Análisis ascendente bidireccional de TAG dirigido por el núcleo TIG *

Vicente Carrillo, Víctor J. Díaz
Universidad de Sevilla
Avda. Reina Mercedes s/n
41012 Sevilla
carrillo@lsi.us.es

Miguel Ángel Alonso
Universidade da Coruña
Campus de Elviña s/n
15071 La Coruña
alonso@udc.es

Resumen: Definimos un analizador tabular para gramáticas de adjunción de árboles (TAG) con estrategia de análisis ascendente y recorrido bidireccional de la cadena de entrada. Este analizador es el resultado de la fusión del analizador ascendente bidireccional ya definido para TAG con el nuevo analizador para gramáticas de inserción de árboles (TIG) que presentamos también en este trabajo. Mostramos como el nuevo algoritmo combinado presenta una reducción de la complejidad teórica media respecto al analizador original para TAG.
Palabras clave: análisis sintáctico, gramáticas de adjunción de árboles, gramáticas de inserción de árboles

Abstract: We define a bidireccional bottom-up parser for Tree Adjoining Grammars (TAG). This parser is a mixture of a defined bidireccional bottom-up parser for TAG and a new parser for Tree Insertion Grammars (TIG) that we present here. We show that the new mixed parser for TAG presents a significative reduction of the theoretical complexity.
Keywords: parsing, tree adjoining grammars, tree insertion grammars

1 Introducción

Las gramáticas de adjunción de árboles (*Tree Adjoining Grammars*, TAG) (Joshi y Schabes, 1997) es un formalismo con la potencia generadora y expresividad adecuadas para la descripción de la sintaxis de los lenguajes naturales. Sin embargo, el coste computacional teórico de su análisis sintáctico es elevado, estableciéndose la cota de complejidad temporal en $\mathcal{O}(n^6)$ y la espacial en $\mathcal{O}(n^4)$, donde n es la longitud de la cadena de entrada. Recientemente se han descrito muchas aproximaciones que intentan mejorar las prestaciones de los analizadores para TAG, entre ellas podemos citar las basadas en restricciones sobre el formalismo, como las propuestas descritas en (Schabes y Waters, 1995) y (Rogers, 1994).

Las gramáticas de inserción de árboles (*Tree Insertion Grammar*, TIG) (Schabes y Waters, 1995) se obtienen mediante la aplicación de ciertas restricciones sobre las TAG. Se pueden analizar con el mismo coste que las gramáticas incontextuales *Context Free Grammars*, CFG) y presentan muchas de las ventajas lingüísticas de las TAG, aunque su potencia expresiva se limita a lenguajes incontextuales.

Los *esquemas de análisis sintáctico* (Sikkel, 1997) constituyen un excelente método para describir algoritmos de análisis sintáctico. Podemos encontrar un completo conjunto de algoritmos de análisis para TAG derivados a partir de analizadores para CFG y descritos mediante esquemas de análisis en (Alonso et al., 1999), concretamente, se definen dos ascendentes y dos predictivos con recorridos inidireccionales de la cadena de entrada. Este conjunto se extiende posteriormente con la definición de un esquema ascendente con recorrido bidireccional (Díaz, Carrillo, y Alonso, 2000) basado en el algoritmo de análisis definido para CFG por de Vreught y Honig (de Vreught y Honig, 1989). Para el caso de las TIG, en (Carrillo, Díaz, y Alonso, 2002) encontramos un conjunto de esquemas con diversas estrategias y recorridos unidireccionales, concretamente, tres ascendentes y uno predictivo.

La importancia fundamental del formalismo TIG se basa en el hecho de que la ma-

* Parcialmente financiado por el Ministerio de Ciencia y Tecnología (proyectos FIT-150500-2002-416 y TIC2000-0370-C02-01 ; acciones integradas HF2002-81 y HP2001-0044) y Xunta de Galicia (proyectos PGIDIT02SIN01E y PGIDT01PXI10506PN)

yor parte de las estructuras de los árboles definidos para gramáticas de amplia cobertura se corresponden con estructuras de árboles TIG. Un claro ejemplo lo tenemos en la gramática del inglés XTAG (Christy et al., 1994), donde el 99% de los árboles y adjunciones permitidas existentes se encuentran dentro del formalismo TIG. En este trabajo proponemos un analizador para TAG que saca provecho de esta circunstancia, de forma que durante su ejecución se comporta de forma dinámica, trabajando como analizador para TIG cuando analiza árboles TAG que cumplen las restricciones impuestas en el formalismo TIG o como analizador para TAG cuando analiza otro tipo de árboles. Al tratarse de analizadores para TAG, sus cotas de complejidad teórica permanecen en las estándar para las mismas. Sin embargo, veremos como dichas cotas se ven significativamente reducidas cuando los algoritmos analizan árboles con estructuras TIG.

La construcción de analizadores para TAG que funcionen de la forma descrita requiere la fusión de esquemas para TAG y TIG que empleen la misma estrategia de análisis. En (Alonso, Carrillo, y Díaz, 2002) se definió un analizador predictivo que hacía uso de esta técnica. Aquí presentamos un esquema con estrategia ascendente bidireccional, resultado de la integración del esquema para TAG definido en (Díaz, Carrillo, y Alonso, 2000) con el nuevo esquema para TIG que también presentamos en este trabajo.

El artículo se organiza de la siguiente forma. En la sección 2 se introducen los formalismo TAG, TIG y los esquemas de análisis sintáctico. En la sección 3 se muestra la definición del esquema ascendente bidireccional para TAG. En la sección 4 se define un esquema ascendente bidireccional para TIG. En la sección 5 se define el esquema para TAG resultado de la fusión de los esquemas de las secciones 3 y 4.

2 Fundamentos

Una TAG es una 5-tupla $(V_N, V_T, S, \boldsymbol{I}, \boldsymbol{A})$, donde V_N es un conjunto de símbolos no terminales, V_T es un conjunto de símbolos terminales, $S \in V_N$ es el axioma, $\boldsymbol{I}$ es un conjunto finito de *árboles iniciales* finitos y $\boldsymbol{A}$ es un conjunto finito de *árboles auxiliares* finitos. Al conjunto $\boldsymbol{I} \cup \boldsymbol{A}$ se le denomina *árboles elementales*. Nos referiremos a la raíz de un árbol elemental γ como $\mathbf{R}^\gamma$. En cada árbol elemental, los nodos de la frontera se etiquetan con símbolos terminales, la palabra vacía (ε) o símbolos no terminales marcados para sustitución, excepto un nodo en cada árbol auxiliar, cuya etiqueta es la misma que la de la raíz y que se denomina nodo *pie*. Denotaremos como $\mathbf{F}^\beta$ al nodo pie de un árbol auxiliar β. Denominamos *espina* al camino de la raíz al pie de un árbol auxiliar. Usaremos $label(M^\gamma)$ para denotar la etiqueta asociada al nodo M^γ.

