

Análisis sintáctico

Procesamiento del Lenguaje Natural

José María Gómez Hidalgo

<http://www.esp.uem.es/jmgomez/>

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

Índice

1. Representación del conocimiento sintáctico *
2. Gramáticas de estructura de frase independientes del contexto *
3. **Poder expresivo de los formalismos gramaticales**
4. **Algoritmos de análisis**
5. Gramáticas de cláusulas definidas (DCGs) *
6. Gramáticas basadas en restricciones
7. El papel del léxico

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

Análisis Sintáctico

1. Representación del conocimiento sintáctico

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

1. Representación del conocimiento sintáctico

- **Naturaleza del conocimiento sintáctico**
 - Estructura de las expresiones de un lenguaje
 - Si es un lenguaje natural, estructura de las oraciones, es decir, cómo se relacionan las palabras de una oración entre sí
 - Cómo se organizan las palabras en grupos o sintagmas
 - Qué palabras o grupos modifican a otras palabras o grupos
 - Qué palabras o grupos son los más importantes de la oración

1. Representación del conocimiento sintáctico

- Ejemplo
 - Oración: "Juan compra una novela en la librería"
 - "En la librería" es un sintagma preposicional que modifica al núcleo de la oración, que es el verbo "compra"
 - Agrupamiento (sintagma preposicional)
 - Modificación (modifica al verbo)
 - Importancia (el verbo es el núcleo)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

1. Representación del conocimiento sintáctico

- Tipos de conocimiento sintáctico
 - Formalismo gramatical
 - Lenguaje de representación de información sintáctica
 - Gramática - G
 - Representación de la estructura de un lenguaje, en un formalismo gramatical
 - Árbol de análisis
 - Representación de la estructura de una oración concreta de un lenguaje concreto (caracterizado por una gramática expresada en un formalismo)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

1. Representación del conocimiento sintáctico

- Análisis sintáctico y traducción
 - Dada la gramática G que caracteriza al lenguaje L , algunas expresiones pertenecen a L y otras no
 - **Reconocimiento** = decidir si una expresión pertenece al lenguaje caracterizado por G
 - **Análisis (*parsing*)** = decidir si una expresión pertenece al lenguaje representado por G , y en caso afirmativo, asignar a la expresión una estructura acorde a G
 - Análisis = traducción de expresiones en lenguaje natural a estructuras sintácticas

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

1. Representación del conocimiento sintáctico

- Características deseables en los formalismos gramaticales
 - Naturalidad lingüística
 - Conceptos lingüísticos fácilmente expresables
 - Poder expresivo
 - Capacidad de representar lenguajes (matemáticamente complejos)
 - Efectividad computacional
 - Gramática leída por hombre y máquina
 - Diseño de lenguajes formales

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

1. Representación del conocimiento sintáctico

- Características deseables en las gramáticas
 - General
 - Debe cubrir el mayor subconjunto posible del idioma
 - Selectiva
 - Debe minimizar el número de problemas que encuentra en la oraciones no válidas
 - Comprensible
 - Debe ser lo más sencilla posible

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

Análisis Sintáctico

2. Gramáticas de estructura de frase independientes del contexto

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

2. Gramáticas de estructura de frase independientes del contexto

- Gramática de estructura de frase independiente del contexto, CF-PSG (Context-free phrase structure grammar)
- Una tupla $G = (T, NT, S, P)$
 - T es un conjunto finito de símbolos terminales o léxicos
 - NT es un conjunto finito disjunto de T de elementos no terminales o categorías
 - P es un conjunto finito de producciones o reglas de reescritura de la forma $A \rightarrow a$ donde A es una cat. y a es una secuencia de categorías y símbolos léxicos
 - S es un símbolo de NT llamado axioma o símbolo inicial, y no aparece en lado derecho de ninguna regla

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

2. Gramáticas de estructura de frase independientes del contexto

- Ejemplo (G1)
 - $T = \{el, la, perro, salchicha, come\}$
 - $NT = \{O, SN, SV, V, Art, N\}$
 - $S = O$
 - $P = \{O \rightarrow SN\ SV, SN \rightarrow Art\ N, SV \rightarrow V\ SN, Art \rightarrow el, Art \rightarrow la, V \rightarrow come, N \rightarrow perro, N \rightarrow salchicha\}$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

2. Gramáticas de estructura de frase independientes del contexto

- Una CF-PSG G sirve para
 - Definir un conjunto de frases (subconjuntos de T^*) aceptables en un lenguaje o gramaticales = el lenguaje generado (representado) por la gramática G , $L(G)$
 - Asignar una $o +$ estructuras sintácticas a frases gramaticales de la gramática, frases de $L(G)$
 - Las estructuras asignadas son árboles o "grafos acíclicos dirigidos" (Directed acyclic graphs, DAGs)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

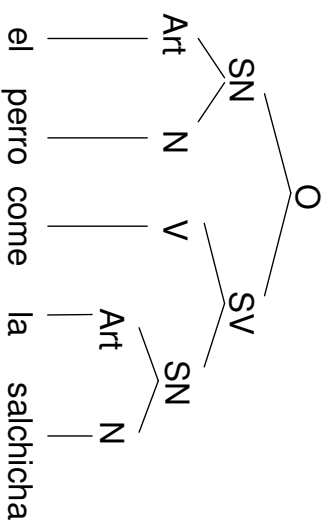
2. Gramáticas de estructura de frase independientes del contexto

- Ejemplo
 - La frase $F =$ "el perro come la salchicha" es aceptada por G_1 , es decir, $F \in L(G_1)$
 - A la frase F se le puede asignar el árbol siguiente
 - Representado como lista:
 - (O (SN (Art el) (N perro))) (SV (V come) (SN (Art la) (N salchicha))))

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

2. Gramáticas de estructura de frase independientes del contexto

- Ejemplo
 - Árbol representado gráficamente



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

2. Gramáticas de estructura de frase independientes del contexto

- Un árbol de análisis representa dos tipos de relaciones entre símbolos
 - Dominio
 - Una categoría domina a otra o a un símbolo léxico cuando es su nodo padre en el árbol de análisis
 - Ej. Art domina a "el", y SV domina a V
 - Precedencia
 - Un símbolo precede a otro cuando es un hermano suyo a la izquierda en el árbol de análisis
 - Ej. "el" precede a "perro", y V precede a SN

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

2. Gramáticas de estructura de frase independientes del contexto

- Sobregeneración
 - La gramática puede generar frases u oraciones no deseables
 - G1 genera "la salchicha come el perro"
 - Puede no ser problema si se pretende reconocer frases, pero lo es si se pretende producir frases

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

2. Gramáticas de estructura de frase independientes del contexto

- Infrageneración
 - La gramática puede no ser capaz de generar oraciones deseables
 - G1 no genera "el perro comió la salchicha"
 - Puede no ser problema para producir frases, pero lo es para reconocer frases

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

2. Gramáticas de estructura de frase independientes del contexto

- Asignación de estructuras correctas
 - La gramática puede no asignar las estructuras deseadas por el desarrollador de la misma
 - Si se pretende asignar la estructura de función (p. ej., SN Objeto o SN Sujeto), G1 no es capaz de hacerlo
 - Las estructuras dependen de la lingüística, donde no hay acuerdos definitivos

