

“Hacia la adaptación al crédito europeo en Autómatas y
Lenguajes Formales”

Ana Sánchez Ortega

UPV/EHU / LSI / TR 01-2007

Hacia la adaptación al crédito europeo en Autómatas y Lenguajes Formales

Ana Sánchez Ortega
Dpto. de Lenguajes y Sistemas Informáticos Informática
Facultad de Informática - Universidad del País Vasco
Pº Manuel Lardizabal 1, 20018 San Sebastián
ana.sanchez@ehu.es

Resumen

Durante el curso 2006-2007 la asignatura Autómatas y Lenguajes Formales se ha incorporado al proyecto SICRE (Seguimiento para la Introducción de los CRéditos Europeos). El presente documento recoge cuál ha sido la orientación de la asignatura, principalmente en cuanto a actividades realizadas y resultados de las mismas.

Introducción

La asignatura Autómatas y Lenguajes Formales (ALF) que se imparte en segundo curso de la Ingeniería Técnica en Informática de Sistemas es una de las asignaturas que se han incorporado al plan que la UPV-EHU ha implantado en el marco de la adaptación al crédito europeo ECTS (European Credit Transfer System). Durante el año 2005-06 dentro del programa AICRE (Programa de Asesoramiento para la Introducción del Crédito Europeo) se definieron las competencias, temario, bibliografía, actividades y evaluación según el protocolo definido en este plan.

El curso 2006-07 dentro del programa SICRE (Seguimiento para la Introducción de los CRéditos Europeos) ha sido el año de implantación del plan. Aunque se trata de un marco que pretende dar continuidad y apoyo al planteamiento hecho en AICRE, ha existido total libertad para incluir algunas variaciones respecto a lo previsto para cada asignatura en dicho programa. En el caso de ALF las modificaciones no han afectado ni al temario impartido ni a las competencias trabajadas. El cambio principal ha sido en la evaluación, al eliminar la obligación de un examen final, que en el planteamiento inicial tenía un peso del 65% sobre la nota final de la asignatura. La necesidad de presentar un protocolo de manera coherente y como una alternativa de evaluación continua, en oposición al examen final ha influido en la decisión de que fuera una opción completa. Por otro lado la estimación del tiempo de dedicación de la única profesora de ALF que ha implantado este enfoque en la asignatura, ha obligado a dar prioridad a las actividades previstas frente al modelo de comunicación, dejando el uso de Moodle, inicialmente previsto, para cursos posteriores y utilizando el correo electrónico como herramienta de comunicación no presencial entre la profesora y los estudiantes durante este curso.

Este documento es una recopilación de las actividades realizadas y sus resultados por lo que no entra en una valoración sobre las mejoras de aprendizaje obtenidas, ni en las posibles ventajas o inconvenientes.

Planteamiento

ALF es una asignatura troncal y sus contenidos teóricos están claramente definidos y se incluyen dentro de las ramas de la Informática Teórica. Los temas que se tratan están relacionados por un lado con la Jerarquía de lenguajes de Chomsky y por otro con los límites de los sistemas informáticos. Se estudian los reconocedores (autómatas) y generadores (gramáticas) de cada clase de lenguajes junto con los algoritmos de equivalencia entre distintos modelos, y también la existencia de problemas no computables. Se han definido siete competencias relacionadas con estos conceptos:

1. Definir formalmente lenguajes asociados a formatos de datos, instrucciones, ...
2. Utilizar diferentes modelos formales para describir lenguajes concretos

3. Describir formalmente dispositivos abstractos de cómputo
4. Clasificar un lenguaje dentro de la jerarquía de Chomsky
5. Aplicar algoritmos de transformación entre diferentes modelos formales
6. Desarrollar fragmentos de código basados en una notación formal para resolver problemas de reconocimiento de patrones
7. Conocer la existencia de límites teóricos para la resolución computacional de problemas

Para trabajar las competencias y también para evaluarlas se han propuesto distintas actividades teniendo en cuenta el tipo específico más adecuado. El peso de las competencias no es igual y por ello los instrumentos de evaluación influyen con distinto porcentaje en la nota final. Además de las actividades relacionadas con la evaluación las otras tareas recogidas en el plan se refieren a las realizadas en las clases presenciales: clases expositivas con estudio posterior y resolución, corrección y discusión de problemas planteados a lo largo de todo el curso.

