Capítulo 3

Esquemas de análisis sintáctico

En el contexto computacional, la definición de algoritmos para el análisis sintáctico utiliza esencialmente una estrategia constructiva. Un analizador de este género calcula una serie de resultados intermedios que son utilizados para obtener, sucesivamente, nuevos resultados intermedios más avanzados — en el sentido de más cercanos a una posible solución —. Si la oración es gramatical, este proceso debería concluir obteniendo un resultado final a partir del cual podríamos recuperar sus árboles sintácticos. Aunque esta es fundamentalmente la forma en que proceden la mayor parte de los analizadores descritos en la literatura, la estrategia utilizada y la descripción final de los algoritmos oculta esta similitud.

En este capítulo introduciremos la teoría general de Sikkel que permitirá estudiar con uniformidad las diversas estrategias aplicadas en la definición de algoritmos para el análisis sintáctico [83]. La idea principal en la que se basa esta teoría consiste en relacionar el problema de determinar si una oración es gramatical y el problema de demostrar fórmulas en sistemas deductivos. Fruto de esta asociación nacen los denominados sistemas y esquemas de análisis sintáctico. Además de presentar los fundamentos en los que se apoya este nuevo marco teórico, destacaremos sus ventajas a la hora de especificar y comparar, de forma homogénea, distintas estrategias de análisis. Aunque en este capítulo nos centraremos en el estudio de esquemas para gramáticas independientes del contexto, en los siguientes veremos cómo puede extenderse de forma natural al caso de las gramáticas de adjunción de árboles.

3.1 El análisis interpretado como deducción

El enfoque deductivo se caracteriza por relacionar el proceso de búsqueda de los árboles sintácticos con la demostración de teoremas en un sistema deductivo. La novedad del planteamiento radica en que las fórmulas del sistema deductivo, que denominaremos ítems, tendrán una interpretación específica relacionada con el problema del análisis sintáctico. La forma en que calculamos los ítems en un sistema deductivo de esta índole es similar a la forma en que son deducidas las fórmulas en cualquier otro sistema deductivo. Partiendo de un conjunto de ítems, aplicamos sucesivamente un conjunto de reglas deductivas (o pasos deductivos) de forma que sean calculados todos los ítems que pueden ser deducidos por el sistema. La presencia o no de algunos ítems determinará si una oración es gramatical. En el caso de que lo sea, un examen cuidadoso del conjunto de los ítems calculados nos dará la oportunidad de construir todos sus árboles sintácticos. Una consecuencia interesante de este hecho es la interpretación ambivalente del sistema deductivo como reconocedor o analizador. El propio bosque sintáctico puede ser, a su vez, interpretado mediante una gramática independiente del contexto que genera exclusivamente la oración de entrada [15].

Ejemplo 3.1 Definiremos un sistema deductivo para el método CYK descrito para gramáticas independientes del contexto. En lo que sigue, subrayaremos los nombres de los analizadores aplicados a gramáticas independientes del contexto con objeto de distinguir nítidamente que son aplicados justamente a esa clase de gramáticas.

Dada $G = (V_T, V_N, S, P)$ en forma normal de Chomsky y una cadena de entrada $a_1, \ldots a_n$ con $n \geq 1$, el conjunto de ítems $\mathcal{I}_{\underline{\text{CYK}}}$ se define de la siguiente forma:

$$\mathcal{I}_{\underline{\text{CYK}}} = \{[A, i, j] \mid A \in V_N,\ 0 \leq i \leq j\}$$

Cada ítem de este conjunto representa que el símbolo no terminal A reconoce el segmento de la cadena de entrada comprendido entre las posiciones $i+1$ y j, es decir, $A \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_j$. No se exige que $j \leq n$ con objeto de que la definición del conjunto de ítems no dependa del tamaño de la cadena de entrada. De esta forma, la definición es general y aplicable a cualquier cadena de entrada. Si la oración es gramatical, y sólo en ese caso, deberá ser deducido el ítem $[S, 0, n]$.

El método CYK utiliza una estrategia ascendente pura: parte de los símbolos terminales incluidos en la entrada y culmina alcanzando la raíz de todos los árboles sintácticos de la oración, si existen. Tan sólo dos reglas deductivas son necesarias en el método que nos atañe. Las denominaremos reglas deductivas de comienzo $\mathcal{D}^{\text{Ini}}_{\underline{\text{CYK}}}$ y compleción $\mathcal{D}^{\text{Cmp}}_{\underline{\text{CYK}}}$.

La regla deductiva de comienzo no presenta antecedentes y se limita a calcular aquellos ítems relacionados con producciones $A \to a \in P$ cuyo símbolo terminal en el lado derecho concuerde con algún símbolo de la cadena de entrada a_i.

$$\mathcal{D}^{\text{Ini}}_{\underline{\text{CYK}}} = \frac{}{[A, i-1, i]} \quad A \to a \in P,\ a = a_i$$

Dada la producción $A \to BC \in P$, suponiendo que han sido deducidos dos ítems asociados a los no terminales B y C tales que reconocen segmentos de cadena colindantes, la regla deductiva de compleción los agrupará en un ítem. Este nuevo ítem estará asociado al no terminal A manifestando que éste último reconoce el segmento total reconocido por B y C.

$$\mathcal{D}^{\text{Cmp}}_{\underline{\text{CYK}}} = \frac{[B, i, j]\ [C, j, k]}{[A, i, k]} \quad A \to BC \in P$$

¶

3.1.1 Aportación del enfoque deductivo

La relación descubierta entre el análisis y la deducción lleva consigo interesantes consecuencias. Los sistemas deductivos, al ser presentados mediante una notación rigurosa y con alto nivel de abstracción, son lenguajes muy adecuados para la especificación de problemas. Como toda especificación formal, la descripción de analizadores sintácticos mediante sistemas deductivos facilita el estudio de su corrección debido, en gran medida, a que parte de una notación simple y precisa que ignora gran cantidad de detalles. Por ejemplo, bajo este enfoque se ignoran los siguientes aspectos relacionados con la implementación:

- Las estructuras de datos que soportarán la representación, almacenamiento y acceso de los ítems.
- El flujo de control, ya que sólo se definen las reglas deductivas sin imponer la forma en que éstas deberán ser aplicadas.

- Los mecanismos de comunicación, en el caso de que estemos definiendo analizadores en entornos de computación concurrente.

El modelo propuesto es suficientemente general como para poder adoptar distintas estrategias de resolución bajo un mismo prisma. Distintas estrategias obligarán a definir distintos conjuntos de ítems y reglas deductivas. Sin embargo, mantendremos el mismo mecanismo de deducción.

Todas las características anteriores son el fruto de que este enfoque establece de forma rigurosa la teoría en la que se fundamenta la descripción de algoritmos de análisis. Una consecuencia adicional es la oportunidad de establecer relaciones matemáticas entre los analizadores. Las relaciones definidas entre los algoritmos abrirán las puertas para estudiar de forma comparada los distintos algoritmos presentes en la literatura.

Bajo la perspectiva de la implementación también presenta ventajas, ya que aprovechando que el proceso de deducción es común a todos los sistemas podemos definir un programa independiente dedicado a esta tarea. El algoritmo en el que se basa este programa presenta analogías evidentes con los sistemas para la demostración de teoremas. Sin embargo, su precedente directo hay que buscarlo en los algoritmos *chart*, descritos por Kay, y en los algoritmos tabulares basados en programación dinámica. También podemos traducir las reglas deductivas en un programa lógico, de modo que combinando ambos programas podremos construir prototipos rápidos donde la eficiencia no sea un factor crítico.

3.1.2 Naturaleza de los ítems

Hemos dicho que la construcción del bosque sintáctico se apoya en las producciones de la gramática y la cadena de entrada. Si somos capaces de caracterizar tanto las producciones como la cadena de entrada mediante árboles, el análisis sintáctico se reduce a un problema de composición de árboles gobernado por un conjunto de reglas deductivas que son las que imponen las normas del juego. Con cada nueva aplicación de una regla deductiva se obtendrá un nuevo árbol sintáctico incompleto o parcial, que podrá ser usado en sucesivas aplicaciones de reglas deductivas hasta alcanzar, si existen, todos los árboles sintácticos de la oración respecto de la gramática.

En el contexto de las gramáticas independientes del contexto, la interpretación anterior es clara ya que las producciones $A \rightarrow \delta$ pueden ser consideradas como árboles de altura unidad cuya raíz estará etiquetada con el no terminal A y los nodos inmediatamente dominados por A estarán etiquetados con los símbolos incluidos en δ. Asimismo, para cada símbolo a_i con $1 \leq i \leq n$ de la cadena de entrada $a_1 \ldots a_n$, podemos asociar una pseudo-producción de la forma $a \rightarrow a_i$ donde relacionamos los símbolos terminales con la posición que ocupan en la oración de entrada. A partir de este género especial de producciones, siguiendo el mismo razonamiento que con las otras producciones, obtendremos árboles de altura unidad.