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

Análisis Sintáctico

3. Poder expresivo de los formalismos gramaticales

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

3. Poder expresivo de los formalismos gramaticales

- Capacidad matemática para expresar lenguajes
 - Formalismos
 - Lenguajes naturales
- Existen diversos formalismos gramaticales (no sólo CF-PSGs) y cada uno es capaz de expresar ciertos tipos de lenguajes y no otros

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

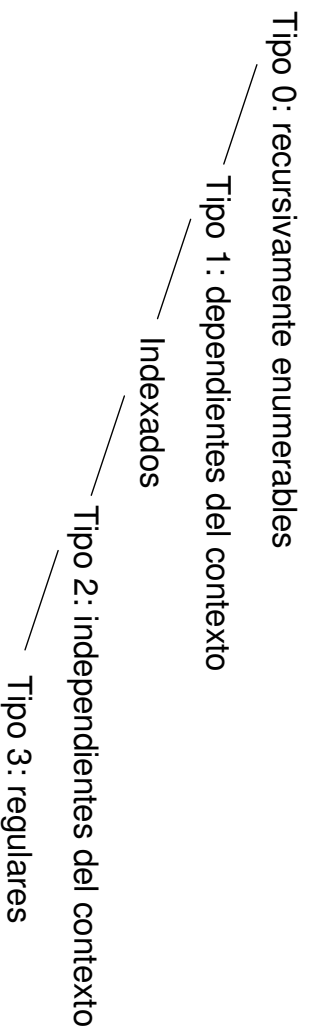
3. Poder expresivo de los formalismos gramaticales

- Los lenguajes se clasifican según la **jerarquía de Chomsky**
 - Los define en términos de los formalismos gramaticales que son capaces de generarlos
 - El tipo de lenguaje que es capaz de expresar un formalismo determina su poder expresivo o capacidad generativa
 - Los formalismos se suelen diferenciar por la forma de las **reglas de reescritura**

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

3. Poder expresivo de los formalismos gramaticales

- Jerarquía de Chomsky



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

3. Poder expresivo de los formalismos gramaticales

- Tipo 3: lenguajes regulares
 - Caracterizados por gramáticas (regulares) con reglas de la forma $A \rightarrow aB$ donde A y B son categorías y a es terminal
 - Ejemplo: $L = \{a^*b^*\}$
- Tipo 2: lenguajes independientes del contexto
 - Caracterizados por gramáticas (independientes del contexto) con reglas de la forma $A \rightarrow \alpha$ donde A es una categoría y α es una secuencia (posiblemente vacía) de categorías y/o terminales
 - Ejemplo: $L = \{a^n b^n\}$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

3. Poder expresivo de los formalismos gramaticales

- Lenguajes indexados
 - Caracterizados por gramáticas (indexadas) que permiten asignar atributos a los símbolos y funciones para calcular los valores de los atributos
 - Se utilizan para definir la semántica de las CFGs en teoría de compiladores y lenguajes formales
 - Ejemplo: $L = \{a^n b^n c^n\}$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

3. Poder expresivo de los formalismos gramaticales

- Tipo 1: Lenguajes dependientes del contexto
 - Caracterizados por gramáticas (dependientes del contexto) con reglas de la forma $\alpha A \beta \rightarrow \alpha \psi \beta$ donde α , β y ψ son secuencias de símbolos y ψ es no vacía
 - Las cadenas α y β determinan el contexto en el cual es aplicable la regla
 - Ejemplo: $L = \{a^n b^n c^n\}$
- Tipo 0: Lenguajes/conjuntos recursivamente enumerables
 - Caracterizados por gramáticas con reglas de reescritura sin restricciones
 - Potencia equivalente a las máquinas de Turing

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

3. Poder expresivo de los formalismos gramaticales

- La familia de lenguajes de tipo i contiene a la familia de tipo $i+1$, es decir, los lenguajes regulares son un subconjunto propio de los dependientes del contexto
- Según el tipo de lenguajes, existen algoritmos de análisis eficientes, o no eficientes, o no existen

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

3. Poder expresivo de los formalismos gramaticales

- Complejidad del lenguaje natural como lenguaje formal
 - La inmensa mayoría de los fenómenos que se encuentran en los LN se puede expresar por medio de CF-PSGs
- Hay fenómenos infrecuentes que no expresables por CF-PSGs
 - Ej. Dialecto del alemán hablado en Zurich, Suiza
 - Permite construcciones del tipo $Npa^m Npb^n Vam Vb^n$ que no son dependientes del contexto

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

3. Poder expresivo de los formalismos gramaticales

- Se pueden diseñar gramáticas de tipo CF-PSG que cubran un porcentaje muy alto de un LN
- El uso de formalismos con mayor capacidad expresiva permite expresar de manera **más intuitiva** fenómenos del lenguaje que son complejos de expresar en CF-PSGs

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

Análisis Sintáctico

4. Algoritmos de análisis

4. Algoritmos de análisis

- Un algoritmo de análisis (parser) es un algoritmo que permite decidir si una expresión en LN pertenece al lenguaje generado por una gramática y, en caso afirmativo, asigna una o más estructuras (árboles) de análisis a la misma
- Nos concentramos primero en reconocimiento para CF-PSGs, y luego en la obtención del árbol

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Tipos de análisis
 - Dirección horizontal (precedencia)
 - Dirección vertical (dominio)
- Según la dirección horizontal (precedencia)
 - De izquierda a derecha y viceversa
- Según la vertical
 - Análisis ascendente (bottom-up)
 - Análisis descendente (top-down)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Según la dirección vertical (dominio)
 - Análisis ascendente (bottom-up)
 - Trata de construir el árbol de análisis de una expresión de abajo hacia arriba, partiendo de la expresión y llegando al símbolo inicial
 - Tiene problemas de recursión infinita con las producciones vacías ($A \rightarrow \epsilon$)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Según la dirección vertical (dominio)
 - Análisis descendente (top-down)
 - Trata de construir el árbol de análisis de una expresión de arriba hacia abajo, partiendo del símbolo inicial y llegando a la expresión en LN
 - Tiene problemas de recursión infinita con la recursión a izquierdas o la recursión a derechas, según la dirección horizontal del análisis ($A \rightarrow Aa$ ó $A \rightarrow aA$)
- Se pueden combinar (análisis ascendente y descendente, análisis basado en el núcleo)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Algoritmos simples de análisis
 - Nos concentramos sólo en reconocer
 - Trabajamos sobre la gramática G32
 - $P = \{(1) O \rightarrow SN SV, (2) SV \rightarrow V, (3) SV \rightarrow V SN, (4) SN \rightarrow pedro, (5) SN \rightarrow arroz, (6) V \rightarrow come, (7) V \rightarrow comía\}$
 - Análisis descendente por la izquierda con vuelta atrás (Análisis Descendente Simple, ADS)
 - Análisis ascendente por la izquierda con vuelta atrás (Left Corner Parsing, LCP)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Algoritmos simples de análisis
 - Estructuras de datos
 - Funciones
 - Heurísticas de control
 - Criterio de éxito
 - Inicio

4. Algoritmos de análisis

- Análisis descendente por la izquierda con vuelta atrás (Análisis Descendente Simple, ADS)
 - Estructuras de datos
 - A = lista de análisis
 - E = lista de entrada
 - Funciones
 - Exp. (X) = expandir la regla X
 - Red. = consumir (reducir) un símbolo de E - scanning

