$),  
  ciudad($GIJON (ASTURIAS)$)  
])
```

```
titulo($ADICCIONES. REVISTA DE SOCIDROGALCOHOL$),
```

```
issn($0214-4840$),
```

```
editor($SOCIODROGALCOHOL$),
```

```
donde([  
  direccion($RAMBLA, 15, 2° Y 3° $),  
  c_postal($07303$),  
  ciudad($PALMA DE MALLORCA$)  
])
```

titulo(\$ADUANAS. REVISTA DE COMERCIO INTERNACIONAL Y ESTUDIOS
FISCALES\$),

issn(\$0400-5732\$),

editor(\$CONS. DE ESTUDIOS ADUANEROS Y COMERCIO EXTERIOR\$),

donde(
 direccion(\$EVARISTA SAN MIGUEL, 10\$),
 c_postal(\$28008\$),
 ciudad(\$MADRID\$)
)

... 129

¹²⁹Otros documentos analizados se encuentran en la p. 326

Conclusiones

El objetivo del trabajo consistía en el desarrollo de un sistema de análisis documental automático capaz de enfrentarse a clases distintas de documentos caracterizados por una estructura o formato describible gramaticalmente. Dos tareas fundamentales se hacían necesarias:

- Definir un lenguaje que permitiera describir clases distintas de documentos de manera sencilla y ágil
- Desarrollar la herramienta que -incorporando un intérprete de las descripciones- generase un analizador para cada una de las familias de documentos, expresándolos finalmente en un formato uniforme de representación

Una vez definido LENDEX e implementado S-Lendex, se ha hecho preciso probar la capacidad del lenguaje y del programa con clases diversas de documentos. Se ha procurado para ello contemplar familias de características muy distintas: catálogos de publicaciones, índices, anuncios de prensa, resoluciones del BOE, entradas de diccionario, etc.

Los resultados son alentadores. Con descripciones sencillas se ha conseguido un análisis satisfactorio de cada una de las clases, observándose la flexibilidad del lenguaje para adaptarse -con poco esfuerzo- a distintos fines, así como su capacidad para sobreponerse a incidencias en el análisis de índole diversa: categorías gramaticales fallidas, secuencias de texto indeterminadas, etc.

Por otra parte, hemos desarrollado dos versiones de S-Lendex:

1. Una primera con dos características fundamentales:

- 1.1. La interpretación de las expresiones LENDEX se sostiene hasta el final, con el consiguiente gasto de recursos y la ventaja del desarrollo más directo
- 1.2. El programa genera automáticamente gramáticas reconocedoras para analizar las áreas DD
2. Otra en que:
 - 2.1. Se efectúa una compilación previa de las expresiones específicas del lenguaje con el fin de evitar repetir operaciones posiblemente costosas
 - 2.2. Las gramáticas generadas por el programa para las áreas DD son analizadoras

La evaluación de los resultados con una y otra versión no deja lugar a dudas: la segunda supone un considerable aumento de eficacia del sistema. Para la misma clase de documentos e idéntica descripción, la diferencia ronda en ocasiones el 50% en tiempo de ejecución.

Varias expectativas de desarrollo quedan abiertas. La primera se relaciona con una mejora de la capacidad expresiva de LENDEX: observamos que un buen número de descripciones admitiría una simplificación aún mayor. Pensemos por ejemplo en el modo de expresar las áreas DD, casi todas ellas con el rasgo true asociado. Asimismo puede ser interesante aumentar el conjunto de predefinidos LENDEX con el fin de evitar al usuario definir lexemas tan comunes como artículos, conjunciones y preposiciones.

Otra línea de desarrollo -comentada ya en su momento- es la relativa al carácter más o menos determinista del sistema. Como entonces indicamos, se ha procurado efectuar un desarrollo flexible del mismo desde dos puntos de vista:

1. El tratamiento de los fines de áreas DD se hace en función de la presencia o no en la GDU de una condición fuerte asociada al área: si semejante condición existe, se genera una gramática menos determinista que en caso contrario
2. Se ha dotado a LENDEX de procedimientos gramaticales de tres tipos: determinista, semideterminista e indeterminista con el fin de que sea el usuario quien decida en cada caso cuál es el tratamiento más adecuado para la clase de documentos a analizar

Ya señalamos a este respecto la posibilidad de hacer a S-Lendex más adaptativo dotándole de mecanismos que le permitan decidir cuál es el grado posible de *backtracking* en función de los documentos y la complejidad de la descripción.

Otra extensión interesante de S-Lendex pasa por aumentar el grado de parametrización del resultado del análisis. Esto es: permitir al usuario decidir aún más qué áreas se recuperan y de qué manera. La distinción entre áreas R y RR es un primer paso en esta dirección. Sin embargo, hay una asunción implícita que puede modificarse: áreas DD y ED constituyen indefectiblemente un nodo en el árbol de análisis final. Puede interesar en ocasiones que esto no suceda. Para ello habría que enriquecer la gramática de LENDEX asociando a cada área la indicación correspondiente a este hecho o bien pensar en algún modificador que lo expresara de manera semejante a como están tratadas las áreas R.

Relacionado con este problema, podría también plantearse determinar en la GDU el grado de profundidad general deseado en el árbol de análisis: desde el nivel uno que indicase la descripción plana en términos sólo de las áreas presentes en la producción principal hasta el nivel actual en que se reflejan todas las áreas del documento.

Los problemas derivados de la lectura automática de los documentos por un sistema de *scanner* + OCR han sido obviados. Probada ya la capacidad de S-Lendex para enfrentarse al análisis de textos estructurados carentes de este tipo de

incidencia, una nueva línea de desarrollo es la consistente en ampliar su funcionalidad con herramientas capaces de enfrentarse a este género de problemas: identificación flexible de secuencias, medida y evaluación de semejanzas, etc. Así como integrar en él sistemas de probada utilidad en la identificación de áreas mediante técnicas gramaticales de evaluación de indicios léxicos.

Un asunto vinculado a éste -de indudable relevancia-, es el de la definición, creación y mantenimiento de bases de datos léxicas especializadas en dominios diversos. Concebido S-Lendex como un sistema de propósito general para el análisis automático de documentos estructurados, resulta imposible incorporar *a priori* léxicos adaptados a cada una de las clases por las múltiples formas que éstas pueden presentar: distintos ámbitos, materias e idiomas. Parece más ajustado pensar en el desarrollo de técnicas -en lo posible automáticas- de generación de los léxicos necesarios para el análisis de los documentos.

Todo lo cual invita a contemplar S-Lendex como un primer paso en una línea de investigación con interesantes expectativas de desarrollo en el futuro.

Anexo 1: Una GDU

/* Primera producción de una gramática LENDEX: descripción del documento o clase de documentos a tratar */

```
documento ==> [* espaciofi, sigs, enc, tit, +/- ed, pub, df, serie, isbn,
                &\ secundaria, regs].
```

/* Aserción de las áreas débilmente definidas del documento con indicación del rasgo asociado */

```
area_dd(tit, true).
area_dd(enc, true).
area_dd(ed, true).
area_dd(pub, true).
area_dd(ser, true).
area_dd(sec, true).
area_dd(enc, en_may).
```

/* Aserción de los fines correspondientes a cada una de las áreas DD citadas en la sección anterior, con indicación del tipo de fin: interno, externo_homo o externo_hetero */

```
fin(enc, fl, interno).
fin(ser, parentesis_dsp, externo_hetero).
fin(tit, df, externo_hetero).
fin(tit, desde_d(3,punto), interno).
fin(pub, df, externo_hetero).
fin(pub, (numeral^entre(numeral,1900,1995),*futil), interno).
fin(sec, romano^member(romano,[$!$, $!|$, $!||$]), externo_homo).
fin(sec, regs, externo_hetero).
fin(X, fa, interno).
fin(X, isbn, externo_hetero).
```

/* Aserción de las condiciones asociadas a áreas DD y ED, con indicación de su tipo: fuerte o débil*/

condicion(enc, Z^(algun Z es Q^lex(Q, en_may)), fuerte).
 condicion(ed, Z^(algun Z es Q^(Q = \$ed\$), length(Z,N), N<15), fuerte).

/* Producciones correspondientes a las áreas ED */

```

serie =>   parentesis_esp, ser, parentesis_dsp.

isbn  =>   * espaciofl,
          ? $ISBN$,
          +/ dospuntos_sp,
          &/ (numeral, +/ guion_sp).

sigs  =>   &/ ( +/ (lexema^(lexema= $L$;lexema=$D$)),
              * futil,
              numeral,
              * espaciofl).

regs  =>   $R$, +/ punto_sp,
          &/ (numeral, * espaciofl).

fa    =>   desde_d(2, fl) ;
          (guion, guion) ;
          (punto_sp, &/ guion_sp) ;
          (punto_sp, fl, * espacio).

df    =>   +/ romano, +/ coma_sp, * futil,
          pag, +/ puntoycoma_sp, dim, * espaciofl.

dim   =>   * espacio, numeral^entre(numeral,10,40),
          +/ coma_sp, $5$, * espacio, cm.

pag   =>   numeral^mayor(numeral,21), espaciofl, pag.

cm    =>   $cm$, +/punto.

pg    =>   $p$, +/punto.
    