Ha sido mayor el porcentaje de estudiantes que ha optado por cursar la asignatura de esta manera (63%) que los que han elegido la forma más tradicional. Ha supuesto un grupo de 22 estudiantes, que en su mayoría (77%) no ha necesitado realizar el examen final.

En los siguientes apartados se detallan los resultados de cada actividad y algunos comentarios sobre su desarrollo.

Propuesta metodológica

El plan completo se incluye en el anexo e incluye ocho actividades de evaluación, cinco de ellas individuales y tres para realizar en grupos pequeños de dos o tres personas. Las actividades que se han realizado en grupo han tenido un enfoque más práctico y han requerido el uso de su ordenador personal o de los disponibles en el centro.

Las actividades individuales son cuestionarios sobre cada uno de los temas e incluyen preguntas sobre cada una de las competencias trabajadas. Estas actividades pueden considerarse también de aprendizaje, ya que, el hecho de que sean corregidas inmediatamente y comentados los fallos con los estudiantes les permite aprender de sus propios errores o de los compañeros.

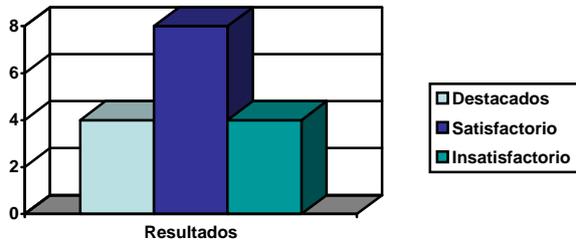
Para evaluar cada una de las tareas se han utilizado tres valores: insatisfactorio, estándar (o satisfactorio) y destacado. Con cada actividad se ha solicitado a cada estudiante información sobre las horas que han dedicado a su preparación o realización y también sobre su asistencia y participación en clase. Hemos de aclarar que no se exigía ningún porcentaje mínimo de asistencia. Generalmente las horas de trabajo personal han quedado por debajo de lo recomendado para preparar las actividades individuales y se han superado (de media) las horas recomendadas en las de grupo.

Se ha realizado también una tutoría personalizada (AE8) con cada alumno que ha seguido el programa, aunque sin influencia en la evaluación. Se ha llevado a cabo cuando ya se habían realizado cinco actividades y se ha aprovechado para explicarles de nuevo, y de manera individual, los fallos que se habían detectado en las actividades realizadas hasta ese momento y también para dialogar sobre su situación para superar la asignatura sin necesidad de realizar el examen final.

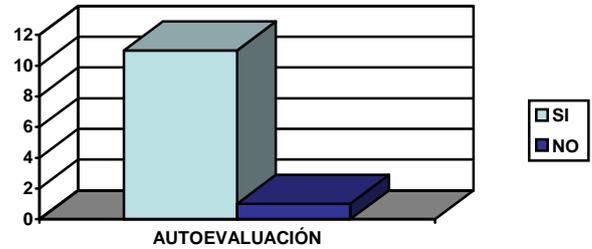
Actividad AE1

La primera actividad es un cuestionario directamente relacionado con la primera competencia: definir formalmente lenguajes asociados a formatos de datos o instrucciones. Se trata de utilizar operaciones de lenguajes y palabras, conceptos básicos para el resto de la asignatura. De la misma manera que ocurre en otras actividades, los estudiantes que mejores resultados obtuvieron no fueron los que más horas de preparación declararon necesitar, pero sí los que registraron mayor porcentaje de asistencia y participación.