Los árboles auxiliares en los cuales todo nodo frontera no vacío está a la izquierda (resp. derecha) del nodo pie se denominan *árboles auxiliares izquierdos (resp. derechos)*. El resto de árboles auxiliares se denominan *árboles wrapping*. Usaremos $\boldsymbol{A_L}$ y $\boldsymbol{A_R}$ para denotar los conjuntos de árboles auxiliares izquierdos y derechos, respectivamente.

Una derivación TAG comienza con un árbol inicial cuya raíz está etiquetada por S. Este árbol se extiende repetidamente usando las operaciones de *adjunción* y *sustitución*. La *adjunción* inserta un árbol auxiliar β en el nodo M^γ de un árbol γ que tenga la misma etiqueta que $\mathbf{R}^\beta$. En concreto, M^γ es reemplazado por β y $\mathbf{F}^\beta$ es reemplazado por el subárbol dominado por M^γ. Usaremos $\beta \in adj(M^\gamma)$ para denotar que un árbol $\beta \in \boldsymbol{A}$ puede ser adjuntado en un nodo M^γ, es decir, M^γ es un nodo de adjunción. Si la adjunción no es obligatoria en M^γ entonces $\mathbf{nil} \in adj(M^\gamma)$, donde **nil** es un símbolo vacío. La adjunción de un árbol auxiliar izquierdo (resp. derecho) se denomina *adjunción izquierda (resp. derecha)*. Usaremos $\beta \in \text{ladj}(M^\gamma)$ (resp. $\beta \in \text{radj}(M^\gamma)$) para denotar que $\beta \in \boldsymbol{A_L}$ ($\beta \in \boldsymbol{A_R}$) se puede adjuntar en el nodo M^γ, es decir, M^γ es un nodo de adjunción izquierda (resp. derecha). Si una adjunción izquierda (resp. derecha) no es obligatoria en el nodo M^γ entonces $\mathbf{nil} \in \text{ladj}(M^\gamma)$ ($\mathbf{nil} \in \text{radj}(M^\gamma)$). La *sustitución* es una operación obligatoria y reemplaza un nodo marcado para sustitución M^γ con una copia de un árbol inicial α cuya raíz esté etiquetada igual que M^γ. Usamos $\alpha \in \text{subst}(M^\gamma)$ para indicar que el nodo M^γ puede ser sustituido por el árbol $\alpha \in \boldsymbol{I}$.

Hasta este punto las definiciones de TIG y TAG son iguales. Sin embargo, las TIG no permiten: (1) árboles auxiliares *wrapping*, (2) la adjunción de un árbol auxiliar izquierdo (resp. derecho) en la espina de un árbol auxiliar derecho (resp. izquierdo), (3) la adjun-

ción en los nodos situados a la derecha (resp. izquierda) de la espina de los árboles auxiliares izquierdos (resp. derechos) y (4) la adjunción en los nodos raíz y pie de los árboles auxiliares. Para incrementar los árboles que se pueden generar, TIG permite un número arbitrario de adjunciones simultáneas sobre un mismo nodo.

Con objeto de representar los árboles de análisis parciales, definimos una producción $N^\gamma \rightarrow N_1^\gamma \dots N_g^\gamma$ para cada nodo N^γ y sus secuencia ordenada de g hijos $N_1^\gamma \dots N_g^\gamma$ en un árbol elemental. Denotaremos el conjunto de producciones asociado a un árbol elemental γ como $\mathcal{P}(\gamma)$. Por razones técnicas consideramos las producciones adicionales $\top \rightarrow \mathbf{R}^\alpha$, $\top \rightarrow \mathbf{R}^\beta$ y $\mathbf{F}^\beta \rightarrow \bot$ para cada árbol inicial α y cada árbol auxiliar β. Se prohíbe la adjunción y sustitución en los nodos $\top$ y $\bot$.

Los algoritmos de análisis sintáctico se pueden definir como sistemas deductivos (Shieber, Schabes, y Pereira, 1995), donde las fórmulas, llamadas ítems, son conjuntos de constituyentes completos o incompletos. Los *esquemas de análisis* fueron introducidos en (Sikkel, 1997) como un método de alto nivel para la descripción de algoritmos de análisis. Un sistema de análisis abstrae los detalles de implementación, como las estructuras de control y datos.

Formalmente, un *sistema de análisis* para una gramática G y una cadena $a_1 \dots a_n$ es un triple $< \mathcal{I}, \mathcal{H}, \mathcal{D} >$, donde $\mathcal{I}$ es un conjunto de *ítems* que representa resultados de análisis intermedios, $\mathcal{H}$ es el conjunto inicial de ítems llamado *hipótesis* que codifica la frase que se va a analizar, y $\mathcal{D}$ un conjunto de *pasos deductivos* que permite que se generen nuevos ítems a partir de los existentes. Los pasos deductivos son de la forma $\frac{\eta_1,\dots,\eta_k}{\xi}$ *cond*, que significa que si están presentes todos los antecedentes η_i de un paso deductivo y se satisface la condición *cond*, entonces el consecuente ξ debería ser generado por el analizador. El conjunto $\mathcal{F} \subseteq \mathcal{I}$ de *ítems finales* representa el reconocimiento de una frase. Un *esquema de análisis* es un sistema de análisis parametrizado para una gramática y una frase.

