



UNIVERSIDADE DA CORUÑA
Departamento de Computación

TESIS DOCTORAL

**INTERPRETACIÓN TABULAR DE AUTÓMATAS PARA
LENGUAJES DE ADJUNCIÓN DE ÁRBOLES**

Autor: MIGUEL A. ALONSO PARDO

Directores: MANUEL VILARES FERRO
ERIC VILLEMONTÉ DE LA CLERGERIE

25 de septiembre de 2000

Dr. Manuel Vilares Ferro
Profesor Titular de Universidad
Departamento de Computación
Universidad de La Coruña
España

Dr. Eric Villemonte de la Clergerie
Chargé de Recherche
INRIA
Francia

CERTIFICAN:

Que la memoria titulada “Interpretación tabular de autómatas para lenguajes de adjunción de árboles” ha sido realizada por D. Miguel A. Alonso Pardo bajo nuestra dirección en el Departamento de Computación de la Universidad de La Coruña y concluye la Tesis que presenta para optar al grado de Doctor.

La Coruña, 31 de marzo de 2000

Dr. Manuel Vilares Ferro
Director de la Tesis Doctoral

Dr. Eric Villemonte de la Clergerie
Director de la Tesis Doctoral

Tesis Doctoral: INTERPRETACIÓN TABULAR DE AUTÓMATAS PARA LENGUAJES DE ADJUNCIÓN DE ÁRBOLES

Autor: D. Miguel A. Alonso Pardo

Directores: Dr. Manuel Vilares Ferro
Dr. Eric Villemonte de la Clergerie

Fecha: 25 de septiembre de 2000

Tribunal

Presidente: Josep Miró (Universitat de les Illes Balears, España)

Vocal 1º: José Mira Mira (UNED, España)

Vocal 2º: Pierre Boullier (INRIA, Francia)

Vocal 3º: Mark-Jan Nederhof (DFKI, Alemania)

Secretario: Antonio Blanco Ferro (Universidade da Coruña, España)

Calificación: Sobresaliente Cum Laude

A Adriana

Agradecimientos

Deseo expresar mi gratitud a todos los que han contribuido a crear un agradable ambiente de trabajo para el desarrollo de esta tesis, tanto en la Facultad de Informática de La Coruña como en el centro de investigación del INRIA¹ en Rocquencourt, Francia. En La Coruña, destacar la ayuda prestada por los miembros que a lo largo del tiempo ha tenido el grupo de investigación COLE², con mención especial a David Cabrero, Jorge Graña y Manuel Vilares, paciente director de esta tesis. En el INRIA, destacar la paciencia de mi compañero de despacho, Frédéric Tendeau, y la colaboración del resto de componentes del proyecto Atoll³, con mención especial a Pierre Boullier, por sus valiosos comentarios, Eric de la Clergerie, por haber aceptado codirigir esta tesis, y Bernard Lang, por haber aceptado ser el director de mi estancia.

Fructíferas han sido también las discusiones con Tilman Becker, Vicente Carrillo, Víctor Díaz, Patrice Lopez, Mark-Jan Nederhof y Giorgio Satta, así como los comentarios aportados por numerosos revisores anónimos de las publicaciones resultantes del trabajo realizado en esta tesis.

Especialmente importante ha sido el apoyo prestado por Adriana Dapena durante todo este tiempo.

Agradezco también a la Xunta de Galicia la financiación de parte de la investigación mediante una beca predoctoral, a Caixa Galicia la financiación de mi estancia en Francia y a la Universidade da Coruña la financiación parcial de los desplazamientos a congresos.

El proyecto de investigación en el que se enmarca la realización de esta tesis ha sido financiado, en el periodo en que he participado, en sucesivas fases por los fondos FEDER de la Unión Europea (proyecto 1FD97-0047-C04-02), la Xunta de Galicia (proyectos XUGA10501B93, XUGA20403B95, XUGA10505B96, XUGA20402B97, PGIDT99XI10502B), los Ministerio de Educación y Cultura y Asuntos Exteriores (acciones integradas HF 96-36 y HF 97-223) y el Centro Ramón Piñeiro para a Investigación en Humanidades.

¹Institut National de Recherche en Informatique et en Automatique (<http://www.inria.fr/>).

²Compiladores y Lenguajes (<http://coleweb.dc.fi.udc.es/>).

³Atelier d'Outils Logiciels pour le Langage Naturel (<http://atoll.inria.fr/>).

Resumen

Las gramáticas de adjunción de árboles son una extensión de las gramáticas independientes del contexto que utilizan árboles en vez de producciones como estructuras elementales y que resultan adecuadas para la descripción de la mayor parte de las construcciones sintácticas presentes en el lenguaje natural. Los lenguajes generados por esta clase de gramáticas se denominan lenguajes de adjunción de árboles y son equivalentes a los lenguajes generados por las gramáticas lineales de índices y otros formalismos suavemente dependientes del contexto.