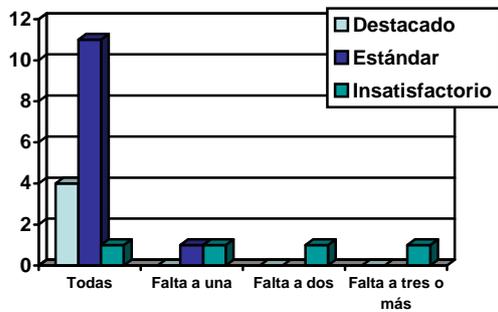
RESULTADOS AE1



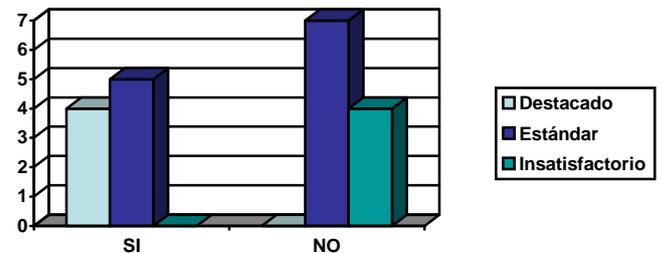
AUTOEVALUACIÓN AE1



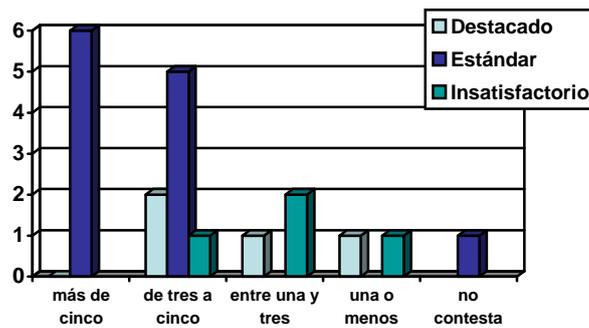
ASISTENCIA AE1



PARTICIPACIÓN AE1



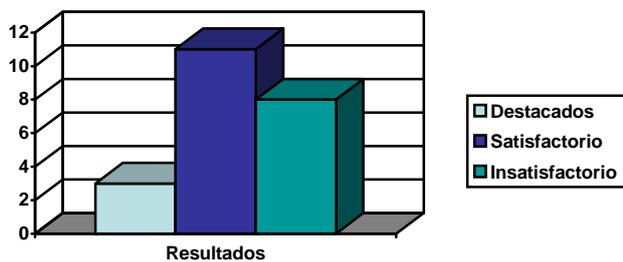
COSTE HORAS AE1



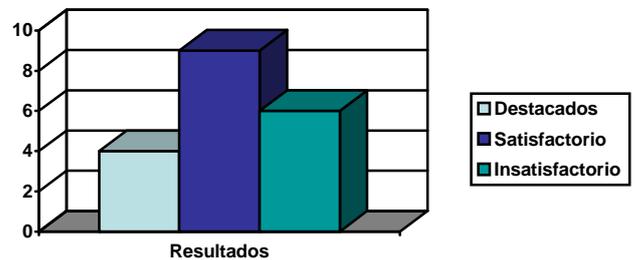
Actividades AE4 y AE6

Las actividades AE4 y AE6 también son cuestionario relacionados con las competencias dos (utilizar diferentes modelos formales para describir lenguajes concretos), tres (describir formalmente dispositivos abstractos de cómputo) y cinco (aplicar algoritmos de transformación entre diferentes modelos formales). Pero la actividad AE4 lo hace con los modelos y algoritmos relacionados con los lenguajes regulares y la actividad AE6 con los lenguajes independientes de contexto. La actividad AE6 también incluye la competencia cuatro (clasificar un lenguaje dentro de la jerarquía de Chomsky). A medida que avanzó el curso la participación en clase subió. Pero estas dos actividades hubieran necesitaban más horas de preparación de las que, en general dedicaron, y esta es, seguramente la razón de que se obtuvieran los peores resultados de todas las actividades realizadas. Se refleja también en la autoevaluación que las competencias relacionadas con la aplicación de algoritmos son mejor resueltas que las de carácter más teórico.