```

Anexo 2: Documentos Input

Este anexo consiste en la recopilación de documentos pertenecientes a clases distintas en formato input. Para cada una de las clases aquí contempladas se propone una GDU en el apartado *Aplicaciones*, p.236, resultado de cuyo análisis es el Anexo 3.

Base de datos documental

d(3700,\$ □Signatura=4747 □CDU: □Encabezamiento=BLASCO GASCO, Francisco de P. □Título_Uniforme= □Título_Mención=Objeto de hipoteca y ejecución hipotecaria : determinación legal... el objeto hipotecado : doctrina y jurisprudencia / Francisco de P. Blasco Gasco ; prólogo Vicente Montés Penedés □Edición: □Publicación=Valencia : Edit. General de Derecho, D.L. 1994 □Colación:208 p.; 24 cm. □Serie= □Notas= □ISBN/ISSN=84-605-0043-8 □NIPO= □Materias=1. Hipotecas - Espana □Secundarias=I. Título □Registro/s=4747 □Anotaciones= □\$).

d(3701,\$ □Signatura=4772 □CDU: □Encabezamiento=BODEN, Margaret A. □Título_Uniforme= □Título_Mención=La mente creativa : mitos y mecanismos / por Margaret A. Boden □Edición: □Publicación=Barcelona : Gedisa, D.L. 1994 □Colación:404 p. ; 23 cm. □Serie=(Hombre y sociedad. Gladema) □Notas= □ISBN/ISSN=84-7432-500-5 □NIPO= □Materias=1. Creatividad 2. Conocimiento (Psicología) □Secundarias=I. Título □Registro/s=4772 □Anotaciones= □\$).

d(3702,\$ □Signatura=2308, cª 410, p.2307 □CDU: □ Encabezamiento=BORRAJO DACRUZ,Efrén □Título_Uniforme= □Título_Mención=Introducción al derecho espanol del trabajo / Efrén Borrajo Dcruz □Edición:4ª ed. reimp. □Publicación=Madrid : Técnos, 1978 □Colación:XXIII, 455 p. ; 22 cm. □Serie=(Biblioteca Universitaria) □Notas= □ISBN/ISSN=84-309-0549-9 □NIPO= □Materias=1.Derecho del trabajo - Espana □Secundarias=I.Título □Registro/s=410,2307,2308 □Anotaciones= □\$).

d(3703,\$ □Signatura=4385/1, 4386/1 □CDU: □Encabezamiento= BORRELL FONTELLES, José □Titulo_Uniforme= □Titulo_Mención=Métodos matemáticos para la economía / José Borrell Fontelles □Edición: □Publicación=Madrid : Pirámide, 1990- □Colación:V. ; 23cm. □Serie=(Ciencia y técnica) □Notas=T.I: Campos y autosistemas. -- 4ª ed. -- 1990. -- 334 p. -- ISBN:84-368-0182-2 □ISBN/ISSN= □NIPO= □Materias=1. Matemáticas □Secundarias=I. Título □Registro/s=4385, 4386 □Anotaciones= □\$).

d(3704,\$ □Signatura=4767 □CDU: □Encabezamiento=BOTELLA, Juan □Titulo_Uniforme= □Titulo_Mención=Análisis de datos en psicología I / Juan Botella, Orfelio G. León, Rafael San Martín □Edición: □Publicación=Madrid : Pirámide, D.L. 1992 □Colación:399 p. ; 23 cm. □Serie=(Psicología) □Notas= □ISBN/ISSN=84-368-0717-0 (Tomo I) □NIPO= □Materias=1. Estadística matemática □Secundarias=I. León, Orfelio G. II. San Martín, Rafael III. Título □Registro/s=4767 □Anotaciones= □\$).

Índice

d(0,\$MACCORMAC, Pat: 4, 1.868.\$).
d(1,\$MACDONALD, Jeannette: 518, 806.\$).
d(2,\$MACINTIRE, Lani: 5.497.\$).
d(3,\$MACIVER, Allan: 3.829, 4.128,\$).
d(4,\$MACKENZIE, Mike: 6.100.\$).
d(5,\$MACKERSIE, David: 4.304, 4.900, 5.280, 5.306.\$).
d(6,\$MACKINNEY'S COTTON PICKERS: 242.\$).
d(7,\$MACKINTOSH, Ken: 5.000, 5.346, 5.489, 5.537, □ 5.655, 5.741, 5.750, 5.788, 5.955, 6.036.\$).
d(8,\$THE MACKPIESEN: 5.000.\$).
d(9,\$MACHADO: 3.988, 4.111, 4.356, 4.788.\$).
d(10,\$MACHADO, Armando: 3.142.\$).
d(11,\$MACHADO, Leocadio R.: 4.094, 4.178, 4.293.\$).
d(12,\$MACHIN, Antonio: 889, 1.188, 1.236, 1.365, 1.391, □ 1.392, 1.727, 1.766, 1.845, 1.855, 1.925, 2.134, □ 2.142, 2.163, 2.174, 2.188, 2.221, 2.300, 2.305, □ 2.379, 2.395, 2.427, 2.428, 2.439, 2.457, 2.504, □ 2.515, 2.534, 2.540, 2.635, 2.647, 2.659, 2.663, □ 2.765, 2.823, 2.827, 2.837, 2.880, 2.923, 2.930, □ 2.931, 2.949, 2.978, 2.981, 2.994, 3.064, 3.108, □ 3.147, 3.154, 3.165, 3.306, 3.392, 3.448, 3.452, □ 3.520, 3.547, 3.588, 3.647, 3.907, 3.912, 3.929, □ 3.973, 4.187, 4.236, 4.345, 4.506, 4.820, 4.927, □ 5.037, 5.038, 5.335, 5.470, 5.476, 6.175, 6.213, □ 6.256.\$).
d(13,\$MACHITO: 3.189, 3.861, 4.343.\$).
d(14,\$MADRID, Manolito de: 1.460.\$).
d(15,\$MADRIGUERA, Enric: 1.157, 2.875, 2.906, 2.966.\$).
d(16,\$MAGOLI: 4.170.\$).
d(17,\$MAGYARI, Marika: 2.456, 2.541, 2.604, 2.633, □ 2.662, 2.743.\$).
d(18,\$MAIRENA, Antonio: 3.531, 3.616, 4.125, 4.227.\$).
d(19,\$MAIRENA, Pepe: 6.233.\$).

- d(20,\$MALANDO Y SU ORQUESTA: 5.698.\$).
- d(21,\$MALCUZYNSKI, Witold: 3.353, 3.467, 3.507, 3.562, □ 3.563, 3.594, 4.688.\$).
- d(22,\$MALDONADO, F. Z.: 3.982, 4.228.\$).
- d(23,\$MALERAS, Emma: 2.943, 3.493, 3.663, 5.041, 5.477 □ 5.524, 5.557, 5.573, 5.732, 5.804, 5.810, 5.872, □ 6.007.\$).
- d(24,\$MALERBA, Alfredo: 4.131.\$).
- d(25,\$MALKO, Nicolai: 3.217, 3.480, 3.769, 4.018.\$).
- d(26,\$MALVIN, Artie: 3.407.\$).
- d(27,\$MANDARINO: 707, 1.236.\$).
- d(28,\$MANGANO, Silvana: 4.837.\$).
- d(29,\$MANIET, Roger: 4.651.\$).
- d(30,\$MANNARINI, Ida: 1.249, 1.614, 1.675.\$).
- d(31,\$MANOLO EL DE HUELVA: 543, 647.\$).
- d(32,\$MANOLO "EL GAFAS": 1.187, 1.190, 1.621.\$).
- d(33,\$MANOLO EL GRANAINO: 5.351.\$).
- d(34,\$MANOLO EL MALAGUENO: 1.398, 1.488, 1.729, □ 1.854, 2.261, 2.492, 2.498, 2.563, 2.728, 2.746, □ 2.899, 3.019, 3.719, 4.392, 4.395, 4.527, 5.238, □ 5.351, 5.521, 5.527, 5.625, 5.700, 5.721.\$).
- d(35,\$MANONE, Wingie: 661, 1.280, 3.465.\$).
- d(36,\$MANTOVANI, Annunzio: 1.184, 1.308, 1.618, 1.771, □ 1.948, 3.844, 4.205, 4.354, 4.359, 4.504, 4.590, □ 4.611, 4.657, 4.828, 4.905, 5.060, 5.317, 5.405, □ 5.593, 5.645, 5.735, 5.777, 5.831, 5.985.\$).
- d(37,\$MANZANILLA, Manolo: 2.063, 2.263, 2.387, 4.496.\$).
- d(38,\$MANZANITO DE CASTUERA: 2.675.\$).
- d(39,\$MAÑEZ, Encarnita: 2.872, 3.239.\$).
- d(40,\$MARAVIGLIA: 4.206, 4.569, 5.441.\$).
- d(41,\$MARAVILLA, Juanito: 3.253, 3.285, 3.523.\$).
- d(42,\$MARAVILLA, Luis: 1.187, 1.190, 1.288, 1.301, 1.302, □ 1.621, 2.436, 2.568, 2.820, 2.861, 3.953, 4.272, □ 4.582, 4.814, 4.816, 6.243.\$).
- d(43,\$MARCEN, Eduardo: 1.380.\$).

d(44,\$MARC, Jean: 3.257.\$).

d(45,\$MARCO, Jean: 3.221, 4.030.\$).

d(46,\$MARCO, Salome: 5.380.\$).

d(47,\$MARCO, V.: 6.450.\$).

d(48,\$MARCHAND, Colette: 4.902, 5.126.\$).

d(49,\$MARCHENA, Antonio de: 2.736.\$).

Villancicos

d(0,\$
263

SALAZAR, Diego José. Letras de los villancicos, que se cantaron en la... Iglesia Metropolitana y Patriarcal... de Sevilla, en los solemnes Maytines del Nacimiento de Nuestro Señor Jesu-Christo... de 1706 / compuestos por D. Diego Joseph de Salazar, Racionero y Maestro de Capilla...—Impresso en Sevilla: por Juan Francisco de Blas, 1706.—[8] p.; 4º
Sign.: A*-4.—Datos de pie de imp. tomados del colofón.—Texto a dos col.

Contiene:

1. Villancico Primero. De Kalenda: "Para libertar su Reyno..." [Int., Estr. y Coplas]
2. Villancico II: "Zagalejas al Niño adoremos..." [Estr. y Coplas]
3. Villancico III: "Con varias mercaderias..." [Int., Estr. y Coplas]
4. Villancico IV: "Pajarillo canoro..." [Estr. y Coplas]
5. Villancico V. Negro. "A ziolo Flazico ezcucheme?..." [Estr. y Coplas]
6. Villancico VI: "Cytaras de cristal..." [Estr. y Coplas]
7. Villancico VII: "Por divertir à el Dios Niño..." [Int., Estr. y Coplas]
8. Villancico VIII: "Niño mio, que al yelo..." [Estr. y Coplas]
9. Villancico IX: "A Què tocan?..." [Estr. y Coplas]

Aguilar: Imp. sev., 88.

· VE/ 531-37

\$).

d(1,\$264

MENDOZA, Gabriel José. Letras de los villancicos que se cantaron en la... Iglesia Colegial de Nuestro Señor San Salvador... de Sevilla, en la Kalenda y Maytines del Nacimiento de nuestro Señor Jesu-christo... de 1706 / por Don Gabriel Joseph de Mendoza, Maestro de Capilla...—[Sevilla?: s. n., 1706?]-[8] p.; 4º

Sign.: A*-4.—Datos de pie de imp. deducidos del tit.—Texto a dos col.—Port. con orla tip. y viñeta xil. de la Adoración de los Pastores.

Contiene:

1. Villancico I de Kalenda y primero de Maytines: "Ha de la Cuna del Sol..." [Estr. y Coplas]
2. Villancico II: "Ha Señorito..." [Estr. y Coplas]
3. Villancico III: "A Belèn vãn Peregrinos..." [Int., Estr. y Coplas]
4. Villancico IV: "Niño tierno, que rendido..." [Estr. y Coplas]
5. Villancico V: "Pasqual, q es archivo andante..." [Int., Estr. y Coplas]
6. Villancico VI: "Los Pastores esta noche..." [Int., Estr. y Coplas]
7. Villancico VII: "A siolos Molenos..." [Estr. y Coplas]
8. Villancico VIII. De la Obra: "Ay, què dolor!..." [Estr. y Coplas]

Aguilar: Imp. sev., 2138.

· VE/ 531-46

\$).

d(2,\$265

SALAZAR, Diego José. Letras de los villancicos que se cantaron en la... Iglesia Metropolitana y Patriarcal de Sevilla, en los... Maytines de la Purissima é Inmaculada Concepcion de Maria..., este presente año de 1706 / compuestos por D. Diego Joseph de Salazar, Racionero y Maestro de Capilla...—Impresso en Sevilla: por Ivan Francisco de Blas. [1706?].—[4] p.; 4º

Sign.: []*-2.—Lugar de imp. e impresor tomados del colofón y fecha deducida del

tit.--Texto a dos col.

Contiene:

1. Villancico Primero de Kalenda: "Oy que el Padre Omnipotente..." [Int., Estr. y Coplas]
2. Villancico II: "Oygan, atiendan..." [Estr. y Coplas]
3. Villancico III: "Como serà que las sombras..." [Estr. y Coplas]
4. Villancico IV: "Sagradas Luces, silencio..." [Estr. y Coplas]
5. Villancico V: "Corran, corran las Fuentes..." [Estr. y Coplas]

· R/ 34199*-21

\$).

d(3,\$266

LETRAS de los villancicos que se han de cantar en los solemnes Maytines y celebre Fiesta que consagra y dedica la Hermandad de N. Señora de los Dolores en la Parroquia Iglesia... de... S. Marcos, este año 1706: dirigidas por la misma Hermandad al señor D. Juan de Arjal... / cantadas por la Capilla de Musica de la Colegial de N. Señor S. Salvador.--En Sevilla: por Jvan de la Pverta..., [1706?].--[8] p.; 4°

Sign.: []*-4.--Fecha de imp. deducida del tit.--Texto a dos col.--Port. con orla tip. y viñeta xil. de la Virgen de los Dolores.

Contiene:

1. Villancico I: "Los Serafines alados..." [Int., Estr. y Coplas]
2. Villancico II: "Moradores del mundo..." [Estr. y Coplas]
3. Villancico III: "Al arma, al arma, al arma..." [Estr. y Coplas]
4. Villancico IV: "Amante Aurora mia..." [Estr. y Coplas]
5. Villancico V: "Los Cielos Soberanos..." [Estr. y Coplas]
6. Villancico VI: "Hermosa Arquitectura..." [Estr. y Coplas]
7. Villancico VII: "De los rebatos confusos..." [Int., Estr. y Coplas]
8. Villancico VIII: "Qve dulce se muestra..." [Estr. y Coplas]
9. Villancico IX: "Soberanos Espiritus..." [Estr. y Coplas]

· VE/ 1201-30

\$).

d(4,\$267

SALAZAR, Diego José. Letras de los villancicos que se cantaron en la... Iglesia Metropolitana y Patriarcal... de Sevilla, en los solemnes Maytines de la Venida de los Santos Reyes...

de 1707/compuestos por Don Diego Joseph de Salazar, Racionero y Maestro de Capilla...--En Sevilla: por Juan Francisco de Blas..., 1707.--[8] p.; 4°-

Sign.: A*-4.--Datos de pie de imp. tomados del colofón.--Texto a dos col.--Port. con orla tip. y viñeta xil. de la Giralda.

Contiene:

1. Villancico Primero de Kalenda: "La brillante bellissima estrella..." [Estr. y Coplas]
2. Villancico II: "Pastorcita soy..." [Estr. y Coplas]
3. Villancico III: "Viendo que Reyes adoran..." [Estr. y Coplas]
4. Villancico IV: "Gallegos baylarines..." [Estr. y Coplas]
5. Villancico V: "Pluma de oro que rasgos dilatas..." [Estr. y Coplas]
6. Villancico VI: "Dos Negritos tan acordes..." [Int., Estr. y Coplas]
7. Villancico VII: "Las Gitanas desde Egipto..." [Int., Estr. y Coplas]
8. Villancico VIII: "Celebren los sentidos..." [Estr. y Coplas]
9. Villanaco IX: "El cuento de la lalila..." [Int., Estr. y Coplas]

· VE/ 531-36

\$).

d(5,\$268

LETRAS de los villancicos que en los solemnnes Maytines de la Resurreccion de N. Señor Jesu-Christo cantaron los Niños Seyses, en la... Metropolitana y Patriarcal Iglesia de Sevilla, este año de 1707 / puestos en mvsica por su Maestro D. Juan Donoso Cabeza de Baca...--En Sevilla: por Ivan Francisco de Blas..., [1707?].--[4] p.; 4º

Sign.: []*-2.-- Fecha de imp. deducida del tit.--Texto a dos col.--Port. con orla tip. y viñeta xil. de Cristo resucitado.

Contiene:

1. Villancico Primero: "Aquel Rey Soberano..." [Int., Estr. y Coplas]
2. Villancico Segvndo: "Oy que resucita amante..." [Int., Estr. y Coplas]
3. Villancico Tercero de Clarin: "Del Alva luciente..." [Estr. y Coplas]

Aguilar: Imp. sev., 99.

· VE/ 531-61

\$).

d(6,\$269

SALAZAR, Diego José. Letras de los villancicos, que se cantaron en la... Patriarcal Iglesia desta Ciudad de Sevilla, en los... Maytines del Nacimiento de N. Señor Jesu-Christo, este año de 1707 / compuesto por D. Diego Joseph de Salazar, Racionero y Maestro de Capilla de dicha S. Iglesia.--En Sevilla: por Juan Francisco de Blas..., 1707.--[8] p; 4º

Sign.: A*-4.--Datos de pie de imp. tomados del colofón.--Texto a dos col.--Port. con orla tip. y viñeta xil. que representa el Nacimiento.

Contiene:

1. Villancico Primero de Kalenda. Primero Noctvrno: "O venturoso dia!..." [Estr. y Coplas]
2. Villancico II: "Ay, que vna Andaluza..." [Estr. y Coplas]
3. Villancico III: "Los Vizcaynos, que es gente..." [Int., Estr. y Coplas]
4. Villancico IV: "Oygan el agudeza..." [Estr. y Coplas]
5. Villancico V: "Ay Jesus, ay mi bien, ay mi Dios..." [Estr. y Coplas]
6. Villancico VI. Negro: "Qvatlo Negliyoç venimoz..." [Estr. y Coplas]
7. Villancico VII: "Pves nace mi Niño,..." [Estr. y Coplas]
8. Villancico VIII: "Este es el Solio en q nace el amor,..." [Estr. y Coplas]
9. Villancico IX: "Alerta Zagal,..." [Estr. y Coplas]

· VE/ 1189-36

\$).

d(7,\$270

SALAZAR, Diego José. Letras de los villancicos que se cantaron [sic] en la... Metropolitana y Patriarcal Iglesia... de Sevilla, en los... Maytines de la Purissima Concepcion de Maria... / compuestos por Don Diego Joseph de Salazar, Racionero y Maestro de Capilla...--En Sevilla: por Juan Francisco de Blas..., [1707?].-- [4] p; 4º

Sign.: []*-2.--Lugar de imp. e impresor tomados del colofón y fecha deducida de la actividad del Maestro de Capilla y de la fiesta.--Texto a dos col.

Contiene

1. Villancico Primero: "Oy sobre los montes Santos,..." [Int., Estr. y Coplas]
2. Villancico II: "Albricias, Pastores,..." [Estr. y Coplas]
3. Villancico III. A la Purissima Concepcion: "Celebre en el Impireo..." [Estr. y Coplas]
4. Villancico IV: "Ay como cantan,..." [Estr. y Coplas]
5. Villancico V: "Paxarillos amantes,..." [Estr. y Coplas]

· VE/ 1305-115 Foliación ms.: "475-476".--(Barbieri)

\$).

d(8,\$271

SALAZAR, Diego José. Letras de los villancicos que se cantaron en la... Metropolitana y Patriarcal Iglesia... de Sevilla, en los solemnnes Maytines de la Venida de los Santos Reyes...

de 1708 / compuestos por D. Diego Joseph de Salazar, Racionero y Maestro de Capilla...--En Sevilla: por Juan Francisco de Blas..., [1708?].--[8] p.; 4º

Sign.: A*-4.--Lugar de imp. e impresor tomados del colofón y fecha deducida del tit.--Texto a dos col.--Port. con orla tip. y viñeta xii. de la Giralda

Contiene

1. Villancico Primero: "Salva clarines sonoros..." [Estr. y Coplas]
2. Villancico II: "Con todos sus instrumentos..." [Int., Estr. y Coplas]
3. Villancico III: "Pasito, silencio..." [Estr. y Coplas]
4. Villancico IV: "Coronadas de Rosas..." [Estr. y Coplas]
5. Villancico V: Morisco: "Meguelejo..." [Estr. y Coplas]
6. Villancico VI: "Vv Astro flamante..." [Estr. y Coplas]
7. Villancico VII: "Ay Zagalejas, ay..." [Estr. y Coplas]
8. Villancico VIII: "Vn Barquillero, à Belen..." [Int., Estr. y Coplas]
9. Villancico IX: "Viendo algunas rudas Aves..." [Int., Estr. y Coplas]

Aguilar: Imp., sev., 124

· VE/ 1310-8 (Barbieri)

\$).

d(9,\$272

SANTISSO BERMUDEZ, Gregorio. Letras de los villancicos que se cantaron en los... Maytines de la Purissima Concepcion... en la... Iglesia Metropolitana y Patriarcal de... Sevilla / compuestos (vacante el Magisterio de Capilla) por D. Gregorio Santisso Bermudez, Presbytero, Maestro de Seises de dicha Santa Iglesia.--[Sevilla?: s.n., 1709?].-- [4] p; 4º Sign.: []*-2,.- Lugar de imp. deducido del tit. y fecha deducida del autor y del periodo en que esta vacante el magisterio de capilla.--Texto a dos col.

Contiene:

1. Villancico I: "La Gracia, y la Embidia salen..." [Int., Estr. y Coplas]
2. Villancico II: "Ay que me Abrasa vna llama..." [Estr. y Coplas]
3. Villancico III: "Celebre en el Impireo..." [Estr. y Coplas]
4. Villancico IV: "Los nuevos Cantores..." [Estr. y Coplas]
5. Villancico V: "Avecita, que en dulce armonia..." [Estr. y Coplas]

· VE/ 1305-112 Foliación ms.: "22-23".--(Barbieri)

\$).

Diccionario

[DRAE,70]. Se han conservado los fines de línea originales -propios del encolumnado en que el diccionario se encuentra impreso-, para ver mejor la estructura general de las entradas.

d(1,\$florales. (Del lat. *florales ludi*, juegos florales.) adj. pl. Aplícase a las fiestas o juegos que celebraban los gentiles en honor de la diosa Flora. A su imitación se han instituido después en Provenza y en otras partes. || 2. V. juegos florales.\$).

d(2,\$florar. intr. Dar flor. Dícese de los árboles y las plantas, singularmente de los que se cultivan para cosechar sus frutos.\$).

d(3,\$flordelisado, da. adj. Blas. V. cruz flordelisada.\$).

d(4,\$flordelisar. tr. Blas. Adornar con flores de lis una cosa.\$).

d(5,\$floreado, da. p. p de florear. || 2. adj. V. pan floreado.\$).

d(6,\$florear. (Del fr. *floréal*.) m. Octavo mes del calendario republicano francés, cuyos días primero y último coincidían, respectivamente, con el 20 de abril y el 19 de mayo.\$).

d(7,\$florear. tr. Adornar y guarnecer con flores. || 2. Tratándose de la harina,

sacar la primera y más sutil por medio del cedazo más espeso. || 3. Disponer el naipe para hacer trampa. || 4. intr. Vibrar, mover la punta de la espada. || 5. Tocar dos o tres cuerdas de la guitarra con tres dedos sucesivamente sin parar, formando así un sonido continuado. || 6. fam. Echar a una mujer piropos. || 7. Ar., Chile y Sal. Escoger lo mejor de una cosa.\$).

d(8,\$florecedor, ra. adj. Que florece.\$).

d(9,\$floreecer. (De *florescer*.) intr. Echar o arrojar flor. ú. t. c. tr. || 2. fig. Prosperar crecer en riqueza o reputación. Dícese también de los entes morales; como la justicia, las ciencias, etc. || 3. fig. Existir una persona o cosa insigne en un tiempo o época determinada. || 4. prnl. Hablando de algunas cosas, como el queso, pan, etc., ponerse mohosas.\$).

d(10,\$florecente. p. a. de florecer. Que florece. || 2. fig. Favorable, venturoso, próspero.\$).

d(11,\$floreecimiento. m. Acción y efecto de florecer o florecerse.\$).

d(12,\$Florencia. n. p. V. raja de Florencia.\$).

d(13,\$florentin. adj. florentino. Apl. a pers., ú.t.c.s.\$).

d(14,\$florentino, na. (Del lat. florentinus.) adj. Natural de Florencia. ú. t. c. s. || 2. Perteneciente a esta ciudad de Italia.\$).

d(15,\$florentísimo, ma. (Del lat. florentissimus.) adj. sup. de floreciente. Que prospera o florece con excelencia.\$).

d(16,\$floreo. (De florear,) m. fig. Conversación vana y de pasatiempo. || 2. fig. Dicho vano y superfluo empleado sin otro fin que el de hacer alarde de ingenio, o el de halagar o lisonjear al oyente, o sólo por mero pasatiempo. || 3. Danza. En la danza española, movimiento de un pie en el aire cuando el otro permanece en el suelo, y el cuerpo sostenido sobre él. || 4. Esgr. Vibración o movimiento de la punta de la espada. || 5. Mús. Acción de florear en la guitarra.\$).

d(17,\$florería. f. Tienda donde se venden flores y plantas de adorno.\$).

d(18,\$florero, ra. adj. fig. Que usa de palabras chistosas y lisonjeras. ú. t. c. s. || 2. m. y f. El que o la que vende flores. || 3. m. Vaso para poner flores naturales o artificiales. || 4. Maceta o tiesto con flores. || 5. Armario, caja o lugar destinado para guardar flores. || 6. Germ. Fullero que hace trampas floreando el

naipe. || 7. Pint. Cuadro en que sólo se representan flores.\$).

d(19,\$florescencia. f. eflorescencia. || 2. Bot. Acción de florecer. || 3. Bot. Época en que las plantas florecen, o aparición de las flores en cada vegetal.\$).

d(20,\$florescer. (Del lat. florescere.) intr. ant. florecer.\$).

d(21,\$floresta. (Del b. lat. foresta, y éste de foras, de fuera.) f. Terreno frondoso y ameno poblado de árboles. || 2. fig. Reunión de cosas agradables y de buen gusto.\$).

d(22,\$florete. (d. de flor.) s. Entre guarnicioneros, bordadura sobrepuesta que sirve de fuerza y adorno en los extremos de las cinchas. || 2. Danza. En la danza española, tejido o movimiento que se hacia con ambos pies.\$).

d(23,\$floreteada. (De florete.) f. ant. Papirote dado en la frente.\$).

d(24,\$floreteazo. m. Golpe dado con el florete.\$).