En la primera parte de esta memoria se presenta el problema del análisis sintáctico de los lenguajes de adjunción de árboles. Para ello, se establece un camino evolutivo continuo en el que se sitúan los algoritmos de análisis sintáctico que incorporan las estrategias de análisis más importantes, tanto para el caso de las gramáticas de adjunción de árboles como para el caso de las gramáticas lineales de índices.

En la segunda parte se definen diferentes modelos de autómatas que aceptan exactamente los lenguajes de adjunción de árboles y se proponen técnicas que permiten su ejecución eficiente. La utilización de autómatas para realizar el análisis sintáctico es interesante porque permite separar el problema de la definición de un algoritmo de análisis sintáctico del problema de la ejecución del mismo, al tiempo que simplifica las pruebas de corrección. Concretamente, hemos estudiado los siguientes modelos de autómatas:

- Los autómatas a pila embebidos descendentes y ascendentes, dos extensiones de los autómatas a pila que utilizan como estructura de almacenamiento una pila de pilas. Hemos definido nuevas versiones de estos autómatas en las cuales se simplifica la forma de las transiciones y se elimina el control de estado finito, manteniendo la potencia expresiva.
- La restricción de los autómatas lógicos a pila para adaptarlos al reconocimiento de las gramáticas lineales de índices, obteniéndose diferentes tipos de autómatas especializados en diversas estrategias de análisis según el conjunto de transiciones permitido.
- Los autómatas lineales de índices, tanto los orientados a la derecha, adecuados para estrategias en las cuales las adjunciones se reconocen de manera ascendente, los orientados a la izquierda, aptos para estrategias de análisis en las que las adjunciones se tratan de forma descendente, como los fuertemente dirigidos, capaces de incorporar estrategias de análisis en las cuales las adjunciones se tratan de manera ascendente y/o descendente.
- Los autómatas con dos pilas, una extensión de los autómatas a pila que trabaja con una pila maestra encargada de dirigir el proceso de análisis y una pila auxiliar que restringe las transiciones aplicables en un momento dado. Hemos descrito dos versiones diferentes de este tipo de autómatas, los autómatas con dos pilas fuertemente dirigidos, aptos para describir estrategias de análisis arbitrarias, y los autómatas con dos pilas ascendentes, adecuados para describir estrategias de análisis en las cuales las adjunciones se procesan ascendentemente.

Hemos definido esquemas de compilación para todos estos modelos de autómeta. Estos esquemas permiten obtener el conjunto de transiciones correspondiente a la implantación de una determinada estrategia de análisis sintáctico para una gramática dada.

Todos los modelos de autómeta pueden ser ejecutados en tiempo polinomial con respecto a la longitud de la cadena de entrada mediante la aplicación de técnicas de interpretación tabular. Estas técnicas se basan en la manipulación de representaciones colapsadas de las configuraciones del autómeta, denominadas ítems, que se almacenan en una tabla para su posterior reutilización. Con ello se evita la realización de cálculos redundantes.

Finalmente, hemos analizado conjuntamente los diferentes modelos de autómeta, los cuales se pueden dividir en tres grandes grupos: la familia de los autómetas generales, de la que forman parte los autómetas lineales de índices fuertemente dirigidos y los autómetas con dos pilas fuertemente dirigidos; la familia de los autómetas descendentes, en la que se encuadran los autómetas a pila embebidos y los autómetas lineales de índices orientados a la izquierda; y la familia de los autómetas ascendentes, en la que se enmarcan los autómetas a pila embebidos ascendentes, los autómetas lineales de índices orientados a la derecha y los autómetas con dos pilas ascendentes.

Abstract

Tree adjoining grammars are an extension of context-free grammars that use trees instead of productions as the primary representing structure and that are considered to be adequate to describe most of syntactic phenomena occurring in natural languages. These grammars generate the class of tree adjoining languages, which is equivalent to the class of languages generated by linear indexed grammars and other mildly context-sensitive formalisms.

In the first part of this dissertation, we introduce the problem of parsing tree adjoining grammars and linear indexed grammars, creating, for both formalisms, a continuum from simple pure bottom-up algorithms to complex predictive algorithms and showing what transformations must be applied to each one in order to obtain the next one in the continuum.

In the second part, we define several models of automata that accept the class of tree adjoining languages, proposing techniques for their efficient execution. The use of automata for parsing is interesting because they allow us to separate the problem of the definition of parsing algorithms from the problem of their execution. We have considered the following types of automata:

- Top-down and bottom-up embedded push-down automata, two extensions of push-down automata working on nested stacks. A new definition is provided in which the finite-state control has been eliminated and several kinds of normalized transition have been defined, preserving the equivalence with tree adjoining languages.
- Logical push-down automata restricted to the case of tree adjoining languages. Depending on the set of allowed transitions, we obtain three different types of automata.
- Linear indexed automata, left-oriented and right-oriented to describe parsing strategies in which adjunctions are recognized top-down and bottom-up, respectively, and strongly-driven to define parsing strategies recognizing adjunctions top-down and/or bottom-up.
- 2-stack automata, an extension of push-down automata working on a pair of stacks, a master stack driving the parsing process and an auxiliary stack restricting the set of transitions that can be applied at a given moment. Strongly-driven 2-stack automata can be used to describe bottom-up, top-down or mixed parsing strategies for tree adjoining languages with respect to the recognition of the adjunctions. Bottom-up 2-stack automata are specifically designed for parsing strategies recognizing adjunctions bottom-up.