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Heurísticas de control
 - Sólo se puede expandir el primer símbolo de A, y sólo cuando es no terminal
 - Sólo se puede consumir o reducir cuando el primer símbolo de A y de E son el mismo (terminal)
 - Se prefiere consumir a expandir
- Criterio de éxito
 - A = (), E = ()
- Inicio
 - A = (O), E = (pedro comía arroz)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

– Ejecución

- Exp. (1) A = (SN SV), E = (pedro comía arroz)
- Exp. (4) A = (pedro SV), E = (pedro comía arroz)
- Red. A = (SV), E = (comía arroz)
- Exp. (2) A = (V), E = (comía arroz)
- Exp. (6) A = (come), E = (comía arroz) FALLO
- Exp. (7) A = (comía), E = (comía arroz)
- Red. A = (), E = (arroz) FALLO

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

– ... Ejecución

- Exp. (3) A = (V SN), E = (comía arroz)
- Exp. (6) A = (come SN), E = (comía arroz) FALLO
- Exp. (7) A = (comía SN), E = (comía arroz)
- Red. A = (SN), E = (arroz)
- Exp. (4) A = (pedro), E = (arroz) FALLO
- Exp. (5) A = (arroz), E = (arroz)
- Red. A = (), E = () ÉXITO

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Análisis ascendente por la izquierda con vuelta atrás (Left Corner Parsing, LCP)
 - Se suele llamar "left corner parsing" porque las reglas están indexadas por su lado derecho más que por el izquierdo

A
↙
B C D ...

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Estructuras de datos
 - E = lista de entrada
- Funciones
 - Rec. (X) = reconocer la regla X
- Criterio de éxito
 - E = (O)
- Inicio
 - E = (pedro comía arroz)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Heurísticas de control
 - Se reconoce la estructura más a la izquierda, usando la primera regla disponible
 - Si $E = (A B C \dots)$, se busca (por este orden):
 - La primera regla con lado derecho que empiece por A y se ajuste
 - Si se falla, la siguiente regla en estas condiciones
 - Si se falla con A, se prueba con B

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Ejecución
 - Rec. (4) $E = (SN \text{ comía arroz})$
 - Rec. (7) $E = (SN V \text{ arroz})$
 - Rec. (2) $E = (SN SV \text{ arroz})$
 - Rec. (1) $E = (O \text{ arroz})$
 - Rec. (5) $E = (O SN) \text{ FALLO}$
 - Rec. (5) $E = (SN SV SN)$
 - Rec. (1) $E = (O SN) \text{ FALLO}$
 - Rec. (5) $E = (SN V SN)$
 - Rec. (2) $E = (SN SV SN)$
 - Rec. (1) $E = (O SN) \text{ FALLO}$
 - Rec. (3) $E = (SN SV)$
 - Rec. (1) $E = (O) \text{ ÉXITO}$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Estrategias de búsqueda
 - Análisis = búsqueda en un árbol de opciones
 - Opción = ¿Qué regla aplico ahora?
 - Búsquedas
 - En profundidad
 - En anchura
 - Mixtas y variaciones

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Complejidad
 - Independientemente de la estrategia de búsqueda, exponenciales
 - Muy ineficientes
 - Problema: se repiten cálculos (ramas enteras de los árboles de búsqueda duplicadas)
 - Solución: guardar cálculos parciales => análisis basado en diagramas

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

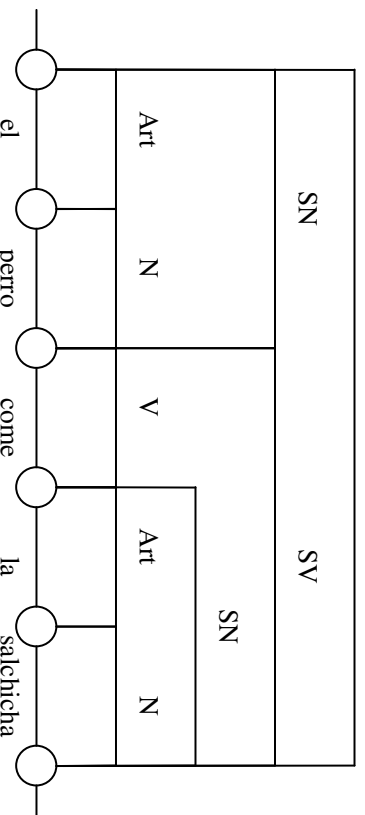
4. Algoritmos de análisis

- Análisis basado en diagramas
 - Estrategia = almacenar resultados parciales
- Versión simple
 - Estructura = diagrama o "chart" = máquina de estados
 - Almacenar símbolos gramaticales

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Frase de entrada
el perro come la salchicha
- Categorías
o



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

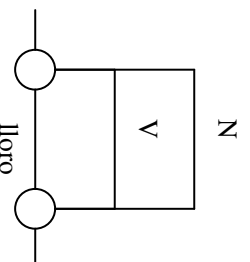
4. Algoritmos de análisis

- **Ventajas**
 - **Eficiencia**
 - No se repiten cálculos
 - Se agregan arcos sólo si no están
 - **Capacidad de reconstrucción del árbol de análisis**
 - Si hay una O desde el estado <1> al <6>, es que debe haber un SN entre el <1> y algún <X>, y un SV entre el estado <X> y el <6>

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- **Capacidad de representar la ambigüedad estructural**



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Representamos hipótesis y objetivos
 - Estructura = máquina de estados con etiquetas avanzadas
 - Etiquetas = reglas con el metasímbolo "."
 - $A \rightarrow a \cdot b$ desde el estado $\langle i \rangle$ al $\langle j \rangle$ ($j \geq i$)
 - Significa que desde $\langle i \rangle$ hasta $\langle j \rangle$ hemos conseguido encontrar a , y que para completar A precisamos encontrar b hasta un estado $\langle k \rangle$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

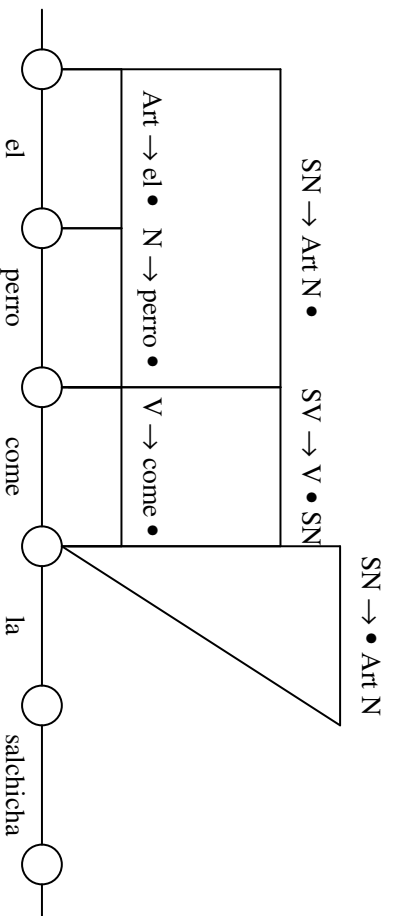
4. Algoritmos de análisis

- Casos especiales de etiquetas
 - $A \rightarrow \cdot a$ desde $\langle i \rangle$ hasta $\langle i \rangle$
 - En $\langle i \rangle$ va a comenzar algo que será una A
 - $A \rightarrow a \cdot$ desde $\langle i \rangle$ hasta $\langle j \rangle$ ($j \geq i$)
 - Entre $\langle i \rangle$ y $\langle j \rangle$ hemos encontrado una A
 - ARCOS INACTIVOS
 - En gramáticas con reglas de la forma $A \rightarrow e$, aparecerán arcos con etiquetas de la forma $A \rightarrow \cdot$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Ejemplo