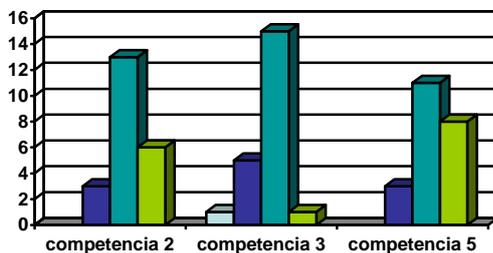
RESULTADOS AE4



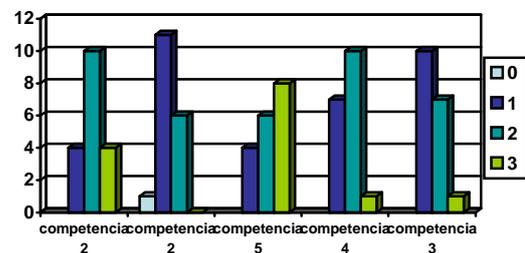
RESULTADOS AE6



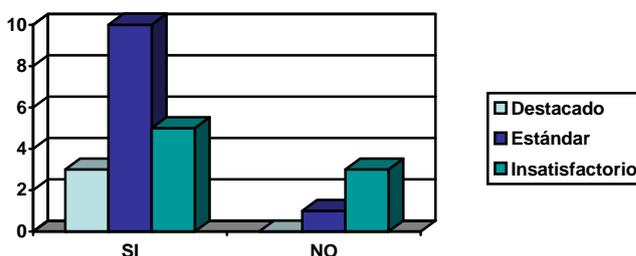
AUTOEVALUACIÓN AE4



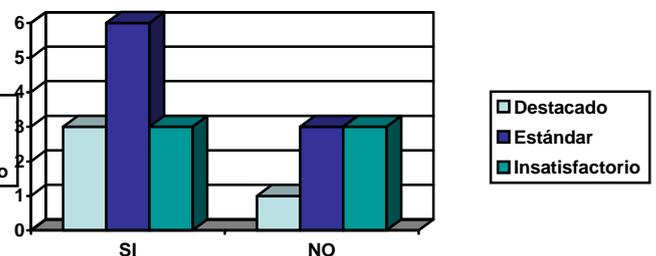
AUTOEVALUACIÓN AE6



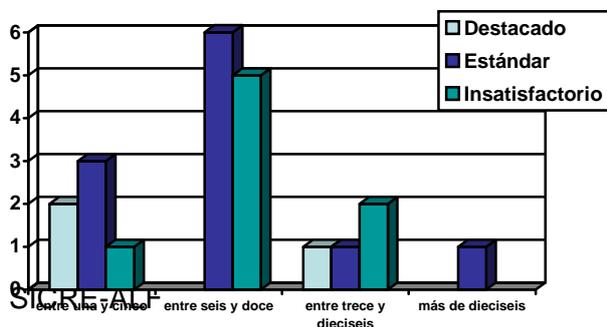
PARTICIPACIÓN AE4



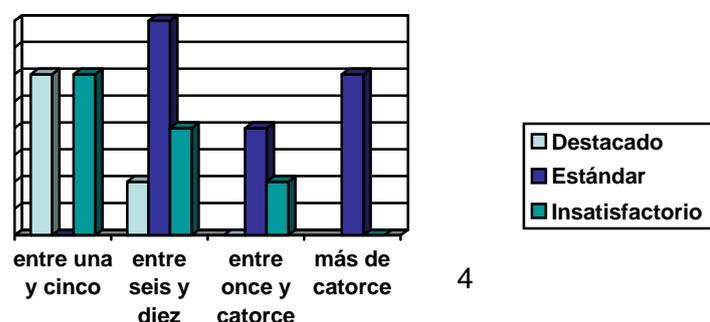
PARTICIPACIÓN AE6



COSTE HORAS AE4



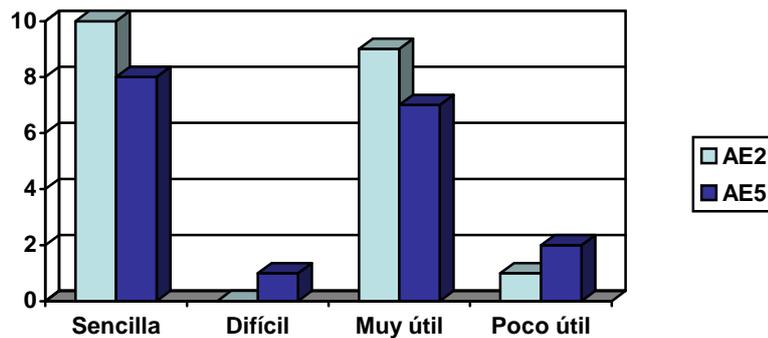
COSTE HORAS AE6



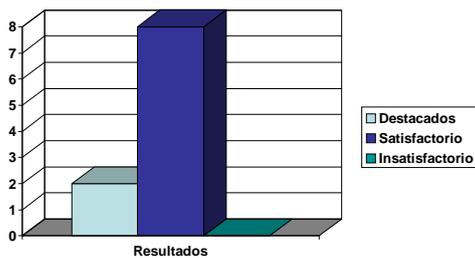
Actividades AE2 y AE5

Las actividades AE2 y AE5 trabajan también las competencias dos y cinco pero utilizando la herramienta gráfica JFLAP [2]. La primera lo hace con los lenguajes regulares y la segunda con los independientes de contexto. En general la herramienta les pareció sencilla y útil. En relación con el planteamiento de las actividades, hubo una mejora en la AE5, al observar que en el caso de la AE2 habían usado la herramienta para aplicar el algoritmo, pero sin reflexionar sobre el modo en que se hacía. Este hecho influyó también en que no valoraran suficientemente la herramienta para comprender mejor el funcionamiento de los algoritmos, como puede verse en la gráfica de autoevaluación.