3 Esquema ascendente bidireccional para TAG

Definimos ahora el esquema **dVH** presentado en (Díaz, Carrillo, y Alonso, 2000). Se trata de un esquema con estrategia ascendente y recorrido bidireccional de la cadena de entrada que está basado en el algoritmo para gramáticas independientes del contexto definido por De Vreught y Honig para CFG. Al tratarse de un analizador ascendente sin ningún tipo de predicción, puede producir un mayor número de ítems durante el proceso de análisis, sin embargo, los estudios preliminares mostrados en (Díaz y Alonso, 2000) demuestran que su comportamiento en casos prácticos es similar al de analizadores que incorporan predicción. Las ventajas fundamentales que ofrece este esquema son dos: sus pasos deductivos requieren un máximo de dos antecedentes y permite obtener mayor información parcial para entradas incorrectas. Así, mientras lo primero disminuye el espacio de búsqueda del algoritmo y aumenta su velocidad, lo segundo hace de este algoritmo una opción adecuada para realizar análisis sintáctico fragmental.

Puesto que se trata de un algoritmo bidireccional, el reconocimiento puede comenzar en cualquier posición de las reglas y expandirse en ambas direcciones. Esto nos obliga a modificar la forma general de regla punteada usada en los esquemas de tipo Earley, de manera que introducimos un punto adicional para delimitar la parte reconocida dentro de una regla. Entonces el dominio del esquema **dVH** se define mediante ítems de la forma:

$$\mathcal{I}_{\text{dVH}} = \{[N^\gamma \rightarrow \nu \bullet \delta \bullet \omega, i, j \mid p, q]\}$$

donde $N^\gamma \rightarrow \nu\delta\omega \in \mathcal{P}(\gamma)$ y $\gamma \in \boldsymbol{I} \cup \boldsymbol{A}$. Los índices $0 \leq i \leq j$ establecen las posiciones de la cadena de entrada que delimitan el fragmento reconocido por δ. Si p y q presentan un valor definido, entonces N^γ pertenece a la espina de un árbol auxiliar, δ domina el nodo pie de γ y se cumple $(p, q) \leq (i, j)$.

Los pasos deductivos del esquema son:

$$\mathcal{D}_{\text{dVH}}^{\text{Scan}} = \frac{[a, j, j+1]}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, j, j+1 \mid -, -]} \; label(M^\gamma) = a$$

$$\mathcal{D}_{\text{dVH}}^{\varepsilon} = \frac{}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, j, j \mid -, -]} \; label(M^\gamma) = \varepsilon$$

$$\mathcal{D}_{\text{dVH}}^{\text{Comp}} = \frac{[M^\gamma \rightarrow \bullet\delta\bullet, i, j \mid p, q]}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, i, j \mid p, q]} \; \mathbf{nil} \in \text{adj}(M^\gamma)$$

$$\mathcal{D}_{\text{dVH}}^{\text{Con}} = \frac{[N^\gamma \rightarrow \nu \bullet \delta \bullet \delta'\omega, i, j' \mid p, q] \quad [N^\gamma \rightarrow \nu\delta \bullet \delta' \bullet \omega, j', j \mid p', q']}{[N^\gamma \rightarrow \nu \bullet \delta\delta' \bullet \omega, i, j \mid p \cup p', q \cup q']}$$

$$\mathcal{D}_{\text{dVH}}^{\text{Foot}} = \frac{}{[\mathbf{F}^\beta \rightarrow \bullet\bot\bullet, k, l \mid k, l]} \; \beta \in \text{adj}(M^\gamma)$$

$$\mathcal{D}_{\text{dVH}}^{\text{AdjComp}} = \frac{[\top \rightarrow \bullet\mathbf{R}^\beta\bullet, j, m \mid k, l] \quad [M^\gamma \rightarrow \bullet\delta\bullet, k, l \mid p, q]}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, j, m \mid p, q]} \; \beta \in \text{adj}(M^\gamma)$$

El reconocimiento ascendente se inicia con los pasos deductivos $\mathcal{D}^{\text{Scan}}_{\text{dVH}}$ y $\mathcal{D}^{\varepsilon}_{\text{dVH}}$, los cuales introducen en el análisis los subárboles que dominan directamente nodos etiquetados con símbolos terminales que coincida con algún símbolo de la cadena de entrada ($\mathcal{D}^{\text{Scan}}_{\text{dVH}}$) o nodos etiquetados con la palabra vacía ($\mathcal{D}^{\varepsilon}_{\text{dVH}}$).

El paso $\mathcal{D}^{\text{Comp}}_{\text{dVH}}$ continúa el reconocimiento del superárbol respecto a un nodo una vez que el subárbol dominado por él ha sido completamente reconocido, siempre que la adjunción no sea obligatoria en dicho nodo.

Cuando dos fragmentos adyacentes en una regla hayan reconocido segmentos colindantes en la cadena de entrada, el paso $\mathcal{D}^{\text{Con}}_{\text{dVH}}$ se encarga de concatenarlos.

El paso deductivo $\mathcal{D}^{\text{Foot}}_{\text{dVH}}$ lleva a cabo el reconocimiento del nodo pie de los árboles auxiliares entre todas las posibles posiciones de la cadena de entrada. Evidentemente, este paso se puede filtrar incluyendo un antecedente que refleje el hecho de que el subárbol que pende del nodo pie ya haya sido reconocido, sin embargo, no consideraremos esta mejora incluida en la definición original.

El paso deductivo $\mathcal{D}^{\text{AdjComp}}_{\text{dVH}}$ continúa el reconocimiento del superárbol respecto a M^{γ} donde se ha efectuado la adjunción del árbol auxiliar β, una vez que éste ha sido completamente reconocido.

El conjunto de ítems finales se define mediante:

$$\mathcal{F}_{\text{dVH}} = \{[\top \rightarrow \bullet \mathbf{R}^{\alpha} \bullet, 0, n \mid -, -] \mid \alpha \in \boldsymbol{I}\}$$

4 *Esquema ascendente bidireccional para TIG*

En esta sección definimos un nuevo esquema para TIG con estrategia ascendente y recorrido bidireccional de la cadena de entrada, al que denominaremos $\mathbf{dVH^i}$.