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

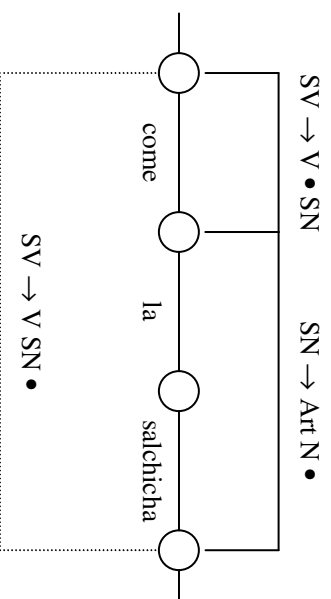
4. Algoritmos de análisis

- Representación de un arco
 - $A \rightarrow a \cdot b$ desde el estado $\langle i \rangle$ al $\langle j \rangle$ ($j \geq i$)
 - Se representa por medio del "ítem" $[i, j, A \rightarrow a \cdot b]$
- Reglas de manipulación de los arcos
 - Regla fundamental
 - Reglas adicionales para conectar diagrama y gramática
- Dos tipos de reglas adicionales
 - Regla ascendente
 - Regla descendente

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

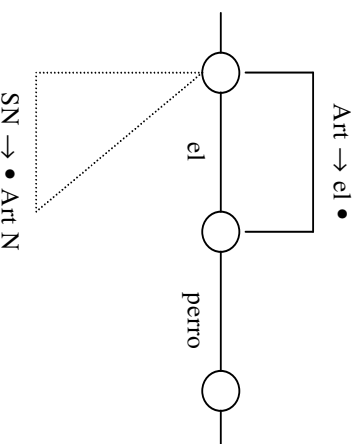
- **Regla fundamental**
 - Si el diagrama contiene un par de arcos de la forma $[i, j, A \rightarrow a \cdot B b]$ y $[j, k, B \rightarrow g \cdot j]$, entonces agregar $[i, j, A \rightarrow a B \cdot b]$



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- **Regla ascendente**
 - Si el diagrama contiene un arco de la forma $[i, j, A \rightarrow a \cdot j]$, entonces agregar y $[i, i, B \rightarrow \cdot A g]$ para toda regla de la gramática de la forma $B \rightarrow A g$
 - Precisa inicializar los arcos léxicos



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Admite todos los tipos de análisis
 - Descendente, ascendente, mixto
 - Izquierda a derecha, viceversa, basado en islas...
- Diversas estrategias de control
 - Control de búsqueda = gestión del conjunto de arcos activos
 - Conjunto como cola o como pila
 - Combinaciones entre dirección del análisis y control

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Algoritmo de Earley
 - Estrategia de control descendente para análisis basado en diagramas
 - Dos tipos de estructuras para representar el diagrama
 - Ítems de la forma $[A \rightarrow a \cdot b, i]$
 - Secuencia de conjuntos I_j con $j = 0, \dots, n$, siendo n el número de elementos léxicos de la frase a reconocer
 - $[A \rightarrow a \cdot b, i] \in I_j$ con $j \geq i \Leftrightarrow$ el diagrama contiene el arco $[i, j, A \rightarrow a \cdot b]$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Esqueleto del algoritmo
 - Objetivo = construir los conjuntos I_j para una frase de entrada $a_1 \dots a_n$ y verificar que el ítem $[S \rightarrow g \cdot, 0]$ pertenece a I_n , para alguna cadena g y siendo S el símbolo inicial
 - 1. Contruir I_0
 - 2. Construir I_j en términos de I_0, \dots, I_{j-1} , para $j = 1, \dots, n$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Construcción de I_0
 - (1) Para cada producción $S \rightarrow a$ con $S =$ símbolo inicial, agregar el ítem $[S \rightarrow \cdot a, 0]$
 - Repetir los pasos (2) y (3) hasta que no se pueda agregar ningún ítem
 - (2) Si el ítem $[B \rightarrow g \cdot, 0]$ pertenece a I_0 , agregar $[A \rightarrow a \cdot B \cdot b, 0]$ por cada ítem de la forma $[A \rightarrow a \cdot B b, 0]$ que esté en I_0
 - (3) Si el ítem $[A \rightarrow a \cdot B b, 0]$ está en I_0 , entonces para cada regla de la forma $B \rightarrow g$ en la gramática, agregar el ítem $[B \rightarrow \cdot g, 0]$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Construcción de I_j en términos de I_0, \dots, I_{j-1}
 - (4) Para cada ítem de la forma $[A \rightarrow a \cdot c b, i]$ en I_{j-1} , con $c = a_j$ en la cadena de entrada, agregar $[A \rightarrow a c \cdot b, i]$ a I_j
 - Repetir los pasos (5) y (6) hasta que no se pueda agregar ningún ítem a I_j
 - (5) Si el ítem $[B \rightarrow g \cdot, i]$ pertenece a I_j , agregar $[A \rightarrow a B \cdot b, k]$ por cada ítem de la forma $[A \rightarrow a \cdot B b, k]$ que esté en I_i
 - (6) Si el ítem $[A \rightarrow a \cdot B b, i]$ está en I_j , entonces para cada regla de la forma $B \rightarrow g$ en la gramática, agregar el ítem $[B \rightarrow \cdot g, j]$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Complejidad
 - Para una frase cualquiera de longitud n , y suponiendo acceso en tiempo constante a la gramática, la complejidad de Earley es de $O(n^3)$
 - Para una frase cualquiera de longitud n y una gramática no ambigua accesible en tiempo constante, la complejidad de Earley es de $O(n^2)$

4. Algoritmos de análisis

- Construcción del árbol de análisis
 - Puede hacerse
 - Simultáneamente con el análisis
 - Posteriormente al análisis
 - Debe dar cuenta de la ambigüedad
 - Análisis basado en diagramas mantiene los resultados parciales
 - Si hay que construir todos los árboles de análisis
 - Aun después de Earley, tiempo exponencial

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

4. Algoritmos de análisis

- Algoritmos más eficientes
 - Existen para gramáticas no ambiguas (LR(0), LR(1), etc.) - Teoría de procesadores de lenguaje
 - Para gramáticas ambiguas, algoritmo de Tomita que es una extensión de LR(1)

Análisis Sintáctico

5. Gramáticas de cláusulas definidas

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Descripción general
- Notación
- Uso
- Análisis gramatical

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Definite Clause Grammars (DCGs)
- Extensión natural de las CF-PSGs
- Mejoran a las CF-PSGs
 - Proporcionan dependencia del contexto
 - Construcción de estructuras
 - Sintácticas
 - Semánticas
 - Condiciones adicionales => cálculos auxiliares

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Heredan las propiedades de las CF-PSGs
 - Claridad y modularidad
 - Recursividad
 - Sobre todo relativos a algoritmos de análisis
 - Resultados teóricos
- Básicamente CF-PSGs con extensiones
- Azúcar sintáctico sobre Prolog

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Notación básica
 - Símbolos con minúsculas
 - Símbolos léxicos entre corchetes (listas)
 - Símbolos léxicos con mayúscula entre comillas simples
 - Reglas incluyen "-->", " y "."
- Ejemplo: DCG31 (G31)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Extensiones de la notación
 - Inclusión de variables en las categorías
 - Primera letra con mayúsculas
 - Llamadas a Prolog
 - Entre llaves
- Ejemplo
 - nombre(N) --> [Palabra], {raiz(Palabra,N),es_nombre(N)}.
 - Significa que la palabra Palabra es un nombre N si la raíz de Palabra es N y N es un nombre.

