

Generador de Evaluador estático



Tesis de la carrera Licenciatura en Ciencias de la Computación.

Autores:

KILMURRAY, Gerardo Luis

PICCO, Gonzalo Martin

Departamento de Computación
Facultad de Ciencias Exactas, Físico-Químicas y Naturales
Universidad Nacional de Río Cuarto - Argentina.

Director:

Mg. Arroyo Marcelo.

Agradecimientos

A toda la familia.

Índice general

1. Introducción	1
1.1. Gramática de Atributos	1
1.2. Métodos de Evaluación	2
1.2.1. Evaluación dinámica	2
1.2.2. Evaluación estática	3
1.2.3. Evaluacion de la familia ANCAG	3
1.3. Secuencia de visita	3
1.4. Generación de evaluadores para GA bien definidas	3
1.5. Evaluación durante el parsing	3
2. Clasificación de AG	5
2.1. Clasificación basada en la estrategia de evaluación	5
2.2. Clasificación basada en dependencias	5
2.3. Clasificación de Knuth	5
2.3.1. Árbol sintáctico atribuido	5
3. Multi-plans attribute grammar	7
3.1. Clasificación de Wu Yang	7
3.1.1. Gramaticas no circulates(NC)	7
3.1.2. ANCAG	7
4. Evaluacion estatica de MAG	9
4.1. Algoritmos para grafos: DP, DOWN, DCG y ADP	9
4.2. Algoritmo de computo de planes	9
4.3. Algoritmo de computo de secuencia de visita	9
5. Diseño de MagGen	11
5.1. Lenguaje de especificación de las MAG	11
5.2. Estructuras internas	11
5.3. Diseño del evaluador estático generado	11
6. Detalles de Implementación	13
6.1. GNU/linux	13
6.2. Herramientas utilizadas	13
6.2.1. Eclipse	13
6.2.2. Boost	13
6.2.3. Doxygen	14
6.2.4. Eclipse	14
6.2.5. Graphviz	14
6.2.6. Dia	14

6.2.7. Latex	14
6.2.8. g++, make, gdb, Nemiver	14
6.2.9. Cloc, CCCC, Gcov	14
6.3. Algoritmo de generación de secuencia de visita	14
6.4. Algoritmo de generación de código	14
7. Usos	15
7.1. Uso de MagGen	15
7.2. Uso del evaluador generado	15
8. Conclusión	17
8.1. Conclusión	17
8.2. Trabajos futuros	17
A. Appendix Example	19
A.1. Appendix Example section	19

Introducción

Contents

1.1. Gramática de Atributos	1
1.2. Métodos de Evaluación	2
1.2.1. Evaluación dinámica	2
1.2.2. Evaluación estática	3
1.2.3. Evaluación de la familia ANCAG	3
1.3. Secuencia de visita	3
1.4. Generación de evaluadores para GA bien definidas	3
1.5. Evaluación durante el parsing	3

Desde que D. Knuth introdujo en 1966 las gramáticas de atributos (GA), estas se han utilizado ampliamente para el desarrollo de herramientas de procesamiento de lenguajes formales como compiladores, intérpretes, traductores como también para especificar la semántica de lenguajes de programación. Las gramáticas de atributos son un formalismo simple para la especificación de la semántica de lenguajes formales, como los lenguajes de programación o de especificación. Integran la modularidad que brindan las gramáticas libres de contexto y la expresividad de un lenguaje funcional.

1.1. Gramática de Atributos

En una gramática de atributos, se relaciona con cada símbolo de una gramática libre de contexto un conjunto de atributos. Cada regla o producción tiene asociados un conjunto de reglas semánticas que toman la forma de asignación a atributos de valores denotados por la aplicación de una función, la cual puede tomar como argumentos instancias de atributos pertenecientes a los símbolos que aparecen en la producción. Las reglas semánticas inducen dependencias entre los atributos que ocurren en la producción. El orden de evaluación es implícito (si existe) y queda determinado por las dependencias entre las instancias de los atributos. Una regla semántica se podrá evaluar cuando las instancias de los atributos que aparecen como sus argumentos estén evaluadas. Un evaluador de gramáticas de atributos debe tener en cuenta las dependencias entre las instancias de atributos para seguir un orden consistente de evaluación de los mismos. Si una GA contiene dependencias circulares no podría ser evaluada ya que no podría encontrarse un orden de evaluación.

Existen numerosas herramientas basadas en este formalismo o en alguna de sus extensiones, entre las cuales podemos mencionar yacc, Yet Another Compiler-Compiler [29], desarrollado por ATT, AntLR, JavaCC, JavaCUP, ELI y muchas otras.

1.2. Métodos de Evaluación

Los métodos estáticos deben tener en cuenta todos los posibles árboles sintácticos posibles a ser generados por la gramática y calcular todas las posibles dependencias entre las instancias de los atributos. Además, se deberán detectar las posibles dependencias circulares, para informar la viabilidad de su evaluación. Esto se conoce como el problema de la circularidad, el cual se ha demostrado ser intrínsecamente exponencial [20]. El problema de la circularidad ha motivado que muchos investigadores hayan realizado esfuerzos en la búsqueda e identificación de familias o subgrupos de gramáticas de atributos, para las cuales puedan detectarse circularidades con algoritmos de menor complejidad (polinomial o lineal). Estas familias imponen restricciones sobre la gramática de atributos o sobre las dependencias entre sus atributos para garantizar que una GA no sea circular, con el costo de restringir su poder expresivo. Las clases de familias de gramáticas de atributos que se han utilizado para el desarrollo de herramientas eficientes y que se encuentran ampliamente analizadas en la bibliografía especializada, encontramos las s-atribuidas², l-atribuidas, las gramáticas de atributos ordenadas (OAG) y las absolutamente no circulares (ANCAG)[2]. En 1980, Uwe Kastens[23] caracterizó las gramáticas de atributos ordenadas y propuso un método para su evaluación, denominado secuencias de visita. Estas son secuencias de operaciones que conducen el recorrido del árbol sintáctico atribuido y realizan la evaluación de las instancias de los atributos. Kastens propone un método para generar las secuencias de visita en tiempo polinomial para la familia OAG. Mas recientemente, en 1999, se han propuesto nuevas familias de GA para las que se pueden implementar evaluadores eficientes basado en métodos estáticos y con un mayor poder expresivo que las utilizadas tradicionalmente[44].