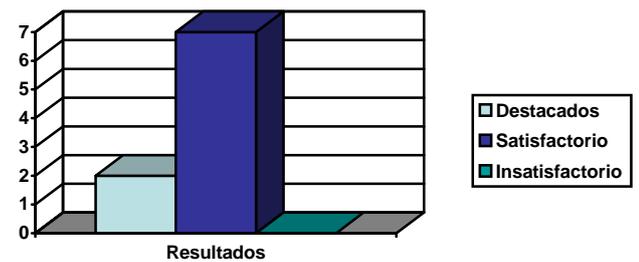
JFLAP



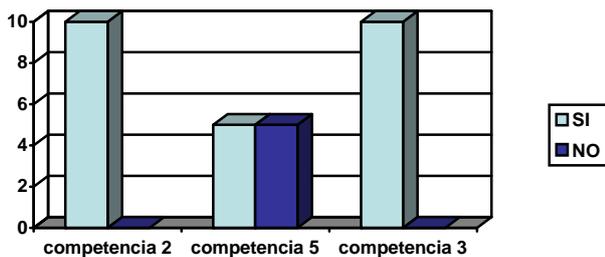
RESULTADOS AE2



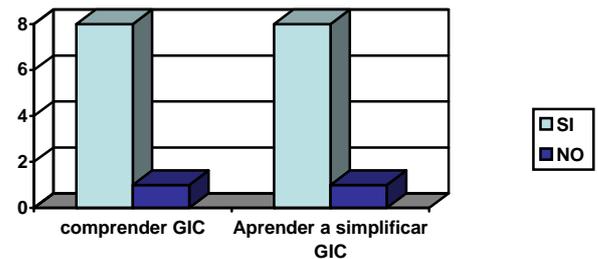
RESULTADOS AE5



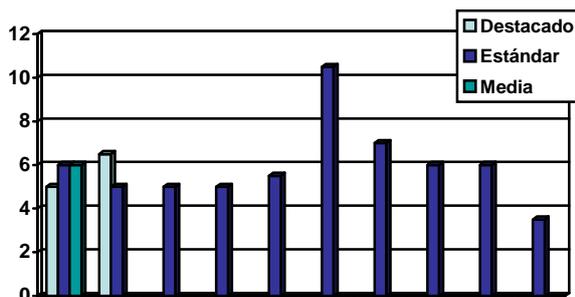
AUTOEVALUACIÓN AE2



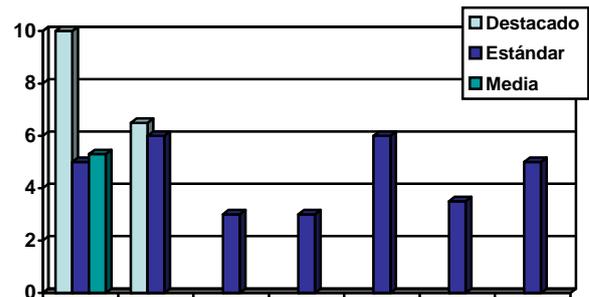
AUTOEVALUACIÓN AE5



COSTE HORAS AE2



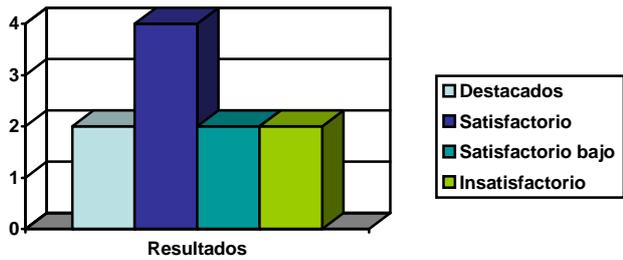
COSTE HORAS AE5



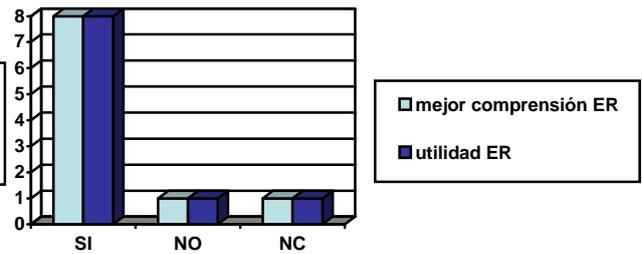
Actividad AE3

La actividad AE3 trata de desarrollar directamente la competencia seis (desarrollar fragmentos de código basados en una notación formal para resolver problemas de reconocimiento de patrones) utilizando concretamente las expresiones regulares de Javascript. El objetivo es que de esta manera comprendieran mejor las expresiones regulares y fueran conscientes también de su utilidad práctica. En el caso de esta actividad se solicitó a algunos grupos que modificaran la expresión regular utilizada para valorarla como satisfactoria. La introducción de esta competencia ha sido lo más novedoso del planteamiento en la asignatura [1].

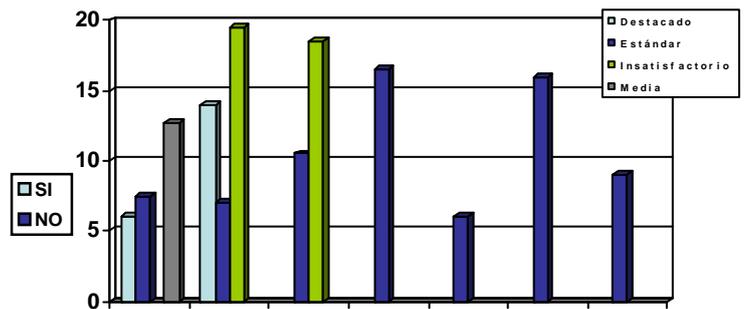
RESULTADOS AE3



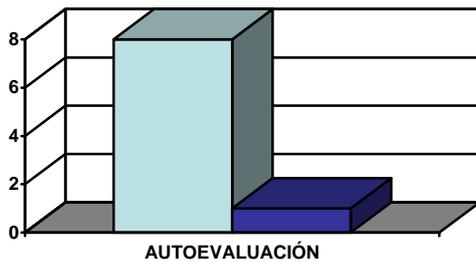
VALORACIÓN AE3



COSTE AE3



AUTOEVALUACIÓN AE3



Actividad AE7

La actividad AE7 trata de comprobar si se ha adquirido la competencia siete (conocer la existencia de límites teóricos para la resolución computacional de problemas) y también reafirmar la competencia cuatro (clasificar un lenguaje dentro de la jerarquía de Chomsky) ahora que ha visto la clasificación completa de lenguajes.



Conclusiones

Tras la puesta en marcha del plan según se ha expuesto en este informe es necesario una reflexión sobre las modificaciones necesarias para su posible continuidad. Los resultados obtenidos en cuanto a porcentaje de aprobados respecto a años anteriores, han mejorado pues sin necesidad de hacer el examen final ya han superado la asignatura el 46% de los estudiantes matriculados, frente al 42% del año anterior. Si embargo no es el único dato a tener en cuenta. Otro dato importante es la valoración de la enseñanza recibida que realizan los estudiantes a través de las encuestas que realiza la universidad, pero los resultados no se conocerán hasta septiembre. Otro factor importante es la carga de trabajo para la profesora de la asignatura, que ha sido claramente superior a años anteriores, aunque no han aumentado significativamente las tutorías. En cualquier caso este informe trata únicamente de recoger el plan puesto en marcha.