Este esquema presenta las siguientes características: (i) incorpora la operación de sustitución; (ii) para evitar la adjunción simultánea, tal como se indica en (Schabes y Waters, 1996), permite como máximo la adjunción de un árbol auxiliar derecho y otro izquierdo sobre un nodo y la adjunción en las raíces de los árboles auxiliares; (iii) para facilitar la integración con el esquema $\mathbf{dVH}$, no permitimos que se adjunte sobre un nodo hasta que el subárbol que domina haya sido completamente reconocido. La restricción (ii) implica la necesidad de llevar un control del número de adjunciones que se han efectuado en un nodo, mientras que la (iii) provoca que el dominio del esquema $\mathbf{buE^i}$ se divida en dos grupos de ítems: los que representan el reconocimiento parcial y los que representan el reconocimiento completo de una regla.

$$\mathcal{I}_{\text{dVH}^{\text{i}}} = \mathcal{I}^{(\text{i})}_{\text{dVH}^{\text{i}}} \cup \mathcal{I}^{(\text{ii})}_{\text{dVH}^{\text{i}}}$$

$$\mathcal{I}^{(\text{i})}_{\text{dVH}^{\text{i}}} = \{[M^{\gamma} \rightarrow \nu \bullet \delta \bullet \omega, i, j, \emptyset]\}$$

donde $M^{\gamma} \rightarrow \nu\delta\omega \in \mathcal{P}(\gamma)$ y $\gamma \in \boldsymbol{I} \cup \boldsymbol{A}$. Los índices $0 \leq i \leq j$ establecen las posiciones de la cadena de entrada que delimitan el fragmento reconocido por δ. Para evitar el reconocimiento completo de la regla se requiere que $\nu \neq \varepsilon$ o $\omega \neq \varepsilon$. El valor $\emptyset$ indica que no se ha completado ninguna adjunción en el nodo M^{γ}.

$$\mathcal{I}^{(\text{ii})}_{\text{dVH}^{\text{i}}} = \{[M^{\gamma} \rightarrow \bullet\delta\bullet, i, j, \mathit{code}]\}$$

donde $M^{\gamma} \rightarrow \delta \in \mathcal{P}(\gamma)$ y $\gamma \in \boldsymbol{I} \cup \boldsymbol{A}$. Los índices $0 \leq i \leq j$ establecen las posiciones de la cadena de entrada que delimitan el fragmento reconocido por δ. Y $\mathit{code} = \emptyset$ si no se completó ninguna adjunción sobre M^{γ}, $\mathit{code} = \{L\}$ si se completó una adjunción izquierda sobre M^{γ}, $\mathit{code} = \{R\}$ si se completó una adjunción derecha sobre M^{γ} y $\mathit{code} = \{L, R\}$ si se completó una adjunción izquierda y otra derecha sobre M^{γ}.

Los pasos deductivos son los siguientes:

$$\mathcal{D}^{\text{Scan}}_{\text{dVH}^{\text{i}}} = \frac{[a, j, j+1]}{[N^{\gamma} \rightarrow \nu \bullet M^{\gamma} \bullet \omega, j, j+1, \emptyset]} \; \mathit{label}(M^{\gamma}) = a$$

$$\mathcal{D}^{\varepsilon}_{\text{dVH}^{\text{i}}} = \frac{}{[N^{\gamma} \rightarrow \nu \bullet M^{\gamma} \bullet \omega, j, j, \emptyset]} \; \begin{array}{l} \mathit{label}(M^{\gamma}) = \varepsilon \text{ o} \\ \mathit{label}(M^{\gamma}) = \bot \end{array}$$

$$\mathcal{D}^{\text{Comp}}_{\text{dVH}^{\text{i}}} = \frac{[M^{\gamma} \rightarrow \bullet\delta\bullet, i, j, \mathit{code}]}{[N^{\gamma} \rightarrow \nu \bullet M^{\gamma} \bullet \omega, i, j, \emptyset]}$$

donde se debe cumplir:
(i) ($\mathbf{nil} \in \text{ladj}(M^{\gamma})$ y $L \notin \mathit{code}$) o
($\beta \in \text{ladj}(M^{\gamma})$ y $L \in \mathit{code}$),
(ii) ($\mathbf{nil} \in \text{radj}(M^{\gamma})$ y $R \notin \mathit{code}$) o
($\beta \in \text{radj}(M^{\gamma})$ y $R \in \mathit{code}$).

$$\mathcal{D}^{\text{Con}}_{\text{dVH}^{\text{i}}} = \frac{\begin{array}{c}[N^{\gamma} \rightarrow \nu \bullet \delta \bullet \delta'\omega, i, j', \emptyset] \\ [N^{\gamma} \rightarrow \nu\delta \bullet \delta' \bullet \omega, j', j, \emptyset]\end{array}}{[N^{\gamma} \rightarrow \nu \bullet \delta\delta' \bullet \omega, i, j, \emptyset]}$$

$$\mathcal{D}^{\text{LAdj}}_{\text{dVH}^{\text{i}}} = \frac{\begin{array}{c}[\top \rightarrow \bullet\mathbf{R}^{\beta}\bullet, i, j, \emptyset] \\ [M^{\gamma} \rightarrow \bullet\delta\bullet, j, k, \mathit{code}]\end{array}}{[M^{\gamma} \rightarrow \bullet\delta\bullet, i, k, \{L\} \cup \mathit{code}]} \; \begin{array}{l} \beta \in \text{ladj}(M^{\gamma}) \\ L \notin \mathit{code} \end{array}$$

$$\mathcal{D}^{\text{RAdj}}_{\text{dVH}^{\text{i}}} = \frac{\begin{array}{c}[\top \rightarrow \bullet\mathbf{R}^{\beta}\bullet, j, k, \emptyset] \\ [M^{\gamma} \rightarrow \bullet\delta\bullet, i, j, \mathit{code}]\end{array}}{[M^{\gamma} \rightarrow \bullet\delta\bullet, i, k, \{R\} \cup \mathit{code}]} \; \begin{array}{l} \beta \in \text{radj}(M^{\gamma}) \\ R \notin \mathit{code} \end{array}$$

$$\mathcal{D}^{\text{Subs}}_{\text{dVH}^{\text{i}}} = \frac{[\top \rightarrow \bullet\mathbf{R}^{\alpha}\bullet, j, k, \emptyset]}{[N^{\gamma} \rightarrow \delta \bullet M^{\gamma} \bullet \nu, j, k, \emptyset]} \; \alpha \in \text{subst}(M^{\gamma})$$

Los pasos deductivos $\mathcal{D}^{\text{Scan}}_{\text{dVH}^{\text{i}}}$ y $\mathcal{D}^{\varepsilon}_{\text{dVH}^{\text{i}}}$ son los que inician el reconocimiento ascendente desde los nodos etiquetados con símbolos terminales que coincida con algún símbolo de la

cadena de entrada y nodos etiquetados con la palabra vacía o $\perp$, respectivamente.