Compilation schemata for these models of automata have been defined. A compilation schema allow us to obtain the set of transitions corresponding to the implementation of a parsing strategy for a given grammar.

All the presented automata can be executed in polynomial time with respect to the length of the input string by applying tabulation techniques. A tabular technique makes possible to interpret an automaton by means of the manipulation of collapsed representation of configurations (called items) instead of actual configurations. Items are stored into a table in order to be reused, avoiding redundant computations.

Finally, we have studied the relations among the different classes of automata, the main difference being the storage structure used: embedded stacks, indices lists or coupled stacks. According to the strategies that can be implemented, we can distinguish three kinds of automata: bottom-up automata, including bottom-up embedded push-down automata, bottom-up restricted logic push-down automata, right-oriented linear indexed automata and bottom-up 2-stack automata; top-down automata, including (top-down) embedded push-down automata, top-down restricted logic push-down automata and left-oriented linear indexed automata; and general automata, including strongly-driven linear indexed automata and strongly-driven 2-stack automata.

Índice general

. Índice de Figuras	XXI
. Índice de Tablas	XXIII
1. Introducción	1
1.1. El lenguaje natural	1
1.2. La programación dinámica en el análisis sintáctico	2
1.3. Formalismos gramaticales	3
1.4. Ámbito de la tesis	4
1.5. Estructura de la memoria	5
1.6. Difusión de resultados	7
1.7. Comunicación con el autor	9
I Lenguajes de adjunción de árboles	11
2. Lenguajes de adjunción de árboles	13
2.1. Lenguajes suavemente dependientes del contexto	13
2.2. Gramáticas de adjunción de árboles	15
2.2.1. La operación de adjunción	16
2.2.2. Árbol de derivación	17
2.2.3. TAG lexicalizadas	19
2.2.4. Propiedades	20
2.2.5. Relevancia lingüística	21
2.3. Formalismos derivados de TAG	24
2.3.1. TAG basadas en unificación	24
2.3.2. TAG estocásticas	24
2.3.3. TAG con dominación local y precedencia lineal	25
2.3.4. TAG síncronas	25
2.3.5. TAG multicomponente	26
2.3.6. Gramáticas de descripción de árboles	26
2.3.7. Gramáticas de inserción de árboles	27
2.3.8. TAG en forma regular	29
2.4. Gramáticas lineales de índices	29
2.4.1. Propiedad de independencia del contexto de LIG	33
2.4.2. Extensiones a la notación	34
2.4.3. Conversión de TAG a LIG	36
2.5. Formalismos derivados de LIG	39
2.5.1. LIG estocásticas	39

2.5.2.	Gramáticas parcialmente lineales de índices	39
2.5.3.	Gramáticas parcialmente lineales de árboles	39
2.5.4.	LIG multiconjunto	40
2.5.5.	Gramáticas pila-lineales independientes del contexto	40
2.6.	Otros formalismos gramaticales que generan TAL	40
2.6.1.	Gramáticas categoriales combinatorias	40
2.6.2.	Gramáticas de núcleo	42
2.6.3.	Sistemas de matrices recurrentes independientes del contexto de índice 2	45
2.6.4.	Gramáticas de concatenación de rangos positivas simples de aridad 2	46
2.6.5.	Gramáticas independientes del contexto acopladas de rango 2	48
2.6.6.	Gramáticas de dos niveles	50
2.7.	Complejidad del análisis de los lenguajes de adjunción de árboles	52
2.7.1.	Complejidad temporal	52
2.7.2.	Complejidad espacial	53
3.	Algoritmos de análisis sintáctico para TAG	55
3.1.	Introducción	55
3.2.	Algoritmo de tipo CYK	58
3.3.	Algoritmos de tipo Earley ascendente	60
3.4.	La propiedad del prefijo válido en los algoritmos de análisis	68
3.5.	Algoritmos de tipo Earley sin la propiedad del prefijo válido	69
3.6.	Algoritmos de tipo Earley con la propiedad del prefijo válido	76
3.7.	Análisis sintáctico de TAG lexicalizadas	84
3.8.	El bosque de análisis	85
3.8.1.	Gramáticas independientes del contexto como bosque de análisis	85
3.8.2.	Gramáticas lineales de índices como bosque de análisis	86
3.9.	Otros algoritmos de análisis sintáctico para TAG	87
3.9.1.	El algoritmo de Lang	87
3.9.2.	Algoritmos bidireccionales	97
3.9.3.	Algoritmos de varias fases	98
3.9.4.	Algoritmos basados en LIG	99
3.9.5.	Algoritmos LR	100
3.9.6.	Algoritmos paralelos	108
4.	Algoritmos de análisis sintáctico para LIG	111
4.1.	Algoritmo de tipo CYK	111
4.2.	Algoritmo de tipo Earley ascendente	113
4.3.	Algoritmo de tipo Earley sin la propiedad del prefijo válido	118
4.4.	Algoritmos de tipo Earley con la propiedad del prefijo válido	120
4.5.	El bosque de análisis	126
4.6.	Comparación entre los algoritmos de análisis sintáctico para LIG y TAG	130
4.7.	Otros algoritmos de análisis sintáctico para LIG	132
4.7.1.	Algoritmo bidireccional	133
4.7.2.	Algoritmo de reconocimiento de Boullier	134
4.7.3.	Algoritmo de análisis sintáctico de Boullier	135