Si optamos por este nuevo planteamiento basado en combinar árboles, el trabajo necesario para obtener un árbol sintáctico puede ser significativamente aliviado si, en vez de preocuparnos de obtener dichos árboles, nos preocupamos tan sólo en demostrar que existen. Con esta nueva estrategia, podemos agrupar conjuntos de árboles de forma que compartan determinadas características comunes. Ahora, los pasos deductivos combinarán conjuntos de árboles que consideramos similares bajo determinadas restricciones. La búsqueda de los árboles sintácticos es sustituida por la búsqueda de un conjunto de árboles donde estén incluidos dichos árboles.

Interpretaremos un *ítem* como una restricción que delimita un conjunto de árboles. Con objeto de aumentar la precisión, las restricciones impuestas por un ítem incluyen información relacionada con posiciones relativas a la cadena de entrada. Estas posiciones caracterizarán qué símbolos de la cadena de entrada son reconocidos por los árboles que representa el ítem.

Ejemplo 3.2 Un ítem $[A, i, j]$ en el conjunto de ítems $\mathcal{I}_{\underline{\text{CYK}}}$ representa a aquellos árboles cuya raíz sea el no terminal A y cuya cosecha se corresponda con el segmento de la cadena de entrada comprendido entre las posiciones $i+1$ y j. Al ignorarse la forma interna del árbol, en principio, podrán existir un número indeterminado de árboles que cumplan dicha condición.

Como dijimos la cadena de entrada $a_1 \ldots a_n$ puede ser interpretada mediante una colección de pseudo-producciones de la forma $a \rightarrow a_i$, y de aquí, como una colección de árboles con altura unitaria. Dado el símbolo a_i con $1 \leq i \leq n$, podemos definir el ítem trivial $[a, i-1, i]$ para representar el árbol relacionado con dicho símbolo. Este ítem representa un árbol cuya raíz está etiquetada con el símbolo terminal a que reconoce el símbolo de la cadena de entrada comprendido entre las posiciones $i-1$ e i. ¶

Atendiendo al género de los árboles que pueden pertenecer a un ítem, éstos últimos pueden ser clasificados en nulos, finales, intermedios o mixtos:

- Un ítem es nulo cuando representa una colección vacía de árboles. Este caso es perfectamente posible ya que podemos imponer una restricción que no satisfaga ningún árbol sintáctico parcial o completo obtenido a partir de una gramática.
- Un ítem es final cuando todos los árboles que incluye son árboles sintácticos para alguna oración.
- Un ítem es intermedio cuando todos los árboles que incluye no son árboles sintácticos para ninguna cadena de entrada. Podemos interpretar que los árboles contenidos son árboles sintácticos parciales que durante el proceso de reconocimiento podrán dar lugar a árboles sintácticos para alguna oración.
- Un ítem es mixto cuando incluye árboles que pueden pertenecer a un ítem final o a un ítem intermedio.

Un conjunto de ítems es regular si no contiene ítems mixtos ni nulos. Un conjunto de ítems es semiregular si no contiene ítems mixtos. Para garantizar la corrección de los analizadores definidos mediante sistemas deductivos, el conjunto de ítems debe ser regular. Informalmente, este requisito garantiza que durante el proceso de análisis no exista confusión entre árboles sintácticos de análisis parciales (resultados intermedios) y árboles sintácticos completos (resultados finales).

La mayoría de los analizadores definidos en la literatura utilizan como base conjuntos de ítems semiregulares. Esta situación es debida a que, si se admiten ítems nulos, la definición del conjunto de ítems de un analizador es menos engorrosa. Afortunadamente, la corrección de un analizador no se ve alterada por un conjunto de ítems semiregulares siempre que se garantice que los ítems nulos no intervengan en el proceso de reconocimiento. Siempre que se tenga esta precaución en mente, la definición de analizadores deductivos mediante conjuntos de ítems regulares o semiregualres no presenta ningún problema práctico.

3.2 Sistemas y esquemas

3.2.1 Definición

Una vez introducidas las nociones básicas, veremos formalmente cómo se definen los sistemas deductivos adaptados para el análisis sintáctico. La definición de estos sistemas requiere de dos argumentos: una gramática G perteneciente a una clase de gramáticas CG y una cadena de entrada (oración) que denotaremos mediante $a_1 \ldots a_n$ siendo $0 \leq n$ el número de símbolos de la cadena. Según el número de argumentos requeridos por el sistema deductivo, se distingue entre:

- sistemas de análisis no instanciados: la oración y la gramática son conocidos a priori, por tanto, su definición no requiere de argumentos.
- sistemas de análisis instanciados: el único argumento requerido es el de la oración, ya que la gramática es conocida a priori.
- esquemas de análisis: es el sistema deductivo más general, ya que su definición requiere de ambos argumentos.

Definición 3.1 *Dada una gramática $G \in GC$ y una cadena de entrada $a_1, \ldots, a_n$, un sistema de análisis no instanciado $\mathbb{P}$ es una tupla $\mathbb{P} = (\mathcal{I}, \mathcal{H}, \mathcal{D})$ donde:*

- *$\mathcal{I}$ es un conjunto de ítems, denominado dominio de $\mathbb{P}$*
- *$\mathcal{H}$ es un conjunto finito de elementos a los que se denominan hipótesis*
- *$\mathcal{D} \subseteq \wp_{fin}(\mathcal{H} \cup \mathcal{I}) \times \mathcal{I}$ es un conjunto de reglas deductivas o pasos deductivos, donde $\wp_{fin}(\mathcal{H} \cup \mathcal{I})$ es el conjunto potencia formado tan sólo por conjuntos finitos descritos a partir de $\mathcal{H} \cup \mathcal{I}$.*

Un sistema de análisis $\mathbb{P}$, o simplemente sistema, se denomina regular si su conjunto de ítems $\mathcal{I}$ es regular. De igual forma, $\mathbb{P}$ será semiregular si $\mathcal{I}$ lo es. Según vemos en la definición, el conjunto $\mathcal{H}$ de hipótesis no tiene porqué ser un subconjunto de $\mathcal{I}$. Consideraremos, como veremos posteriormente, que las hipótesis serán ítems especiales cuya información estará relacionada con la cadena de entrada.

Denotaremos las reglas deductivas incluidas en $\mathcal{D}$ mediante:

$$\frac{\xi_1\, \xi_2 \ldots \xi_k}{\eta}$$

donde $\xi_1\, \xi_2 \ldots \xi_k$ con $k \geq 0$ son denominados los antecedentes de la regla deductiva y η su consecuente. Podemos observar, por un lado, que es perfectamente posible una regla deductiva que no presente ningún antecedente (axiomas) y, por otro, que las hipótesis tan sólo pueden participar en las reglas deductivas como antecedentes. La aplicación de las reglas deductivas puede estar sujeta a que una determinada condición evalúe a cierto. Debemos hacer hincapié en que una regla deductiva es realmente un conjunto donde se representan un número potencialmente infinito de ejemplares. Cada ejemplar se corresponde con la propia regla deductiva pero aplicada respecto a un conjunto de ítems o hipótesis conocidos.

Definición 3.2 *Dada una gramática $G \in CG$, un sistema $\mathbb{P} = (\mathcal{I}, \mathcal{H}, \mathcal{D})$ se denominará instanciado cuando definimos una función $\mathcal{H}_0$ que asocia un conjunto de hipótesis con cada cadena de entrada $a_1 \ldots a_n$ de forma que $\mathbb{P} = (\mathcal{I}, \mathcal{H}_0(a_1 \ldots a_n), \mathcal{D})$ es un sistema.*

Consideraremos que la función $\mathcal{H}_0$ será idéntica para todos los sistemas y vendrá definida mediante:

$$\mathcal{H}_0 = \{[a, i-1, i] \,|\, a = a_i\} \cup \{[\#, -1, 0]\} \cup \{[\$, n, n+1]\}$$

De la definición, vemos que la función $\mathcal{H}_0$ se limita a establecer entre qué dos posiciones se encuentran situados los símbolos participantes en la cadena de entrada. Con objeto de dar una definición general, se han considerado dos símbolos terminales nuevos $\#, \$ \notin V$ que delimitan a izquierda y derecha la cadena. Debido a que la función $\mathcal{H}_0$ será constante para cada sistema, no haremos ninguna distinción entre sistemas instanciados y no instanciados. Por la misma razón, no distinguiremos entre $\mathcal{H}_0$ y $\mathcal{H}$.