d(25,\$florete. (Del fr. fleuret, y este del ital. fioretto, del lat. flos, floris.) adj. V. azúcar, papel florete. || 2. m. Esgrima con espadín. || 3. Espadín destinado a la enseñanza o ejercicio de este juego; es de cuatro aristas, y no suele tener aro en la empuñadura. || 4. Lienzo o tela entrefina de algodón.\$).

d(26,\$floretear. (De floreta.) tr. Adornar y guarnecer con flores una cosa. || 2. intr. Manejar el florete.\$).

d(27,\$floreteo. m. Acción y efecto de floretear.\$).

d(28,\$floreteista. m. El que es diestro en el juego del florete.\$).

d(29,\$floricultor, ra. (Del lat. flos, floris, flor, y cultos, -oris, cultivador.) m. y f. Persona dedicada a la floricultura.\$).

d(30,\$floricultura. (Del lat. Los, floris, flor, y cultura, cultivo.) f. Cultivo de las flores. || 2. Arte que lo enseña.\$).

d(31,\$floridamente. adv. m. fig. Con elegancia y gracia.\$).

d(32,\$floridano, na. adj. Natural de la Florida. ú. t. c. s. || 2. Perteneciente a este Estado de América del Norte.\$).

d(33,\$floridez. (De florido.) f. Abundancia de flores. La FLORIDEZ de la primavera. || 2. fig. Calidad de florido, en el estilo. florido. da. adj. Que tiene flores- || 2. V. junco florido. || 3. V. letra, pascua florida. || 4. fig. Dícese de lo más escogido de alguna cosa. || 5. fig. Dícese del lenguaje o estilo amena y profusamente exornado de galas retóricas. || 6. Germ. Rico, opulento.\$).

d(34,\$florífero, ra. (Del lat. florifer, -era; de

flos, floris, flor, y ferre, llevar.) adj. Que lleva o produce flores.\$).

d(35,\$florífero, ra. (Del lat. florifer, -era; de flos, floris, flor, y gerere. llevar.) adj. poét. florífero.\$).

d(36,\$florilegio. (Del lat. Los, floris, flor, y lege- re, escoger.) m. fig. Colección de trozos selectos de materias literarias.\$).

d(37,\$florín. (Del ital. fiorino, moneda florentina marcada con el lirio de los Médicis.) m. Moneda de plata equivalente al escudo de España, que se usa en algunos países, especialmente en Austria y Holanda, y estuvo marcada antiguamente con una flor de lis. || 2. Moneda de oro mandada acuñar por los reyes de Aragón copiando los florines o ducados de Florencia, su valor sufrió grandes alteraciones por efecto del abuso de liga en el metal.\$).

d(38,\$floripondio. (Quizá de flor y el dialect. predio, del lat. *pendidus, pendiente.) m. Arbusto del Perú, de la familia de las solanáceas, que crece hasta tres metros de altura con tronco leñoso, hojas grandes, alternas, oblongas, enteras y vellosas, flores solitarias, blancas, en forma de embudo, de unos tres decímetros de largo, de olor delicioso, pero perjudicial si se aspira mucho tiempo, y fruto elipsoidal, con muchas semillas pequeñas de figura de riñón. || 2. fig. despect. Flor grande que suele figurar en adornos de mal gusto.\$).

d(39,\$florista. com. Persona que fabrica flores de mano. || 2. Persona que vende flores.\$).

d(40,\$floristería. f. florería.\$).

d(41,\$florisar. tr. Blas. flordelisar.\$).

d(42,\$florón. m. aum. de flor || 2. Adorno hecho a manera de flor muy grande, que se usa en pintura y arquitectura en el centro de los techos de las habitaciones, etc. || 3. Blas. Adorno, a manera de flor, que se pone en el círculo de\$).

BOE

d(1,\$Resolución de 5 de mayo de 1995, de la Universidad de Vigo, por la que se nombra Catedrática de Escuela Universitaria del área de conocimiento de "Filología Francesa", del departamento de Filología Francesa, a doña Terencia Inés Silva Rojas\$).

d(2,\$Resolución de 20 de abril de 1995, de la Universidad de Málaga, por la que se nombra Profesora de la misma a doña Eva Millán Valídeperas\$).

d(3,\$Resolución de 20 de abril de 1995, de la Universidad de Málaga, por la que se nombra Profesor de la misma a don Carlos Rossi Jiménez\$).

d(4,\$Resolución de 24 de abril de 1995, de la Universidad de Málaga, por la que se nombra Profesora de la misma a doña María Lourdes Rey Borrego\$).

d(5,\$Resolución de 24 de abril de 1995, de la Universidad de Málaga, por la que se nombra Profesor de la misma a don José Ramón Díaz Fernández\$).

d(6,\$Resolución de 28 de abril de 1995, de la Universidad Rovira - Virgili, de Tarragona, por la que se nombran diversos Profesores de cuerpos docentes universitarios\$).

d(7,\$Resolución de 3 de mayo de 1995, de la Universidad de Cádiz, por la que se nombran profesores universitarios.\$).

d(8,\$Resolución de 8 de mayo de 1995, de la Universidad Pública de Navarra, por la que se nombra, en virtud de concurso, a doña Irene Zabala Martínez Profesora titular de Universidad en el área de conocimiento de "Electrónicas"\$).

d(9,\$Resolución de 8 de mayo de 1995, de la Universidad Pública de Navarra, por la que se nombra, en virtud de concurso, a don Gabriel Lera Carreras Profesor titular de Universidad en el área de conocimiento de "Ingeniería de Sistemas y Automática"\$).

d(10,\$Resolución de 8 de mayo de 1995, de la Universidad Pública de Navarra, por la que se nombra, en virtud de concurso, a doña Blanca Fernández Viguera Profesora titular de Escuela Universitaria en el área de conocimiento de "Trabajo Social y Servicios Sociales". \$).

d(11,\$Resolución de 8 de mayo de 1995, de la Universidad Pública de Navarra, por la que se nombra, en virtud de concurso, a doña Concepción Corera Oroz Profesora titular de Escuela Universitaria en el área de conocimiento de "Trabajo Social y Servicios Sociales". \$).

d(12,\$Resolución de 8 de mayo de 1995, de la Universidad Pública de Navarra, por la que se nombra, en virtud de concurso, a don José Basilio Galván Herrera Profesor titular de Universidad en el área de conocimiento de "Ingeniería de Sistemas y Automática"\$).

d(13,\$Resolución de 9 de mayo de 1995, de la Universidad de Santiago de Compostela, por la que se nombra Profesor titular de Universidad del área de conocimiento de "Cirugía", del departamento de Cirugía, a don Julián Alvarez Escudero\$).

d(14,\$Resolución de 9 de mayo de 1995, de la Universidad de Huelva, por la que se nombra, en virtud de concurso, a don Emilio Pascual Martínez Catedrático de Universidad del área de conocimiento de "Petrología y Geoquímica", adscrita al departamento de Geología. \$).

d(15,\$Resolución de 3 de mayo de 1995, de la Universidad de León, por la que se nombra a don Eduardo Sánchez Compadre Profesor titular de Escuela Universitaria en el área de conocimiento de "Biología Animal"\$).

d(16,\$Resolución de 5 de mayo de 1995, de la Universidad de Málaga, por la que se nombra Profesor de la misma a don Alfredo Rubio Díaz\$).

d(17,\$Resolución de 5 de mayo de 1995, de la Universidad de Vigo, por la que se nombra Profesora titular de Universidad del área de conocimiento de "Matemática Aplicada", del departamento de Matemática Aplicada a doña María del Carmen Suárez Rodríguez\$).

d(18,\$Resolución de 5 de mayo de 1995, de la Universidad de Vigo, por la que se nombra Catedrática de Escuela Universitaria del área de conocimiento de "Química Orgánica", del departamento de Química Pura y Aplicada, a doña María Generosa de los Angeles Gómez Pacios\$).

d(19,\$Resolución de 5 de mayo de 1995, de la Universidad de Vigo, por la que se nombra Profesora titular de Universidad del área de conocimiento de "Biología Vegetal".\$).

d(20,\$ Resolución de 8 de mayo de 1995, de la Universidad Pública de Navarra, por la que se nombra, en virtud de concurso, a don Gabriel Lera Carreras Profesor titular de Universidad en el área de conocimiento de "Ingeniería de Sistemas y Automática"\$).

d(21,\$ Resolución de 8 de mayo de 1995, de la Universidad Pública de Navarra, por la que se nombra, en virtud de concurso, a doña Blanca Fernández Viguera Profesora titular de Escuela Universitaria en el área de conocimiento de "Trabajo Social y Servicios Sociales"\$).

d(22,\$ Resolución de 8 de mayo de 1995, de la Universidad Pública de Navarra, por la que se nombra, en virtud de concurso, a doña Concepción Corera Oroz Profesora titular de Escuela Universitaria en el área de conocimiento de "Trabajo Social y Servicios Sociales"\$).

d(23,\$Resolución de 5 de mayo de 1995, de la Universidad de Vigo, por la que se nombra Profesora titular de Universidad del área de conocimiento de "Biología Vegetal", del departamento de Recursos Naturales y Medio Ambiente, a doña Aida García Molares\$).

Anuncios de prensa

d(2,\$SCANILLEJAS, San Hilario, cuarto, exterior, 120 metros. cuatro dormitorios, dos baños, agua y calefacción central, ascensores, pisazo. 23.000.000. T. 91/4305412.\$).

d(3,\$ARTURO SORIA, principio, cuatro dormitorios, dos baños, jardines, piscina. T. 91/5165921.\$).

d(4,\$TETUAN, exterior, 105 útiles. 27 de terraza. gran trastero. Verdadera inversión. 17.900.000. T. 91/5718466\$).

d(5,\$VAGUADA, 65 metros, planta 100, dos dormitorios, ascensor, vistas. 13.000.000. Facilidades.\$).

d(6,\$MORATALAZ, precioso piso, todo nuevo, cuatro dormitorios, dos baños, calefacción, cuarta planta, sin ascensor. 14.950.000. T. 91/4305412.\$).

d(7,\$MORATALAZ, Pico Artilleros, cuarto exterior, 120 metros, tres dormitorios, antes cuatro, descomunal salón, dos baños, calefacción, ascensores. Fenomenal piso. 24.250.000.\$).

d(8,\$SIMANCAS, calle Yecla, segundo, exterior, dos dormitorios. Sólo 8.800.000. T. 91/4305412.\$).

d(9,\$MUSAS, cuarto, exterior, dos baños, calefacción, Garaje incluido en precio. 13.900.000. T. 91/4305412.\$).

d(10,\$CENTRIQUISIMO, metro Lavapies, tres dormitorios, todo exterior, 1.500.000. Grandísimas facilidades. T. 5763030.\$).

d(11,\$PACIFICO, Luis Mitjans, tercero. exterior. tres dormitorios, gas natural ascensor. 12.800.000. T. 91/4305412.\$).

d(12,\$RETIRO, Juan de Urbietta, segundo, exterior, dos dormitorios. Para entrar. 14.500.000. \$).

d(13,\$CANILLAS, calle Tribaldos, 120 metros, octavo, exterior, tres dormitorios, antes cuatro, gran salón, calefacción central. Ocasión. 19.800.000. T. 91/436412.\$).

d(14,\$ZONA norte, 120 metros, cuatro dormitorios, dos baños, exterior, calefacción, ascensor, 18.500.000\$).

d(15,\$MANUEL BECERRA, 70 metros, reforma, 12.300.000. T. 91/4477458.\$).

d(16,\$ANTON MARTIN, dúplex. estrenar, 11.700.000. T. 91/4477458.\$).

d(17,\$BETANZOS, tres dormitorios. exterior, luminosos 13.800.000. T. 91/5537308.\$).

d(18,\$JERONIMA LLORENTE. dos dormitorios, totalmente exterior, edificio rehabilitado 13.000.000. T. 91/5535151. \$).

d(19,\$ALAMEDA OSUNA, 100 metros, reformar, 14.000.000. T. 91/4460029.\$).

d(20,\$ARAVACA, céntrico, dos dormitorios, reformado, 12.000.000. T. 91/3574029.\$).

d(21,\$ARAVACA, tres dormitorios, dos baños, calefacción central, 16.500.000. T. 91/3574034.\$).

d(22,\$POZUELO, estudio, salón, baño, pis-

cina, garaje, 11.700.000.
T. 91/3574029.\$).
d(23,\$LISTA, dos dormitorios, 8.500.000.
tercera planta, soleado.
T. 91/5753388.\$).
d(24,\$BATAN mismo, Paseo Extremadura.
Ocasión, gran salón, reformar. Vistas.
Precio-llaves negociar.
T. 91/5111699.\$).
d(25,\$BATAN, vistas Casa Campo, 90
metros, aproximados, tres dormito-
rios, dos baños, uno incorporado,
entrar vivir. Negociamos.\$).
d(26,\$PUERTA ANGEL, ocasión, urge,
buena distribución, vistas. tres dormi-
torios. Todo negociable.\$).
d(27,\$MISMO Paseo Extremadura, oca-
sión, luminoso. reformadísimo. Llaves
convenir. T. 91/3547390\$).
d(28,\$CAMPAMENTO, del cine, estrenar,
urge piscina, dos dormitorios, dos

baños. Todo negociable.
91/5777699.\$).
d(29,\$BARATISIMO, cerca Avenida Opor-
to, reformar. 500.000 entrada. Total
5.100.000. Llaves convenir.
T. 91/5779366.\$).
d(30,\$INTERESANTE oferta, General
Ricardos-Salaberry, soleado, tranqui-
lísimo, tres dormitorios, dos dobles,
terrazas. Condiciones convenir.
T. 91/5779366.\$).
d(31,\$SUPERECONOMICO, Antonio Ley-
va, luminoso reformado, 115 metros,
aproximados, tres dormitonos, baño,
aseo. Entrada convenir. Facilidades.
T. 91/5779366\$).
d(32,\$INTERESANTE, metro Elíptica, exte-
rior, 90 metros, aproximados, tres
dormitorios, salón, jar-
dines. Grandes facilidades.
T. 91/5983722.\$).

Tractatus

- d(1,\$2.025 Sie ist Form und Inhalt.\$).
- d(2,\$2.0251 Raum, Zeit und Farbe (Färbigkeit) sind Formen der Gegenstände.\$).
- d(3,\$2.026 Nur wenn es Gegenstände gibt, kann es eine feste Form der Welt geben.\$).
- d(4,\$2.027 Das Feste, das Bestehende und der Gegenstand sind Eins.\$).
- d(5,\$2.0271 Der Gegenstand ist das Feste, Bestehende; die Konfiguration ist das Wechselnde, Unbeständige.\$).
- d(6,\$2.0272 Die Konfiguration der Gegenstände bildet den Sachverhalt.\$).
- d(7,\$2.03 Im Sachverhalt hängen die Gegenstände ineinander, wie die Glieder einer Kette.\$).
- d(8,\$2.031 Im Sachverhalt verhalten sich die Gegenstände in bestimmter Art und Weise zueinander.\$).
- d(9,\$2.032 Die Art und Weise, wie die Gegenstände im Sachverhalt zusammenhängen, ist die Struktur des Sachverhaltes.\$).
- d(10,\$2.033 Die Form ist die Möglichkeit der Struktur.\$).
- d(11,\$2.034 Die Struktur der Tatsache besteht aus den Strukturen der Sachverhalte.\$).
- d(12,\$2.04 Die Gesamtheit der bestehenden Sachverhalte ist die Welt.\$).
- d(13,\$2.05 Die Gesamtheit der bestehenden Sachverhalte bestimmt auch, welche Sachverhalte nicht bestehen.\$).
- d(14,\$2.06 Das Bestehen und Nichtbestehen von Sachverhalten ist die Wirklichkeit.
(Das Bestehen von Sachverhalten nennen wir auch eine positive, das Nichtbestehen eine negative Tatsache.)\$).
- d(15,\$2.061 Die Sachverhalte sind von einander unabhängig.\$).
- d(16,\$2.062 Aus dem Bestehen oder Nichtbestehen eines Sachverhaltes kann nicht auf das Bestehen oder Nichtbestehen eines anderen geschlossen werden.\$).
- d(17,\$2.063 Die gesamte Wirklichkeit ist die Welt.\$).
- d(18,\$2.1 Wir machen uns Bilder der Tatsachen.\$).

- d(19,\$2.11 Das Bild stellt die Sachlage im logischen Raume, das Bestehen und Nichtbestehen von Sachverhalten vor.\$).
- d(20,\$2.12 Das Bild ist ein Modell der Wirklichkeit.\$).
- d(21,\$2.13 Den Gegenständen entsprechen im Bilde die Elemente des Bildes.\$).
- d(22,\$2.131 Die Elemente des Bildes vertreten im Bild die Gegenstände.\$).
- d(23,\$2.14 Das Bild besteht darin, dass sich seine Elemente in bestimmter Art und Weise zu einander verhalten.\$).
- d(24,\$2.141 Das Bild ist eine Tatsache.\$).
- d(25,\$2.15 Dass sich die Elemente des Bildes in bestimmter Art und Weise zu einander verhalten, stellt vor, dass sich die Sachen so zu einander verhalten.
Dieser Zusammenhang der Elemente des Bildes hei3se seine Struktur und ihre Möglichkeit seine Form der Abbildung.\$).
- d(26,\$2.151 Die Form der Abbildung ist die Möglichkeit, dass sich die Dinge so zu einander verhalten, wie die Elemente des Bildes.\$).
- d(27,\$2.1511 Das Bild ist so mit der Wirklichkeit verknüpft; es reicht bis zu ihr.\$).
- d(28,\$2.1512 Es ist wie ein Masstab an die Wirklichkeit angelegt.\$).
- d(29,\$2.15121 Nur die äussersten Punkte der Teilstriche berühren den zu messenden Gegenstand.\$).
- d(30,\$2.1513 Nach dieser Auffassung gehört also zum Bilde auch noch die abbildende Beziehung, die es zum Bild macht.\$).
- d(31,\$2.1514 Die abbildende Beziehung besteht aus den Zuordnungen der Elemente des Bildes und der Sachen.\$).
- d(32,\$2.1515 Diese Zuordnungen sind gleichsam die Fühler der Bildelemente, mit denen das Bild die Wirklichkeit berührt.\$).
- d(33,\$2.16 Die Tatsache muss, um Bild zu sein, etwas mit dem Abgebildeten gemeinsam haben.\$).
- d(34,\$2.161 In Bild und Abgebildetem muss etwas identisch sein, damit das eine überhaupt ein Bild des anderen sein kann.\$).
- d(35,\$2.17 Was das Bild mit der Wirklichkeit gemein haben muss, um sie auf seine Art und Weise -richtig oder falsch-abbilden zu können, ist seine Form der Abbildung.\$).
- d(36,\$2.171 Das Bild kann jede Wirklichkeit abbilden, deren Form es hat.
Das räumliche Bild alles Räumliche, das farbige alles Farbige, etc.\$).

- d(37,\$2.172 Seine Form der Abbildung aber kann das Bild nicht abbilden; es weist sie auf.\$).
- d(38,\$2.173 Das Bild stellt sein Objekt von ausserhalb dar (sein Standpunkt ist seine Form der Darstellung), darum stellt das Bild sein Objekt richtig oder falsch dar.\$).
- d(39,\$2.174 Das Bild kann sich aber nicht ausserhalb seiner Form der Darstellung stellen.\$).
- d(40,\$2.18 Was jedes Bild, welcher Form immer, mit der Wirklichkeit gemein haben muss, um sie überhaupt-richtig oder falsch-abbilden zu können, ist die logische Form, das ist, die Form der Wirklichkeit.\$).
- d(41,\$2.181 Ist die Form der Abbildung die logische Form, so heisst das Bild das logische Bild.\$).
- d(42,\$2.182 Jedes Bild ist auch ein logisches. (Dagegen ist z.B. nicht jedes Bild ein raumliches.)\$).
- d(43,\$2.19 Das logische Bild kann die Welt abbilden.\$).
- d(44,\$2.2 Das Bild hat mit dem Abgebildeten die logische Form der Abbildung gemein.\$).
- d(45,\$2.201 Das Bild bildet die Wirklichkeit ab, indem es eine Möglichkeit des Bestehens und Nichtbestehens von Sachverhalten darstellt.\$).
- d(46,\$2.202 Das Bild stellt eine mögliche Sachlage im logischen Raume dar.\$).
- d(47,\$2.203 Das Bild enthält die Möglichkeit der Sachlage, die es darstellt.\$).
- d(48,\$2.21 Das Bild stimmt mit der Wirklichkeit überein oder nicht; es ist richtig oder unrichtig, wahr oder falsch.\$).
- d(49,\$2.22 Das Bild stellt dar, was es darstellt, unabhängig von seiner Wahr- oder Falschheit, durch die Form der Abbildung.\$).

Revistas

- d(1,\$ABACO 0213-6252 CENTRO DE ESTUDIOS ECONOMICOS Y SOCIALES SAN BERNARDO, 40, 5°, D. 33201 GIJON (ASTURIAS)\$).
- d(2,\$ ADICCIONES. REVISTA DE SOCIDRO- GALCOHOL 0214-4840 SOCIODROGALCOHOL RAMBLA, 15, 2° Y 3° 073003 PALMA DE MALLORCAS\$).
- d(3,\$ ADUANAS. REVISTA DE COMERCIO IN-TERNACIONAL Y ESTUDIOS FISCALES 0400-5732 CONS. DE ESTUDIOS ADUANEROS Y CO-MERCIO EXTERIOR EVARISTA SAN MIGUEL, 10 28008 MADRID\$).
- d(4,\$ AFFERS INTERNACIONALS 0212-1786 CENTRE D'INFORMACIO Y DOCUMENTA-CIO INTERNACIONALS ELISABETS, 12 08001 BARCELONA\$).
- d(5,\$ ACRICULTURA Y SOCIEDAD 0211-8394 MINIST. DE AGRICULTURA. SERV. DE PU-BLICACIONES PASEO INFANTA ISABEL, 1 28007 MADRID\$).
- d(6,\$ ALFOZ 0401-3239 CENTRO DE INVESTIGACION Y DOCU-MENTACION URBANA Y RURAL (CIDUR) INFANTAS, 13 28004 MADRID\$).
- d(7,\$ ALTA DIRECCION 0002-6549 ALTA DIRECCION S.A. AVDA. INFANTA CARLOTA, 6 08029 BARCELONA\$).
- d(8,\$ AMERICA LATINA, HOY. REVISTA DE CIENCIAS SOCIALES 1130-2887 UNIV. COMPLUTENSE DE MADRID. SEMI-ÑARIO ESTUDIOS POLITICOS SOBRE AME-RICA LATINA FAC. CC. POLITICAS. CAMPUS DE SOMOSAGUAS 28023 MADRID\$).
- d(9,\$ ANALES DE ESTUDIOS ECONOMICOS Y EMPRESARIALES 0213-7569 UNIV. DE VALLADOLID. SECR. DE PUBLI-CACIONES RAMON Y CAJAL, 7 47005 VALLADOLID\$).
- d(10,\$ ANALES DE LA REAL ACADEMIA DE CIENCIAS MORALES Y POLITICAS 0210-4121 REAL ACAD. DE CIENCIAS MORALES Y POLITICAS PLAZA DE LA VILLA, 2 28005 MADRID\$).
- d(11,\$ ANALISI 0211-2175 UNIV. AUTONOMA DE BARCELONA. SERV. DE PUBLICACIONES CAMPUS UNIVERSITARIO DE BELLATERRA 08193 BELLATERRA (BARCELONA)\$).
- d(12,\$ ANTHROPOS. SUPLEMENTOS 1130-2089 ANTHROPOS VIA AUGUSTA, 64 08006 BARCELONA\$).
- d(13,\$ ARCHIPIELAGO 0214-2686 EDITORIAL PAMELA PLAZUELA DEL CONSEJO, 3, 4°. 31001 PAMPLONA\$).
- d(14,\$ AREAS. REVISTA DE CIENCIAS SOCIA-LES 0211-6707 EDITORA REGIONAL DE MURCIA ISAAC ALBENIZ 30009 MURCIA\$).
- d(15,\$ AYER MARCIAL PONS PLAZA DEL CONDE DEL VALLE DE SUCHIL, 8 29015 MADRID\$).
- d(16,\$ BANCA ESPAÑOLA 0210-1688 REMARCA CONDE DE PEÑALVER, 52 28006 MADRID\$).
- d(17,\$ BOLETIN DE ECONOMIA DE NAVARRA 1131-5490 GOBIERNO DE NAVARRA. SERV. DE ECO-NOMIA ARSIETA, 8, 6° 31002 PAMPLONA (NAVARRA)\$).
- d(18,\$ BOLETIN DE ESTUDIOS ECONOMICOS 0006-6249 UNIV. DE DEUSTO HERMANOS AGUIRRE, 2. APTDO. 20044 48080 BIL-BAO\$).

d(19,\$□□□BOLETIN DE LA INSTITUCION LIBRE□DE ENSEÑANZA□□0214-1302□□FUNDACION FRANCISCO GINER DE LOS□RIOS□□PASEO DEL GENERAL MARTINEZ CAMPOS, 14□28010 MADRID\$).

d(20,\$□□□BOLETIN DE PSICOLOGIA□□0212-8179□□UNIV. DE VALENCIA. FAC. DE FILOSOFIA□Y CIENCIAS DE LA EDUCACION□□AVDA. BLASCO IBAÑEZ, 21 46010 VALENCIAS\$).

d(21,\$□□□BOLETIN DE RELACIONES ECONOMI-□CAS INTERNACIONALES□□MINIST. DE ASUNTOS EXTERIORES. SECR.□GENERAL TEC.□□PLAZA DE LA MARINA ESPAÑOLA 28013 MADRID\$).

d(22,\$□□□BOLETIN ECONOMICO DE ANDALUCIA□□0212-6621□□JUNTA DE ANDALUCIA. CONS. DE ECO-□NOMIA, INDUSTRIA Y ENERGIA□□AVDA REPUBLICA ARGENTINA, 81 41071 SEVILLAS\$).

d(23,\$□□□BOLETIN ECONOMICO DE ICE□□0213-3768□□MINIST. DE COMERCIO Y TURISMO□□PASEO DE LA CASTELLANA, 152, PLANTA 16 28016 □MADRID□□\$).

Anexo 3: Documentos Output

Lo que sigue es el resultado de aplicar las GDUs del apartado *Aplicaciones*, p.236, a los documentos input del Anexo 2.

Base de datos documental