II	Modelos de autómatas para los lenguajes de adjunción de árboles	137
5.	Autómatas a pila	139
5.1.	Definición	139
5.1.1.	Definición con estados	139
5.1.2.	Definición sin estados	140
5.2.	Esquemas de compilación	141
5.2.1.	Estrategia descendente	142
5.2.2.	Estrategia Earley	143
5.2.3.	Estrategia ascendente	143
5.3.	Tabulación	146
5.3.1.	La técnica de Lang	146
5.3.2.	La técnica de Nederhof	147
6.	Autómatas a pila embebidos	149
6.1.	Introducción	149
6.2.	Autómatas a pila embebidos	149
6.3.	Autómatas a pila embebidos sin estados	153
6.4.	Equivalencia entre autómatas a pila embebidos sin estados y con estados	158
6.5.	Esquemas de compilación de gramáticas independientes del contexto	160
6.6.	Esquemas de compilación de gramáticas de adjunción de árboles	161
6.6.1.	Estrategia descendente	164
6.6.2.	Estrategia Earley	164
6.6.3.	Estrategia ascendente	166
6.7.	Esquemas de compilación de gramáticas lineales de índices	166
6.7.1.	Estrategia descendente	168
6.7.2.	Estrategia Earley	168
6.7.3.	Estrategia ascendente	169
6.8.	Lenguajes de adjunción de árboles y EPDA	169
6.9.	Tabulación	172
7.	Autómatas a pila embebidos ascendentes	187
7.1.	Introducción	187
7.2.	Definición con estados	187
7.3.	Autómatas a pila embebidos ascendentes sin estados	193
7.4.	Equivalencia entre autómatas a pila embebidos sin estados y con estados	198
7.5.	Esquemas de compilación de gramáticas independientes del contexto	201
7.6.	Esquemas de compilación de gramáticas de adjunción de árboles	202
7.6.1.	Estrategia descendente	204
7.6.2.	Estrategia Earley	204
7.6.3.	Estrategia ascendente	204
7.7.	Esquemas de compilación de gramáticas lineales de índices	206
7.7.1.	Estrategia descendente	208
7.7.2.	Estrategia Earley	208
7.7.3.	Estrategia ascendente	208
7.8.	Lenguajes de adjunción de árboles y BEPDA	210
7.9.	Tabulación	213

8. Autómatas lógicos a pila restringidos	219
8.1. Introducción	219
8.2. Autómatas lógicos a pila	220
8.3. Esquemas de compilación de gramáticas lineales de índices	222
8.3.1. Estrategia genérica	223
8.3.2. Estrategias *-ascendentes	224
8.3.3. Estrategias *-Earley	226
8.3.4. Estrategias *-descendentes	230
8.4. Estrategias de análisis de gramáticas de adjunción de árboles	232
8.4.1. Estrategia genérica	233
8.4.2. Estrategias *-ascendentes	234
8.4.3. Estrategias *-Earley	236
8.4.4. Estrategias *-descendentes	240
8.5. Tabulación de los autómatas lógicos a pila restringidos	240
8.5.1. Tabulación de estrategias *-ascendentes	243
8.5.2. Tabulación de estrategias ascendentes-ascendentes	254
8.5.3. Tabulación de estrategias *-Earley	255
8.5.4. Tabulación de estrategias *-descendentes	267
9. Autómatas lineales de índices	283
9.1. Introducción	283
9.2. Autómatas lineales de índices orientados a la derecha	284
9.2.1. Esquemas de compilación	285
9.2.2. Tabulación	286
9.3. Autómatas lineales de índices orientados a la izquierda	286
9.3.1. Esquemas de compilación de gramáticas lineales de índices	288
9.3.2. Esquemas de compilación de gramáticas de adjunción de árboles	291
9.3.3. L-LIA y los lenguajes de adjunción de árboles	292
9.3.4. Tabulación	296
9.4. Autómatas lineales de índices fuertemente dirigidos	310
9.4.1. Esquemas de compilación de gramáticas lineales de índices	314
9.4.2. Esquemas de compilación de gramáticas de adjunción de árboles	314
9.4.3. SD-LIA y los lenguajes de adjunción de árboles	316
9.4.4. Tabulación	323
10. Autómatas con dos pilas	333
10.1. Introducción	333
10.2. Autómatas con dos pilas	333
10.3. Autómatas con dos pilas fuertemente dirigidos	336
10.3.1. Esquemas de compilación de gramáticas lineales de índices	339
10.3.2. Esquemas de compilación de gramáticas de adjunción de árboles	339
10.3.3. SD-2SA y los lenguajes de adjunción de árboles	342
10.3.4. Tabulación	346
10.4. Autómatas con dos pilas ascendentes	366
10.4.1. Esquemas de compilación de gramáticas lineales de índices	368
10.4.2. Esquemas de compilación para gramáticas de adjunción de árboles	368
10.4.3. BU-2SA y los lenguajes de adjunción de árboles	369
10.4.4. Tabulación	372