Definición 3.3 *Un esquema de análisis* **P** *para una clase de gramáticas CG es una función que asocia un sistema* $\mathbb{P} = (\mathcal{I}, \mathcal{H}, \mathcal{D})$ *a cada gramática* $G \in CG$.

De esta forma dada un gramática $G \in CG$ y un esquema de análisis **P**, o simplemente esquema, tenemos que $\mathbf{P}(G)(a_1 \dots a_n)$ es un sistema. De forma análoga a los sistemas, hablaremos de esquemas regulares y semiregulares.

Ejemplo 3.3 El esquema **CYK** para gramáticas independientes del contexto se define a partir del sistema $\mathbb{P}_{\underline{\text{CYK}}} = (\mathcal{I}_{\underline{\text{CYK}}}, \mathcal{H}, \mathcal{D}_{\underline{\text{CYK}}})$ donde $\mathcal{D}_{\underline{\text{CYK}}} = \mathcal{D}^{\text{Ini}}_{\underline{\text{CYK}}} \cup \mathcal{D}^{\text{Cmp}}_{\underline{\text{CYK}}}$. ¶

La distinción entre sistema y esquema es de índole conceptual. Si una cierta propiedad es cumplida en un sistema $\mathbb{P}$ para cualquier gramática $G \in CG$ y cadena de entrada $a_1 \dots a_n$ entonces la propiedad será cumplida para su esquema correspondiente **P**. Si procuramos que tan sólo el conjunto de hipótesis dependa de la cadena de entrada, un sistema o esquema queda completamente determinado tan sólo con definir su conjunto de ítems y reglas deductivas.

3.2.2 Deducción

Una vez definido el concepto de sistema, veremos ahora cómo podemos caracterizar el conjunto de ítems que deduce. La deducción es el mecanismo fundamental por el que simulamos las composiciones que se realizan durante el proceso de reconocimiento. Es decir, interpretaremos que el análisis de una oración respecto a una gramática equivale al cálculo de todos los ítems deducidos por un sistema de análisis para dicha oración y gramática.

Definición 3.4 *Sea* $\mathbb{P} = (\mathcal{I}, \mathcal{H}, \mathcal{D})$ *un sistema donde* $Y, Y' \subseteq (\mathcal{H} \cup \mathcal{I})$. *La relación de inferencia* $\vdash \subseteq \wp(\mathcal{H} \cup \mathcal{I}) \times \mathcal{I}$ *se define mediante:*

$$Y \vdash \eta \text{ si y sólo si } (Y', \eta) \in \mathcal{D} \text{ dado que } Y' \subseteq Y$$

Es decir, la relación $\vdash$ es el cierre debido a la adición de antecedentes en las reglas deductivas del sistema. Según vemos de la definición, el número de antecedentes añadido no tiene porqué ser finito.

Una secuencia de deducciones en un sistema $\mathbb{P}$ es un par $(Y; \xi_1, \dots, \xi_j)$ definido sobre el conjunto $\wp_{fin}(\mathcal{H} \cup \mathcal{I}) \times \mathcal{I}^+$ que cumple que para todo $1 \leq i \leq j$:

$$Y \cup \{\xi_1, \dots, \xi_{j-1}\} \vdash \xi_j$$

Abusando de la notación, denotaremos una secuencia de deducciones mediante:

$$Y \vdash \xi_1 \vdash \dots \vdash \xi_j$$

Denotaremos mediante $\triangle(\mathbb{P})$, o alternativamente $\triangle_{\mathbb{P}}$, el conjunto de todas las secuencias de deducción definidas a partir del sistema $\mathbb{P} = (\mathcal{I}, \mathcal{H}, \mathcal{D})$.

La relación $\vdash^*$ será el cierre reflexivo y transitivo de la relación de inferencia $\vdash$.

$$Y \vdash^* \xi \text{ cuando } \xi \in Y \text{ o } Y \vdash \dots \vdash \xi$$

Esta relación nos servirá para determinar los ítems deducidos por un sistema, es decir, aquellos ítems que podemos obtener mediante las reglas deductivas partiendo de alguna de sus hipótesis.

Definición 3.5 *Dado un sistema* $\mathbb{P} = (\mathcal{I}, \mathcal{H}, \mathcal{D})$ *definimos su conjunto de ítems válidos mediante:*

$$\mathcal{V} = \{\xi \in \mathcal{I} \mid \mathcal{H} \vdash^* \xi\}$$

Ejemplo 3.4 Dada $G = (\{a\}, \{A, B, S\}, S, \{S \to AA, S \to AB, A \to a, B \to a\})$ y la entrada aa, el conjunto de ítems deducidos por el sistema $\mathbb{P}_{\underline{\text{CYK}}}$ sería el mostrado en la tabla 3.1

#ítem	ítem	regla deductiva aplicada
1	$[A, 0, 1]$	Comienzo
2	$[A, 1, 2]$	Comienzo
3	$[B, 0, 1]$	Comienzo
4	$[B, 1, 2]$	Comienzo
5	$[S, 0, 2]$	Compleción(1,2) Compleción(1,4)

Tabla 3.1: Deducción en el método CYK

Podemos caracterizar de forma global los ítems válidos deducidos por el sistema $\mathbb{P}_{\underline{\text{CYK}}}$ de la siguiente forma:

$$\mathcal{V}_{\underline{\text{CYK}}} = \{[A, i, j] \mid A \stackrel{*}{\Rightarrow} a_{i+1} \dots a_j\}$$

¶

3.2.3 Corrección

Una vez conocidos todos los ítems válidos de un sistema, la presencia o no de algunos ítems permitirá determinar si una oración es gramatical. En vez de introducir el concepto de validez semántica a la hora de definir la corrección de un sistema o esquema, nos bastará con distinguir de forma adecuada qué género de ítems consideramos que conducen o no a un resultado satisfactorio. Puesto que el problema a resolver está fijado de antemano, la distinción de los ítems vendrá motivada claramente según el género de árboles representado por los ítems.

Como vimos al estudiar las distintas clases de ítems, éstos se dividían en nulos, finales, intermedios y mixtos. Dado un sistema (semi)regular $\mathbb{P} = (\mathcal{I}, \mathcal{H}, \mathcal{D})$ dividiremos su conjunto de ítems en dos:

- el conjunto de ítems *finales* $\mathcal{F} \subseteq \mathcal{I}$;
- y el conjunto de ítems *intermedios* $\mathcal{I} - \mathcal{F}$.

El conjunto $\mathcal{F}$, lógicamente incluirá todos los ítems que sean finales. Es decir, aquellos que representen árboles de análisis para alguna cadena de entrada.

En general, estamos interesados en el bosque sintáctico asociado a una sola entrada. Para distinguir esta situación, dividiremos el conjunto de ítems finales en dos subconjuntos disjuntos, denominados conjuntos de ítems finales correctos e incorrectos. El conjunto de ítems *finales correctos* $\mathcal{C}$ tan sólo incluirá los árboles de análisis para una determinada cadena de entrada cuando ésta sea, efectivamente, gramatical.

Ejemplo 3.5 Siguiendo con el sistema $\mathbb{P}_{\underline{\text{CYK}}}$, su conjunto de ítems finales $\mathcal{F}_{\underline{\text{CYK}}}$ vendrá determinado por el ítem $[S, 0, n]$, mientras que el conjunto de ítems finales correctos será $\mathcal{C}_{\underline{\text{CYK}}} = \mathcal{F}_{\underline{\text{CYK}}}$ si la oración de entrada es gramatical y $\mathcal{C}_{\underline{\text{CYK}}} = \emptyset$ en el caso de que no lo sea. ¶

Definición 3.6 *Sea $\mathcal{V}$ el conjunto de ítems deducidos por el sistema $\mathbb{P}$:*

- *Un sistema $\mathbb{P}$ será consistente si $\mathcal{F} \cap \mathcal{V} \subseteq \mathcal{C}$.*
- *Un sistema $\mathbb{P}$ será completo si $\mathcal{F} \cap \mathcal{V} \supseteq \mathcal{C}$.*
- *Un sistema $\mathbb{P}$ será correcto si es completo y consistente, es decir, si $\mathcal{F} \cap \mathcal{V} = \mathcal{C}$.*

Una vez definido qué entendemos por sistemas correctos, la extensión de este concepto a esquemas es como cabría esperar. Un esquema (semi)regular $\mathbf{P}$ será consistente, completo o correcto para una clase de gramáticas CG si, respectivamente, $\mathbf{P}(G)(a_1 \dots a_n)$ es consistente, completo o correcto para toda gramática $G \in CG$ y cadena de entrada $a_1 \dots a_n$.