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Usos de las DCGs
 - Dependencia del contexto
 - Concordancia
 - Restricciones selectivas
 - Construcción de estructuras
 - Sintácticas
 - Semánticas
 - Realización de cómputos auxiliares
 - Búsqueda en el diccionario y análisis morfológico

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Concordancia
 - Es una dependencia contextual en **sentido lingüístico**
 - No es una dependencia contextual en sentido de **poder expresivo**
 - Ej. "los hombres"
 - "hombres" tiene que concordar con el número del contexto marcado por "los"
- Se puede tratar con CF-PSGs

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- La gramática
- se convierte en

sintagma_nominal →
artículo nombre
artículo → el
artículo → los
nombre → hombre
nombre → hombres

sintagma_nominal →
artículo_singular
nombre_singular
sintagma_nominal → artículo_plural
nombre_plural
artículo_singular → el
artículo_plural → los
nombre_singular → hombre
nombre_plural → hombres

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Concordancia
 - Se simplifica el tratamiento con DCGs
 - Ejemplo
 - sintagma_nominal --> artículo(Numero), nombre(Numero).
 - artículo(singular) --> [e].
 - artículo(plural) --> [los].
 - nombre(singular) --> [hombre].
 - nombre(plural) --> [hombres].

5. Gramáticas de cláusulas definidas

- **Restricciones selectivas**
 - Se refieren a dependencias semánticas (actor, instrumento = casos)
 - Imponen chequeos semánticos para disminuir
 - el número de árboles sintácticos generados
 - el grado de regeneración de una gramática
 - Por ejemplo, G31 reconoce "la salchicha come el perro"
=> se convierte en DCG32 que no reconozca la expresión
 - Ejemplos: DCG32 y DCG33

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- **Construcción de estructuras**
 - Sintácticas
 - árbol de análisis
 - Semánticas
 - representación del significado
 - forma lógica

5. Gramáticas de cláusulas definidas

- **Construcción del árbol de análisis**
 - *La estructura representada con términos*
 - *Ejemplo: "El perro come la salchicha" en G31*

o(sn(art(el),
n(perro))),
sv(v(come),
sn(art(la),
n(salchicha))))

– Ejemplo: DCG34

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- **Construcción de la forma lógica (DCG35)**
 - *Modificamos G31 => G33*

O → SN SV
SN → Det N SV → V SN
Det → un Det → todo
Det → una Det → toda
V → come
N → perro N → salchicha

5. Gramáticas de cláusulas definidas

– Forma lógica

- Queremos representar la frase "todo perro come una salchicha" como
$$\forall X (\text{perro}(X) \rightarrow \exists Y (\text{salchicha}(Y) \rightarrow \text{come}(X, Y)))$$
- Los **nombres** se representan por medio de predicados, propiedades sobre variables lógicas
- Los **determinantes** se representan por medio de cuantificadores
- Los **verbos** se representan por medio de predicados

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

– Notación de las fórmulas

- \forall y \exists son operadores binarios \Rightarrow todo(X, P(X)) y existe(X, P(X)) para algún predicado P
- \rightarrow es un operador binario \Rightarrow implica(P, Q)
- Ej. La fórmula anterior sería

todo(X,
 implica(perro(X),
 existe(Y,
 implica(salchicha(Y),
 come(X, Y))))))

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Cómputos auxiliares
 - Ejemplo
nombre(N) --> [Palabra], {raiz(Palabra,N), es_nombre(N)}.
 - Se obtiene la raíz de la palabra por medio de raiz/2
 - Se busca en el diccionario una raíz por medio de es_nombre/1

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- Análisis gramatical en Prolog
 - Prolog convierte automáticamente las reglas a una representación interna operativa
 - DCG = azúcar sintáctico sobre Prolog
 - De manera inmediata, DCG = analizador / generador

5. Gramáticas de cláusulas definidas

- Representación interna similar a los grafos del análisis basado en diagramas
 - Estado = par de listas diferencia
 - Ejemplos
 - oracion --> sintagma_nominal, sintagma_verbal.
 - oracion(S0, S2) :- sintagma_nominal(S0, S1),
sintagma_verbal(S1, S2).
 - nombre(N) --> [W],{raiz((W, N), es_nombre(N))}.
 - nombre(N, [W|X], X) :- raiz(W, N), es_nombre(N).

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- DCG como programa
 - Invocación = objetivo
 - Se puede usar como analizador / generador
 - El tipo de uso está limitado por el comportamiento de los predicados extragramaticales
 - Ejemplo:
 - oracion(o(SN, SV)) --> sintagma_nominal(SN),
sintagma_verbal(SV).

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- **Uso como analizador**
 - oracion(X, [aquí, va, la, lista, de, entrada],[[]]).
 - X toma como valor con el árbol de análisis si la entrada es gramatical, o se produce fallo
 - Por cada valor de X, un árbol de análisis

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- **Uso como generador**
 - oracion(o(sn(...)), X,[[]]).
 - X toma como valor una oración cuyo árbol de análisis es el primer argumento
 - Esto es posible porque dado un significado, hay varias formas de realizarlo superficialmente (con activa, pasiva, elipsis, etc.)
 - Se puede especificar parcialmente el árbol

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- DCG como analizador
 - Tipo de análisis = descendente, de izquierda a derecha y con vuelta atrás, es decir, similar a ADS => ineficiente
 - Ventaja: depuración de la gramática más rápida
 - Se gana en tiempo de desarrollo y se pierde en ejecución (preferible, hardware más barato que software)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

5. Gramáticas de cláusulas definidas

- DCG como analizador
 - Problema en reglas con recursión por la izquierda
 - ¡¡¡Evitarlas!!!
 - Pasar a recursión por la derecha
 - Gestionar con argumentos y otros predicados la construcción de las estructuras deseables
 - Predicado auxiliar de ayuda
 - Para gramáticas con símbolos sin argumentos
phrase(P, L) :- Goal =.. [P,L,[]], call(Goal).