11. Recapitulación	385
11.1. Autómatas generales	385
11.1.1. SD-2SA y SD-LIA	385
11.2. Autómatas descendentes	389
11.2.1. EPDA y L-LIA	389
11.2.2. RLPDA *-descendentes y L-LIA	390
11.3. Autómatas ascendentes	391
11.3.1. BEPDA y R-LIA	391
11.3.2. BU-2SA y R-LIA	392
12. Conclusiones	397
12.1. Trabajo futuro	398
III Apéndices	401
A. Esquemas de análisis sintáctico	403
A.1. Esquemas de análisis sintáctico para CFG	403
A.2. Sistemas de deducción gramatical	404
A.3. Transformación de esquemas de análisis sintáctico	405
A.3.1. Generalización	405
A.3.2. Filtrado	406
A.4. Esquemas de análisis sintáctico para autómatas a pila	407
A.5. Esquemas de análisis sintáctico para TAG y LIG	407
A.6. Análisis de complejidad	408
A.6.1. Complejidad espacial	408
A.6.2. Complejidad temporal	408
B. Algoritmos de análisis sintáctico para CFG	411
B.1. El algoritmo CYK	411
B.2. Una versión ascendente del algoritmo de Earley	412
B.3. El algoritmo de Earley	413
C. Análisis sintáctico LR generalizado	415
C.1. Introducción	415
C.2. La relación entre los algoritmos de Earley y LR	416
C.3. Análisis LR con preanálisis: SLR(1) y LR(1)	419
C.4. LR(1) y LALR(1) con tablas precompiladas	423
C.5. LR(1) y LALR(1) con complejidad $\mathcal{O}(n^3)$	424
C.5.1. Análisis de complejidad	427
C.6. Tabulación del autómata a pila LR	427
C.7. Análisis sintáctico LR Generalizado para DCG	432
C.8. Análisis sintáctico LR Generalizado para Gramáticas Lineales de Índices	434
C.9. Conclusiones	439
· Bibliografía	441
· Índice Onomástico	459
· Índice de Materias	459

Índice de figuras

2.1. Estructuras asociadas a la cadena aaa por \mathcal{G}_1 y por \mathcal{G}_2	15
2.2. Operación de adjunción	17
2.3. Gramática de adjunción de árboles que genera el lenguaje $a^n b^m c^n d^m$	18
2.4. Árbol derivado (izquierda) y de derivación (derecha) en TAG para $aabbccddd$	18
2.5. Relaciones cruzadas en la cadena $aabbccddd$	19
2.6. TAG lexicalizada que genera el lenguaje $a^n b^m c^n d^m$	20
2.7. Gramática de adjunción de árboles para $a^n b^n e$ y árbol derivado para $aabbe$	21
2.8. Dominio extendido de localidad de las TAG	22
2.9. Dominio de localidad de la dependencia entre who_i y ϵ_i	23
2.10. Dependencia de larga distancia entre who_i y ϵ_i	23
2.11. Gramática de inserción de árboles	28
2.12. Árbol derivado en LIG para la cadena $aabbccddd$	32
2.13. Árbol derivado en LIG para la cadena $aabbe$	32
2.14. Árbol derivado en CCG para la cadena $aabbccddd$	42
2.15. Árbol de derivación en HG para la cadena $aabbccddd$	44
2.16. cf-RMS que genera el lenguaje $\{a^n b^m c^n d^m\}$	46
2.17. Derivación de la cadena $aabbccddd$ en cf-RMS	46
3.1. Ejemplos de árbol inicial y auxiliar	57
3.2. Descripción gráfica de un paso \mathcal{D}_{CYK}^{Adj}	60
3.3. Descripción gráfica de un paso $\mathcal{D}_{buE}^{AdjComp}$	66
3.4. Descripción gráfica de un paso $\mathcal{D}_{Ear}^{AdjComp^1}$	73
3.5. Descripción gráfica de un paso $\mathcal{D}_{Earley}^{AdjComp^1}$	78
3.6. Descripción gráfica de la aplicación consecutiva de los pasos $\mathcal{D}_{Nederhof}^{AdjComp^0}$ y $\mathcal{D}_{Nederhof}^{AdjComp^1}$	83
3.7. Descripción gráfica de la aplicación de un paso $\mathcal{D}_{Lang}^{SpineFootPred}$	96
3.8. Descripción gráfica de la aplicación de un paso $\mathcal{D}_{Lang}^{SpineFootComp}$	96
6.1. Autómata a pila embebido	151
6.2. Autómata a pila embebido después de una transición	151
6.3. Transiciones SWAP, PUSH y POP	154
6.4. Transiciones WRAP-A y WRAP-B	154
6.5. Transición UNWRAP	154
6.6. Reglas de compilación para TAG	162
6.7. Derivaciones de llamada en EPDA	176
6.8. Derivaciones de retorno en EPDA	176
6.9. Derivaciones de puntos especiales en EPDA	176
7.1. Una configuración de un BEPDA	188