Según hemos visto, la corrección de un sistema permite establecer si una oración es gramatical. Si el objetivo final es la obtención de los árboles sintácticos debemos posteriormente considerar el conjunto de ítems válidos. Veremos esquemáticamente dos métodos clásicos para la obtención de dichos árboles. Partiendo de los ítems finales, el primer método consiste en construir los árboles sintácticos de forma descendente aplicando las reglas deductivas en orden inverso a como fueron aplicadas. Esta método se aprovecha de que los ítems representan árboles sintáctico incompletos y que las reglas deductivas reflejan cómo han sido combinados éstos últimos. El segundo método consiste en anotar los ítems válidos con la información de cómo han sido deducidos. La información deseada, que se encuentra dispersa entre los ítems válidos, es organizada mediante un grafo implícito que nos dice cómo están relacionados los ítems.

3.3 Relaciones entre esquemas

Sin duda, una de las mayores ventajas ofrecidas por la definición de analizadores sintácticos mediante sistemas deductivos es la capacidad de establecer relaciones matemáticas entre ellos. Aunque la mayor parte de las relaciones también pueden ser definidas como cabría esperar para sistemas, en este caso es más interesante su definición para esquemas, ya que el estudio de las mismas se centra sobre los propios analizadores independientemente de la gramática y cadena de entrada. Una vez definida la relación entre los esquemas, al igual que con cualquier otra relación matemática, podemos estudiar qué propiedades cumplen.

Otra consecuencia añadida es que las relaciones inducen la construcción de una red donde se ponen de manifiesto las similitudes entre los analizadores. De esta forma los sistema deductivos, al mismo tiempo que permiten la especificación y estudio de analizadores de forma aislada, permiten el estudio global y comparado de los mismos.

Las propias relaciones pueden ser utilizadas de forma activa interpretándolas como transformaciones. Así, podemos obtener nuevos analizadores a partir de uno dado, o incluso mejorar el rendimiento computacional de un analizador. Desde esta perspectiva, es interesante conocer qué propiedades respecto a la corrección son preservadas por una relación. De esta forma, podemos conocer de antemano si un nuevo analizador producto de una transformación es consistente o completo.

Considerando que el conjunto de hipótesis es común a todos los esquemas, las relaciones vendrán descritas tan sólo a través de los conjunto de ítems y de reglas deductivas. En lo que sigue, consideraremos los siguientes esquemas: $\mathbf{P_1} = (\mathcal{I}_1, \mathcal{H}, \mathcal{D}_1)$ y $\mathbf{P_2} = (\mathcal{I}_2, \mathcal{H}, \mathcal{D}_2)$. Denotaremos mediante $\vdash_1$ y $\vdash_2$, respectivamente, las relaciones de inferencia definidas sobre los esquemas $\mathbf{P_1}$ y $\mathbf{P_2}$. Del mismo modo usaremos $\mathcal{V}_1$ y $\mathcal{V}_2$ para referirnos a sus respectivos conjuntos de ítems válidos.

Podemos definir funciones que transforman ítems de un esquema en ítems de otro, es decir, funciones del tipo: $f : \mathcal{I}_1 \to \mathcal{I}_2$. Veremos primero una clase de función entre ítems, denominada regular, que será de especial interés a la hora de relacionar esquemas. Las funciones regulares se caracterizan por conservar la información representada por los ítems. Es decir, todos los árboles que son representados por un ítem están contenidos en la imagen del ítem. Posteriormente, generalizaremos la función f para que sea aplicable a conjuntos de ítems, reglas deductivas y secuencias deductivas.

Definición 3.7 *Decimos que una función entre ítems $f : \mathcal{I}_1 \to \mathcal{I}_2$ es regular, si para todo ítem $i \in \mathcal{I}_1$ y para todo árbol $t \in i$, se verifica que $t \in f(i)$.*

La función f, regular o no, extendida para cubrir conjuntos de ítems, dado $Y \subseteq \mathcal{I}_1$, se define mediante:

$$f(Y) = \{\xi \in \mathcal{I}_2 \mid \exists \eta \in Y : f(\eta) = \xi\}$$

es decir, $f(Y)$ sería el subconjunto de $\mathcal{I}_2$ que contiene aquellos ítems que son imagen de algún ítem contenido en Y.

Asumiendo que el conjunto de hipótesis es disjunto con respecto al conjunto de los ítems en $\mathcal{I}_1$ y $\mathcal{I}_2$, y dado $f(h) = h$ para todo $h \in \mathcal{H}$, la función f extendida a relaciones de inferencia o reglas deductivas se define:

$$f(\eta_1 \dots \eta_k \vdash \xi) = f(\eta_1) \dots f(\eta_k) \vdash f(\xi)$$

De forma análoga, podemos extender f a secuencias deductivas, de modo que la notación

$$f(\triangle(\mathbf{P_1})) = \triangle(\mathbf{P_2})$$

será una forma concisa de representar:

$$Y_2 \vdash_2 x_1 \vdash_2 \dots \vdash_2 x_j$$

si y solo si existen $Y_1 \in \wp(\mathcal{H} \cup \mathcal{I}_1)$ con $f(Y_1) = Y_2$ y $x'_1, \dots x'_j \in \mathcal{I}_1$ con $f(x'_i) = x_i$ tales que

$$Y_1 \vdash_1 x'_1 \vdash_1 \dots \vdash_1 x'_j$$

Tras introducir los conceptos preliminares necesarios, veremos su utilidad a la hora de establecer relaciones entre esquemas. Agruparemos dichas relaciones en dos grandes grupos según sean producto de una generalización o de un filtro.

- La generalización es el fruto del refinamiento y/o la extensión de un esquema. La generalización consiste en añadir más detalles a un analizador en el sentido de ampliar su conjunto de ítems, reglas deductivas o la clase de gramáticas a la que es aplicable. La ampliación puede resultar útil desde una perspectiva computacional si conduce a mejoras cualitativas.

- Un filtro es, en general, la relación inversa al refinamiento. Ahora el propósito es conseguir mejoras cuantitativas en un analizador disminuyendo su número de ítems, reglas deductivas o secuencias deductivas. La disminución será posible si podemos asegurar que el filtro aplicado no influye en la corrección del esquema.

3.3.1 Generalizaciones

Dentro de las generalizaciones, podemos encontrarnos con las siguientes relaciones: refinamiento de ítems, refinamiento de reglas deductivas y extensiones. Primero veremos cada una de ellas con cierto detalle para finalmente poner un ejemplo que involucra a todas.

Un esquema $\mathbf{P_2}$ es un refinamiento de los ítems de un esquema $\mathbf{P_1}$ si un ítem del esquema $\mathbf{P_1}$ es partido en varios ítems en el esquema $\mathbf{P_2}$. La partición puede obligar a que tengamos que adaptar también el conjunto de reglas deductivas del esquema $\mathbf{P_2}$. Un caso trivial de refinamiento de ítems consiste simplemente en adoptar una nueva notación de los mismos. En este caso, las reglas deductivas deberán asumir esta nueva notación en sus antecedentes y consecuentes.

Definición 3.8 *Decimos que el esquema* $\mathbf{P_2}$ *es un refinamiento de los ítems del esquema* $\mathbf{P_1}$, *denotado mediante* $\mathbf{P_1} \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{P_2}$, *si existe una función regular entre ítems* $f : \mathcal{I}_2 \to \mathcal{I}_1$ *tal que:*

1. $\mathcal{I}_1 = f(\mathcal{I}_2)$ *(la función cubre todos los ítems de* $\mathcal{I}_1$*)*
2. $\triangle(\mathbf{P_1}) = f(\triangle(\mathbf{P_2}))$

En general, la relación inversa al refinamiento de ítems se denomina contracción de ítems y la representaremos mediante $\overset{\mathbf{ic}}{\Longrightarrow}$. Por tanto, si se verifica $\mathbf{P_2} \overset{\mathbf{ic}}{\Longrightarrow} \mathbf{P_1}$, entonces $\mathbf{P_1} \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{P_2}$. La propia filosofía en la que se basan los esquemas de análisis se corresponde con una contracción de ítems, ya que éstos comprimen en un sólo objeto todo un conjunto de árboles.

Un esquema $\mathbf{P_2}$ es un refinamiento de las reglas deductivas de un esquema $\mathbf{P_1}$ si una regla deductiva del esquema $\mathbf{P_1}$ es descompuesta en varias reglas deductivas en el esquema $\mathbf{P_2}$. Es posible que esta descomposición obligue a introducir nuevos ítems en el esquema $\mathbf{P_2}$ para almacenar los resultados intermedios debido al refinamiento efectuado.