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

Análisis Sintáctico

6. Gramáticas basadas en restricciones

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Introducción
- PATR
- Estructuras de rasgos (ERs)
- Unificación
- ERs frente a términos
- Análisis gramatical en GBRs
- Poder expresivo de las GBRs

6. Gramáticas basadas en restricciones

- Serie de formalismos gramaticales surgidos en los 80
 - Lexical Functional Grammar, LFG (1982)
 - Definite Clause Grammar, DCG (1982)
 - Generalized Phrase Structure Grammar, GPSG (1982)
 - Functional Unification Grammar, FUG (1983)
 - Head-driven Phrase Structure Grammar, HPSG (1985)

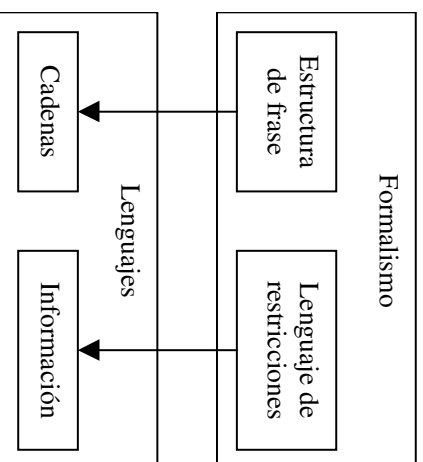
Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Ideas centrales
 - Un lenguaje no es un conjunto de frases aceptables (gramaticalmente), sino una relación entre cadenas y su información lingüística asociada
 - Una gramática es un conjunto de restricciones de dos tipos
 - Restricciones sobre las cadenas y su combinación
 - Restricciones sobre los elementos informativos asociados a las cadenas

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- PATR
 - Formalismo de carácter teórico
 - Proporciona un marco unificador de las GBRs
 - Extrae resultados teóricos sobre análisis gramatical
 - Las reglas describen la relación entre las cadenas y su información asociada simultáneamente
 - La información se representa por medio de estructuras de grafos o de rasgos

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Cada regla consta de
 - Una parte independiente del contexto que indica como se concatenan las cadenas para producir una componente
 - Un conjunto de ecuaciones que restringen los tipos permitidos de información de los constituyentes

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- La regla $O \rightarrow SN SV$ se escribe en PATR como

$X0 \rightarrow X1 X2$
 $<0 \text{ cat}> = O$
 $<1 \text{ cat}> = SN$
 $<2 \text{ cat}> = SV$

6. Gramáticas basadas en restricciones

- Indica que la cadena $X0$ se puede construir por medio de la concatenación de $X1$ y $X2$ en este orden si además se satisface que
 - la información etiquetada como cat en $X0$ es O
 - la información etiquetada como cat en $X1$ es SN
 - la información etiquetada como cat en $X2$ es SV

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Se pueden imponer restricciones más complejas como la concordancia

$X0 \rightarrow X1 X2$

$\langle 0 \text{ cat} \rangle = O$

$\langle 1 \text{ cat} \rangle = SN$

$\langle 2 \text{ cat} \rangle = SV$

$\langle 1 \text{ conc} \rangle = \langle 2 \text{ conc} \rangle$

6. Gramáticas basadas en restricciones

- Se representan restricciones sobre las cadenas atómicas (palabras)

$X_0 \rightarrow \text{'perro'}$

$\langle 0 \text{ cat} \rangle = N$

$\langle 0 \text{ conc gen} \rangle = \text{masc}$

$\langle 0 \text{ conc num} \rangle = \text{sing}$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Convención
 - las etiquetas 'cat' se representan en la parte independiente del contexto
- Ejemplo: PATR31
 - equivalente a DCG33
 - concordancia + restricciones selectivas

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

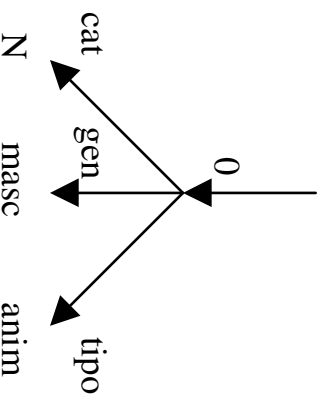
6. Gramáticas basadas en restricciones

- Las cadenas llevan asociados elementos de información representados por medio de grafos (DAGs)
 - Grafos con raíz, conexos y dirigidos con arcos etiquetados desordenados y terminales u hojas etiquetados
 - Las etiquetas de los arcos que surgen de un nodo deben ser distintas

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

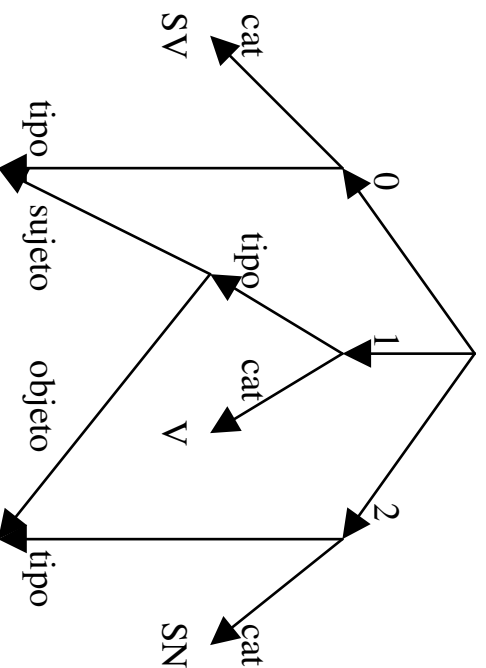
- Ejemplo (R7)



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Ejemplo (R3)



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Restricciones de las ecuaciones sobre los DAGs
 - Un grafo satisface la ecuación $\langle f1 \dots fm \rangle = \langle g1 \dots gk \rangle$ si y sólo si el nodo alcanzado desde la raíz siguiendo los arcos etiquetados con $f1$ hasta fm es el mismo que el alcanzado siguiendo $g1$ hasta gk desde la raíz
 - Un grafo satisface la ecuación $\langle f1 \dots fm \rangle = c$ si y sólo si el nodo alcanzado desde la raíz atravesando $f1$ hasta fm es terminal y su etiquetaa es c

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Estructuras de rasgos (ERs)
 - Son otra forma de representar la información asociada a las cadenas en las GBRS
 - Proviene de la fonética
 - Son equivalentes a los DAGs

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Las estructuras de rasgos son matrices de pares atributo-valor
- Ejemplo (estructura para la cadena de la regla R7)

$$\begin{bmatrix} \text{cat} = \text{N} \\ \text{gen} = \text{masc} \\ \text{tipo} = \text{anim} \end{bmatrix}$$

6. Gramáticas basadas en restricciones

- Admiten estructuración compleja (anidamiento en las estructuras)
- Ejemplo (estructura para la cadena de la regla R6)

$$\left[\begin{array}{l} \text{cat} = V \\ \text{tipo} = \left[\begin{array}{l} \text{sujeto} = \text{anim} \\ \text{objeto} = \text{alim} \end{array} \right] \end{array} \right]$$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Admiten reentrada (etiquetación y coreferencia)
- Ejemplo (estructura para la regla R2)

$$\left[\begin{array}{l} 0 = \left[\begin{array}{l} \text{cat} = \text{SN} \\ \text{tipo} = \$1 \end{array} \right] \\ 1 = \left[\begin{array}{l} \text{cat} = \text{Art} \\ \text{gen} = \$2 \end{array} \right] \\ 2 = \left[\begin{array}{l} \text{cat} = \text{N} \\ \text{tipo} = \$1 \\ \text{gen} = \$2 \end{array} \right] \end{array} \right]$$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Algunos formalismos manipulan las estructuras de rasgos directamente, en lugar de separarlas del formalismo como PATR
 - Tradición de HPSG, LFG y FUG
 - Supongamos que representamos las estructuras como listas entre paréntesis
 - La regla R3 pasa a ser
 - (SV (tipo \$1)) → (V (tipo (sujeto \$1) (objeto \$2))) (SN (tipo \$2))