```
d(3700,[campo([nombre($Signatura$),valor($4747$)]),campo([nombre($CDU$),valor($$)]),campo
([nombre($Encabezamiento$),valor($BLASCO GASCO, Francisco de P.$)]),
campo([nombre($Título_Uniforme$), valor($$)]), campo([nombre($Título_Mención$,
valor($Objeto de hipoteca y ejecución hipotecaria : determinación legal... el objeto hipotecado :
doctrina y jurisprudencia / Francisco de P. Blasco Gasco ; prólogo Vicente Montés Penedés$)]),
campo([nombre($Edición$),valor($$)]), campo([nombre($Publicación$),valor($Valencia : Edit.
General de Derecho, D.L. 1994$)]),campo([nombre($Colación$),valor($208 p.; 24 cm$)]),
campo([nombre($Serie$),valor($$)]), campo([nombre($Notas$), valor($$)]),
campo([nombre($ISBN/ISSN$), valor($84-605-0043-8$)]), campo([nombre($NIPO$), valor($$)]),
campo([nombre($Materias$), valor($1. Hipotecas - Espana$)]), campo([nombre($Secundarias$,
valor($I. Título$)]), campo([nombre($Registro/s$, valor($4747$)]),
campo([nombre($Anotaciones$, valor($$)]))].
```