La función del paso $\mathcal{D}^{\text{Comp}}_{\text{dVH}^{\text{i}}}$ es continuar el reconocimiento del superárbol respecto a un nodo una vez que el subárbol dominado por él ha sido completamente reconocido. Se acompaña una condición lateral a este paso para garantizar que el nodo que domina el subárbol no presente adjunción obligatoria y, en el caso de presentarla, que se haya efectuado.

El paso $\mathcal{D}^{\text{Con}}_{\text{dVH}^{\text{i}}}$ concatena dos fragmentos adyacentes en una regla que reconocen segmentos colindantes en la cadena de entrada. Obsérvese como en ambos antecedentes el componente *adj* se encuentra a $false$, ésto es debido a que ninguno de ellos representa el reconocimiento completo de un subárbol dominado por un nodo, y tal como se establece en la definición del dominio de este esquema, ésta es una condición necesaria para completar una adjunción en un nodo.

Cuando se ha reconocido un árbol auxiliar izquierdo (resp. derecho), el paso $\mathcal{D}^{\text{LAdj}}_{\text{dVH}^{\text{i}}}$ (resp. $\mathcal{D}^{\text{RAdj}}_{\text{dVH}^{\text{i}}}$) efectúa la adjunción en un nodo de adjunción izquierda (resp. derecha), siempre que el reconocimiento lo haya alcanzado y no haya sido ya adjuntado por la izquierda (resp. derecha).

La función del paso $\mathcal{D}^{\text{Subs}}_{\text{dVH}^{\text{i}}}$ es continuar el reconocimiento del superárbol respecto a un nodo de sustitución una vez completado el análisis de un árbol inicial que se pueda sustituir en él.

El conjunto de ítems finales es:

$$\mathcal{F}_{\text{dVH}^{\text{i}}} = \{[\top \rightarrow \bullet \mathbf{R}^{\alpha} \bullet, 0, n, \emptyset] \mid \alpha \in \boldsymbol{I}, label(\mathbf{R}^{\alpha}) = S\}$$

5 *Integrando los esquemas para TIG y TAG*

Nuestro objetivo es integrar la funcionalidad de los esquemas presentados para TIG con sus homónimos para TAG. Recordemos las diferencias fundamentales entre TIG y TAG para tomar las decisiones oportunas de cara a realizar la mencionada integración.

Desde el punto de vista de los árboles elementales, los árboles TIG son un subconjunto de los árboles TAG que cumplen una serie de restricciones. Cuando definimos el formalismo TAG dividimos los árboles elementales en tres grupos: iniciales ($\boldsymbol{I}$), auxiliares izquierdos ($\boldsymbol{A_L}$) y auxiliares derechos ($\boldsymbol{A_R}$), puesto que la definición no permitía otro tipo de estructura distinta a la de éstos. Sin embargo, el formalismo TAG también permite árboles auxiliares wrapping, incluyendo en este grupo tanto los árboles auxiliares con nodos frontera no vacíos a la derecha e izquierda del nodo pie, como aquellos árboles que no cumplen las restricciones de adjunción impuestas en la definición de TIG, aunque presenten estructura de árbol auxiliar izquierdo o derecho.

Puesto que nuestro objetivo es que el análisis varíe de forma dinámica según se aplique sobre árboles TIG o TAG, debemos identificar dentro del conjunto de árboles elementales de cualquier gramática TAG sobre la que se va aplicar este tipo de analizadores cual es el núcleo de árboles que responden a la definición de TIG. Para ello dividimos el conjunto de árboles elementales en cuatro grupos disjuntos: iniciales ($\boldsymbol{I}$), *auxiliares fuertemente izquierdos* ($\boldsymbol{A_{SL}}$), *auxiliares fuertemente derechos*($\boldsymbol{A_{SR}}$) y auxiliares wrapping ($\boldsymbol{A_W}$). Un árbol auxiliar es fuertemente izquierdo (resp. derecho) si todos sus nodos frontera no vacíos se encuentran a la izquierda (resp. derecha) de su nodo pie y prohíbe: (1) la adjunción en su nodo pie, (2) la adjunción en los nodos situados a la derecha (resp. izquierda) de la espina, y (3) la adjunción de árboles fuertemente derechos (resp. izquierdos) y árboles auxiliares wrapping en su espina. Los árboles auxiliares de wrapping son aquellos que no se encuentran en el conjunto $\boldsymbol{A_{SL}} \cup \boldsymbol{A_{SR}}$.

Estos cambios nos obligan a incluir alguna notación nueva en las TAG. El conjunto de árboles auxiliares de una gramática TAG será $\boldsymbol{A} = \boldsymbol{A_{SL}} \cup \boldsymbol{A_{SR}} \cup \boldsymbol{A_W}$. Si un árbol auxiliar β es un árbol auxiliar fuertemente izquierdo, entonces $\beta \in \boldsymbol{A_{SL}}$. Si un árbol auxiliar β es un árbol auxiliar fuertemente derecho, entonces $\beta \in \boldsymbol{A_{SR}}$. Si un árbol auxiliar fuertemente izquierdo $\beta \in \boldsymbol{A_{SL}}$ es adjuntable en un nodo M^{γ} lo denotamos como $\beta \in \text{ladj}(M^{\gamma})$. Si un árbol auxiliar fuertemente derecho $\beta \in \boldsymbol{A_{SR}}$ es adjuntable en un nodo M^{γ} lo denotamos como $\beta \in \text{radj}(M^{\gamma})$. Si un árbol auxiliar wrapping $\beta \in \boldsymbol{A_W}$ es adjuntable en un nodo M^{γ} lo denotamos como $\beta \in \text{wadj}(M^{\gamma})$. Denominamos *adjunción izquierda* (resp. *derecha*) a la adjunción de un árbol auxiliar fuertemente izquierdo (resp. derecho) en un nodo adjuntable. Denominamos *adjunción wrapping* a la adjunción de un árbol auxiliar wrapping en un nodo adjuntable. Cuando la adjunción izquierda (resp. derecha) no es obligatoria en un nodo

M^γ lo denotamos como **nil** $\in$ ladj(M^γ) (resp. **nil** $\in$ radj(M^γ)). Cuando la adjunción wrapping no es obligatoria en un nodo M^γ lo denotamos como **nil** $\in$ wadj(M^γ). Cuando un nodo M^γ no presenta adjunciones izquierda, derecha y wrapping obligatorias (**nil** $\in$ ladj(M^γ),**nil** $\in$ radj(M^γ), **nil** $\in$ wadj(M^γ)) lo denotamos como **nil** $\in$ adj(M^γ).