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Las estructuras de rasgos se pueden interpretar como **funciones parciales** de los rasgos a sus valores, que son otras estructuras de rasgos o átomos
 - Por ejemplo, la estructura A

tipo = persona
nombre = juan
esposa = [nombre = maria]

6. Gramáticas basadas en restricciones

- Corresponde a una función con dominio $\text{dom}(A) = \{\text{tipo, nombre, esposa}\}$ y tal que
 - $A(\text{tipo}) = \text{persona}$
 - $A(\text{nombre}) = \text{juan}$
 - $A(\text{esposa}) = B$ con B una función de dominio $\text{dom}(B) = \{\text{nombre}\}$ y $B(\text{nombre}) = \text{maria}$
- Se admiten caminos
 - $A(\langle \text{esposa nombre} \rangle) = \text{maria}$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- La unificación
 - Ha dado nombre a este tipo de gramáticas (gramáticas de unificación)
 - La unificación se utiliza para combinar ERS o DAGs en la aplicación de las reglas de la gramática en el análisis
 - Se basa en la operación de subsunción

6. Gramáticas basadas en restricciones

- Una ER A subsume a otra B si y sólo si
 - 1. Para cada $R \in \text{dom}(A)$, $A(R)$ subsume a $B(R)$
 - 2. Para todo par de caminos p y q en A con $A(p) = A(q)$, se cumple que $B(p) = B(q)$
- Intuitivamente, A subsume a B cuando A contiene un subconjunto de la información que contiene B
- Se nota por \subseteq

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

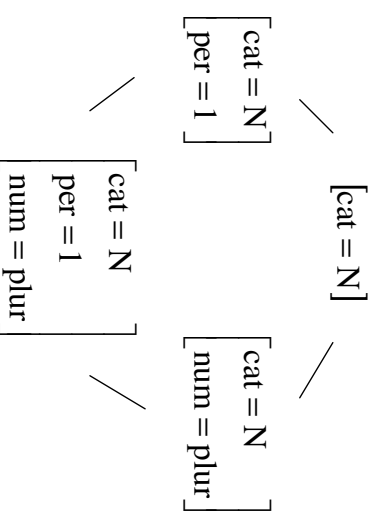
$[0 = [\text{edad} = 19]]$

$\subseteq \left[0 = \left[\begin{array}{l} \text{edad} = 19 \\ \text{edad} - \text{hermano} = 19 \end{array} \right] \right]$

$\subseteq \left[0 = \left[\begin{array}{ll} \text{edad} = \$1 & 19 \\ \text{edad} - \text{hermano} = \$1 & \end{array} \right] \right]$

6. Gramáticas basadas en restricciones

- Define un **orden parcial** entre ERs que da lugar a una estructura de **retículo**



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- La unificación de dos ERs A y B es la estructura C (si existe) que cumple que
 - 1. Tanto A como B subsumen a C ($A \subseteq C$, $B \subseteq C$)
 - 2. Para toda ER D tal que A y B la subsumen, entonces C la subsume (si existe D tal que $A \subseteq D$ y $B \subseteq D$, entonces $C \subseteq D$)
 - Es decir, C es la menor ER subsumida por A y B
 - O de otra forma, C es la ER con menor información que es compatible con A y B simultáneamente

6. Gramáticas basadas en restricciones

- Algoritmo para la unificación de DAGs
- Datos dos DAGs R1 y R2
 1. Si R1 y R2 son hojas
 - 1a. Si sus etiquetas coinciden, devolver R3 con la etiqueta.
 - 1b. Si uno no tiene etiqueta, devolver R3 con la etiqueta del otro.
 - 1c. Si ninguno tiene etiqueta, devolver R3 vacío.
 - 1d. Si las etiquetas no coinciden, devolver fallo.

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

2. Si R1 no es hoja
 - 2a. Si R2 es hoja y está etiquetado, devolver fallo.
 - 2b. Si R2 es hoja sin etiqueta, hacer 2c2.
 - 2c. Si R2 no es hoja, entonces por cada etiqueta L1 \in dom(R1) que llega a R1 hacer:
 - 2c1. Si L1 \in dom(R2) que llega a R21, entonces devolver un nuevo DAG con raíz R3, etiqueta L1 y nodo el resultado de unificar R11 y R21.
 - 2c2. Si L1 \notin dom(R2), entonces devolver un nuevo DAG con raíz R3, etiqueta L1 y nodo una copia de R11.

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

2d. Para cada $L2 \in \text{dom}(R2)$ - $\text{dom}(R1)$ llegando a $R22$, crear un nuevo DAG con raíz $R3$, etiqueta $L2$ y nodo una copia de $R22$.

3. Para garantizar que se mantiene la compartición de valores

3a. Si en algún momento se llega a un nodo ya visitado en algún grafo, el último arco atravesado en ese grafo debe apuntar en $R3$ al nodo equivalente.

3b. Si se llega en los dos grafos a la vez a nodos ya visitados, los nodos correspondientes de $R3$ deben mezclarse recursivamente.

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- La comprobación de aparición
 - Se pueden crear estructuras con ciclos
 - La comprobación de aparición (*occur check*) evita que se unifiquen estructuras con subestructuras propias
 - Es una comprobación costosa que se evita en la implementación de la unificación

6. Gramáticas basadas en restricciones

- Unificación constructiva y destructiva
 - Unificación constructiva = se crea un nuevo DAG
 - Muy costosa
 - Unificación destructiva = se copia del segundo DAG en el primero
 - Más eficiente pero cuidado con los DAG antiguos
 - ¿Se van a volver a usar?
 - ¿Y si falla la unificación?

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Construcción de DAGs para reglas PATR
 - Dada una regla de la forma
$$X_0 \rightarrow X_1 \dots X_n$$
ecuación₁
...
ecuación_M
 - Se construyen las $n+1$ arcos iniciales etiquetadas con 0 hasta $n+1$, y se procesa cada ecuación como una unificación entre el DAG correspondiente a la ecuación y el anterior

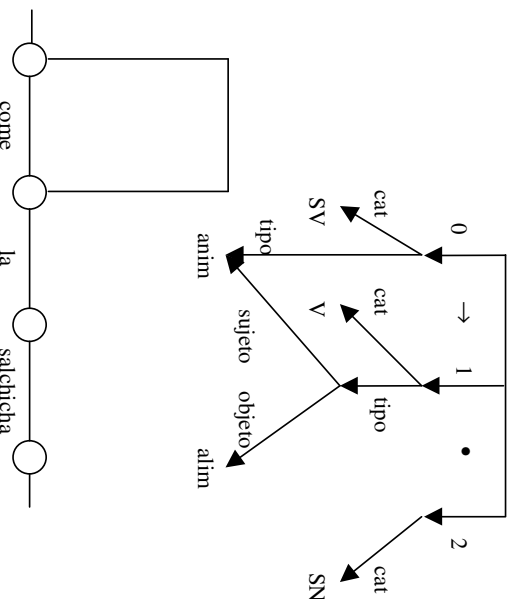
Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Análisis en GBRS
 - Verificación de que 'el perro come la salchicha' es generada por PATR31
 - Tipos de análisis
 - Todos los posibles, aunque especialmente adecuado el ascendente de izquierda a derecha, basado en diagramas
 - Se etiquetan los arcos con DAGs
 - Cada vez que se opera con una regla sobre un diagrama, o bien se construye un DAG o bien se unifican dos DAGs