```
d(3701,[campo([nombre($Signatura$), valor($4772$)]), campo([nombre($CDU$), valor($$)]),
campo([nombre($Encabezamiento$, valor($BODEN, Margaret A.$)]),
campo([nombre($Título_Uniforme$, valor($$)]), campo([nombre($Título_Mención$, valor($La
mente creativa : mitos y mecanismos / por Margaret A. Boden$)]), campo([nombre($Edición$,
valor($$)]), campo([nombre($Publicación$, valor($Barcelona : Gedisa, D.L. 1994$)]),
campo([nombre($Colación$,valor($404 p. ; 23 cm$)]), campo([nombre($Serie$, valor($Hombre
y sociedad. Gladema$)]), campo([nombre($Notas$, valor($$)]), campo([nombre($ISBN/ISSN$,
valor($84-7432-500-5$)]), campo([nombre($NIPO$,valor($$)]), campo([nombre($Materias$,
valor($1. Creatividad 2. Conocimiento (Psicología$)]), campo([nombre($Secundarias$, valor($I.
Título$)]), campo([nombre($Registro/s$, valor($4772$)]), campo([nombre($Anotaciones$,
valor($$)]))].
```

```
d(3702,[campo([nombre($Signatura$,valor($2308, cª 410, p.2307$)]), campo([nombre($CDU$,
valor($$)]), campo([nombre($Encabezamiento$, valor($BORRAJO DACRUZ, Efrén$)]),
campo([nombre($Título_Uniforme$, valor($$)]), campo([nombre($Título_Mención$,
```

valor(\$Introducción al derecho español del trabajo / Efrén Borrajo Dcruz\$)),
 campo([nombre(\$Edición\$), valor(\$4ª ed. reimp\$)]), campo([nombre(\$Publicación\$),
 valor(\$Madrid : Técnos, 1978\$)]), campo([nombre(\$Colación\$), valor(\$XXIII, 455 p. ; 22 cm\$)]),
 campo([nombre(\$Serie\$), valor(\$Biblioteca Universitaria\$)]), campo([nombre(\$Notas\$),
 valor(\$\$)]), campo([nombre(\$ISBN/ISSN\$), valor(\$84-309-0549-9\$)]), campo([nombre(\$NIPO\$),
 valor(\$\$)]), campo([nombre(\$Materias\$), valor(\$1. Derecho del trabajo - Espana\$)]),
 campo([nombre(\$Secundarias\$), valor(\$I. Título\$)]), campo([nombre(\$Registro/s\$),
 valor(\$410,2307,2308\$)]), campo([nombre(\$Anotaciones\$), valor(\$\$)]).

d(3703,[campo([nombre(\$Signatura\$), valor(\$4385/1, 4386/1\$)]), campo([nombre(\$CDU\$),
 valor(\$\$)]), campo([nombre(\$Encabezamiento\$), valor(\$BORRELL FONTELLES, José\$)]),
 campo([nombre(\$Título_Uniforme\$), valor(\$\$)]), campo([nombre(\$Título_Mención\$),
 valor(\$Métodos matemáticos para la economía / José Borrell Fontelles\$)]),
 campo([nombre(\$Edición\$), valor(\$\$)]), campo([nombre(\$Publicación\$), valor(\$Madrid :
 Pirámide, 1990\$)]), campo([nombre(\$Colación\$), valor(\$V. ; 23cm\$)]), campo([nombre(\$Serie\$),
 valor(\$Ciencia y técnica\$)]), campo([nombre(\$Notas\$), valor(\$T.I: Campos y autosistemas. -- 4ª
 ed. -- 1990. -- 334 p. -- ISBN:84-368-0182-2\$)]), campo([nombre(\$ISBN/ISSN\$), valor(\$\$)]),
 campo([nombre(\$NIPO\$), valor(\$\$)]), campo([nombre(\$Materias\$), valor(\$1. Matemáticas\$)]),
 campo([nombre(\$Secundarias\$), valor(\$I. Título\$)]), campo([nombre(\$Registro/s\$), valor(\$4385,
 4386\$)]), campo([nombre(\$Anotaciones\$), valor(\$\$)]).

d(3704,[campo([nombre(\$Signatura\$), valor(\$4767\$)]), campo([nombre(\$CDU\$), valor(\$\$)]), campo
 ([nombre(\$Encabezamiento\$), valor(\$BOTELLA, Juan\$)]), campo([nombre(\$Título_Uniforme\$),
 valor(\$\$)]), campo([nombre(\$Título_Mención\$), valor(\$Análisis de datos en psicología I / Juan
 Botella, Orfelio G. León, Rafael San Martín\$)]), campo([nombre(\$Edición\$), valor(\$\$)]),
 campo([nombre(\$Publicación\$), valor(\$Madrid : Pirámide, D.L. 1992\$)]),
 campo([nombre(\$Colación\$), valor(\$399 p. ; 23 cm\$)]), campo([nombre(\$Serie\$),
 valor(\$Psicología\$)]), campo([nombre(\$Notas\$), valor(\$\$)]), campo([nombre(\$ISBN/ISSN\$),
 valor(\$84-368-0717-0 (Tomo Y\$)]), campo([nombre(\$NIPO\$), valor(\$\$)]),
 campo([nombre(\$Materias\$), valor(\$1. Estadística matemática\$)]),
 campo([nombre(\$Secundarias\$), valor(\$I. León, Orfelio G. II. San Martín, Rafael III. Título\$)]),
 campo([nombre(\$Registro/s \$), valor(\$4767\$)]), campo([nombre(\$Anotaciones\$), valor(\$\$)]).

Índice

d(0,[autor([inv([apell(\$MACCORMAC\$),nombre(\$Pat\$))]),regs([reg(\$4\$),reg(\$1868\$)))]).

d(1,[autor([inv([apell(\$MACDONALD\$),nombre(\$Jeannette\$))]),regs([reg(\$518\$),reg(\$806\$)))]).

d(2,[autor([inv([apell(\$MACINTIRE\$),nombre(\$Lani\$))]),regs([reg(\$5497\$)))]).

d(3,[autor([inv([apell(\$MACIVER\$),nombre(\$Allan\$))]),regs([reg(\$3829\$),reg(\$4128\$)))]).

d(4,[autor([inv([apell(\$MACKENZIE\$),nombre(\$Mike\$))]),regs([reg(\$6100\$)))]).

d(5,[autor([inv([apell(\$MACKERSIE\$),nombre(\$David\$))]),regs([reg(\$4304\$),reg(\$4900\$),reg(\$5280\$),reg(\$5306\$)))]).

d(6,[autor([dir(\$MACKINNEY'S COTTON PICKERS\$))],regs([reg(\$242\$)))]).

d(7,[autor([inv([apell(\$MACKINTOSH\$),nombre(\$Ken\$))]),regs([reg(\$5000\$),reg(\$5346\$),reg(\$5489\$),reg(\$5537\$),reg(\$5655\$),reg(\$5741\$),reg(\$5750\$),reg(\$5788\$),reg(\$5955\$),reg(\$6036\$)))]).

d(8,[autor([dir(\$THE MACKPIESEN\$))],regs([reg(\$5000\$)))]).

d(9,[autor([dir(\$MACHADO\$))],regs([reg(\$3988\$),reg(\$4111\$),reg(\$4356\$),reg(\$4788\$)))]).

d(10,[autor([inv([apell(\$MACHADO\$),nombre(\$Armando\$))]),regs([reg(\$3142\$)))]).

d(11,[autor([inv([apell(\$MACHADO\$),nombre(\$Leocadio R.\$))]),regs([reg(\$4094\$),reg(\$4178\$),reg(\$4293\$)))]).

d(12,[autor([inv([apell(\$MACHIN\$),nombre(\$Antonio\$))]),regs([reg(\$889\$),reg(\$1188\$),reg(\$1236\$),reg(\$1365\$),reg(\$1391\$),reg(\$1392\$),reg(\$1727\$),reg(\$1766\$),reg(\$1845\$),reg(\$1855\$),reg(\$1925\$),reg(\$2134\$),reg(\$2142\$),reg(\$2163\$),reg(\$2174\$),reg(\$2188\$),reg(\$2221\$),reg(\$2300\$),reg(\$2305\$),reg(\$2379\$),reg(\$2395\$),reg(\$2427\$),reg(\$2428\$),reg(\$2439\$),reg(\$2457\$),reg(\$2504\$),reg(\$2515\$),reg(\$2534\$),reg(\$2540\$),reg(\$2635\$),reg(\$2647\$),reg(\$2659\$),reg(\$2663\$),reg(\$2765\$),reg(\$2823\$),reg(\$2827\$),reg(\$2837\$),reg(\$2880\$),reg(\$2923\$),reg(\$2930\$),reg(\$2931\$),reg(\$2949\$),reg(\$2978\$),reg(\$2981\$),reg(\$2994\$),reg(\$3064\$),reg(\$3108\$),reg(\$3147\$),reg(\$3154\$),reg(\$3165\$),reg(\$3306\$),reg(\$3392\$),reg(\$3448\$),reg(\$3452\$),reg(\$3520\$),reg(\$3547\$),reg(\$3588\$),reg(\$3647\$),reg(\$3907\$),reg(\$3912\$),reg(\$3929\$),reg(\$3973\$),reg(\$4187\$),reg(\$

\$4236\$),reg(\$4345\$),reg(\$4506\$),reg(\$4820\$),reg(\$4927\$),reg(\$5037\$),reg(\$5038\$),reg(\$5335\$),reg(\$5470\$),reg(\$5476\$),reg(\$6175\$),reg(\$6213\$),reg(\$6256\$)))]).

d(13,[autor([dir(\$MACHITO\$)],regs([reg(\$3189\$),reg(\$3861\$),reg(\$4343\$)))]).

d(14,[autor([inv([apell(\$MADRID\$),nombre(\$Manolito de\$)))]),regs([reg(\$1460\$)))]).

d(15,[autor([inv([apell(\$MADRIGUERAS\$),nombre(\$Enric\$)))]),regs([reg(\$1157\$),reg(\$2875\$),reg(\$2906\$),reg(\$2966\$)))]).

d(16,[autor([dir(\$MAGOLI\$)],regs([reg(\$4170\$)))]).

d(17,[autor([inv([apell(\$MAGYARI\$),nombre(\$Marika\$)))]),regs([reg(\$2456\$),reg(\$2541\$),reg(\$2604\$),reg(\$2633\$),reg(\$2662\$),reg(\$2743\$)))]).

d(18,[autor([inv([apell(\$MAIRENAS\$),nombre(\$Antonio\$)))]),regs([reg(\$3531\$),reg(\$3616\$),reg(\$4125\$),reg(\$4227\$)))]).

d(19,[autor([inv([apell(\$MAIRENAS\$),nombre(\$Pepe\$)))]),regs([reg(\$6233\$)))]).

d(20,[autor([dir(\$MALANDO Y SU ORQUESTA\$)],regs([reg(\$5698\$)))]).

d(21,[autor([inv([apell(\$MALCUZYNSKI\$),nombre(\$Witold\$)))]),regs([reg(\$3353\$),reg(\$3467\$),reg(\$3507\$),reg(\$3562\$),reg(\$3563\$),reg(\$3594\$),reg(\$4688\$)))]).

d(22,[autor([inv([apell(\$MALDONADO\$),nombre(\$F. Z.\$)))]),regs([reg(\$3982\$),reg(\$4228\$)))]).

d(23,[autor([inv([apell(\$MALERAS\$),nombre(\$Emma\$)))]),regs([reg(\$2943\$),reg(\$3493\$),reg(\$3663\$),reg(\$5041\$),reg(\$5477\$),reg(\$5524\$),reg(\$5557\$),reg(\$5573\$),reg(\$5732\$),reg(\$5804\$),reg(\$5810\$),reg(\$5872\$),reg(\$6007\$)))]).

d(24,[autor([inv([apell(\$MALERBAS\$),nombre(\$Alfredo\$)))]),regs([reg(\$4131\$)))]).

d(25,[autor([inv([apell(\$MALKO\$),nombre(\$Nicolai\$)))]),regs([reg(\$3217\$),reg(\$3480\$),reg(\$3769\$),reg(\$4018\$)))]).

d(26,[autor([inv([apell(\$MALVINS\$),nombre(\$Artie\$)))]),regs([reg(\$3407\$)))]).

d(27,[autor([dir(\$MANDARINO\$)],regs([reg(\$707\$),reg(\$1236\$)))]).

d(28,[autor([inv([apell(\$MANGANOS\$),nombre(\$Silvana\$)))]),regs([reg(\$4837\$)))]).

d(29,[autor([inv([apell(\$MANIETS\$),nombre(\$Roger\$)))]),regs([reg(\$4651\$)))]).

d(30,[autor([inv([apell(\$MANNARINI\$),nombre(\$Ida\$))]),regs([reg(\$1249\$),reg(\$1614\$),reg(\$1675\$)))]).

d(31,[autor([dir(\$MANOLO EL DE HUELVA\$))],regs([reg(\$543\$),reg(\$647\$)))]).

d(32,[autor([dir(\$MANOLO "EL GAFAS"\$))],regs([reg(\$1187\$),reg(\$1190\$),reg(\$1621\$)))]).

d(33,[autor([dir(\$MANOLO EL GRANAINO\$))],regs([reg(\$5351\$)))]).

d(34,[autor([dir(\$MANOLO EL MALAGUENO\$))],regs([reg(\$1398\$),reg(\$1488\$),reg(\$1729\$),reg(\$1854\$),reg(\$2261\$),reg(\$2492\$),reg(\$2498\$),reg(\$2563\$),reg(\$2728\$),reg(\$2746\$),reg(\$2899\$),reg(\$3019\$),reg(\$3719\$),reg(\$4392\$),reg(\$4395\$),reg(\$4527\$),reg(\$5238\$),reg(\$5351\$),reg(\$5521\$),reg(\$5527\$),reg(\$5625\$),reg(\$5700\$),reg(\$5721\$)))]).

d(35,[autor([inv([apell(\$MANONE\$),nombre(\$Wingie\$))]),regs([reg(\$661\$),reg(\$1280\$),reg(\$3465\$)))]).

d(36,[autor([inv([apell(\$MANTOVANI\$),nombre(\$Annunzio\$))]),regs([reg(\$1184\$),reg(\$1308\$),reg(\$1618\$),reg(\$1771\$),reg(\$1948\$),reg(\$3844\$),reg(\$4205\$),reg(\$4354\$),reg(\$4359\$),reg(\$4504\$),reg(\$4590\$),reg(\$4611\$),reg(\$4657\$),reg(\$4828\$),reg(\$4905\$),reg(\$5060\$),reg(\$5317\$),reg(\$5405\$),reg(\$5593\$),reg(\$5645\$),reg(\$5735\$),reg(\$5777\$),reg(\$5831\$),reg(\$5985\$)))]).

d(37,[autor([inv([apell(\$MANZANILLA\$),nombre(\$Manolo\$))]),regs([reg(\$2063\$),reg(\$2263\$),reg(\$2387\$),reg(\$4496\$)))]).

d(38,[autor([dir(\$MANZANITO DE CASTUERA\$))],regs([reg(\$2675\$)))]).

d(39,[autor([inv([apell(\$MAÑEZ\$),nombre(\$Encarnita\$))]),regs([reg(\$2872\$),reg(\$3239\$)))]).

d(40,[autor([dir(\$MARAVIGLIA\$))],regs([reg(\$4206\$),reg(\$4569\$),reg(\$5441\$)))]).

d(41,[autor([inv([apell(\$MARAVILLA\$),nombre(\$Juanito\$))]),regs([reg(\$3253\$),reg(\$3285\$),reg(\$3523\$)))]).

d(42,[autor([inv([apell(\$MARAVILLA\$),nombre(\$Luis\$))]),regs([reg(\$1187\$),reg(\$1190\$),reg(\$1288\$),reg(\$1301\$),reg(\$1302\$),reg(\$1621\$),reg(\$2436\$),reg(\$2568\$),reg(\$2820\$),reg(\$2861\$),reg(\$3953\$),reg(\$4272\$),reg(\$4582\$),reg(\$4814\$),reg(\$4816\$),reg(\$6243\$)))]).

d(43,[autor([inv([apell(\$MARCEN\$),nombre(\$Eduardo\$))]),regs([reg(\$1380\$)))]).

d(44,[autor([inv([apell(\$MARC\$),nombre(\$Jean\$))]),regs([reg(\$3257\$)))]).

d(45,[autor([inv([apell(\$MARCO\$),nombre(\$Jean\$))]),regs([reg(\$3221\$),reg(\$4030\$)))]).

d(46,[autor([inv([apell(\$MARCO\$),nombre(\$Salome\$))]),regs([reg(\$5380\$)])]).

d(47,[autor([inv([apell(\$MARCO\$),nombre(\$V.\$))]),regs([reg(\$6450\$)])]).

d(48,[autor([inv([apell(\$MARCHAND\$),nombre(\$Colette\$))]),regs([reg(\$4902\$),reg(\$5126\$)])]).

d(49,[autor([inv([apell(\$MARCHENA\$),nombre(\$Antonio de\$))]),regs([reg(\$2736\$)])]).

Villancicos

d(0,[registro(\$263\$),autor([apell(\$SALAZAR\$),nombre(\$Diego José\$)]),titulo(\$Letras de los villancicos, que se cantaron en la... Iglesia Metropolitana y Patriarcal... de Sevilla, en los solemnes Maytines del Nacimiento de Nuestro Señor Jesu-Christo... de 1706\$), resp(\$compuestos por D. Diego Ioseph de Salazar, Racionero y Maestro de Capilla\$), pie_imprenta(\$Impresso en Sevilla: por Juan Francisco de Blas, 1706\$), d_fisica(\$[8] p.; 4º\$), signatura(\$Sign.: A*-4\$),notas(\$Datos de pie de imp. tomados del colofón.—Texto a dos col\$), v([num(\$1\$), villa(\$"Para libertar su Reyno..."\$), partes(\$[Int., Estr. y Coplas]\$)], v([num(\$2\$), villa(\$"Zagalejas al Niño adoremos..."\$), partes(\$[Estr. y Coplas]\$)], v([num(\$3\$), villa(\$"Con varias mercaderias..."\$), partes(\$[Int., Estr. y Coplas]\$)], v([num(\$4\$), villa(\$"Pajarillo canoro..."\$), partes(\$[Estr. y Coplas]\$)], v([num(\$5\$), villa(\$"Cytaras de cristal..."\$), partes(\$[Estr. y Coplas]\$)], v([num(\$7\$), villa(\$"Por divertir à el Dios Niño..."\$), partes(\$[Int., Estr. y Coplas]\$)], v([num(\$8\$), villa(\$"Niño mio, que al yelo..."\$), partes(\$[Estr. y Coplas]\$)], v([num(\$9\$), villa(\$"A Què tocan?..."\$), partes(\$[Estr. y Coplas]\$)], resto(\$Aguilar: Imp. sev., 88. · VE/ 531-37\$)).

d(1,[registro(\$264\$),autor([apell(\$MENDOZA\$),nombre(\$Gabriel José\$)]),titulo(\$Letras de los villancicos que se cantaron en la... Iglesia Colegial de Nuestro Señor San Salvador... de Sevilla, en la Kalenda y Maytines del Nacimiento de nuestro Señor Jesu-christo... de 1706\$), resp(\$por Don Gabriel Joseph de Mendoza, Maestro de Capilla\$), pie_imprenta(\$[Sevilla?: s.n., 1706?]\$), d_fisica(\$[8] p.\$; 4º\$), signatura(\$Sign.: A*-4\$), notas(\$Datos de pie de imp. deducidos del tit.—Texto a dos col.—Port. con orla tip. y viñeta xil. de la Adoración de los Pastores\$), v([num(\$1\$), villa(\$"Ha de la Cuna del Sol..."\$), partes(\$[Estr. y Coplas]\$)], v([num(\$2\$), villa(\$"Ha Señorito..."\$), partes(\$[Estr. y Coplas]\$)], v([num(\$3\$), villa(\$"A Belèn vãn Peregrinos..."\$), partes(\$[Int., Estr. y Coplas]\$)], v([num(\$4\$), villa(\$"Niño tierno, que rendido..."\$), partes(\$[Estr. y Coplas]\$)], v([num(\$5\$), villa(\$"Pasqual, q es archivo andante..."\$), partes(\$[Int., Estr. y Coplas]\$)], v([num(\$6\$), villa(\$"Los Pastores esta noche..."\$), partes(\$[Int., Estr. y Coplas]\$)], v([num(\$7\$), villa(\$"A siolos Molenos..."\$), partes(\$[Estr. y Coplas]\$)], v([num(\$8\$), villa(\$"Ay, què dolor!..."\$), partes(\$[Estr. y Coplas]\$)], resto(\$Aguilar: Imp. sev., 2138. · VE/ 531-46\$)).

d(2,[registro(\$265\$),autor([apell(\$SALAZAR\$),nombre(\$Diego José\$)]),titulo(\$Letras de los villancicos que se cantaron en la... Iglesia Metropolitana y Patriarcal de Sevilla, en los... Maytines de la Purissima é Inmaculada Concepcion de Maria..., este presente año de 1706\$), resp(\$compuestos por D. Diego Ioseph de Salazar, Racionero y Maestro de Capilla\$), pie_imprenta(\$Impresso en Sevilla: por Ivan Francisco de Blas.... [1706?]\$), d_fisica(\$[4] p.; 4º\$), signatura(\$Sign.: []*-2\$), notas(\$Lugar de imp. e impresor tomados del colofón y fecha deducida del tit.—Texto a dos col\$), v([num(\$1\$), villa(\$"Oy que el Padre Omnipotente..."\$),

partes(\$[Int., Estr. y Coplas]\$)), v([num(\$2\$), villa("\$Oygan, atiendan..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$3\$), villa("\$Como serà que las sombras..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$4\$), villa("\$Sagradas Luces, silencio..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$5\$), villa("\$Corran, corran las Fuentes..."\$), partes(\$[Estr. y Coplas]\$)), resto(\$ R/ 34199*-21\$)).

d(3,[registro(\$266\$), titulo(\$LETRAS de los villancicos que se han de cantar en los solemnes Maytines y celebre Fiesta que consagra y dedica la Hermandad de N. Señora de los Dolores en la Parroquial Iglesia... de... S. Marcos, este año 1706: dirigidas por la misma Hermandad al señor D. Juan de Arajal\$), resp(\$cantadas por la Capilla de Musica de la Colegial de N. Señor S. Salvador\$), pie_imprensa(\$En Sevilla: por Jvan de la Pverta..., [1706?]\$), d_fisica(\$[8] p.; 4°\$), signatura(\$Sign.: []*-4\$), notas(\$Fecha de imp. deducida del tit.—Texto a dos col.—Port. con orla tip. y viñeta xil. de la Virgen de los Dolores\$), v([num(\$1\$), villa("\$Los Serafines alados..."\$), partes(\$[Int., Estr. y Coplas]\$)), v([num(\$2\$), villa("\$Moradores del mundo..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$3\$), villa("\$Al arma, al arma, al arma..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$4\$), villa("\$Amante Aurora mia..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$5\$), villa("\$Los Cielos Soberanos..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$6\$), villa("\$Hermosa Arquitectura..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$7\$), villa("\$De los rebatos confusos..."\$), partes(\$[Int., Estr. y Coplas]\$)), v([num(\$8\$), villa("\$Qve dulce se muestra..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$9\$), villa("\$Soberanos Espiritus..."\$), partes(\$[Estr. y Coplas]\$)), resto(\$ VE/ 1201-30\$)).

d(4,[registro(\$267\$), autor([apell(\$SALAZAR\$), nombre(\$Diego José\$)]), titulo(\$Letras de los villancicos que se cantaron en la... Iglesia Metropolitana y Patriarcal... de Sevilla, en los solemnes Maytines de la Venida de los Santos Reyes... de 1707\$), resp(\$compuestos por Don Diego Joseph de Salazar, Racionero y Maestro de Capilla\$), pie_imprensa(\$En Sevilla: por Juan Francisco de Blas..., 1707\$), d_fisica(\$[8] p.; 4°\$), signatura(\$Sign.: A*-4\$), notas(\$Datos de pie de imp. tomados del colofón.—Texto a dos col.—Port. con orla tip. y viñeta xil. de la Giralda\$), v([num(\$1\$), villa("\$La brillante bellissima estrella..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$2\$), villa("\$Pastorcita soy..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$3\$), villa("\$Viendo que Reyes adoran..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$4\$), villa("\$Gallegos baylarines..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$5\$), villa("\$Pluma de oro que rasgos dilatas..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$6\$), villa("\$Dos Negritos tan acordes..."\$), partes(\$[Int., Estr. y Coplas]\$)), v([num(\$7\$), villa("\$Las Gitanas desde Egipto..."\$), partes(\$[Int., Estr. y Coplas]\$)), v([num(\$8\$), villa("\$Celebren los sentidos..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$9\$), villa("\$El cuento de la laila..."\$), partes(\$[Int., Estr. y Coplas]\$)), resto(\$ VE/ 531-36\$)).