Definición 3.9 *Decimos que el esquema* $\mathbf{P_2}$ *es un refinamiento de las reglas deductivas del esquema* $\mathbf{P_1}$, *denotado mediante* $\mathbf{P_1} \overset{\mathbf{sr}}{\Longrightarrow} \mathbf{P_2}$, *si se cumple:*

1. $\mathcal{I}_1 \subseteq \mathcal{I}_2$
2. $\vdash_1^* \subseteq \vdash_2^*$

Las relaciones anteriores pueden ser compuestas dando lugar al concepto de refinamiento.

Definición 3.10 *Decimos que el esquema* $\mathbf{P_2}$ *es un refinamiento del esquema* $\mathbf{P_1}$, *denotado mediante* $\mathbf{P_1} \overset{\mathbf{ref}}{\Longrightarrow} \mathbf{P_2}$, *si existen esquemas* $\mathbf{P'}$ *o* $\mathbf{P''}$ *tales que:*

1. *o bien* $\mathbf{P_1} \overset{\mathbf{sr}}{\Longrightarrow} \mathbf{P'} \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{P_2}$
2. *o bien* $\mathbf{P_1} \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{P''} \overset{\mathbf{sr}}{\Longrightarrow} \mathbf{P_2}$

Un esquema $\mathbf{P_2}$ es una extensión de $\mathbf{P_1}$ si el esquema $\mathbf{P_2}$ es aplicable a una clase mayor de gramáticas. Esta relación es la única que no puede ser definida para sistemas ya que involucra a toda una clase de gramáticas.

Definición 3.11 *Sea* $\mathbf{P_1}$ *un esquema definido sobre una clase de gramáticas* CG_1 *y* $\mathbf{P_2}$ *un esquema definido sobre una clase de gramáticas* CG_2, *decimos que* $\mathbf{P_2}$ *es una extensión del esquema* $\mathbf{P_1}$, *denotado mediante* $\mathbf{P_1} \overset{\mathbf{ext}}{\Longrightarrow} \mathbf{P_2}$, *si se cumple:*

1. $CG_1 \subseteq CG_2$

2. $\mathbf{P_1}(G)(a_1 \dots a_n) = \mathbf{P_2}(G)(a_1 \dots a_n)$ *para toda* $G \in CG_1$ *y cadena de entrada* $a_1 \dots a_n$.

Definición 3.12 *Un esquema* $\mathbf{P_2}$ *es una generalización de* $\mathbf{P_1}$, *denotado* $\mathbf{P_1} \overset{\mathbf{gen}}{\Longrightarrow} \mathbf{P_2}$, *si es una composición arbitraria de refinamientos o extensiones.*

Enunciaremos ahora las propiedades más destacadas de las generalizaciones (la demostración de dichas propiedades se encuentra en [83]).

Propiedad 3.1 *Las generalizaciones están relacionadas de la siguiente forma: todo refinamiento de ítems o reglas deductivas es un refinamiento, todo refinamiento es una generalización, y toda extensión es una generalización.*

- $\overset{\mathbf{ir}}{\Longrightarrow} \subseteq \overset{\mathbf{ref}}{\Longrightarrow}$
- $\overset{\mathbf{sr}}{\Longrightarrow} \subseteq \overset{\mathbf{ref}}{\Longrightarrow}$
- $\overset{\mathbf{ref}}{\Longrightarrow} \subseteq \overset{\mathbf{gen}}{\Longrightarrow}$
- $\overset{\mathbf{ext}}{\Longrightarrow} \subseteq \overset{\mathbf{gen}}{\Longrightarrow}$

Propiedad 3.2 *Las generalizaciones* $\overset{\mathbf{ir}}{\Longrightarrow}, \overset{\mathbf{ic}}{\Longrightarrow}, \overset{\mathbf{sr}}{\Longrightarrow}, \overset{\mathbf{ref}}{\Longrightarrow}, \overset{\mathbf{ext}}{\Longrightarrow}$ *y* $\overset{\mathbf{gen}}{\Longrightarrow}$ *son relaciones reflexivas y transitivas.*

Propiedad 3.3 *Respecto a la corrección, las generalizaciones verifican:*

- *La relación* $\overset{\mathbf{sr}}{\Longrightarrow}$ *preserva la completud.*
- *Si* $\mathbf{P_1} \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{P_2}$, *entonces la corrección del esquema* $\mathbf{P_2}$ *implica la corrección de* $\mathbf{P_1}$.
- *La relación* $\overset{\mathbf{ic}}{\Longrightarrow}$ *preserva la corrección.*

Ejemplo 3.6 Veremos cómo generalizar el esquema **CYK** para que pueda ser aplicado a cualquier gramática independiente del contexto. Es decir, la gramática no tendrá porqué estar en forma normal de Chomsky. Como veremos posteriormente, el esquema resultante, que denominaremos **buE**, está íntimamente relacionado con el analizador de Earley para gramáticas independientes del contexto.

Para demostrar que se verifica $\mathbf{CYK} \overset{\mathbf{gen}}{\Longrightarrow} \mathbf{buE}$, definiremos una serie de esquemas intermedios que cubrirán todas las relaciones de generalización vistas:

$$\mathbf{CYK} \overset{\mathbf{ir}}{\Longrightarrow} \mathbf{CYK'} \overset{\mathbf{sr}}{\Longrightarrow} \mathbf{ECYK} \overset{\mathbf{ext}}{\Longrightarrow} \mathbf{buE}$$

El esquema **CYK'** es similar al esquema **CYK** salvo que sus ítems son refinados de manera que un ítem CYK, supongamos $[A, i, j]$, es enriquecido mediante $[A \to \delta\bullet, i, j]$. Es decir, hemos detallado la producción asociada al símbolo A del ítem, y por tanto, un ítem original se corresponde potencialmente con un conjunto de ítems en el nuevo esquema. El punto que decora la producción delimita qué sección de su parte derecha reconoce el segmento de la cadena de entrada comprendido entre las posiciones $i+1$ y j. En este esquema, puesto que la sección derecha es vacía, indica que toda la producción es la que ha reconocido el segmento. Esta interpretación está en consonancia con la aportada por los ítems del esquema **CYK**. Las reglas deductivas no requieren ser modificadas salvo que debemos adoptar en los antecedentes y en el consecuente

la nueva definición de ítem. Ahora bien, debido a que los ítems son más explícitos a la hora de determinar tanto los símbolos de la cadena de entrada como las producciones, las condiciones laterales que delimitan la aplicación de las reglas deductivas pueden ser sustancialmente simplificadas.

Dada una gramática independiente del contexto $G = (V_T, V_N, S, P)$ en forma normal de Chomsky, el esquema **CYK′** se define mediante el conjunto de ítems $\mathcal{I}_{\underline{\text{CYK}'}}$ y el conjunto de reglas deductivas $\mathcal{D}_{\underline{\text{CYK}'}} = \mathcal{D}^{\text{Ini}}_{\underline{\text{CYK}'}} \cup \mathcal{D}^{\text{Cmp}}_{\underline{\text{CYK}'}}$ donde:

$$\mathcal{I}_{\underline{\text{CYK}'}} = \{[A \to \delta\bullet, i, j] \mid A \to \delta \in P,\ 0 \leq i \leq j\}$$

$$\mathcal{D}^{\text{Ini}}_{\underline{\text{CYK}'}} = \frac{[a, j-1, j]}{[A \to a\bullet, j-1, j]}$$

$$\mathcal{D}^{\text{Cmp}}_{\underline{\text{CYK}'}} = \frac{[B \to \delta_1\bullet, i, j]\,[C \to \delta_2\bullet, j, k]}{[A \to BC\bullet, i, k]}$$

El siguiente esquema va a refinar la regla deductiva $\mathcal{D}^{\text{Cmp}}_{\underline{\text{CYK}'}}$ de forma que una deducción en el esquema **CYK′**:

$$[B \to \delta_1\bullet, i, j]\,[C \to \delta_2\bullet, j, k] \vdash [A \to BC\bullet, i, k]$$

va a ser obtenida en el nuevo esquema **ECYK** mediante la siguiente secuencia de deducciones:

$$\begin{array}{rl} & \vdash [A \to \bullet BC, i, i] \\ [A \to \bullet BC, i, i]\,[B \to \delta_1\bullet, i, j] & \vdash [A \to B \bullet C, i, j] \\ [A \to B \bullet C, i, j]\,[C \to \delta_2\bullet, j, k] & \vdash [A \to BC\bullet, i, k] \end{array}$$

Para conseguirlo tendremos que reorganizar tanto las reglas deductivas como el conjunto de ítems. Primero, debemos admitir producciones en los ítems cuyo punto no tenga porqué estar al final de su parte derecha. Segundo, debemos sustituir las reglas del esquema **CYK′** por otras, que partiendo el punto desde el principio de las producciones ($\mathcal{D}^{\text{Ini}}_{\underline{\text{ECYK}}}$), lo avance ($\mathcal{D}^{\text{Sc}}_{\underline{\text{ECYK}}}$ y $\mathcal{D}^{\text{Cmp}}_{\underline{\text{ECYK}}}$) hasta alcanzar su final. Una vez terminado el reconocimiento de la producción, se procede de forma ascendente al igual que en los esquemas **CYK** y **CYK′**.