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Estructuras de rasgos y DAGs frente a términos
 - ERs y DAGs son más flexibles que los términos
 - Las subestructuras se etiquetan simbólicamente, no mediante la posición

- Ejemplo

$$\left[\begin{array}{l} \text{tipo} = \text{persona} \\ \text{nombre} = \text{juan} \\ \text{esposa} = [\text{nombre} = \text{maria}] \end{array} \right]$$

estructura(persona,juan,esposa(maria))

- La esposa se accede por la etiqueta "esposa", no por ser el tercer argumento

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Las ERs no requieren aridad fija
 - Dos estructuras pueden unificar sin tener la misma aridad
 - Ejemplo

$$\left[\begin{array}{l} \text{tipo} = \text{persona} \\ \text{nombre} = \text{juan} \end{array} \right] \cup \left[\begin{array}{l} \text{tipo} = \text{persona} \\ \text{edad} = 23 \end{array} \right] \Rightarrow \left[\begin{array}{l} \text{tipo} = \text{persona} \\ \text{nombre} = \text{juan} \\ \text{edad} = 23 \end{array} \right]$$

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- estructura(persona,juan) y estructura(persona,23) no unifican, aunque sí lo hacen

estructura(persona,juan,X)

∪

estructura(persona,Y,23)

=>

estructura(persona,juan,23)

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Se elimina la distinción entre función y argumento
 - En las ERs, todos los elementos tienen igual rango, no se destaca el símbolo de función
- Las variables y la correferencia se tratan por separado
 - Variable = la estructura vacía [], que unifica con todo
 - Correferencia = etiquetas \$X

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

6. Gramáticas basadas en restricciones

- Poder expresivo de las GBRs
 - Depende del formalismo concreto
 - Todos al menos representan lenguajes dependientes del contexto (tipo 1)
 - DCG => lenguajes recursivamente enumerables (tipo 0)
 - PATR => al menos dependientes del contexto

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

Análisis Sintáctico

7. El papel del léxico

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- **Léxico**
 - Es la componente del sistema de PLN que contiene información sobre las palabras del lenguaje (símbolos léxicos)
 - Conviene separarlo del analizador sintáctico-semántico

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- Las GBRs (incluyendo DCGs) están "fuertemente lexicalizadas"
 - La mayor parte de la información sobre las cadenas está en el léxico
 - Las reglas de la gramática imponen restricciones sobre la información sólomente, pero no aportan información apenas

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- El léxico debe contener muchos tipos de información para ser útil
 - Categorías sintácticas
 - Posibilidades de subcategorización
 - Número
 - Tipo de nombre
 - Aspecto
 - Reflexividad
 - Caso
 - Finitud
 - Persona
 - Género
 - Modo
 - etc.

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- Características del léxico
 - Debe tener información semántica para poder deducir el significado de la oración
 - No debe ser muy grande, sino que debe aprovechar la morfología para evitar redundancia

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- **Papel de la morfología**
 - Para cada palabra, se debe incluir al menos una entrada en el léxico
 - Para cada categoría sintáctica de una palabra, una entrada
 - Para cada significado de una palabra, una entrada
 - Se obtienen léxicos enormes

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- **Ejemplo: entrada léxica del verbo "leer" en traductor.pl (extracto)**

```
verbo(esp,read,np(singular,1)) --> [leol].
verbo(esp,read,np(singular,2)) --> [lees].
verbo(esp,read,np(plural,1)) --> [leemos].
verbo(esp,read,np(plural,2)) --> [leeis].
verbo(esp,read,np(plural,3)) --> [leen].
```

7. El papel del léxico

- Sin embargo, la información léxica para "leemos" se puede derivar del verbo "leer" y su conjugación

leemos = le + emos

le = su significado es "leer(X, Y)" y su categoría V

emos = primera persona, plural, género indistinto, tiempo presente, modo indicativo

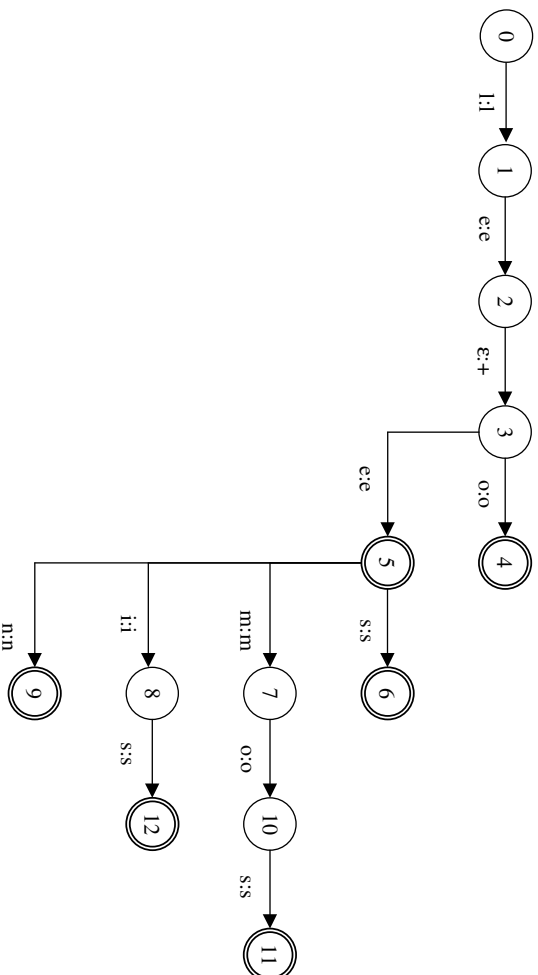
- Sólo hace falta codificar lexemas y morfemas, y reglas que permitan combinarlos

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- **Arquitectura clásica en dos fases**
 - Fase 1 = descomposición por medio de autómata
 - Fase 2 = obtención de información por medio de GBRs

7. El papel del léxico



Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

V → Lex Des

<0 sin subcat> = <1 subcat>

<0 sem> = <1 sem>

<0 sin> = <2 sin>

Lex → 'le'

<0 fn> = leer

<0 arg 1> = anim

<0 subcat arg1> = SN

Des → 'es'

<0 per> = 1

<0 num> = sing

<0 tiempo> = pres

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- Patrones de rasgos
 - Establecen generalizaciones que ayudan a minimizar el léxico
 - En PATR se representan por medio de macros
 - Ejemplo
 - Todo verbo intransitivo admite solo sujeto, y todo verbo transitivo admite también objeto

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

macro sin_IV:

<cat> = V

<arg0> = SN

<arg0 caso> = nom

macro sin_TV:

sin_IV

<arg1 cat> = SN

<arg1 caso> = acus

V → 'ando'

sin_IV

V → 'leo'

sin_TV

Procesamiento del Lenguaje Natural – José María Gómez Hidalgo – U. Europea de Madrid

7. El papel del léxico

- Se pueden expandir
 - Cuando se crea el léxico
 - Cuando se solicita la entrada léxica que llama a la macro
 - Cuando hace falta en una unificación
- Los patrones de rasgos establecen un retículo con herencia, en el que entre ecuaciones incompatibles en un patrón, se prioriza la más específica (similar a la anulación en una jerarquía de programación orientada a objetos, con herencia múltiple)