Respecto a la operación de composición básica, mientras las TAG sólo permiten una adjunción sobre un nodo, la versión de TIG que hemos empleado en este trabajo permite la adjunción de un árbol auxiliar izquierdo y otro derecho sobre un nodo. En este punto se plantean dos alternativas: limitamos a una adjunción en un nodo como obliga el formalismo TAG, o permitimos varias adjunciones como propone el formalismo TIG. Hemos elegido la primera opción fundamentalmente porque no podemos olvidar que nuestro objetivo es diseñar un analizador eficiente para TAG y, en la medida de lo posible, debemos adoptar las restricciones de las mismas. Obviamente esta decisión va a provocar ciertas limitaciones a la hora de definir ciertas formaciones lingüísticas con árboles elementales de tipo TIG.

En las definición del analizador para TIG incluimos la operación de sustitución, sin embargo, por claridad, obviamos dicha operación en el algoritmo para TAG. Dado que la sustitución no modifica la potencia de ninguno de ambos formalismos y puesto que ya la hemos introducido en los esquemas para TIG, vamos a incluirla en el analizador que presentamos en esta sección.

El esquema para TAG $\mathbf{dVH^{Mix}}$, que definiremos a continuación, es el resultado de la fusión de los esquemas $\mathbf{dVH^{i}}$ y **dVH**. Para conseguir un analizador más eficiente que el ya definido para TAG, observemos el paso de adjunción del algoritmo para TAG $\mathcal{D}^{\text{Adj}}_{\text{dVH}}$. Este paso presenta una complejidad temporal de $\mathcal{O}(n^6)$, debido a que en sus antecedentes se incorpora la información referente al fragmento de la cadena de entrada reconocido por el subárbol que pende del nodo pie del árbol auxiliar que se va a adjuntar, concretamente los índices k y l. Sin embargo, cuando se trata de árboles auxiliares fuertemente izquierdos o derechos no es necesario transmitir dicha información. Teniendo ésto en cuenta, el esquema $\mathbf{dVH^{Mix}}$ toma como base el algoritmo para TAG, pero el paso de adjunción lo divide en tres, uno para cada tipo de árbol auxiliar. De esta forma disminuye la complejidad teórica cuando realiza adjunciones izquierdas o derechas y, por tanto, reduce la complejidad teórica media.

Al igual que en los dos esquemas ascendentes anteriores, la adjunción de un árbol auxiliar sobre un nodo no se puede llevar a cabo hasta que no se complete el reconocimiento del subárbol que pende del nodo pie. Esta consideración hace que el dominio del nuevo esquema se pueda considerar como el resultado de la adición de los dos índices correspondientes a la información del pie y la redefinición del parámetro adicional para limitar a una sola adjunción en cada nodo. El conjunto de ítems del nuevo esquema $\mathbf{dVH^{Mix}}$ es:

$$\mathcal{I}_{\text{dVH}^{\text{Mix}}} = \mathcal{I}^{(\text{i})}_{\text{dVH}^{\text{Mix}}} \cup \mathcal{I}^{(\text{ii})}_{\text{dVH}^{\text{Mix}}}$$

$$\mathcal{I}^{(\text{i})}_{\text{dVH}^{\text{Mix}}} = \{[M^\gamma \rightarrow \nu \bullet \delta \bullet \omega, i, j \mid p, q \mid false]$$

donde $M^\gamma \rightarrow \nu\delta\omega \in \mathcal{P}(\gamma)$, $\gamma \in \boldsymbol{I} \cup \boldsymbol{A}$, $\nu \neq \varepsilon$ o $\omega \neq \varepsilon$. Los índices $0 \leq i \leq j$ establecen las posiciones dentro de la cadena de entrada que delimitan el fragmento reconocido por δ. Si p y q presentan un valor definido, entonces M^γ es un nodo de la espina de un árbol auxiliar wrapping ($\gamma \in \boldsymbol{A_W}$) y se cumple $(p, q) \leq (i, j)$. El último parámetro $false$ indica que no se ha completado ninguna adjunción sobre M^γ.

$$\mathcal{I}^{(\text{ii})}_{\text{dVH}^{\text{Mix}}} = \{[M^\gamma \rightarrow \bullet\delta\bullet, i, j \mid p, q \mid adj]$$

donde $M^\gamma \rightarrow \delta \in \mathcal{P}(\gamma)$, siendo $\gamma \in \boldsymbol{I} \cup \boldsymbol{A}$. Los índices $0 \leq i \leq j$ establecen las posiciones dentro de la cadena de entrada que delimitan el fragmento reconocido por δ. Si p y q presentan un valor definido, entonces M^γ es un nodo de la espina de un árbol auxiliar wrapping ($\gamma \in \boldsymbol{A_W}$) y se cumple $(p, q) \leq (i, j)$. El parámetro booleano $adj \in \{true, false\}$ indica si se ha completado alguna adjunción (izquierda, derecha o wrapping) sobre M^γ.