El esquema **ECYK**, que tan sólo es aplicable en el caso gramáticas en forma normal de Chomsky, se define mediante el conjunto de ítems $\mathcal{I}_{\underline{\text{ECYK}}}$ y el conjunto de reglas deductivas $\mathcal{D}_{\underline{\text{ECYK}}} = \mathcal{D}^{\text{Ini}}_{\underline{\text{ECYK}}} \cup \mathcal{D}^{\text{Sc}}_{\underline{\text{ECYK}}} \cup \mathcal{D}^{\text{Cmp}}_{\underline{\text{ECYK}}}$ tales que:

$$\mathcal{I}_{\underline{\text{ECYK}}} = \{[A \to \nu \bullet \omega, i, j] \mid A \to \nu\omega \in P,\ 0 \leq i \leq j\}$$

$$\mathcal{D}^{\text{Ini}}_{\underline{\text{ECYK}}} = \frac{}{[A \to \bullet\delta, j, j]}$$

$$\mathcal{D}^{\text{Sc}}_{\underline{\text{ECYK}}} = \frac{[A \to \nu \bullet a\omega, i, j]\,[a, j, j+1]}{[A \to \nu a \bullet \omega, i, j+1]}$$

$$\mathcal{D}^{\text{Cmp}}_{\underline{\text{ECYK}}} = \frac{[B \to \delta\bullet, j, k]\,[A \to \nu \bullet B\omega, i, j]}{[A \to \nu B \bullet \omega, i, k]}$$

¶

La definición del esquema **buE** es análoga a la del esquema **ECYK**. Por tanto se cumple: $\mathcal{I}_{\underline{\text{buE}}} = \mathcal{I}_{\underline{\text{ECYK}}}$ y $\mathcal{D}_{\underline{\text{buE}}} = \mathcal{D}_{\underline{\text{ECYK}}}$. La diferencia radica en que el nuevo esquema es aplicable a cualquier género de gramática independiente del contexto. En general, como veremos en los capítulos siguientes, la extensión de esquemas no es necesariamente una relación tan trivial como la mostrada.

3.3.2 Filtros

Dentro de los filtros, podemos encontrarnos con las siguientes relaciones: filtro estático, filtro dinámico y contracción de secuencias deductivas. Veremos cada una con cierto detalle al mismo tiempo que presentamos ejemplos.

Un esquema $\mathbf{P_2}$ es un filtro estático del esquema $\mathbf{P_1}$ si son eliminados ítems o reglas deductivas redundantes del esquema $\mathbf{P_1}$. Por tanto, al suprimir los ítems o las reglas deductivas no alteramos el número de árboles sintácticos que pudiéramos obtener, aunque sí podemos reducir el número de ítems válidos. Estos filtros se denominan estáticos, debido a que la reducción puede ser efectuada previamente al propio proceso de deducción.

Definición 3.13 *Decimos que el esquema* $\mathbf{P_2}$ *es un filtro estático del esquema* $\mathbf{P_1}$*, denotado mediante* $\mathbf{P_1} \overset{\mathbf{sf}}{\Longrightarrow} \mathbf{P_2}$*, si se cumple:*

1. $\mathcal{I}_1 \supseteq \mathcal{I}_2$
2. $\mathcal{D}_1 \supseteq \mathcal{D}_2$

Existe un género trivial de filtro estático, que denominaremos eliminación de redundancia y denotaremos mediante $\overset{\mathbf{re}}{\Longrightarrow}$, que se caracteriza por que mantiene el conjunto de ítems válidos. Un filtro de esta clase simplemente elimina aquellos ítems que jamás serán deducidos o suprime aquellas reglas inútiles que no se aplicarán. Decimos que $\mathbf{P_1} \overset{\mathbf{re}}{\Longrightarrow} \mathbf{P_2}$ si se cumple $\mathbf{P_1} \overset{\mathbf{sf}}{\Longrightarrow} \mathbf{P_2}$ y además $\mathcal{V}_1 = \mathcal{V}_2$.

Ejemplo 3.7 Veremos un ejemplo sencillo de filtro estático basándonos en el esquema **buE**. Dada una gramática $G = (V_T, V_N, S, P)$ y un símbolo no terminal $A \in V_N$, éste se denomina reducido si cumple lo siguiente: (i) existen $\nu, \omega \in (V_T \cup V_N)^*$ tales que $S \overset{*}{\Rightarrow} \nu A \omega$ y (ii) existe $w \in V_T^*$ tal que $A \overset{*}{\Rightarrow} w$.

Dada una gramática $G = (V_T, V_N, S, P)$ podemos obtener, a partir de ella, otra gramática reducida $G' = (V_T, V_N', S, P')$. Para conseguirlo basta, por una parte, con que V_N' incluya tan sólo los símbolos no terminales reducidos de V_N y, por otra, que P' contenga todas las producciones de P salvo aquellas que incluyen algún símbolo no terminal que no sea reducido.

Se puede demostrar que las gramáticas son equivalentes, es decir $L(G) = L(G')$, y que dada una oración, perteneciente a ambos lenguajes, obtendremos los mismos árboles sintácticos. La reducción tiene como cometido obtener una nueva gramática donde han sido eliminadas derivaciones que sabemos de antemano que no conducen a ninguna oración gramatical. Si analizamos una gramática no reducida, serán calculadas todas las derivaciones, incluyendo las innecesarias. El filtro estático simplemente eliminará aquellos ítems que sabemos no serán de utilidad. El esquema $\underline{\mathbf{buE'}}$, que presentamos a continuación, es un filtro estático del esquema **buE**, es decir, $\underline{\mathbf{buE}} \overset{\mathbf{sf}}{\Longrightarrow} \underline{\mathbf{buE'}}$. Se define mediante el siguiente conjunto de ítems:

$$\mathcal{I}_{\underline{\text{buE}'}} = \{[A \to \nu \bullet \omega] \mid A \to \nu\omega \in P',\ 0 \leq i \leq j\}$$

El conjunto de reglas deductivas es igual al del esquema **buE**, por tanto $\mathcal{D}_{\underline{\text{buE}'}} = \mathcal{D}_{\underline{\text{buE}}}$. ¶

Un esquema $\mathbf{P_2}$ es un filtro dinámico del esquema $\mathbf{P_1}$ si la validez de algunos ítems del esquema $\mathbf{P_1}$ pueden depender de la validez de otros ítems del mismo esquema. Para conseguirlo se hace uso de información contextual añadiendo nuevos ítems antecedentes o nuevas condiciones laterales en las reglas deductivas del esquema $\mathbf{P_2}$. La diferencia de estos filtros con respecto a los estáticos, es que la reducción realizada requiere ser efectuada durante el proceso de deducción. Un ejemplo clásico de filtro dinámico es la utilización de símbolos de lectura adelantada (*look-ahead*). En este caso, los ítems añadidos a las reglas deductivas se corresponden con hipótesis que permiten restringir su aplicación siempre que un determinado símbolo de la palabra de entrada esté a continuación del esperado.