7.2.	Configuración de un BEPDA tras aplicar una transición del primer tipo	188
7.3.	Otra configuración de un BEPDA	190
7.4.	Configuración de un BEPDA tras aplicar una transición del segundo tipo	190
7.5.	Transiciones UNWRAP-A y UNWRAP-B	195
7.6.	Transición WRAP	195
7.7.	Derivaciones de llamada en BEPDA	214
7.8.	Derivaciones de retorno en BEPDA	214
8.1.	Derivaciones de llamada en estrategias *-ascendentes	246
8.2.	Derivaciones de retorno en estrategias *-ascendentes	247
8.3.	Derivaciones de llamada en estrategias *-Earley	260
8.4.	Derivaciones de retorno en estrategias *-Earley	260
8.5.	Derivaciones de puntos especiales en estrategias *-Earley	260
8.6.	Regla de combinación para las transiciones $B[\circ\circ_1] C[\circ\circ_2\gamma'] \mapsto F[\circ\circ_2]$	262
8.7.	Regla de combinación para las transiciones $B[\circ\circ_1\gamma] C[\circ\circ_2] \mapsto F[\circ\circ_2\gamma]$	262
8.8.	Derivaciones de llamada en estrategias *-descendentes	273
8.9.	Derivaciones de retorno en estrategias *-descendentes	273
8.10.	Derivaciones de puntos especiales en estrategias *-descendentes	273
9.1.	Construcción <i>orientada a la derecha</i> de las pilas de índices	284
9.2.	Construcción <i>orientada a la izquierda</i> de las pilas de índices	288
9.3.	Derivaciones de llamada en L-LIA	298
9.4.	Derivaciones de retorno en L-LIA	298
9.5.	Derivaciones de puntos especiales en L-LIA	298
9.6.	Derivaciones de llamada en SD-LIA	325
9.7.	Derivaciones de retorno en SD-LIA	325
9.8.	Derivaciones de puntos especiales en SD-LIA	325
10.1.	Derivaciones de llamada en SD-2SA	348
10.2.	Derivaciones de retorno en SD-2SA	351
10.3.	Derivaciones de puntos especiales en SD-2SA	353
10.4.	Derivaciones de llamada en BU-2SA	375
10.5.	Derivaciones de retorno en BU-2SA	376
10.6.	Derivaciones de puntos especiales en BU-2SA	376
C.1.	Representación gráfica del algoritmo de Earley	418
C.2.	Representación gráfica del algoritmo LR(0)	420
C.3.	Representación gráfica del algoritmo LR(1)	422
C.4.	Representación gráfica del algoritmo LR(1) con complejidad cúbica	426
C.5.	Representación gráfica de la evolución de la pila en un algoritmo LR	428
C.6.	Transiciones para la cadena de entrada <i>cda</i> en el autómata LALR(1) de la gramática \mathcal{G}_1	430
C.7.	Transiciones de un ciclo en el autómata LALR(1) de la gramática \mathcal{G}_2	431