Referencias

1. Blanco J.M., Sánchez A. formularios web y expresiones regulares en actividades prácticas de Autómatas y Lenguajes Formales. JENUI 2007
2. Rodger S. H., Finley T. W. JFLAP: An Interactive Formal Languages and Automata Package. Jones and Bartlet Publishers. 2006



informatika
fakultatea



facultad de
informática

Departamento de Lenguajes y
Sistemas Informáticos

Autómatas y Lenguajes Formales

9 créditos = 6 T + 3 P

Ingeniería Técnica en Informática de Sistemas

Programa del curso 2006-2007 - Grupo 16

Página web de la asignatura: <http://ji.ehu.es/alf/>

Profesora: Ana Sánchez

Despacho: 234

Dirección electrónica: ana.sanchez@ehu.es

Horarios de tutorías 1^{er} cuatrimestre:

Lunes 11:00 a 12:30 y 15:30-17:00

Miércoles: 11:00 a 12:30

Jueves: 11:00 a 12:30

TEMARIO

- 1.- Introducción.
- 2.- Jerarquía de Chomsky
 - 2.1. Lenguajes regulares.
 - 2.1.1. Autómatas finitos y gramáticas de tipo 3.
 - 2.1.2. Expresiones regulares.
 - 2.1.3. Aplicaciones.
 - 2.2. Lenguajes independientes del contexto.
 - 2.2.1. Gramáticas independientes del contexto.
 - 2.2.2. Autómatas con pila
 - 2.2.3. Aplicaciones.
 - 2.3.- Lenguajes recursivos y recursivamente enumerables.
 - 2.3.1. Autómatas de Turing.
 - 2.3.2. Gramáticas generales.
- 3.- Introducción a la Teoría de la Computabilidad.
 - 3.1. Problema de parada.
 - 3.2. Otros problemas no computables.
 - 3.3. Computabilidad y modelos abstractos de cómputo

Objetivos generales:

1. Formalización del concepto de computación a través de modelos abstractos: lenguajes formales, autómatas y gramáticas.
2. Comprensión de la existencia de límites intrínsecos a los procesos computacionales.
3. Visión de la influencia de los resultados sobre modelos abstractos en el resto de las ramas de la Informática y principalmente en Compilación.

Objetivos específicos:

1. Soltura en la manipulación de los modelos descritos para la resolución de problemas específicos.
2. Utilización de una batería de algoritmos de transformación de los modelos abstractos.
3. Jerarquización de los mismos según su capacidad de cómputo.

Bibliografía Básica:

1. J. E. HOPCROFT, R. MOTWANI, J. D. ULLMAN. "Introducción a la teoría de Autómatas, Lenguajes y Computación", 2. edición. Pearson educación, 2002
2. D. KELLEY; "Teoría de Autómatas y Lenguajes Formales". Prentice-Hall, 1.995
3. R. BRENA; "Autómatas y Lenguajes. Un enfoque de diseño" . Tec de Monterrey. 2003. [on line] Disponible en <http://lizt.mty.itesm.mx/~rbrena/AyL.html>

Bibliografía Complementaria:

1. H.R. LEWIS; C.H. PAPADIMTRIOU; " Elements of the Theory of Computation", 2. edición. Prentice-Hall, 1998
2. M. SIPSER; " Introduction to the Theory of Computation ". PWS, 1997
3. P. ISASI; P. MARTÍNEZ; D. BORRAJO; "Lenguajes, Gramáticas y Autómatas. Un enfoque práctico". Addison-Wesley, 1.997
4. T. A. SUDKAMP; "Languages and Machines". Addison-Wesley, 1988
5. D. WOOD; "Theory of computation". John Wiley & Sons, Inc., 1987.