Definición 3.14 *Decimos que el esquema* $\mathbf{P_2}$ *es un filtro dinámico del esquema* $\mathbf{P_1}$, *denotado mediante* $\mathbf{P_1} \overset{\mathbf{df}}{\Longrightarrow} \mathbf{P_2}$, *si se cumple:*

1. $\mathcal{I}_1 \supseteq \mathcal{I}_2$
2. $\vdash_1 \supseteq \vdash_2$

Ejemplo 3.8 Pasaremos a ver el esquema basado en el conocido método de Earley para gramáticas independientes del contexto. Mostraremos que el nuevo esquema **Earley** es un filtro dinámico del esquema **buE**, es decir **buE** $\overset{\mathbf{df}}{\Longrightarrow}$ **Earley**. Aunque el conjunto de ítems de ambos esquemas va a ser el mismo, $\mathcal{I}_{\underline{\text{buE}}} = \mathcal{I}_{\underline{\text{Earley}}}$, las reglas deductivas van a ser ampliadas. Vimos que el esquema **buE** deducía todos aquellos ítems de la forma $[B \to \bullet\delta, j, j]$ con $0 \leq j$. Ahora, el esquema **Earley** deducirá un ítem de este conjunto tan sólo si es válido el ítem $[A \to \nu \bullet B\omega, i, j]$. Vemos que el filtro consiste en añadir información descendente. O sea, mientras que el esquema **buE** es estrictamente ascendente, el esquema **Earley** es ascendente con predicción. Para que el reconocimiento pueda comenzar debemos añadir otra regla deductiva que realice la predicción sobre las producciones cuyo lado izquierdo sea el axioma. Por tanto,

$$\mathcal{D}_{\underline{\text{Earley}}} = \mathcal{D}^{\text{Ini}}_{\underline{\text{Earley}}} \cup \mathcal{D}^{\text{Sc}}_{\underline{\text{Earley}}} \cup \mathcal{D}^{\text{Pre}}_{\underline{\text{Earley}}} \cup \mathcal{D}^{\text{Cmp}}_{\underline{\text{Earley}}}$$

definidas de la siguiente forma:

$$\mathcal{D}^{\text{Ini}}_{\underline{\text{Earley}}} = \frac{}{[S \to \bullet\delta, 0, 0]}$$

$$\mathcal{D}^{\text{Sc}}_{\underline{\text{Earley}}} = \frac{[A \to \nu \bullet a\omega, i, j]\,[a, j, j+1]}{[A \to \nu a \bullet \omega, i, j+1]}$$

$$\mathcal{D}^{\text{Pre}}_{\underline{\text{Earley}}} = \frac{[A \to \nu \bullet B\omega, i, j]}{[B \to \bullet\delta, j, j]}$$

$$\mathcal{D}^{\text{Cmp}}_{\underline{\text{Earley}}} = \frac{[B \to \delta\bullet, j, k]\,[A \to \nu \bullet B\omega, i, j]}{[A \to \nu B \bullet \omega, i, k]}$$

donde podemos observar que $\mathcal{D}^{\text{Sc}}_{\underline{\text{Earley}}} = \mathcal{D}^{\text{Sc}}_{\underline{\text{buE}}}$ y que $\mathcal{D}^{\text{Cmp}}_{\underline{\text{Earley}}} = \mathcal{D}^{\text{Cmp}}_{\underline{\text{buE}}}$. ¶

Un esquema $\mathbf{P_2}$ es una contracción de secuencias deductivas del esquema $\mathbf{P_1}$ si sustituimos toda una secuencia deductiva del esquema $\mathbf{P_1}$ por otra secuencia de menor longitud en el esquema $\mathbf{P_2}$. Esta clase de filtro es la que mayor grado de reducción efectúa debido a que sustituye toda una secuencia de ítems.

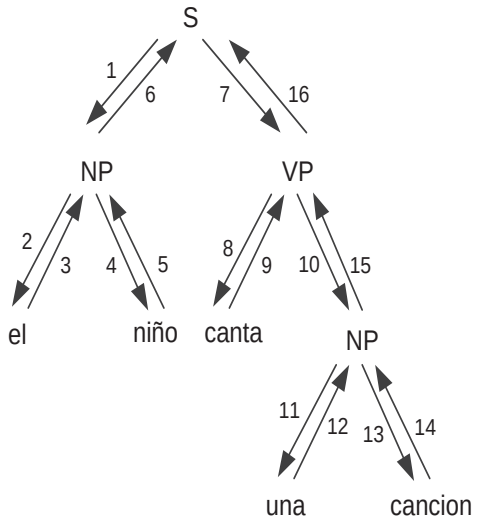

Figura 3.1: Recorrido en el método Earley

Definición 3.15 *Decimos que el esquema* $\mathbf{P_2}$ *es una contracción de secuencias o de pasos deductivos del esquema* $\mathbf{P_1}$*, denotado mediante* $\mathbf{P_1} \overset{\mathbf{sc}}{\Longrightarrow} \mathbf{P_2}$*, si se cumple:*

1. $\mathcal{I}_1 \supseteq \mathcal{I}_2$
2. $\vdash_1^* \supseteq \vdash_2^*$

Ejemplo 3.9 Veremos ahora el esquema **LC** basado en el método de la esquina izquierda. Este nuevo esquema realiza una contracción de secuencias deductivas del esquema **Earley**, es decir, **Earley** $\overset{\mathbf{sc}}{\Longrightarrow}$ **LC**. La contracción consiste en reducir el número de ítems deducidos durante la fase de predicción. Un ejemplo de dicha contracción podemos verlo en las figuras 3.1 y 3.2 donde se muestra la forma en que es recorrido un determinado árbol sintáctico por ambos métodos indicando mediante líneas de punto las predicciones que son suprimidas. Para simplificar las predicciones, el esquema **LC** se apoya en la relación denominada esquina izquierda.

Dada una gramática independiente del contexto $G = (V_T, V_N, S, P)$, denominamos esquina izquierda de una producción al símbolo situado más a la izquierda de la parte derecha de dicha producción. Es decir, dada la producción $C \to X\delta \in P$, su esquina izquierda será el símbolo terminal o no terminal X. Una producción nula $C \to \epsilon$ tendrá como esquina izquierda a la palabra vacía. La relación $>_\ell$ definida en $V_N \times (V \cup \{\epsilon\})$, con $V = V_T \cup V_N$, se define:

$$C >_\ell U \text{ si existe } C \to \delta \in P \text{ tal que } U \text{ es su esquina izquierda}$$

Denotamos mediante $>_\ell^*$ el cierre reflexivo y transitivo de $>_\ell$.

A partir de la relación anterior podemos ver que un ítem $[C \to \bullet D\mu, i, i]$ será deducido por el esquema **Earley** tan sólo cuando es válido el ítem $[A \to \nu \bullet B\omega, h, i]$ con $B >_\ell^* C$. La idea general del esquema consiste en suprimir todos aquellos ítems de la forma $[C \to \bullet D\mu, i, i]$ salvo que $i = 0$ y $C = S$. Por tanto, $\mathcal{I}_{\underline{\text{LC}}} \subseteq \mathcal{I}_{\underline{\text{Earley}}}$. Para garantizar la corrección del algoritmo, introduciremos reglas deductivas asociadas a cada clase de esquina izquierda: cuando ésta es un símbolo no terminal, un terminal o la palabra vacía. El dominio del esquema vendría dado por

$$\mathcal{I}_{\underline{\text{LC}}} = \mathcal{I}_{(1)} \cup \mathcal{I}_{(2)} \cup \mathcal{I}_{(3)}$$

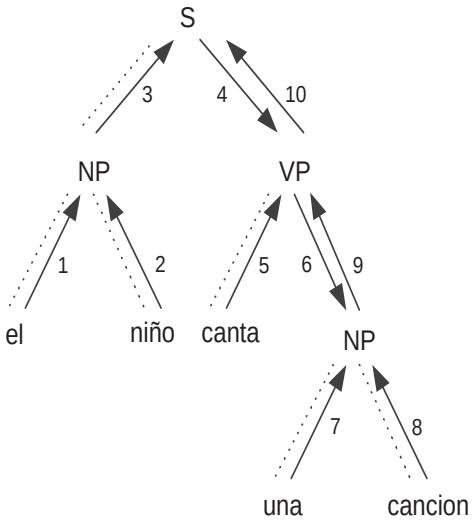

Figura 3.2: Recorrido en el método de la esquina izquierda

donde

$$\mathcal{I}_{(1)} = \{[A \to X\nu \bullet \omega, i, j] \,|\, A \to X\nu\omega \in P,\ 0 \leq i \leq j\}$$
$$\mathcal{I}_{(2)} = \{[A \to \bullet, j, j] \,|\, A \to \epsilon \in P,\ 0 \leq j\}$$
$$\mathcal{I}_{(3)} = \{[S \to \bullet\delta, 0, 0] \,|\, S \to \delta \in P\}$$

$$\mathcal{D}_{\underline{\mathrm{LC}}} = \mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{Ini}} \cup \mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{Sc}} \cup \mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{LC(C)}} \cup \mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{LC(a)}} \cup \mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{LC}(\epsilon)} \cup \mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{Cmp}}$$

donde

$$\mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{Ini}} = \frac{}{[S \to \bullet\delta, 0, 0]}$$

$$\mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{Sc}} = \frac{[A \to \nu \bullet a\omega, i, j]\,[a, j, j+1]}{[A \to \nu a \bullet \omega, i, j+1]}$$

$$\mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{LC(C)}} = \frac{[A \to \nu \bullet B\omega, h, i]\,[D \to \delta\bullet, i, j]}{[C \to D \bullet \mu, i, j]} \quad B >_\ell^* C$$

$$\mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{LC(a)}} = \frac{[A \to \nu \bullet B\omega, h, i]\,[a, i, i+1]}{[C \to a \bullet \mu, i, i+1]} \quad B >_\ell^* C$$

$$\mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{LC}(\epsilon)} = \frac{[A \to \nu \bullet B\omega, h, i]}{[C \to \bullet, i, i]} \quad B >_\ell^* C$$

$$\mathcal{D}_{\underline{\mathrm{LC}}}^{\mathrm{Cmp}} = \frac{[B \to \delta\bullet, j, k]\,[A \to \nu \bullet B\omega, i, j]}{[A \to \nu B \bullet \omega, i, k]}$$