d(5,[registro(\$268\$), titulo(\$LETRAS de los villancicos que en los solemnes Maytines de la Resurreccion de N. Señor Jesu-Christo cantaron los Niños Seyses, en la... Metropolitana y Patriarcal Iglesia de Sevilla, este año de 1707\$), resp(\$puestos en mvica por su Maestro D. Juan Donoso Cabeza de Baca\$), pie_imprensa(\$En Sevilla: por Ivan Francisco de Blas..., [1707?]\$), d_fisica(\$[4] p.\$), signatura(\$4° Sign.: []*-2\$), notas(\$Fecha de imp. deducida del tit.—Texto a dos

col.--Port. con orla tip. y viñeta xil. de Cristo resucitado\$,v([num(\$1\$),villa("\$Aquel Rey Soberano..."\$),partes(\$[Int., Estr. y Coplas]\$)),v([num(\$2\$),villa("\$Oy que resucita amante..."\$),partes(\$[Int., Estr. y Coplas]\$)),v([num(\$3\$),villa("\$Del Alva luciente..."\$),partes(\$[Estr. y Coplas]\$)),resto(\$Aguilar: Imp. sev., 99. · VE/ 531-61\$)).

d(6,[registro(\$269\$),autor([apell(\$SALAZAR\$),nombre(\$Diego José\$)),titulo(\$Letras de los villancicos, que se cantaron en la... Patriarcal Iglesia desta Ciudad de Sevilla, en los... Maytines del Nacimiento de N. Señor Iesu-Christo, este año de 1707\$), resp(\$compuesto por D. Diego Joseph de Salazar, Racionero y Maestro de Capilla de dicha S. Iglesia\$), pie_imprensa(\$En Sevilla: por Juan Francisco de Blas..., 1707\$),d_fisica(\$[8] p; 4º\$),signatura(\$Sign.: A*-4\$),notas(\$Datos de pie de imp. tomados del colofón.—Texto a dos col.—Port. con orla tip. y viñeta xil. que representa el Nacimiento\$), v([num(\$1\$), villa("\$O venturoso dia!..."\$),partes(\$[Estr. y Coplas]\$)), v([num(\$2\$),villa("\$Ay, que vna Andaluzica..."\$),partes(\$[Estr. y Coplas]\$)), v([num(\$3\$),villa("\$Los Vizcaynos, que es gente..."\$),partes(\$[Int., Estr. y Coplas]\$)), v([num(\$4\$),villa("\$Oygan el agudeza..."\$),partes(\$[Estr. y Coplas]\$)), v([num(\$5\$),villa("\$Ay Jesus, ay mi bien, ay mi Dios..."\$),partes(\$[Estr. y Coplas]\$)), v([num(\$6\$),villa("\$Qvatlo Negliyoz venimoz..."\$),partes(\$[Estr. y Coplas]\$)), v([num(\$7\$), villa("\$Pves nace mi Niño,..."\$),partes(\$[Estr. y Coplas]\$)),v([num(\$8\$),villa("\$Este es el Solio en q nace el amor,..."\$),partes(\$[Estr. y Coplas]\$)),v([num(\$9\$),villa("\$Alerta Zagal,..."\$),partes(\$[Estr. y Coplas]\$)),resto(\$· VE/ 1189-36\$)).

d(7,[registro(\$270\$),autor([apell(\$SALAZAR\$),nombre(\$Diego José\$)),titulo(\$Letras de los villancicos que se cantaror [sic] en la... Metropolitana y Patriarcal Iglesia... de Sevilla, en los... Maytines de la Purissima Concepcion de Maria\$),resp(\$compuestos por Don Diego Joseph de Salazar, Racionero y Maestro de Capilla\$),pie_imprensa(\$En Sevilla: por Juan Francisco de Blas..., [1707?]\$),d_fisica(\$[4] p; 4º\$), signatura(\$Sign.: [*-2\$),notas(\$Lugar de imp. e impresor tomados del colofón y fecha deducida de la actividad del Maestro de Capilla y de la fiesta.—Texto a dos col\$), v([num(\$1\$),villa("\$Oy sobre los montes Santos,..."\$),partes(\$[Int., Estr. y Coplas]\$)), v([num(\$2\$),villa("\$Albricias, Pastores,..."\$),partes(\$[Estr. y Coplas]\$)), v([num(\$3\$), villa("\$Celebre en el Impireo..."\$),partes(\$Estr- y Coplas\$)),v([num(\$4\$),villa("\$Ay como cantan,..."\$),partes(\$[Estr. y Coplas]\$)),v([num(\$5\$),villa("\$Paxarillos amantes,..."\$),partes(\$[Estr. y Coplas]\$)),resto(\$· VE/ 1305-115 Foliación ms.: "475-476".—(Barbieri\$)))).

d(8,[registro(\$271\$),autor([apell(\$SALAZAR\$),nombre(\$Diego José\$)), titulo(\$Letras de los villancicos que se cantaron en la... Metropolitana y Patriarcal Iglesia... de Sevilla, en los solemnes Maytines de la Venida de los Santos Reyes... de 1708\$),resp(\$compuestos por D. Diego Joseph de Salazar, Racionero y Maestro de Capilla\$), pie_imprensa(\$En Sevilla: por Juan Francisco de Blas..., [1708?]\$),d_fisica(\$[8] p.; 4º\$),signatura(\$Sign.: A*-4\$),notas(\$Lugar de imp. e impresor tomados del colofón y fecha deducida del tit.—Texto a dos col.—Port. con orla tip. y viñeta xil. de la Giralda\$), v([num(\$1\$), villa("\$Salva clarines sonoros..."\$),partes(\$[Estr. y Coplas]\$)), v([num(\$2\$),villa("\$Con todos sus instrumentos..."\$),partes(\$[Int., Estr. y Coplas]\$)),

v([num(\$3\$),villa("\$Pasito, silencio..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$4\$), villa("\$Coronadas de Rosas..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$5\$),villa("\$Morisco: Meguelejo..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$6\$),villa("\$Vv Astro flamante..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$7\$),villa("\$Ay Zagalejas, ay..."\$), partes(\$[Estr. y Cop~as]\$)),v([num(\$8\$), villa("\$Vn Barquillero, à Belen..."\$), partes(\$[Int., Estr. y Coplas]\$)), v([num(\$9\$), villa("\$Viendo algunas rudas Aves..."\$),partes(\$[Int., Estr. y Coplas]\$)), resto(\$Aguilar: Imp., sev., 124 · VE/ 1310-8 (Barbieri)\$)).

d(9,[registro(\$272\$),autor([apell(\$SANTISSO BERMUDEZ\$), nombre(\$Gregorio\$)],titulo(\$Letras de los villancicos que se cantaron en los... Maytines de la Purissima Concepcion... en la... Iglesia Metropolitana y Patriarcal de... Sevilla\$),resp(\$compuestos (vacante el Magisterio de Capilla) por D. Gregorio Santisso Bermudez, Presbytero, Maestro de Seises de dicha Santa Iglesia\$), pie_imprenta(\$[Sevilla?: s.n., 1709?]\$), d_fisica(\$[4] p; 4º\$),signatura(\$Sign.: []*-2\$), notas(\$Lugar de imp. deducido del tit. y fecha deducida del autor y del periodo en que esta vacante el magisterio de capilla.—Texto a dos col\$),v([num(\$1\$),villa("\$La Gracia, y la Embidia salen..."\$),partes(\$[Int., Estr. y Coplas]\$)), v([num(\$2\$),villa("\$Ay que me Abrasa vna llama..."\$), partes(\$[Estr. y Coplas]\$)),v([num(\$3\$),villa("\$Celebre en el Impireo..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$4\$),villa("\$Los nuevos Cantores..."\$), partes(\$[Estr. y Coplas]\$)), v([num(\$5\$),villa("\$Avecita, que en dulce armonia..."\$), partes(\$[Estr. y Coplas]\$)),resto(\$· VE/ 1305-112 Foliación ms.: "22-23".--(Barbieri)\$)).

Diccionario

d(1,[palabra(\$florales\$), origen\$(Del lat. florales ludi, juegos florales.)\$, acepcion([aux(\$adj. pl.\$), sigdo(\$Aplicase a las fiestas o juegos que celebraban los gentiles en honor de la diosa Flora. A su imitación se han instituido después en Provenza y en otras partes\$))), acepcion([num(\$2\$), vease([sigdo(\$juegos florales\$))]))).

d(2,[palabra(\$florar\$), acepcion([aux(\$intr.\$), sigdo(\$Dar flor. Dícese de los árboles y las plantas, singularmente de los que se cultivan para cosechar sus frutos\$)))).

d(3,[palabra(\$flordelizado, da\$), acepcion([aux(\$adj. Blas.\$), vease([sigdo(\$cruz flordelizada\$))]))).

d(4,[palabra(\$flordelisar\$), acepcion([aux(\$tr. Blas.\$), sigdo(\$Adornar con flores de lis una cosa\$)))).

d(5,[palabra(\$floreado, da\$), acepcion([aux(\$p. p de florear\$), sigdo(\$\$)), acepcion([num(\$2\$), aux(\$adj.\$), vease([sigdo(\$pan floreado\$))]))).

d(6,[palabra(\$florear\$), origen\$(Del fr. floréal.)\$, acepcion([aux(\$m.\$), sigdo(\$Octavo mes del calendario republicano francés, cuyos días primero y último coincidían, respectivamente, con el 20 de abril y el 19 de mayo\$)))).

d(7,[palabra(\$florear\$), acepcion([aux(\$tr.\$), sigdo(\$Adornar y guarnecer con flores\$))), acepcion([num(\$2\$), sigdo(\$Tratándose de la harina, sacar la primera y más sutil por medio del cedazo más espeso\$))), acepcion([num(\$3\$), sigdo(\$Disponer el naipe para hacer trampa\$))), acepcion([num(\$4\$), aux(\$intr.\$), sigdo(\$Vibrar, mover la punta de la espada\$))), acepcion([num(\$5\$), sigdo(\$Tocar dos o tres cuerdas de la guitarra con tres dedos sucesivamente sin parar, formando así un sonido continuado\$))), acepcion([num(\$6\$), aux(\$fam.\$), sigdo(\$Echar a una mujer piropos\$))), acepcion([num(\$7\$), aux(\$Ar.\$), sigdo(\$Chile y Sal. Escoger lo mejor de una cosa\$)))).

d(8,[palabra(\$florecedor, ra\$), acepcion([aux(\$adj.\$), sigdo(\$Que florece\$)))).

d(9,[palabra(\$florecer\$), origen\$(De florescer.)\$, acepcion([aux(\$intr.\$), sigdo(\$Echar o arrojar flor\$), usase(\$ú. t. c. tr.\$))), acepcion([num(\$2\$), aux(\$fig.\$), sigdo(\$Prosperar crecer en riqueza o reputación. Dícese también de los entes morales; como la justicia, las ciencias, etc\$))), acepcion([num(\$3\$), aux(\$fig.\$), sigdo(\$Existir una persona o cosa insigne en un tiempo o época

determinada\$)), acepcion([num(\$4\$), aux(\$prnl.\$), sigdo(\$Hablando de algunas cosas, como el queso, pan, etc., ponerse mohosas\$)))).

d(10,[palabra(\$florecente\$), acepcion([aux(\$p. a. de florecer\$), sigdo(\$Que florece\$)), acepcion([num(\$2\$), aux(\$fig.\$), sigdo(\$Favorable, venturoso, próspero\$)))]).

d(11,[palabra(\$florecimiento\$), acepcion([aux(\$m.\$), sigdo(\$Acción y efecto de florecer o florecerse\$)))]).

d(12,[palabra(\$Florencia\$), acepcion([aux(\$n. p.\$), vease([sigdo(\$raja de Florencia\$)))]).

d(13,[palabra(\$floreantin\$), acepcion([aux(\$adj.\$), sigdo(\$floreantino. Apl. a pers.\$), usase(\$ú. t. c. s.\$)))]).

d(14,[palabra(\$floreantino, na\$), origen(\$Del lat. florentinus.\$), acepcion([aux(\$adj.\$), sigdo(\$Natural de Florencia\$), usase(\$ú. t. c. s.\$))], acepcion([num(\$2\$), sigdo(\$Perteneiente a esta ciudad de Italia\$)))]).

d(15,[palabra(\$floreantísimo, ma\$), origen(\$Del lat. florentissimus.\$), acepcion([aux(\$adj.\$), sigdo(\$sup. de floreciente. Que prospera o florece con excelencia\$)))]).

d(16,[palabra(\$floreos\$), origen(\$De florear.\$), acepcion([aux(\$m. fig.\$), sigdo(\$Conversación vana y de pasatiempo\$)), acepcion([num(\$2\$), aux(\$fig.\$), sigdo(\$Dicho vano y superfluo empleado sin otro fin que el de hacer alarde de ingenio, o el de halagar o lisonjear al oyente, o sólo por mero pasatiempo\$)), acepcion([num(\$3\$), sigdo(\$Danza. En la danza española, movimiento de un pie en el aire cuando el otro permanece en el suelo, y el cuerpo sostenido sobre él\$)), acepcion([num(\$4\$), sigdo(\$Esgr. Vibración o movimiento de la punta de la espada\$)), acepcion([num(\$5\$), sigdo(\$Mús. Acción de florear en la guitarra\$)))]).

d(17,[palabra(\$florería\$), acepcion([aux(\$f.\$), sigdo(\$Tienda donde se venden flores y plantas de adorno\$)))]).

d(18,[palabra(\$florero, ra\$), acepcion([aux(\$adj. fig.\$), sigdo(\$Que usa de palabras chistosas y lisonjeras\$), usase(\$ú. t. c. s.\$))], acepcion([num(\$2\$), aux(\$m. y f.\$), sigdo(\$El que o la que vende flores\$)), acepcion([num(\$3\$), aux(\$m.\$), sigdo(\$Vaso para poner flores naturales o artificiales\$)), acepcion([num(\$4\$), sigdo(\$Maceta o tiesto con flores\$)), acepcion([num(\$5\$), sigdo(\$Armario, caja o lugar destinado para guardar flores\$)), acepcion([num(\$6\$), sigdo(\$Germ. Fullero que hace trampas floreado el naipe\$)), acepcion([num(\$7\$), sigdo(\$Pint. Cuadro en que sólo se representan flores\$)))]).

- d(19,[palabra(\$florescencia\$),acepcion([aux(\$f.\$),sigdo(\$florescencia\$)]), acepcion([num(\$2\$), sigdo(\$Bot. Acción de florecer\$)]), acepcion([num(\$3\$), sigdo(\$Bot. Época en que las plantas florecen, o aparición de las flores en cada vegetal\$)])))).
- d(20,[palabra(\$florescer\$), origen(\$(Del lat. florescere.)\$), acepcion([aux(\$intr. ant.\$), sigdo(\$florecer\$)])))).
- d(21,[palabra(\$foresta\$), origen(\$(Del b. lat. foresta, y éste de foras, de fuera.)\$), acepcion([aux(\$f.\$), sigdo(\$Terreno frondoso y ameno poblado de árboles\$)]), acepcion([num(\$2\$), aux(\$fig.\$), sigdo(\$Reunión de cosas agradables y de buen gusto\$)])))).
- d(22,[palabra(\$florete\$), origen(\$(d. de flor.)\$), acepcion([aux(\$s.\$), sigdo(\$Entre guarnicioneros, bordadura sobrepuesta que sirve de fuerza y adorno en los extremos de las cinchas\$)]), acepcion([num(\$2\$), sigdo(\$Danza. En la danza española, tejido o movimiento que se hacia con ambos pies\$)])))).
- d(23,[palabra(\$floreteada\$), origen(\$(De florete.)\$), acepcion([aux(\$f. ant.\$), sigdo(\$Papirote dado en la frente\$)])))).
- d(24,[palabra(\$floreteazo\$), acepcion([aux(\$m.\$), sigdo(\$Golpe dado con el florete\$)])))).
- d(25,[palabra(\$florete\$), origen(\$(Del fr. fleuret, y este del ital. fioretto, del lat. flos, floris.)\$), acepcion([aux(\$adj.\$), vease([sigdo(\$azúcar, papel florete\$)]))], acepcion([num(\$2\$), aux(\$m.\$), sigdo(\$Esgrima con espadín\$)]), acepcion([num(\$3\$), sigdo(\$Espadín destinado a la enseñanza o ejercicio de este juego; es de cuatro aristas, y no suele tener aro en la empuñadura\$)]), acepcion([num(\$4\$), sigdo(\$Lienzo o tela entrefina de algodón\$)])))).
- d(26,[palabra(\$floretear\$), origen(\$(De florete.)\$), acepcion([aux(\$tr.\$), sigdo(\$Adornar y guarnecer con flores una cosa\$)]), acepcion([num(\$2\$), aux(\$intr.\$), sigdo(\$Manejar el florete\$)])))).
- d(27,[palabra(\$floreteo\$), acepcion([aux(\$m.\$), sigdo(\$Acción y efecto de floretear\$)])))).
- d(28,[palabra(\$floreteista\$), acepcion([aux(\$m.\$), sigdo(\$El que es diestro en el juego del florete\$)])))).
- d(29,[palabra(\$floricultor, ra\$), origen(\$(Del lat. flos, floris, flor, y cultos, -oris, cultivador.)\$), acepcion([aux(\$m. y f.\$), sigdo(\$Persona dedicada a la floricultura\$)])))).
- d(30,[palabra(\$floricultura\$), origen(\$(Del lat. Los, floris, flor. y cultura, cultivo.)\$), acepcion([aux(\$f.\$), sigdo(\$Cultivo de las flores\$)]), acepcion([num(\$2\$), sigdo(\$Arte que lo enseña\$)])))).

- d(31,[palabra(\$floridamente\$), acepcion([aux(\$adv. m. fig.\$), sigdo(\$Con elegancia y gracia\$))]).
- d(32,[palabra(\$floridano, na\$), acepcion([aux(\$adj.\$), sigdo(\$Natural de la Florida\$), usase(\$ú. t. c. s.\$)), acepcion([num(\$2\$), sigdo(\$Perteneiente a este Estado de América del Norte\$))]).
- d(33,[palabra(\$floridez\$), origen(\$De florido.\$), acepcion([aux(\$f.\$), sigdo(\$Abundancia de flores. La FLORIDEZ de la primavera\$)), acepcion([num(\$2\$), aux(\$fig.\$), sigdo(\$Calidad de florido, en el estilo. florido. da. adj. Que tiene flores\$)), acepcion([num(\$2\$), vease([sigdo(\$junco florido\$))]), acepcion([num(\$3\$), vease([sigdo(\$letra, pascua florida\$))]), acepcion([num(\$4\$), aux(\$fig.\$), sigdo(\$Dícese de lo más escogido de alguna cosa\$)), acepcion([num(\$5\$), aux(\$fig.\$), sigdo(\$Dícese del lenguaje o estilo amena y profusamente exornado de galas retóricas\$)), acepcion([num(\$6\$), sigdo(\$Germ. Rico, opulento\$))]).
- d(34,[palabra(\$florífero, ra\$), origen(\$Del lat. florifer, -era; de flos, floris, flor, y ferre, llevar.\$), acepcion([aux(\$adj.\$), sigdo(\$Que lleva o produce flores\$))]).
- d(35,[palabra(\$florífero, ra\$), origen(\$Del lat, florifer, -era; de flos, floris, flor, y gerere. llevar.\$), acepcion([aux(\$adj.\$), sigdo(\$poét. florífero\$))]).
- d(36,[palabra(\$florilegio\$), origen(\$Del lat. Los, floris, flor, y legere, escoger.\$), acepcion([aux(\$m. fig.\$), sigdo(\$Colección de trozos selectos de materias literarias\$))]).
- d(37,[palabra(\$florín\$), origen(\$Del ital. fiorino, moneda florentina marcada con el lirio de los Médicis.\$), acepcion([aux(\$m.\$), sigdo(\$Moneda de plata equivalente al escudo de España, que se usa en algunos países, especialmente en Austria y Holanda, y estuvo marcada antiguamente con una flor de lis\$)), acepcion([num(\$2\$), sigdo(\$Moneda de oro mandada acuñar por los reyes de Aragón copiando los florines o ducados de Florencia, su valor sufrió grandes alteraciones por efecto del abuso de liga en el metal\$))]).
- d(38,[palabra(\$floripondio\$), origen(\$Quizá de flor y el dialect. predio, del lat. *pendidus, pendiente.\$), acepcion([aux(\$m.\$), sigdo(\$Arbusto del Perú, de la familia de las solanáceas, que crece hasta tres metros de altura con tronco leñoso, hojas grandes, alternas, oblongas, enteras y vellosas, flores solitarias, blancas, en forma de embudo, de unos tres decímetros de largo, de olor delicioso, pero perjudicial si se aspira mucho tiempo, y fruto elipsoidal, con muchas semillas pequeñas de figura de riñón\$)), acepcion([num(\$2\$), aux(\$fig.\$), sigdo(\$despect. Flor grande que suele figurar en adornos de mal gusto\$))]).
- d(39,[palabra(\$florista\$), acepcion([aux(\$com.\$), sigdo(\$Persona que fabrica flores de mano\$)), acepcion([num(\$2\$), sigdo(\$Persona que vende flores\$))]).
- d(40,[palabra(\$floristería\$), acepcion([aux(\$f.\$), sigdo(\$florería\$))]).

d(41,[palabra(\$florlisar\$), acepcion([aux(\$tr. Blas.\$), sigdo(\$flordelisar\$)])]).

d(42,[palabra(\$florón\$), acepcion([aux(\$m. aum. de flor\$), sigdo(\$\$))], acepcion([num(\$2\$), sigdo(\$Adorno hecho a manera de flor muy grande, que se usa en pintura y arquitectura en el centro de los techos de las habitaciones, etc\$)]), acepcion([num(\$3\$), aux(\$Blas.\$), sigdo(\$Adorno, a manera de flor, que se pone en el circulo de\$)]])).