1.2.1. Evaluación dinámica

bla bla

1.2.2. Evaluación estática

1.2.3. Evaluacion de la familia ANCAG

1.3. Secuencia de visita

1.4. Generación de evaluadores para GA bien definidas

1.5. Evaluación durante el parsing

bla bla

Clasificación de AG

Contents

2.1. Clasificación basada en la estrategia de evaluación	5
2.2. Clasificación basada en dependencias	5
2.3. Clasificación de Knuth	5
2.3.1. Árbol sintáctico atribuido	5

bla

2.1. Clasificación basada en la estrategia de evaluación

bla bla

2.2. Clasificación basada en dependencias

bla bla

2.3. Clasificación de Knuth

cla cla

2.3.1. Árbol sintáctico atribuido

Multi-plans attribute grammar

Contents

3.1. Clasificación de Wu Yang	7
3.1.1. Gramaticas no circulates(NC)	7
3.1.2. ANCAG	7

3.1. Clasificación de Wu Yang

3.1.1. Gramaticas no circulates(NC)

bla bla

3.1.2. ANCAG

Evaluacion estatica de MAG

Contents

4.1. Algoritmos para grafos: DP, DOWN, DCG y ADP	9
4.2. Algoritmo de computo de planes	9
4.3. Algoritmo de computo de secuencia de visita	9

bla bla

4.1. Algoritmos para grafos: DP, DOWN, DCG y ADP

bla bla

4.2. Algoritmo de computo de planes

bla bla

4.3. Algoritmo de computo de secuencia de visita

Diseño de MagGen

Contents

5.1. Lenguaje de especificación de las MAG	11
5.2. Estructuras internas	11
5.3. Diseño del evaluador estático generado	11

bla bla

5.1. Lenguaje de especificación de las MAG

bla bla.

5.2. Estructuras internas

bla bla

5.3. Diseño del evaluador estático generado

bla bla

Detalles de Implementación

Contents

6.1. GNU/linux	13
6.2. Herramientas utilizadas	13
6.2.1. Eclipse	13
6.2.2. Boost	13
6.2.3. Doxygen	14
6.2.4. Eclipse	14
6.2.5. Graphviz	14
6.2.6. Dia	14
6.2.7. Latex	14
6.2.8. g++, make, gdb, Nemiver	14
6.2.9. Cloc, CCCC, Gcov	14
6.3. Algoritmo de generación de secuencia de visita	14
6.4. Algoritmo de generación de código	14

bla bla

6.1. GNU/linux

bla bla

6.2. Herramientas utilizadas

bla bla

6.2.1. Eclipse

bla

6.2.2. Boost

bla bla

6.2.3. Doxygen

bla bla

6.2.4. Eclipse

bla

6.2.5. Graphviz

bla bla

6.2.6. Dia

bla bla

6.2.7. Latex

bla bla

6.2.8. g++, make, gdb, Nemiver

bla bla

6.2.9. Cloc, CCCC, Gcov

bla bla

6.3. Algoritmo de generación de secuencia de visita

bla bla

6.4. Algoritmo de generación de código

bla bla

7

Usos

Contents

7.1. Uso de MagGen	15
7.2. Uso del evaluador generado	15

7.1. Uso de MagGen

bla bla

7.2. Uso del evaluador generado

bla bla

8

Conclusión

Contents

8.1. Conclusión	17
8.2. Trabajos futuros	17

bla bla.

8.1. Conclusión

bla bla

8.2. Trabajos futuros

bla bla

A

Appendix Example

A.1. Appendix Example section

And I cite myself to show by bibtex style file (two authors) [?].

This for other bibtex stye file : only one author [?] and many authors [?].

Design and Use of Numerical Anatomical Atlases for Radiotherapy

Abstract: The main objective of this thesis is to provide radio-oncology specialists with automatic tools for delineating organs at risk of a patient undergoing a radiotherapy treatment of cerebral or head and neck tumors.

To achieve this goal, we use an anatomical atlas, i.e. a representative anatomy associated to a clinical image representing it. The registration of this atlas allows to segment automatically the patient structures and to accelerate this process. Contributions in this method are presented on three axes.

First, we want to obtain a registration method which is as independent as possible w.r.t. the setting of its parameters. This setting, done by the clinician, indeed needs to be minimal while guaranteeing a robust result. We therefore propose registration methods allowing to better control the obtained transformation, using outlier rejection techniques or locally affine transformations.

The second axis is dedicated to the consideration of structures associated with the presence of the tumor. These structures, not present in the atlas, indeed lead to local errors in the atlas-based segmentation. We therefore propose methods to delineate these structures and take them into account in the registration.

Finally, we present the construction of an anatomical atlas of the head and neck region and its evaluation on a database of patients. We show in this part the feasibility of the use of an atlas for this region, as well as a simple method to evaluate the registration methods used to build an atlas.

All this research work has been implemented in a commercial software (Imago from DOSIsoft), allowing us to validate our results in clinical conditions.

Keywords: Atlas-based Segmentation, non rigid registration, radiotherapy, atlas creation