El conjunto de pasos deductivos viene dado por:

$$\mathcal{D}^{\text{Scan}}_{\text{dVH}^{\text{Mix}}} = \frac{[a, j, j+1]}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, j, j+1 \mid -, - \mid false]}$$

donde $label(M^\gamma) = a$

$$\mathcal{D}^{\varepsilon}_{\text{dVH}^{\text{Mix}}} = \frac{}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, j, j \mid -, - \mid false]}$$

donde $label(M^\gamma) = \varepsilon$ o ($label(M^\gamma) = \perp$ y $\gamma \in \boldsymbol{A_{SL}} \cup \boldsymbol{A_{SR}}$)

$$\mathcal{D}^{\text{Comp}}_{\text{dVH}^{\text{Mix}}} = \frac{[M^\gamma \rightarrow \bullet\delta\bullet, i, j \mid p, q \mid adj]}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, i, j \mid p, q \mid false]}$$

donde se debe cumplir que ($\mathbf{nil} \in \text{adj}(M^\gamma)$ y $adj = false$) o ($\exists \beta \in \text{adj}(M^\gamma)$ y $adj \neq false$).

$$\mathcal{D}^{\text{Con}}_{\text{dVH}^{\text{Mix}}} = \frac{[N^\gamma \rightarrow \nu \bullet \delta \bullet \delta' \omega, i, j' \mid p, q \mid false] \quad [N^\gamma \rightarrow \nu\delta \bullet \delta' \bullet \omega, j', j \mid p', q' \mid false]}{[N^\gamma \rightarrow \nu \bullet \delta\delta' \bullet \omega, i, j \mid p \cup p', q \cup q' \mid false]}$$

$$\mathcal{D}^{\text{Foot}}_{\text{dVH}^{\text{Mix}}} = \frac{}{[\mathbf{F}^\beta \rightarrow \bullet \bot \bullet, k, l \mid k, l \mid false]} \quad \beta \in \boldsymbol{A_W}$$

$$\mathcal{D}^{\text{LAdj}}_{\text{dVH}^{\text{Mix}}} = \frac{[\top \rightarrow \bullet \mathbf{R}^\beta \bullet, i, j \mid -, - \mid false] \quad [M^\gamma \rightarrow \bullet\delta\bullet, j, k \mid p, q \mid false]}{[M^\gamma \rightarrow \bullet\delta\bullet, i, k \mid p, q \mid true]}$$

donde $\beta \in \text{ladj}(M^\gamma)$

$$\mathcal{D}^{\text{RAdj}}_{\text{dVH}^{\text{Mix}}} = \frac{[\top \rightarrow \bullet \mathbf{R}^\beta \bullet, j, k \mid -, - \mid false] \quad [M^\gamma \rightarrow \bullet\delta\bullet, i, j \mid p, q \mid false]}{[M^\gamma \rightarrow \bullet\delta\bullet, i, k \mid p, q \mid true]}$$

donde $\beta \in \text{radj}(M^\gamma)$

$$\mathcal{D}^{\text{WAdj}}_{\text{dVH}^{\text{Mix}}} = \frac{[\top \rightarrow \bullet \mathbf{R}^\beta \bullet, j, m \mid k, l \mid false] \quad [M^\gamma \rightarrow \bullet\delta\bullet, k, l \mid p, q \mid false]}{[M^\gamma \rightarrow \bullet\delta\bullet, j, m \mid p, q \mid true]}$$

donde $\beta \in \text{wadj}(M^\gamma)$

$$\mathcal{D}^{\text{Subs}}_{\text{dVH}^{\text{Mix}}} = \frac{[\top \rightarrow \bullet \mathbf{R}^\alpha \bullet, j, k \mid -, - \mid false]}{[N^\gamma \rightarrow \nu \bullet M^\gamma \bullet \omega, j, k \mid -, - \mid false]}$$

donde $\alpha \in \text{subst}(M^\gamma)$.

Los pasos deductivos que inician el reconocimiento ascendente son $\mathcal{D}^{\text{Scan}}_{\text{dVH}^{\text{Mix}}}$, $\mathcal{D}^{\varepsilon}_{\text{dVH}^{\text{Mix}}}$ y $\mathcal{D}^{\text{Foot}}_{\text{dVH}^{\text{Mix}}}$. Los pasos $\mathcal{D}^{\text{Scan}}_{\text{dVH}^{\text{Mix}}}$ y $\mathcal{D}^{\varepsilon}_{\text{dVH}^{\text{Mix}}}$ inician el reconocimiento desde los nodos etiquetados con símbolos terminales, la palabra vacía o el nodo pie de los árboles auxiliares en $\boldsymbol{A_{SL}} \cup \boldsymbol{A_{SR}}$. El paso $\mathcal{D}^{\text{Foot}}_{\text{dVH}^{\text{Mix}}}$ completa el reconocimiento de los subárboles que dominan los nodos pie de los árboles auxiliares en $\boldsymbol{A_W}$ entre todas las posibles posiciones de la cadena de entrada.

Una vez completado el reconocimiento de un subárbol el paso deductivo $\mathcal{D}^{\text{Comp}}_{\text{dVH}^{\text{Mix}}}$ continúa el reconocimiento del superárbol. Si dicho subárbol domina el pie de un árbol en $\boldsymbol{A_W}$, entonces se transmite la información del segmento que cuelga del mismo en los índices p y q. Se acompaña una condición lateral a este paso para garantizar que el nodo que domina el subárbol no presente adjunción (izquierda, derecha o wrapping) obligatoria y, en el caso de presentarla, que se haya efectuado.

Cuando el reconocimiento completa dos fragmentos adyacentes en una regla que dominan segmentos colindantes en la cadena de entrada, el analizador los concatena mediante el paso $\mathcal{D}^{\text{Con}}_{\text{dVH}^{\text{Mix}}}$. Si alguno de los dos fragmentos domina el nodo pie de un árbol auxiliar en $\boldsymbol{A_W}$, entonces el antecedente que representa el reconocimiento de dicho fragmento tendrá en sus índices la información del segmento de la cadena de entrada que cuelga del nodo pie.

Los pasos de adjunción se agrupan en tres tipos: $\mathcal{D}^{\text{LAdj}}_{\text{dVH}^{\text{Mix}}}$ para adjunciones de árboles auxiliares izquierdos, $\mathcal{D}^{\text{RAdj}}_{\text{dVH}^{\text{Mix}}}$ para adjunciones de árboles auxiliares derechos y $\mathcal{D}^{\text{WAdj}}_{\text{dVH}^{\text{Mix}}}$ para adjunciones de árboles auxiliares wrapping.