BOE

d(1,[fecha(\$5 de mayo de 1995\$), uni(\$Universidad de Vigo\$),
info([categoria([cargo(\$Catedrática\$), entidad(\$Escuela Universitaria\$)), area(\$"Filología
Francesa"\$), nombre(\$Terencia Inés Silva Rojas\$))]).

d(2,[fecha(\$20 de abril de 1995\$), uni(\$Universidad de Málaga\$),
info([categoria([cargo(\$Profesora\$)), nombre(\$Eva Millán Valideperas\$))]).

d(3,[fecha(\$20 de abril de 1995\$), uni(\$Universidad de Málaga\$),
info([categoria([cargo(\$Profesor\$)), nombre(\$Carlos Rossi Jiménez\$))]).

d(4,[fecha(\$24 de abril de 1995\$), uni(\$Universidad de Málaga\$),
info([categoria([cargo(\$Profesora\$)), nombre(\$María Lourdes Rey Borrego\$))]).

d(5,[fecha(\$24 de abril de 1995\$), uni(\$Universidad de Málaga\$),
info([categoria([cargo(\$Profesor\$)), nombre(\$José Ramón Díaz Fernández\$))]).

d(6,[fecha(\$28 de abril de 1995\$), uni(\$Universidad Rovira - Virgili, de Tarragona\$), ign((info,
ign(? i)))).

d(7,[fecha(\$3 de mayo de 1995\$), uni(\$Universidad de Cádiz\$), ign((info, ign(? i)))).

d(8,[fecha(\$8 de mayo de 1995\$), uni(\$Universidad Pública de Navarra\$), info([nombre(\$Irene
Zabala Martínez\$), categoria([cargo(\$Profesora\$), entidad(\$Universidad\$)),
area(\$"Electrónicas"\$))]).

d(9,[fecha(\$8 de mayo de 1995\$), uni(\$Universidad Pública de Navarra\$), info([nombre(\$Gabriel
Lera Carreras\$), categoria([cargo(\$Profesor titular\$), entidad(\$Universidad\$)), area(\$"Ingeniería
de Sistemas y Automática"\$))]).

d(10,[fecha(\$8 de mayo de 1995\$), uni(\$Universidad Pública de Navarra\$), info([nombre(\$Blanca
Fernández Viguera\$), categoria([cargo(\$Profesora titular\$), entidad(\$Escuela Universitaria\$)),
area(\$"Trabajo Social y Servicios Sociales"\$))]).

d(11,[fecha(\$8 de mayo de 1995\$), uni(\$Universidad Pública de Navarra\$),
info([nombre(\$Concepción Corera Oroz\$), categoria([cargo(\$Profesora titular\$)), area(\$"Trabajo
Social y Servicios Sociales"\$))]).

d(12,[fecha(\$8 de mayo de 1995\$), uni(\$Universidad Pública de Navarra\$), info([nombre(\$José Basilio Galván Herrera\$), categoria([cargo(\$Profesor titular\$), entidad(\$Universidad\$)]), area(\$"Ingeniería de Sistemas y Automática"\$)])))).

d(13,[fecha(\$9 de mayo de 1995\$), uni(\$Universidad de Santiago de Compostela\$), info([categoria([cargo(\$Profesor titular\$), entidad(\$Universidad\$)]), area(\$"Cirugía"\$), nombre(\$Julián Álvarez Escudero\$)])))).

d(14,[fecha(\$9 de mayo de 1995\$), uni(\$Universidad de Huelva\$), info([nombre(\$Emilio Pascual Martínez\$), categoria([cargo(\$Catedrático\$), entidad(\$Universidad\$)]), area(\$"Petrología y Geoquímica"\$)])))).

d(15,[fecha(\$3 de mayo de 1995\$), uni(\$Universidad de León\$), info([nombre(\$Eduardo Sánchez Compadre\$), categoria([cargo(\$Profesor titular\$), entidad(\$Escuela Universitaria\$)]), area(\$"Biología Animal"\$)])))).

d(16,[fecha(\$5 de mayo de 1995\$), uni(\$Universidad de Málaga\$), info([categoria([cargo(\$Profesor\$)]), nombre(\$don Alfredo Rubio Díaz\$)])))).

d(17,[fecha(\$5 de mayo de 1995\$), uni(\$Universidad de Vigo\$), info([categoria([cargo(\$Profesora titular\$), entidad(\$Universidad\$)]), area(\$"Matemática Aplicada"\$), nombre(\$María del Carmen Suárez Rodríguez\$)])))).

d(18,[fecha(\$5 de mayo de 1995\$), uni(\$Universidad de Vigo\$), info([categoria([cargo(\$Catedrática\$), entidad(\$Escuela Universitaria\$)]), area(\$"Química Orgánica"\$), nombre(\$María Generosa de los Angeles Gómez Pacios\$)])))).

d(19,[fecha(\$5 de mayo de 1995\$), uni(\$Universidad de Vigo\$), ign(info), ind(\$\$), ind(\$Profesora titular de Universidad del área de conocimiento de "Biología Vegetal"\$)])))).

d(20,[fecha(\$8 de mayo de 1995\$), uni(\$Universidad Pública de Navarra\$), info([nombre(\$Gabriel Lera Carreras\$), categoria([cargo(\$Profesor titular\$), entidad(\$Universidad\$)]), area(\$"Ingeniería de Sistemas y Automática"\$)])))).

d(21,[fecha(\$8 de mayo de 1995\$), uni(\$Universidad Pública de Navarra\$), info([nombre(\$Blanca Fernández Viguera\$), categoria([cargo(\$Profesora titular\$), entidad(\$Escuela Universitaria\$)]), area(\$"Trabajo Social y Servicios Sociales"\$)])))).

d(22,[fecha(\$8 de mayo de 1995\$), uni(\$Universidad Pública de Navarra\$), info([nombre(\$Concepción Corera Oroz\$), categoria([cargo(\$Profesora titular\$), entidad(\$Escuela Universitaria\$)]), area(\$"Trabajo Social y Servicios Sociales"\$)])))).

d(23,[fecha(\$5 de mayo de 1995\$), uni(\$Universidad de Vigo\$), info([categoria([cargo(\$Profesora titular\$), entidad(\$Universidad\$)]), area(\$"Biología Vegetal"\$), nombre(\$Aida García Molaes\$)])))).

Anuncios de prensa

d(2,[zona(\$CANILLEJAS\$),descripcion([planta(\$cuarto\$),dorms(\$cuatro\$),metros(\$120\$),baños(\$dos\$),situacion(\$exterior\$)]),precio(\$23.000.000\$),telefono([prefijo(\$91\$),tfno(\$4305412\$)]))).

d(3,[zona(\$ARTURO SORIA\$), descripcion([dorms(\$cuatro\$), baños(\$dos\$)]), telefono([prefijo(\$91\$), tfno(\$5165921\$)]))).

d(4,[zona(\$TETUAN\$),descripcion([metros(\$105\$),situacion(\$exterior\$)]),precio(\$17.900.000\$),telefono([prefijo(\$91\$),tfno(\$5718466\$)]))).

d(5,[zona(\$VAGUADA\$),descripcion([dorms(\$dos\$),metros(\$65\$)]),precio(\$13.000.000\$),comentario(\$Facilidades\$)).

d(6,[zona(\$MORATALAZ\$),descripcion([planta(\$cuarta\$),dorms(\$cuatro\$),estado(\$nuevo\$),baños(\$dos\$)]),precio(\$14.950.000\$),telefono([prefijo(\$91\$),tfno(\$4305412\$)]))).

d(7,[zona(\$MORATALAZ\$),descripcion([planta(\$cuarto\$),dorms(\$tres\$),metros(\$120\$),baños(\$dos\$),situacion(\$exterior\$)]),precio(\$24.250.000\$)).

d(8,[zona(\$SIMANCAS\$),descripcion([planta(\$segundo\$),dorms(\$dos\$),situacion(\$exterior\$)]),precio(\$8.800.000\$),telefono([prefijo(\$91\$),tfno(\$4305412\$)]))).

d(9,[zona(\$MUSAS\$),descripcion([planta(\$cuarto\$),baños(\$dos\$),situacion(\$exterior\$)]),precio(\$13.900.000\$),telefono([prefijo(\$91\$),tfno(\$4305412\$)]))).

d(10,[reclamo(\$CENTRIQUISIMOS\$),descripcion([dorms(\$tres\$),situacion(\$exterior\$)]),precio(\$1.500.000\$),comentario(\$Grandísimas facilidades\$),telefono([tfno(\$5763030\$)]))).

d(11,[zona(\$PACIFICO\$),descripcion([planta(\$tercero\$),dorms(\$tres\$),situacion(\$exterior\$)]),precio(\$12.800.000\$),telefono([prefijo(\$91\$),tfno(\$4305412\$)]))).

d(12,[zona(\$RETIRO\$),descripcion([planta(\$segundo\$),dorms(\$dos\$),situacion(\$exterior\$)]),precio(\$14.500.000\$)).

d(13,[zona(\$CANILLAS\$),descripcion([planta(\$octavo\$),dorms(\$tres\$),metros(\$120\$),situacion(\$exterior\$)]),precio(\$19.800.000\$)).

d(14,[zona(\$ZONA norte\$), descripcion([dorms(\$cuatro\$), metros(\$120\$), baños(\$dos\$),situacion(\$exterior\$)],precio(\$18.500.000\$)))).

d(15,[zona(\$MANUEL BECERRA\$), descripcion([metros(\$70\$), estado(\$reforma\$)],precio(\$12.300.000\$),telefono([prefijo(\$91\$),tfno(\$4477458\$)])))).

d(16,[zona(\$ANTON MARTIN\$), descripcion([planta(\$dúplex\$), estado(\$estrenar\$)],precio(\$11.700.000\$),telefono([prefijo(\$91\$),tfno(\$4477458\$)])))).

d(17,[zona(\$BETANZOS\$),descripcion([dorms(\$tres\$),situacion(\$exterior\$)],precio(\$13.800.000\$),telefono([prefijo(\$91\$),tfno(\$5537308\$)])))).

d(18,[zona(\$JERONIMA LLORENTE\$), descripcion([dorms(\$dos\$), estado(\$rehabilitado\$), situacion(\$exterior\$)],precio(\$13.000.000\$),telefono([prefijo(\$91\$),tfno(\$5535151\$)])))).

d(19,[zona(\$ALAMEDA OSUNA\$), descripcion([metros(\$100\$), estado(\$reformular\$)], precio(\$14.000.000\$),telefono([prefijo(\$91\$),tfno(\$4460029\$)])))).

d(20,[zona(\$ARAVACA\$),descripcion([dorms(\$dos\$),estado(\$reformado\$)],precio(\$12.000.000\$),telefono([prefijo(\$91\$),tfno(\$3574029\$)])))).

d(21,[zona(\$ARAVACAS\$),descripcion([dorms(\$tres\$),baños(\$dos\$)],precio(\$16.500.000\$),telefono([prefijo(\$91\$),tfno(\$3574034\$)])))).

d(22,[zona(\$POZUELO\$),descripcion(\$estudio, salón, baño, piscina, garaje\$), precio(\$11.700.000\$), telefono([prefijo(\$91\$),tfno(\$3574029\$)])))).

d(23,[zona(\$LISTA\$),descripcion([dorms(\$dos\$)],precio(\$8.500.000\$),comentario(\$tercera planta, soleado\$),telefono([prefijo(\$91\$),tfno(\$5753388\$)])))).

d(24,[zona(\$BATAN mismo\$), descripcion([estado(\$reformular\$)], telefono([prefijo(\$91\$),tfno(\$5111699\$)])))).

d(25,[zona(\$BATAN\$),descripcion([dorms(\$tres\$),metros(\$90\$),baños(\$dos\$)])))).

d(26,[zona(\$PUERTA ANGEL\$),descripcion([dorms(\$tres\$)])))).

d(27,[zona(\$MISMO Paseo Extremadura\$), descripcion([estado(\$reformad(isirno\$)], telefono([prefijo(\$91\$), tfno(\$3547390\$)])))).

d(28,[zona(\$CAMPAMENTO\$),descripcion([dorms(\$dos\$),estado(\$estrenar\$),baños(\$dos\$)])))).

d(29,[reclamo(\$BARATISIMO\$),descripcion([estado(\$reformular\$)])))).

d(30,[reclamo(\$INTERESANTE\$),descripcion([dorms(\$tres\$)],telefono([prefijo(\$91\$),tfno(\$5779366\$)])))).

d(31,[reclamo(\$SUPERECONOMICO\$),descripcion([dorms(\$tres\$),metros(\$115\$),estado(\$reformado\$)],telefono([prefijo(\$91\$),tfno(\$5779366\$)])))).

d(32,[reclamo(\$INTERESANTE\$),descripcion([dorms(\$tres\$),metros(\$90\$),situacion(\$exterior\$)],telefono([prefijo(\$91\$),tfno(\$5983722\$)])))).

Tractatus

- d(1,[dictum([numero(\$2.025\$),texto(\$Sie ist Form und Inhalt\$),claves([c(\$Form\$),c(\$Inhalt\$))]))]).
- d(2,[dictum([numero(\$2.0251\$),texto(\$Raum, Zeit und Farbe (Färbigkeit) sind Formen der Gegenstände\$), claves([c(\$Raum\$), c(\$Zeit\$),c(\$Farbe\$), c(\$Färbigkeit\$), c(\$Formen\$), c(\$Gegenstände\$))]))]).
- d(3,[dictum([numero(\$2.026\$),texto(\$Nur wenn es Gegenstände gibt, kann es eine feste Form der Welt geben\$),claves([c(\$Gegenstände\$),c(\$feste\$),c(\$Form\$),c(\$Welt\$))]))]).
- d(4,[dictum([numero(\$2.027\$),texto(\$Das Feste, das Bestehende und der Gegenstand sind Eins\$),claves([c(\$Feste\$),c(\$Bestehende\$),c(\$Gegenstand\$))]))]).
- d(5,[dictum([numero(\$2.0271\$),texto(\$Der Gegenstand ist das Feste, Bestehende; die Konfiguration ist das Wechselnde, Unbeständige\$), claves([c(\$Gegenstand\$), c(\$Feste\$), c(\$Bestehende\$),c(\$Konfiguration\$),c(\$Unbeständige\$))]))]).
- d(6,[dictum([numero(\$2.0272\$),texto(\$Die Konfiguration der Gegenstände bildet den Sachverhalt\$), claves([c(\$Konfiguration\$),c(\$Gegenstände\$),c(\$bildet\$),c(\$Sachverhalt\$))]))]).
- d(7,[dictum([numero(\$2.03\$),texto(\$Im Sachverhalt hängen die Gegenstände ineinander, wie die Glieder einer Kette\$),claves([c(\$Sachverhalt\$),c(\$Gegenstände\$))]))]).
- d(8,[dictum([numero(\$2.031\$),texto(\$Im Sachverhalt verhalten sich die Gegenstände in bestimmter Art und Weise zueinander\$), claves([c(\$Sachverhalt\$), c(\$Gegenstände\$), c(\$Art\$), c(\$Weise\$))]))]).
- d(9,[dictum([numero(\$2.032\$),texto(\$Die Art und Weise, wie die Gegenstände im Sachverhalt zusammenhängen, ist die Struktur des Sachverhaltes\$), claves([c(\$Art\$), c(\$Weise\$), c(\$Gegenstände\$), c(\$Sachverhalt\$),c(\$Struktur\$),c(\$Sachverhaltes\$))]))]).
- d(10,[dictum([numero(\$2.033\$),texto(\$Die Form ist die Möglichkeit der Struktur\$), claves([c(\$Form\$), c(\$Möglichkeit\$), c(\$Struktur\$))]))]).
- d(11,[dictum([numero(\$2.034\$),texto(\$Die Struktur der Tatsache besteht aus den Strukturen der Sachverhalte\$),claves([c(\$Struktur\$),c(\$Tatsache\$),c(\$Strukturen\$),c(\$Sachverhalte\$))]))]).

d(12,[dictum([numero(\$2.04\$),texto(\$Die Gesamtheit der bestehenden Sachverhalte ist die Welt\$),claves([c(\$bestehenden\$),c(\$Sachverhalte\$),c(\$Welt\$))]))]).

d(13,[dictum([numero(\$2.05\$),texto(\$Die Gesamtheit der bestehenden Sachverhalte bestimmt auch, welche Sachverhalte nicht bestehen\$),claves([c(\$bestehenden\$),c(\$Sachverhalte\$),c(\$Sachverhalte\$),c(\$bestehen\$))]))]).

d(14,[dictum([numero(\$2.06\$),texto(\$Das Bestehen und Nichtbestehen von Sachverhalten ist die Wirklichkeit. (Das Bestehen von Sachverhalten nennen wir auch eine positive, das Nichtbestehen eine negative Tatsache.)\$),claves([c(\$Bestehen\$),c(\$Nichtbestehen\$),c(\$Sachverhalten\$),c(\$Wirklichkeit\$),c(\$Bestehen\$),c(\$Sachverhalten\$),c(\$Nichtbestehen\$),c(\$Tatsache\$))]))]).

d(15,[dictum([numero(\$2.061\$),texto(\$Die Sachverhalte sind von einander unabhängig\$),claves([c(\$Sachverhalte\$))]))]).

d(16,[dictum([numero(\$2.062\$),texto(\$Aus dem Bestehen oder Nichtbestehen eines Sachverhaltes kann nicht auf das Bestehen oder Nichtbestehen eines anderen geschlossen werden\$),claves([c(\$Bestehen\$),c(\$Nichtbestehen\$),c(\$Sachverhaltes\$),c(\$Bestehen\$),c(\$Nichtbestehen\$))]))]).

d(17,[dictum([numero(\$2.063\$),texto(\$Die gesamte Wirklichkeit ist die Welt\$),claves([c(\$Wirklichkeit\$),c(\$Welt\$))]))]).

d(18,[dictum([numero(\$2.1\$),texto(\$Wir machen uns Bilder der Tatsachen\$),claves([c(\$Bilder\$),c(\$Tatsachen\$))]))]).

d(19,[dictum([numero(\$2.11\$),texto(\$Das Bild stellt die Sachlage im logischen Raume, das Bestehen und Nichtbestehen von Sachverhalten vor\$),claves([c(\$Bild\$),c(\$Sachlage\$),c(\$logischen\$),c(\$Raume\$),c(\$Bestehen\$),c(\$Nichtbestehen\$),c(\$Sachverhalten\$))]))]).

d(20,[dictum([numero(\$2.12\$),texto(\$Das Bild ist ein Modell der Wirklichkeit\$),claves([c(\$Bild\$),c(\$Wirklichkeit\$))]))]).

d(21,[dictum([numero(\$2.13\$),texto(\$Den Gegenständen entsprechen im Bilde die Elemente des Bildes\$),claves([c(\$Gegenständen\$),c(\$Bilde\$),c(\$Elemente\$),c(\$Bildes\$))]))]).

d(22,[dictum([numero(\$2.131\$),texto(\$Die Elemente des Bildes vertreten im Bild die Gegenstände\$),claves([c(\$Elemente\$),c(\$Bildes\$),c(\$Bild\$),c(\$Gegenstände\$))]))]).

d(23,[dictum([numero(\$2.14\$),texto(\$Das Bild besteht darin, dass sich seine Elemente in bestimmter Art und Weise zu einander verhalten\$),claves([c(\$Bild\$),c(\$Elemente\$),c(\$Art\$),c(\$Weise\$))]))]).

- d(24,[dictum([numero(\$2.141\$),texto(\$Das Bild ist eine Tatsache\$), claves([c(\$Bild\$), c(\$Tatsache\$)])))]).
- d(25,[dictum([numero(\$2.15\$),texto(\$Dass sich die Elemente des Bildes in bestimmter Art und Weise zu einander verhalten, stellt vor, dass sich die Sachen so zu einander verhalten. Dieser Zusammenhang der Elemente des Bildes hei3se seine Struktur und ihre Moglichkeit seine Form der
Abbildung\$), claves([c(\$Elemente\$), c(\$Bildes\$), c(\$Art\$), c(\$Weise\$), c(\$Sachen\$), c(\$Elemente\$), c(\$Bildes\$), c(\$Struktur\$), c(\$Moglichkeit\$), c(\$Form\$), c(\$Abbildung\$)])))]).
- d(26,[dictum([numero(\$2.151\$),texto(\$Die Form der Abbildung ist die Moglichkeit, dass sich die Dinge so zu einander verhalten, wie die Elemente des Bildes\$), claves([c(\$Form\$), c(\$Abbildung\$), c(\$Moglichkeit\$), c(\$Elemente\$), c(\$Bildes\$)])))]).
- d(27,[dictum([numero(\$2.1511\$),texto(\$Das Bild ist so mit der Wirklichkeit verknupft; es reicht bis zu ihr\$), claves([c(\$Bild\$), c(\$Wirklichkeit\$)])))]).
- d(28,[dictum([numero(\$2.1512\$),texto(\$Es ist wie ein Masstab an die Wirklichkeit angelegt\$), claves([c(\$Wirklichkeit\$)])))]).
- d(29,[dictum([numero(\$2.15121\$),texto(\$Nur die äussersten Punkte der Teilstriche berühren den zu messenden Gegenstand\$), claves([c(\$Gegenstand\$)])))]).
- d(30,[dictum([numero(\$2.1513\$),texto(\$Nach dieser Auffassung gehort also zum Bilde auch noch die abbildende Beziehung, die es zum Bild macht\$), claves([c(\$Bilde\$), c(\$abbildende\$), c(\$Beziehung\$), c(\$Bild\$)])))]).
- d(31,[dictum([numero(\$2.1514\$),texto(\$Die abbildende Beziehung besteht aus den Zuordnungen der Elemente des Bildes und der Sachen\$), claves([c(\$abbildende\$), c(\$Beziehung\$), c(\$Elemente\$), c(\$Bildes\$), c(\$Sachen\$)])))]).
- d(32,[dictum([numero(\$2.1515\$),texto(\$Diese Zuordnungen sind gleichsam die Fühler der Bildelemente, mit denen das Bild die Wirklichkeit berührt\$), claves([c(\$Bildelemente\$), c(\$Bild\$), c(\$Wirklichkeit\$)])))]).
- d(33,[dictum([numero(\$2.16\$),texto(\$Die Tatsache muss, um Bild zu sein, etwas mit dem Abgebildeten gemeinsam haben\$), claves([c(\$Tatsache\$), c(\$Bild\$), c(\$Abgebildeten\$)])))]).
- d(34,[dictum([numero(\$2.161\$),texto(\$In Bild und Abgebildetem muss etwas identisch sein, damit das eine überhaupt ein Bild des anderen sein kann\$), claves([c(\$Bild\$), c(\$Abgebildetem\$), c(\$Bild\$)])))]).

d(35,[dictum([numero(\$2.17\$),texto(\$Was das Bild mit der Wirklichkeit gemein haben muss, um sie auf seine Art und Weise -richtig oder falsch- abbilden zu können, ist seine Form der Abbildung\$),claves([c(\$Bild\$),c(\$Wirklichkeit\$),c(\$Art\$),c(\$Weise\$),c(\$falsch\$),c(\$abbilden\$),c(\$Form\$),c(\$Abbildung\$)))]])).

d(36,[dictum([numero(\$2.171\$),texto(\$Das Bild kann jede Wirklichkeit abbilden, deren Form es hat. Das räumliche Bild alles Räumliche, das farbige alles Farbige, etc\$), claves([c(\$Bild\$),c(\$Wirklichkeit\$),c(\$abbilden\$),c(\$räumliche\$),c(\$Bild\$),c(\$Räumliche\$),c(\$farbige\$),c(\$Farbige\$)))]])).

d(37,[dictum([numero(\$2.172\$),texto(\$Seine Form der Abbildung aber kann das Bild nicht abbilden; es weist sie auf\$),claves([c(\$Form\$),c(\$Abbildung\$),c(\$Bild\$),c(\$abbilden\$)))]])).