Índice de tablas

3.1. Relación entre \mathcal{D}_E y el algoritmo tipo Earley sin VPP de Schabes	71
3.2. Actividades realizadas por los pasos deductivos del esquema Lang	89
4.1. Producciones propuestas por Schabes y Shieber	120
4.2. Correspondencia de los pasos deductivos para TAG y LIG	131
4.3. Relaciones binarias definidas por el reconocedor de Boullier	134
4.4. Reglas de composición de las relaciones del reconocedor de Boullier	135
4.5. Relaciones binarias definidas por el analizador sintáctico de Boullier	136
5.1. Reglas de los esquemas de compilación de gramáticas independientes del contexto	141
5.2. Parámetros del esquema de compilación genérico de CFG en PDA	143
5.3. Reglas del esquema de compilación descendente de CFG en PDA	144
5.4. Transiciones del autómata a pila con estrategia descendente	144
5.5. Configuraciones del autómata a pila descendente durante el análisis de <i>aabb</i> . . .	145
5.6. Reglas del esquema de compilación Earley de CFG en PDA	146
5.7. Reglas del esquema de compilación ascendente de CFG en PDA	146
6.1. EPDA que acepta $\{a^n b^n c^n d^n \mid n > 0\}$ y derivación de <i>aabbccdd</i>	152
6.2. EPDA sin estados que acepta $\{a^n b^n c^n d^n \mid n > 0\}$ y derivación de <i>aabbccdd</i> . . .	156
6.3. Normalización de una transición compleja de un EPDA	158
6.4. Reglas del esquema de compilación genérico de TAG en EPDA	164
6.5. Reglas del esquema de compilación descendente de TAG en EPDA	165
6.6. Reglas del esquema de compilación Earley de TAG en EPDA	165
6.7. Reglas del esquema de compilación ascendente de TAG en EPDA	166
6.8. Tipos de reglas de los esquemas de compilación de LIG en EPDA	167
6.9. Reglas del esquema de compilación genérico de LIG en EPDA	168
6.10. Reglas del esquema de compilación descendente de LIG en EPDA	169
6.11. Reglas del esquema de compilación Earley de LIG en EPDA	170
6.12. Reglas del esquema de compilación ascendente de LIG en EPDA	171
6.13. Producciones de la LIG derivada del EPDA que acepta $\{a^n b^n c^n d^n\}$	173
6.14. Derivación de la cadena <i>aabbccdd</i>	173
6.15. Tipos de transiciones de los EPDA	174
6.16. Combinación de ítems en EPDA (fase de llamada)	178
6.17. Combinación de ítems en EPDA (fase de retorno)	179
7.1. BEPDA que acepta $\{a^n b^n c^n d^n \mid n > 0\}$ y configuraciones para <i>aabbccdd</i>	194
7.2. BEPDA sin estados que acepta $\{a^n b^n c^n d^n \mid n > 0\}$ y derivación de <i>aabbccdd</i> . . .	196
7.3. Normalización de una transición compleja en un BEPDA	199
7.4. Reglas del esquema de compilación genérico de TAG en BEPDA	203
7.5. Reglas del esquema de compilación descendente de TAG en BEPDA	205

7.6. Reglas del esquema de compilación Earley de TAG en BEPDA	205
7.7. Reglas del esquema de compilación ascendente de TAG en BEPDA	206
7.8. Tipos de reglas de los esquemas de compilación de LIG en BEPDA	207
7.9. Reglas del esquema de compilación genérico de LIG en BEPDA	208
7.10. Esquema de compilación descendente de LIG en BEPDA	209
7.11. Reglas del esquema de compilación Earley de LIG en BEPDA	209
7.12. Reglas del esquema de compilación ascendente de LIG en BEPDA	210
7.13. Producciones de la LIG derivada del BEPDA que acepta $\{a^n b^n c^n d^n\}$	212
7.14. Derivación de la cadena <i>aabbccdd</i>	212
7.15. Combinación de ítems en BEPDA	215
8.1. Traducción de símbolos LIG en símbolos DCG	220
8.2. Traducción de producciones LIG en cláusulas definidas	220
8.3. Reglas para los esquemas de compilación de gramáticas lineales de índices	222
8.4. Reglas del primer esquema de compilación genérico de LIG en RLPDA	225
8.5. Reglas del esquema de compilación genérico de LIG en RLPDA	225
8.6. Parámetros del esquema de compilación genérico de LIG en RLPDA	225
8.7. Reglas del esquema de compilación ascendente-ascendente de LIG en RLPDA	227
8.8. Reglas del esquema de compilación Earley-ascendente de LIG en RLPDA	227
8.9. Reglas del esquema de compilación descendente-ascendente de LIG en RLPDA	227
8.10. Reglas del esquema de compilación ascendente-Earley de LIG en RLPDA	229
8.11. Reglas del esquema de compilación Earley-Earley de LIG en RLPDA	229
8.12. Reglas del esquema de compilación descendente-Earley de LIG en RLPDA	229
8.13. Reglas del esquema de compilación ascendente-descendente de LIG en RLPDA	231
8.14. Reglas del esquema de compilación Earley-descendente de LIG en RLPDA	231
8.15. Reglas del esquema de compilación descendente-descendente de LIG en RLPDA	231
8.16. Reglas para los esquemas de compilación de gramáticas de adjunción de árboles	233
8.17. Reglas del esquema de compilación genérico de TAG en RLPDA	235
8.18. Parámetros del esquema de compilación genérico de TAG en RLPDA	235
8.19. Reglas del esquema de compilación ascendente-ascendente de TAG en RLPDA	237
8.20. Reglas del esquema de compilación Earley-ascendente de TAG en RLPDA	237
8.21. Reglas del esquema de compilación descendente-ascendente de TAG en RLPDA	238
8.22. Reglas del esquema de compilación ascendente-Earley de TAG en RLPDA	238
8.23. Reglas del esquema de compilación Earley-Earley de TAG en RLPDA	239
8.24. Reglas del esquema de compilación descendente-Earley de TAG en RLPDA	239
8.25. Reglas del esquema de compilación ascendente-descendente de TAG en RLPDA	241
8.26. Reglas del esquema de compilación Earley-descendente de TAG en RLPDA	241
8.27. Reglas del esquema de compilación descendente-descendente de TAG en RLPDA	242
8.28. Tipos de transiciones en las estrategias *-ascendentes	244
8.29. Combinación de ítems en las estrategias *-ascendentes	248
8.30. Combinación de ítems en las estrategia ascendente-ascendente	256
8.31. Tipos de transiciones en las estrategias *-Earley	257
8.32. Combinación de ítems en las estrategias *-Earley	261
8.33. Tipos de transiciones en las estrategias *-descendente	268
8.34. Combinación de ítems en las estrategias *-descendentes (fase de llamada)	272
8.35. Combinación de ítems en las estrategias *-descendentes (fase de retorno)	274
9.1. Reglas del esquema de compilación genérico de LIG en R-LIA	287
9.2. Reglas del esquema de compilación genérico de TAG en R-LIA	287
9.3. Transiciones de L-LIA y RLPDA *-descendentes	289