El conjunto de ítems finales se define mediante:

$$\mathcal{F}_{\text{dVH}^{\text{Mix}}} = \{[\top \rightarrow \bullet \mathbf{R}^\alpha \bullet, 0, n \mid -, - \mid false] \mid \alpha \in \boldsymbol{I}, label(\mathbf{R}^\alpha) = S\}$$

Veamos ahora las mejoras que aporta este analizador respecto complejidad teórica media. Los árboles en $\boldsymbol{A_W}$ se analizan con un coste espacial de $\mathcal{O}(n^4)$ en el caso peor y se reduce a $\mathcal{O}(n^2)$ en el resto. La complejidad temporal en el caso peor de:

- Las adjunciones de árboles auxiliares fuertemente izquierdos o derechos en:
  - un nodo de un árbol en $\boldsymbol{I} \cup \boldsymbol{A_{SL}} \cup \boldsymbol{A_{SR}}$ o un nodo que no está en la espina de un árbol en $\boldsymbol{A_W}$ presentan una complejidad $\mathcal{O}(n^3)$, debido a los pasos $\mathcal{D}^{\text{LAdj}}_{\text{dVH}^{\text{Mix}}}$ y $\mathcal{D}^{\text{RAdj}}_{\text{dVH}^{\text{Mix}}}$.
  - un nodo que se encuentre en la espina de un árbol en $\boldsymbol{A_W}$ presentan una complejidad $\mathcal{O}(n^5)$, debido a los pasos $\mathcal{D}^{\text{LAdj}}_{\text{dVH}^{\text{Mix}}}$ y $\mathcal{D}^{\text{RAdj}}_{\text{dVH}^{\text{Mix}}}$.
- Las adjunciones de árboles auxiliares wrapping en:
  - un nodo de un árbol en $\boldsymbol{I} \cup \boldsymbol{A_{SL}} \cup \boldsymbol{A_{SR}}$ o un nodo que no se encuentre en la espina de un árbol auxiliar en $\boldsymbol{A_W}$ presentan una complejidad $\mathcal{O}(n^4)$, debido al paso $\mathcal{D}^{\text{WAdj}}_{\text{dVH}^{\text{Mix}}}$.
  - un nodo de la espina de un árbol auxiliar en $\boldsymbol{A_W}$ presentan la complejidad máxima del algoritmo de $\mathcal{O}(n^6)$, debido al paso $\mathcal{D}^{\text{WAdj}}_{\text{dVH}^{\text{Mix}}}$.
- Las operaciones de sustitución se realizan en tiempo $\mathcal{O}(n^2)$.
- Las operaciones de reconocimiento, compleción y concatenación permanecen en las mismas cotas de complejidad del esquema **dVH**. Aunque las operaciones de compleción en nodos pertenecientes a las espinas de árboles auxiliares en $\boldsymbol{A_{SL}} \cup \boldsymbol{A_{SR}}$ se reducen a $\mathcal{O}(n^2)$ y las operaciones de concatenación en árboles

auxiliares en $\boldsymbol{A_{SL}} \cup \boldsymbol{A_{SR}}$ permanecen en todos los casos a casos en $\mathcal{O}(n^3)$.

6 Conclusión

Hemos presentado un analizador ascendente bidireccional para TAG más eficiente que el ya definido en la literatura. Este analizador se comporta de forma dinámica, sacando partido del hecho de que la mayor parte de las gramáticas de amplia cobertura responden a las restricciones que impone el formalismo TIG. Este esquema mixto se ha construido fusionando el esquema para TAG ya descrito con el nuevo esquema ascendente bidireccional para TIG que presentamos en este trabajo. Finalmente hemos hecho un estudio detallado de la complejidad teórica que aporta cada paso deductivo en el nuevo esquema combinado, poniendo de manifiesto los casos en los que se producen mejoras de comportamiento respecto al esquema original para TAG.

Como trabajos futuros se pueden realizar estudios empíricos para corroborar los resultados teóricos aquí descritos o definir nuevos esquemas obtenidos mediante la aplicación de esta misma técnica y establecer la relaciones formales entre ellos.

Bibliografía

Alonso, M. A., V. Carrillo, y V. J. Díaz, 2002. *Advances in Artificial Intelligence. IBERAMIA 2002*, volumen 2527 of Lecture Notes in Artificial Intelligence, capítulo Mixed Parsing of Tree Insertion and Tree Adjoining Grammars, páginas 694–703. Springer-Verlag.

Alonso, M.A., D. Cabrero, E. de la Clergerie, y M. Vilares. 1999. Tabular algorithms for TAG parsing. En *Proc. of Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, páginas 150–157, Bergen, Norway.

Carrillo, V., V. J. Díaz, y M. A. Alonso. 2002. Algoritmos de análisis para gramáticas de inserción de árboles. *Procesamiento del lenguaje Natural*, 29:89–96.

Christy, D., D. Egedi, B. Hockey, B. Srinivas, y M. Zaidel. 1994. XTAG system — a wide coverage grammar for English. En *Proc. of the 15th International Conference on Computational Linguistics (COLING'94)*, páginas 922–928, Kyoto, Japón.

Díaz, V. J. y M. A. Alonso. 2000. Comparing tabular parsers for tree adjoining grammars. En *Proc. of 2nd International Workshop on Tabulation in Parsing and Deduction (TAPD'2000)*, Vigo, Spain.

Díaz, V. J., V. Carrillo, y M. A. Alonso. 2000. A bidirectional bottom-up parser for tag. En *Proc. of 6th International Workshop on Parsing Technologies (IWPT'2000)*, Trento, Italy.

de Vreught, J. P. M y H. J. Honig. 1989. A tabular bottom-up recognizer. Informe Técnico 89-78, Departament of Applied Mathematics and Informatics. Delft University of Technologies.

Joshi, A.K. y Y. Schabes, 1997. *Handbook of Formal Languages*, volumen 3, capítulo Tree-adjoining grammars, páginas 69–123. G. Rozenberg and A. Salomaa.

Rogers, J. 1994. Capturing cfls with tree adjoining grammars. En *Proc. of 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94)*, páginas 150–157, Las Cruces, México.

Schabes, Y. y R.C. Waters. 1995. Tree insertion grammar: A cubic-time parsable formalism that lexicalizes context-free grammar without changing the trees produced. *Computational Linguistics*, 21(4):479–513.

Schabes, Y. y R.C. Waters, 1996. *Recent advances in parsing technologies*, volumen 1, capítulo 15. Sthocastic lexicalized tree insertion grammars, páginas 69–123. Kluwer Academic Publishers.

Shieber, S.M., Y. Schabes, y F.C.N. Pereira. 1995. Principles and implementation of deductive parsing. *Journal of Logic Programming*, 24(1–2):3–36.

Sikkel, K. 1997. *Parsing schemata — A framework for specification and analysis of parsing algorithms.* Springer-Verlag, Berlin/Heidelberg/New York.