d(38,[dictum([numero(\$2.173\$),texto(\$Das Bild stellt sein Objekt von ausserhalb dar (sein Standpunkt ist seine Form der Darstellung), darum stellt das Bild sein Objekt richtig oder falsch dar\$),claves([c(\$Bild\$),c(\$Objekt\$),c(\$Standpunkt\$),c(\$Form\$),c(\$Bild\$),c(\$Objekt\$),c(\$falsch\$)))]])).

d(39,[dictum([numero(\$2.174\$),texto(\$Das Bild kann sich aber nicht ausserhalb seiner Form der Darstellung stellen\$),claves([c(\$Bild\$),c(\$Form\$)))]])).

d(40,[dictum([numero(\$2.18\$),texto(\$Was jedes Bild, welcher Form immer, mit der Wirklichkeit gemein haben muss, um sie überhaupt-richtig oder falsch-abbilden zu können, ist die logische Form, das ist, die Form der Wirklichkeit\$), claves([c(\$Bild\$), c(\$Form\$), c(\$Wirk\$), c(\$falsch\$), c(\$abbilden\$),c(\$logische\$),c(\$Form\$),c(\$Form\$),c(\$Wirklichkeit\$)))]])).

d(41,[dictum([numero(\$2.181\$),texto(\$Ist die Form der Abbildung die logische Form, so heisst das Bild das logische Bild\$), claves([c(\$Form\$), c(\$Abbildung\$), c(\$logische\$), c(\$Form\$), c(\$Bild\$),c(\$logische\$),c(\$Bild\$)))]])).

d(42,[dictum([numero(\$2.182\$),texto(\$Jedes Bild ist auch ein logisches. (Dagegen ist z.B. nicht jedes Bild ein räumliches.)\$),claves([c(\$Bild\$),c(\$logisches\$),c(\$Bild\$),c(\$räumliches\$)))]])).

d(43,[dictum([numero(\$2.19\$),texto(\$Das logische Bild kann die Welt abbilden\$), claves([c(\$logische\$), c(\$Bild\$), c(\$Welt\$),c(\$abbilden\$)))]])).

d(44,[dictum([numero(\$2.2\$),texto(\$Das Bild hat mit dem Abgebildeten die logische Form der Abbildung gemein\$), claves([c(\$Bild\$), c(\$Abgebildeten\$), c(\$logische\$), c(\$Form\$), c(\$Abbildung\$)))]])).

d(45,[dictum([numero(\$2.201\$),texto(\$Das Bild bildet die Wirklichkeit ab, indem es eine Möglichkeit des Bestehens und Nichtbestehens von Sachverhalten darstellt\$), claves([c(\$Bild\$),

c(\$bildet\$), c(\$Wirklichkeit\$), c(\$Möglichkeit\$), c(\$Bestehens\$), c(\$Nichtbestehens\$),
c(\$Sachverhalten\$)))))).

d(46,[dictum([numero(\$2.202\$),texto(\$Das Bild stellt eine mögliche Sachlage im logischen
Raume dar\$),claves([c(\$Bild\$),c(\$mögliche\$),c(\$Sachlage\$),c(\$logischen\$),c(\$Raume\$))])))).

d(47,[dictum([numero(\$2.203\$),texto(\$Das Bild enthält die Möglichkeit der Sachlage, die es
darstellt\$),claves([c(\$Bild\$),c(\$Möglichkeit\$),c(\$Sachlage\$))])))).

d(48,[dictum([numero(\$2.21\$),texto(\$Das Bild stimmt mit der Wirklichkeit überein oder nicht; es
ist richtig oder unrichtig, wahr oder falsch\$),claves([c(\$Bild\$),c(\$Wirklichkeit\$),c(\$wahr\$),
c(\$falsch\$))])))).

d(49,[dictum([numero(\$2.22\$),texto(\$Das Bild stellt dar, was es darstellt, unabhängig von seiner
Wahr- oder Falschheit, durch die Form der Abbildung\$),claves([c(\$Bild\$),c(\$Wahr\$),
c(\$Falschheit\$),c(\$Form\$),c(\$Abbildung\$))])))).

Revistas

- d(1,[titulo(\$ABACO\$),issn(\$0213-6252\$),editor(\$CENTRO DE ESTUDIOS ECONOMICOS Y SOCIALES\$),donde([direccion(\$SAN BERNARDO, 40, 5°, D.\$), c_postal(\$33201\$), ciudad(\$GIJON (ASTURIAS)\$)))).
- d(2,[titulo(\$ADICCIONES. REVISTA DE SOCIDROGALCOHOL\$),issn(\$0214-4840\$), editor(\$SOCIODROGALCOHOL\$),donde([direccion(\$RAMBLA, 15, 2° Y 3° 07303 PALMA DE MALLORCA\$)))).
- d(3,[titulo(\$ADUANAS. REVISTA DE COMERCIO INTERNACIONAL Y ESTUDIOS FISCALES\$), issn(\$0400-5732\$),editor(\$CONS. DE ESTUDIOS ADUANEROS Y COMERCIO EXTERIOR\$), donde([direccion(\$EVARISTA SAN MIGUEL, 10\$), c_postal(\$28008\$), ciudad(\$MADRID\$)))).
- d(4,[titulo(\$AFFERS INTERNACIONALS\$),issn(\$0212-1786\$),editor(\$CENTRE D'INFORMACIO Y DOCUMENTACIO INTERNACIONALS\$),donde([direccion(\$ELISABETS, 12\$), c_postal(\$08001\$), ciudad(\$BARCELONA\$)))).
- d(5,[titulo(\$AGRICULTURA Y SOCIEDAD\$),issn(\$0211-8394\$),editor(\$MINIST. DE AGRICULTURA. SERV. DE PUBLICACIONES\$), donde([direccion(\$PASEO INFANTA ISABEL, 1\$),c_postal(\$28007\$),ciudad(\$MADRID\$)))).
- d(6,[titulo(\$ALFOZ\$),issn(\$0401-3239\$),editor(\$CENTRO DE INVESTIGACION Y DOCUMENTACION URBANA Y RURAL (CIDUR\$),donde([direccion(\$INFANTAS, 13\$), c_postal(\$28004\$), ciudad(\$MADRID\$)))).
- d(7,[titulo(\$ALTA DIRECCION\$),issn(\$0002-6549\$),editor(\$ALTA DIRECCION S.A.\$),donde([direccion(\$AVDA. INFANTA CARLOTA, 6\$), c_postal(\$08029\$), ciudad(\$BARCELONA\$)))).
- d(8,[titulo(\$AMERICA LATINA, HOY. REVISTA DE CIENCIAS SOCIALES\$),issn(\$1130-2887\$), editor(\$UNIV. COMPLUTENSE DE MADRID. SEMINARIO ESTUDIOS POLITICOS SOBRE AMERICA LATINAS\$), donde([direccion(\$FAC. CC. POLITICAS. CAMPUS DE SOMOSAGUAS\$), c_postal(\$28023\$),ciudad(\$MADRID\$)))).
- d(9,[titulo(\$ANALES DE ESTUDIOS ECONOMICOS Y EMPRESARIALES\$),issn(\$0213-7569\$),editor(\$UNIV. DE VALLADOLID. SECR. DE PUBLICACIONES\$), donde([direccion(\$RAMON Y CAJAL, 7\$),c_postal(\$47005\$),ciudad(\$VALLADOLID\$)))).

- d(10,[titulo(\$ANALES DE LA REAL ACADEMIA DE CIENCIAS MORALES Y POLITICAS\$),
issn(\$0210-4121\$),editor(\$REAL ACAD. DE CIENCIAS MORALES Y POLITICAS\$),
donde([direccion(\$PLAZA DE LA VILLA, 2\$), c_postal(\$28005\$), ciudad(\$MADRID\$))]).
- d(11,[titulo(\$ANALISI\$),issn(\$0211-2175\$),editor(\$UNIV. AUTONOMA DE BARCELONA. SERV.
DE PUBLICACIONES\$),donde([direccion(\$CAMPUS UNIVERSITARIO DE BELLATERRA\$),
c_postal(\$08193\$),ciudad(\$BELLATERRA (BARCELONA)\$))]).
- d(12,[titulo(\$ANTHROPOS. SUPLEMENTOS\$),issn(\$1130-2089\$), editor(\$ANTHROPOS\$),
donde([direccion(\$VIA AUGUSTA, 64\$), c_postal(\$08006\$), ciudad(\$BARCELONA\$))]).
- d(13,[titulo(\$ARCHIPIELAGOS\$),issn(\$0214-2686\$),editor(\$EDITORIAL PAMELA\$),
donde([direccion(\$PLAZUELA DEL CONSEJO, 3, 4\$), c_postal(\$31001\$),
ciudad(\$PAMPLONA\$))]).
- d(14,[titulo(\$AREAS. REVISTA DE CIENCIAS SOCIALES\$),issn(\$0211-6707\$),editor(\$EDITORA
REGIONAL DE MURCIA\$),donde([direccion(\$ISAAC ALBENIZ\$), c_postal(\$30009\$),
ciudad(\$MURCIA\$))]).
- d(15,[titulo(\$AYER\$),ign(issn),editor(\$MARCIAL PONSS\$),donde([direccion(\$PLAZA DEL CONDE
DEL VALLE DE SUCHIL, 8\$),c_postal(\$29015\$),ciudad(\$MADRID\$))]).
- d(16,[titulo(\$BANCA ESPAÑOLA\$),issn(\$0210-1688\$), editor(\$REMARCA\$),
donde([direccion(\$CONDE DE PEÑALVER, 52\$), c_postal(\$28006\$),ciudad(\$MADRID\$))]).
- d(17,[titulo(\$BOLETIN DE ECONOMIA DE NAVARRA\$),issn(\$1131-5490\$),editor(\$GOBIERNO
DE NAVARRA. SERV. DE ECONOMIAS\$),donde([direccion(\$ARSIETA, 8, 6º\$),
c_postal(\$31002\$), ciudad(\$PAMPLONA (NAVARRA)\$))]).
- d(18,[titulo(\$BOLETIN DE ESTUDIOS ECONOMICOS\$),issn(\$0006-6249\$),editor(\$UNIV. DE
DEUSTO\$),donde([direccion(\$HERMANOS AGUIRRE, 2. APTDO\$), c_postal(\$20044\$),
ciudad(\$48080 BILBAO\$))]).
- d(19,[titulo(\$BOLETIN DE LA INSTITUCION LIBRE DE ENSEÑANZA\$),issn(\$0214-1302\$),
editor(\$FUNDACION FRANCISCO GINER DE LOS RIOS\$),donde([direccion(\$PASEO DEL
GENERAL MARTINEZ CAMPOS, 14\$),c_postal(\$28010\$),ciudad(\$MADRID\$))]).
- d(20,[titulo(\$BOLETIN DE PSICOLOGIA\$),issn(\$0212-8179\$),editor(\$UNIV. DE VALENCIA. FAC.
DE FILOSOFIA Y CIENCIAS DE LA EDUCACION\$),donde([direccion(\$AVDA. BLASCO IBAÑEZ,
21\$),c_postal(\$46010\$),ciudad(\$VALENCIA\$))]).

d(21,[titulo(\$BOLETIN DE RELACIONES ECONOMICAS INTERNACIONALES\$),
ign(issn),editor(\$MINIST. DE ASUNTOS EXTERIORES. SECR. GENERAL TEC\$),
donde([direccion(\$PLAZA DE LA MARINA ESPAÑOLA\$),
c_postal(\$28013\$),ciudad(\$MADRID\$))]).

d(22,[titulo(\$BOLETIN ECONOMICO DE ANDALUCIA\$),issn(\$0212-6621\$),editor(\$JUNTA DE
ANDALUCIA. CONS. DE ECONOMIA, INDUSTRIA Y ENERGIA\$),donde([direccion(\$AVDA
REPUBLICA ARGENTINA, 81\$),c_postal(\$41071\$),ciudad(\$SEVILLA\$))]).

d(23,[titulo(\$BOLETIN ECONOMICO DE ICE\$),issn(\$0213-3768\$),editor(\$MINIST. DE
COMERCIO Y TURISMO\$),donde([direccion(\$PASEO DE LA CASTELLANA, 152, PLANTA
16\$),c_postal(\$28016\$),ciudad(\$MADRID\$))]).

Listas

Abreviaturas

área DD	Área débilmente definida
área ED	Área estrictamente definida
área R	Área de sólo reconocimiento
área RR	Área de reconocimiento y recuperación
BNF	Backus-Naur Form
DCG	Definite Clause Grammar
EBNF	Extended Backus-Naur Form
G_{Lex}	Gramática del lexema
G_{Lendex}	Gramática de LENDEX
G_{Input}	Gramática del Input
G_{Output}	Gramática del Output
GDU	Gramática definida por el usuario
ISBD	International Standard Book Description
SGML	Standard Generalized Markup Language

Figuras

FIGURA 1: ESTRUCTURA GENERAL DE UN DOCUMENTO	18
FIGURA 2: UN DOCUMENTO ISBD	22
FIGURA 3: REPRESENTACIÓN EN ÁRBOL DEL ANÁLISIS DE UN DOCUMENTO EN TÉRMINOS DE SUS ÁREAS R Y RR.....	23
FIGURA 4: ESQUEMA GENERAL DE S-LENDEX	66
FIGURA 5: EJEMPLO DE DOCUMENTO INPUT.....	74
FIGURA 6: EJEMPLO DE DOCUMENTO OUTPUT	79
FIGURA 7: REPRESENTACIÓN EN FORMA DE ÁRBOL DEL ANÁLISIS LENDEX DE UN REGISTRO BIBLIOGRÁFICO.....	80
FIGURA 8: SIGNIFICADO DEL SALTO DADO POR HAY_D/5 EN ANALISIS/4	161
FIGURA 9: CUANTIFICADORES LENDEX DE RECONOCIMIENTO Y RECUPERACIÓN	173
FIGURA 10: CUANTIFICADORES LENDEX DE SÓLO RECONOCIMIENTO	194

Tablas

TABLA 1: PRIMER ANÁLISIS DE UN REGISTRO ISBD	24
TABLA 2: SEGUNDO ANÁLISIS DE UN REGISTRO ISBD.....	25
TABLA 3: ANÁLISIS ALTERNATIVO DEL ÁREA ISBN	26
TABLA 4: TIPOS DE LEXEMA: EJEMPLOS.....	28
TABLA 5: COMPILACIÓN DE UNA GDU	225
TABLA 6: TRADUCCIÓN DE LAS SECUENCIAS Y LOS LEXEMAS DESCRITOS	228

Procedimientos

PROBLEMA_INTEGRIDAD/1	85
ASERTA_GRAMATICA/2.....	106
ASERTA_LVACIA/1	107
ASERTA_FINES/1	108
ASERTA_UN_FIN/4.....	109
ASERTA_RECURSION/2.....	112
SATISFACE_RASGO/2	113
META_LENDEX/6.....	123
LEXEMA_DESCRITO/2.....	130
LX_TERMINAL /2.....	131
TERMINO_CNTF/4.....	135
CUANTIFICADOR/2.....	135
META_LENDEX_CNTF/7	137
META_LENDEX_DD/5	142
DCG_TERM/4	143
SUSTITUCION_VLEX/3.....	145
EXT_CI_AUX /3	146
SATISFACE_CONDICION/5	146
SEGMENTO_COMPLEJO/3	148
SEGMENTO_ANTERIOR/3.....	149
RESTA/3.....	150
ANALISIS/4.....	153
HAY/6.....	163
HAY_MULTIPLE/4.....	168
META_DIS / 5	174
@/3	175
(+)/ 5	178
(**)/ 5.....	179
(&&)/ 5.....	180
DESDE / 6.....	182
HASTA/ 6.....	183
ENTRE / 7.....	184
EX / 6	185

Listas y bibliografía

META_DIS / 4	192
(++) / 4	195
(**) / 4	195
(&&) / 4	196
DESDE / 5	197
HASTA / 5	197
ENTRE / 6	198
EX / 5 199	
COMPILACION_PRODS/0	215
COMPILACION_LX_DCG/2	217
COMPILACION_CNTF/5	220

Bibliografía

- [Abramson-Dahl,89]. Abramson, H. and Dahl, V. (1989). *Logic Grammars*. New York: Springer Verlag.
- [Abramson-Rogers,89]. Abramson, H. and Rogers, M.H., eds. (1989). *Meta-Programming in Logic Programming*. Cambridge, MA: The MIT Press
- [Acquilex,89]. Acquilex. "Technical Annex". ESPRIT BRA-3030
- [Adarraga-Zaccagnini,94]. Adarraga, A., Zaccagnini, J.L. (1994). *Psicología e Inteligencia Artificial*. Madrid: Editorial Trotta
- [Akindele-Belaïd,94]. Akindele, O. and Belaïd, A. (1994). En: [Colloque,94].
- [Amble,87]. Amble, T. (1987). *Logic Programming and Knowledge Engineering*. Wokingham: Addison-Wesley Publishing Company
- [André et al.,89]. André, Furuta and Quint. (1989). *Structured Documents*. Cambridge: Cambridge University Press.
- [Arity,92]. Arity Corporation. *The Arity/Prolog Compiler and Interpreter* (1992). Concord, MA: Damonmill Square
- [Barklund,89]. Barklund, J. (1989). What is a Meta-variable in Prolog? En [Abramson-Rogers,89; pp.384-398]
- [Bastos,94]. Bastos, S. (1994). La recuperación automática de información jurídica: metodología de análisis lógico-sintáctico aplicado a la lengua portuguesa. Tesis doctoral. Madrid, Septiembre-94: UCM, Fac. CC. Información.
- [Bratko,86]. Bratko, I. (1986). *Prolog Programming for Artificial Intelligence*. Wokingham: Addison-Wesley

[Bruffaerts-Henin,89]. Bruffaerts, A. and Henin, E. (1989). Negation as Failure: Proofs, Inference Rules and Meta-interpreters. En [Abramson-Rogers,89; pp.169-190]

[Cavalieri et al.,89]. Cavalieri, Lamma, Mello and Natali. (1989). Meta-Programming in Prolog Through Direct Introspection: A comparison with Meta-Interpretation Techniques. En [Abramson-Rogers,89; pp.400-433]

[Chenevoy-Belaïd,94]. Chenevoy, Y. and Belaïd, A. (1994). Une approche objet de la reconnaissance structurelle de documents. En [Colloque,94].

[Clark-Tärnlund,82]. Clark, K.L. and Tärnlund, S.-A., eds. (1982). *Logic Programming*. London: Academic Press, Inc.

[Clocksin-Mellish,84]. Clocksin, W.F. and Mellish, C.S. (1984). *Programming in Prolog*. 2nd edn. Berlin: Springer Verlag

[Cochard-Ingold,84]. Cochard, J.L, Ingold, R. (1984). Le projet ESCROC: Environnement de saisie et de correction de la reconnaissance optique de caractères. [Colloque,94].

[Colloque,94]. 3eme Colloque National sur L'Ecrit et le Document. Roan: 6-8 Juillet 1994.

[Courant-Robin,85]. Courant, M. and Robin, S. (1985). Classified Advertisement analysis in the context of an Expert System in Ad Matching. En [Dahl-St.Dizier,85, pp.33-47].

[Covington,94]. Covington, M. (1994). *Natural Language Processing for Prolog Programmers*. Englewood Cliffs, NJ: Prentice Hall.

[Dahl,94]. Dahl, V. (1994). Natural Language Processing and Logic Programming. En *The Journal of Logic Programming*: 19, 20; pp. 681-714

[Dahl-Massicote,89]. Dahl, V. and Massicotte, P. (1989). Processing Techniques for Discontinuous Grammars. En [Abramson-Rogers,89; pp.141-156]

[Dahl-St.Dizier,85]. Dahl, V. and Saint-Dizier, P. (1985). *Natural Language Understanding and Logic Programming*. Amsterdam: North-Holland

[Deville,90]. Deville, Y (1990). *Logic Programming. Systematic Program Development*. Wokingham: Addison-Wesley

[Discos,88]. Biblioteca Nacional. (1988). *Catálogo de discos de 78 rpm. en la Biblioteca Nacional*. Madrid: Ministerio de Cultura, D.G. del libro y bibliotecas.

[Dodd et al.,91]. Dodd, Owens and Torrance. (1991). *Logic Programming. Expanding the Horizons*. Oxford: Intellect

[DRAE,70]. Diccionario de la Real Academia Española. (1970). Madrid: Espasa-Calpe, S.A.

[García-Lucas,87]. García Gutierrez, A., Lucas Fernández, R. (1987). *Documentación automatizada en los medios informativos*. Madrid: Paraninfo.

[Gal et al.,91]. Gal, Lapalme, Saint-Dizier and Somers (1991). *Prolog for Natural Language Processing*. Chichester: John Wiley & Sons.

[Gazdar-Mellish,89]. Gazdar, G. and Mellish, C. (1989). *Natural Language Processing in Prolog: an Introduction to Computational Linguistics*. Wokingham, England: Addison-Wesley Publishing Company

[Gibbins,88]. Gibbins, P. (1988). *Logic with Prolog*. Oxford: Clarendon Press.

[Hill-Lloyd,89]. Hill, P.M. and Lloyd, J.W. (1989). Analysis of Meta-Programs. En [Abramson-Rogers,89; pp.23-51]

[Indice,91]. ISOC. (1991). *Índice español de Ciencias Sociales*, Volumen XII. Madrid: Consejo Superior de Investigaciones Científicas

[Ingold,89]. Ingold, R. (1989). Text Structure in Optical Reading. En [André et al.,89].

[King,83]. King, M. ed. (1983). *Parsing Natural Language*. London: Academic Press.

[Krulee,91]. Krulee, G.K. (1991). *Computer Processing of Natural Language*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

[Lloyd,87]. Lloyd, J.W. (1987). *Foundations of Logic Programming*. Berlin: Springer Verlag

[McCord,87]. McCord, M. (1987). Natural Language Processing in Prolog. En [Walker et al.,87; pp.291-402].

[NORDInfo,92]. NORDInfo retroconversion seminar. The Royal Library, Copenhagen, 10-11 december 1992

[Obermaier,89]. Obermaier, K. (1989). *Natural Language Processing Technologies in AI*. Chichester: Ellis Horwood Limited.

[Pereira-Shieber,87]. Pereira, F. and Shieber, S. (1987). *Prolog and Natural-Language Analysis*. Menlo Park, CA: CSLI

[Sarabia,92]. Sarabia, J. (1992). Técnicas de metaprogramación en gramáticas lógicas. En VII Congreso de lenguajes naturales y lenguajes formales. Vic-Barcelona, septiembre 1991.

[Sarabia,93]. Sarabia, J. (1993). Procedimientos metagramaticales y léxicos. Informe técnico del laboratorio de Inteligencia Artificial. 1/93

[Sarabia,94]. Sarabia, J. (1994). ¿Cómo hacemos para entender lo que leemos?. *En Information World en Español*, nº 28, octubre 1994.

[Sterling-Shapiro,86]. Sterling, L. and Shapiro, E. (1986). *The Art of Prolog*. Cambridge, MA: The MIT Press.

[Tazi-Virbel,85]. Tazi, S. and Virbel, J. (1985). Formal Representation of Textual Structures for an Intelligent Text-Editing System. En [Dahl-St.Dizier,85, pp.191-205].

[VL,92]. Verba Logica. (1992). Project for Retrospective Conversion using AI techniques. Informe técnico del laboratorio de IA, 1/92

[VL,94]. Verba Logica. (1994). III Informe técnico del laboratorio de Inteligencia Artificial.

[Villancicos,90]. Biblioteca Nacional. (1990). *Catálogo de villancicos y oratorios en la Biblioteca Nacional. Siglos XVIII-XIX*. Madrid: Ministerio de Cultura, D.G. del libro y bibliotecas.

[Walker et al.,87]. Walker, McCord, Sowa and Wilson. (1987). *Knowledge Systems and Prolog. A Logical Approach to Expert Systems and Natural Language Processing*. Reading, MA: Addison-Wesley Publishing Company, Inc.

[Wittgenstein,69]. Wittgenstein, L. (1969). *Tractatus logico-philosophicus*. Frankfurt am Main: Suhrkamp Verlag

[Yalçinalp-Sterling,89]. Yalçinalp, L.Ü. and Sterling, L. An Integrated Interpreter for Explaining Prolog's Successes and Failures. En [Abramson-Rogers,89; pp. 191-203].