9.4. Reglas del esquema de compilación genérico de LIG en L-LIA	290
9.5. Reglas del esquema de compilación ascendente-descendente de LIG en L-LIA . .	290
9.6. Reglas del esquema de compilación Earley-descendente de LIG en L-LIA	290
9.7. Reglas del esquema de compilación descendente-descendente de LIG en L-LIA .	291
9.8. Reglas del esquema de compilación genérico de TAG en L-LIA	292
9.9. Reglas del esquema de compilación ascendente-descendente de TAG en L-LIA . .	293
9.10. Reglas del esquema de compilación Earley-descendente de TAG en L-LIA	294
9.11. Reglas del esquema de compilación descendente-descendente de TAG en L-LIA .	294
9.12. Tipos de transiciones L-LIA	295
9.13. Combinación de ítems en L-LIA (fase de llamada)	300
9.14. Combinación de ítems en L-LIA (fase de retorno)	301
9.15. L-LIA que acepta $\{a^n b^n c^n d^n \mid n > 0\}$ y derivación de <i>aabbccdd</i>	302
9.16. Ítems generados durante el reconocimiento de <i>aabbccdd</i>	303
9.17. derivación de la cadena $a_1^2 \dots a_{2k}^2$ en N-LIA	312
9.18. Reglas del esquema de compilación genérico de LIG en SD-LIA	315
9.19. Parámetros del esquema de compilación genérico de LIG en SD-LIA	315
9.20. Reglas del esquema de compilación genérico de TAG en SD-LIA	317
9.21. Parámetros del esquema de compilación genérico de TAG en SD-LIA	317
9.22. Transiciones del SD-LIA que acepta $\{a^n b^n c^n d^n \mid n > 0\}$	321
9.23. Configuraciones del SD-LIA para la cadena de entrada <i>aaabbbcccddd</i>	321
9.24. Producciones de la LIG obtenida a partir del SD-LIA	322
9.25. Derivación en LIG de la cadena <i>aaabbbcccddd</i>	322
9.26. Combinación de ítems en SD-LIA	326
10.1. Transiciones del SD-2SA que acepta $\{a^n b^n c^n d^n \mid n > 0\}$	340
10.2. Configuraciones del SD-2SA para la cadena de entrada <i>aaabbbcccddd</i>	340
10.3. Reglas del esquema de compilación genérico de LIG en SD-2SA	341
10.4. Parámetros del esquema de compilación genérico de LIG en SD-2SA	341
10.5. Reglas del esquema de compilación genérico de TAG en SD-2SA	343
10.6. Parámetros del esquema de compilación genérico de TAG en SD-2SA	343
10.7. Reglas de combinación de ítems en SD-2SA	352
10.8. Reglas del esquema de compilación genérico de LIG en BU-2SA	368
10.9. Parámetros del esquema de compilación genérico de LIG en SD-2SA	369
10.10 Reglas del esquema de compilación genérico de TAG en BU-2SA	370
10.11 Parámetros del esquema de compilación genérico de TAG en BU-2SA	371
10.12 Transiciones del BU-2SA que acepta $\{a^n b^n c^n d^n \mid n > 0\}$	373
10.13 Configuraciones del BU-2SA para la cadena de entrada <i>aaabbbcccddd</i>	373
10.14 Producciones de la LIG obtenida a partir del BU-2SA	374
10.15 Derivación en LIG de la cadena <i>aaabbbcccddd</i>	374
10.16 Reglas de combinación de ítems en BU-2SA	377
11.1. Equivalencia de las transiciones de SD-LIA y de SD-2SA	386
11.2. Esqueleto independiente del contexto de las transiciones de SD-LIA y SD-2SA .	387
11.3. Tratamiento de los índices en las transiciones de SD-LIA y SD-2SA	387
11.4. Cambios de modo en SD-LIA y SD-2SA	388
11.5. Correspondencia entre las transiciones de los RLPDA *-Earley y los SD-LIA . .	388
11.6. Cambio de notación en las transiciones de los EPDA	390
11.7. Equivalencia entre L-LIA y un subconjunto de los EPDA	390
11.8. Correspondencia entre las transiciones de los RLPDA *-descendentes y de los L-LIA	391
11.9. Equivalencia entre las Transiciones de BEPDA y de R-LIA	392