¶

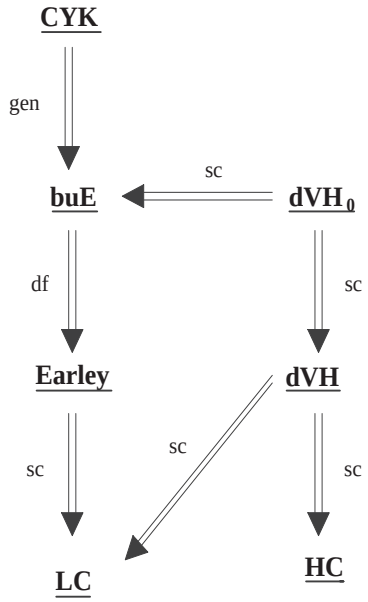

Figura 3.3: Red para gramáticas independientes del contexto

Enunciaremos ahora las propiedades más destacadas de los filtros (la demostración de dichas propiedades se encuentra en [83]).

Propiedad 3.4 *Los filtros se relacionan de la siguiente forma: todo filtro estático es un filtro dinámico, y a su vez todo filtro dinámico es una contracción de secuencias deductivas.*

$$\overset{\mathbf{sf}}{\Longrightarrow} \subseteq \overset{\mathbf{df}}{\Longrightarrow} \subseteq \overset{\mathbf{sc}}{\Longrightarrow}$$

Propiedad 3.5 *Los filtros* $\overset{\mathbf{sf}}{\Longrightarrow}, \overset{\mathbf{df}}{\Longrightarrow}$ *y* $\overset{\mathbf{sc}}{\Longrightarrow}$ *son relaciones reflexivas y transitivas.*

Propiedad 3.6 *Respecto a la corrección los filtros* $\overset{\mathbf{sf}}{\Longrightarrow}, \overset{\mathbf{df}}{\Longrightarrow}$ *y* $\overset{\mathbf{sc}}{\Longrightarrow}$ *preservan la consistencia*

3.3.3 Red de analizadores

Como ya hemos comentado, las relaciones entre esquemas inducen a la construcción de una *red* de analizadores. Podemos considerar que esta red no es sino un resumen conciso donde se determina la forma en que están vinculadas las distintas estrategias. Debido a que el análisis de gramáticas independientes del contexto ha sido objeto de numerosos estudios, la red de referencia es sin duda alguna la que corresponde a este género de analizadores. Sin ánimo de ser exhaustivos ni precisos comentaremos una red simplificada donde tan sólo incluimos algunos de los analizadores más relevantes (véase la figura 3.3).

Partimos del esquema **CYK** que utiliza una estrategia ascendente pura donde la lectura de la cadena de entrada es en una sola dirección, aunque dicha lectura puede empezar por cualquiera de sus símbolos. Este método tiene como limitación que sólo es aplicable a gramáticas independientes del contexto en forma normal de Chomsky. Al generalizar dicho método para que su aplicación cubra cualquier gramática independiente del contexto, obtenemos el esquema **buE** que no es sino una versión ascendente del método de Earley.

Podemos reducir el número de ítems deducidos por el esquema **buE**, si, aun manteniendo la lectura de izquierda a derecha, ésta se efectúa desde el principio hasta el final de la cadena

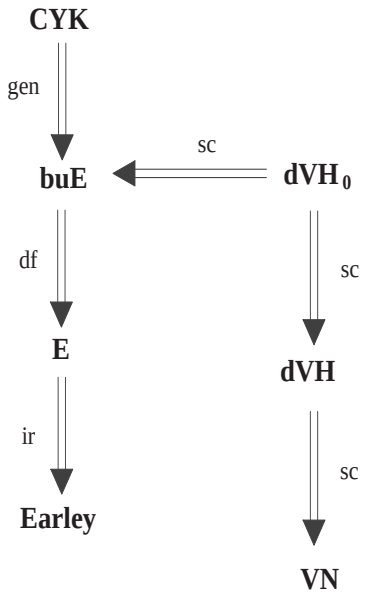

Figura 3.4: Red para gramáticas de adjunción de árboles

de entrada. Para conseguirlo efectuamos un filtro que incluirá información descendente (predicción), lo que nos conduce al esquema **Earley**. Podemos obtener una versión más económica de este último método si comprimimos la etapa de predicción. Para conseguirlo almacenamos la información relacionada con el elemento situado más a la izquierda de cada producción, es decir su esquina izquierda. De este modo, al efectuar una predicción descendemos directamente a través de las esquinas izquierdas de cada producción. Esta estrategia nos conduce al esquema **LC**, que a su vez, puede ser considerado una versión especializada del esquema basado en núcleos **HC** en donde se refina el concepto de esquina. En este caso, la esquina — a la que se denomina núcleo — no necesariamente debe ser el símbolo situado más a la izquierda en la producción. En general, la elección del núcleo vendrá motivada por razones lingüísticas.

Para establecer con mayor rigor la relación entre los esquemas **LC** y **HC** disponemos del esquema **dVH** definido a partir del método descrito por de Vreught y Honig. Los dos esquemas anteriores pueden ser considerados una contracción de este último esquema donde se incorpora información descendente. El esquema **dVH** utiliza una estrategia ascendente donde el reconocimiento puede comenzar sobre cualquier posición de la cadena de entrada y donde la lectura se efectúa en ambos sentidos: es decir, tanto de izquierda a derecha como de derecha a izquierda. Informalmente, si restringimos la lectura para que sea en un sólo sentido de izquierda a derecha nos conducirá al esquema de la esquina izquierda y si seleccionamos los símbolos por donde se empezará el reconocimiento (núcleos) obtendremos una versión basada en núcleos.

La red original para gramáticas independientes del contexto puede ser enriquecida mediante otros esquemas intermedios cuyo interés es más teórico que práctico ya que sirven de nexo para relacionar otros esquemas más destacados. Un ejemplo de esta clase de analizadores es el esquema **dVH0**, al que podemos considerar una versión poco elaborada del esquema **dVH**, cuya utilidad es relacionar las estrategias utilizadas por los esquemas **dVH** y **buE**.

Red para gramáticas de adjunción de árboles

En los dos siguientes capítulos nos centraremos en mostrar los esquemas y las relaciones que originan la red para gramáticas de adjunción de árboles (véase la figura 3.4). Como quedó

dicho, el estudio de las gramáticas de adjunción de árboles ha estado influido en gran medida por los resultados obtenidos para las gramáticas independientes del contexto. La consecuencia más inmediata de esta afirmación es que habrá muchas similitudes entre las redes de ambos formalismos.

Para empezar, la red de las gramáticas de adjunción de árboles incluye también un camino que relaciona los analizadores derivados de CYK y de Earley. También hay un camino que relaciona los analizadores bidireccionales: el método derivado del descrito por de Vreught y Honig para CFG (esquema **dVH**) y el método basado en núcleos debido a Van Noord (esquema **VN**). De forma similar a cómo sucede en las gramáticas independientes del contexto, ambos caminos están relacionados mediante un esquema intermedio $\mathbf{dVH_0}$.

Sin embargo, existen algunas diferencias notables entre ambas redes. La relación entre los esquemas **CYK** y **Earley** no es tan diáfana como cabría esperar. Como veremos posteriormente, la razón se debe a que en las gramáticas de adjunción de árboles, los métodos predictivos no necesariamente garantizan la propiedad del prefijo válido. Por tanto, el método de Earley debe ser divido en dos atendiendo a que dicha propiedad no se cumpla (esquema **E**) o sí se cumpla (esquema **Earley**). También podemos comprobar la ausencia de algunos esquemas que todavía no han sido descritos en la literatura como los esquemas basados en la esquina izquierda.

Como conclusión, podemos destacar que una ventaja adicional del enfoque deductivo precisamente procede de la capacidad para comparar analizadores sintácticos entre distintos formalismos. Esta comparación es factible tanto a nivel teórico (topología de la red) como a nivel empírico (rendimiento de los analizadores).