Herramientas de visualización:

1. Robinson, M.B., Hamshar, J.A., Novillo, J.A., Novillo, J.E., Duchowski, A.T.: "Java Computability Tool kit (JCT)" en <http://humboldt.sunyit.edu/jct/>
2. Susan H. Rodger: Visual and interactive tools (JFLAP) en <http://www.cs.duke.edu/~rodger/tools/>
3. Stephan Diehl, Andreas Kerren, Torsten Weller: Ganifa en: <http://rw4.cs.uni-sb.de/~ganimal/GANIFA/doc>

Evaluación. De cara a la primera convocatoria de febrero de 2007, habrá dos posibilidades, a elección personal de cada estudiante, con la salvedad de que la segunda posibilidad sólo se puede elegir al principio de curso, durante las dos primeras semanas del mismo:

1. Evaluación tradicional mediante un examen final.

Se realizará por medio de un examen global, a realizar el 22 de enero de 2007 (lunes), de una duración aproximada de 3 h, que recogerá ponderadamente todos los aspectos trabajados en la asignatura.

A esta opción se podrán acoger todos/as los/as alumnos/as, tanto quienes lo elijan inicialmente, como quienes no superen el segundo modo de evaluación.

2. Evaluación continua según modelo de créditos europeos (ECTS).

(Sólo válida para la convocatoria de febrero. En junio sólo hay examen)

Se realizará teniendo en cuenta las ocho actividades de evaluación (AE) planteadas a lo largo del 1^{er} cuatrimestre:

a) Cuatro pruebas parciales escritas sobre los siguientes temas:

AE1) Inducción, Palabras y Lenguajes 1/2 hora

AE4) Lenguajes regulares 1 hora

AE6) Lenguajes Independientes de contexto 1 hora

AE7) Funciones Turing computables. 1/2 hora

Las fechas de realización serán, previsiblemente: AE1) 16 de octubre; AE4) 13 de noviembre; AE6) 11 de diciembre. AE7) 12 de enero

b) Trabajo práctico (AE2) consistente en desarrollar fragmentos de código basados en una notación formal para resolver problemas de reconocimiento de patrones. A entregar no más tarde del 13 de noviembre.

c) Dos entregas de problemas resueltos utilizando la herramienta de visualización JFLAP con modelos de lenguajes regulares (AE3) y modelos independientes de contexto (AE5). A entregar no más tarde del 13 de noviembre la AE3 y 4 de diciembre la AE5.

d) Tutoría personalizada (AE8) con cada estudiante en las que la profesora evaluará el nivel de comprensión y participación en las actividades realizadas hasta el momento de la tutoría.

Cada actividad se valorará como destacado, estándar o insatisfactorio. Se permitirá un máximo de tres actividades insatisfactorias siempre que no incluyan simultáneamente la AE4 y AE6 y no sean las tres individuales.

La nota final obtenida por evaluación continua se publicará el 16 de enero de 2007.

2ª convocatoria: Examen para todos/as los/as alumnos/as: 18-junio-2007 (lunes).

Competencias específicas: Tras la superación de las actividades propuestas serás capaz de:

8. Definir formalmente lenguajes asociados a formatos de datos, instrucciones, ...
9. Utilizar diferentes modelos formales para describir lenguajes concretos
10. Describir formalmente dispositivos abstractos de cómputo
11. Clasificar un lenguaje dentro de la jerarquía de Chomsky
12. Aplicar algoritmos de transformación entre diferentes modelos formales
13. Desarrollar fragmentos de código basados en una notación formal para resolver problemas de reconocimiento de patrones
14. Conocer la existencia de límites teóricos para la resolución computacional de problemas

Dedicación aproximada por cada tema y actividades relacionadas:

Tema	Duración	Actividad	Trabajo presencial Clase	Trabajo semipresencial	Método de Evaluación		Porcentaje	
					AE Grupo	AE Individual		
T1-	4 oct — 9 oct	Teoría + ejercicios	6 h	5 h		AE1 - 1/2 h 16 oct	AE8- TUTORÍA Y ASISTENCIA	10%
T2.1-	11 oct — 9 nov	Teoría + ejercicios	21 h	16 h		AE4 - 1 h 13 de nov		20%
		JFLAP	0,5 h	5 h	AE2 – 13 nov Trabajo con JFLAP			10%
		Práctica		10 h	AE3 – 13 nov Desarrollo de la práctica	15%		
T2.2-	10 nov — 11 dic	Teoría + ejercicios	18 h	14 h		AE6 - 1 h 11 de dic		20%
		JFLAP	0,5	5 h	AE5 – 11 dic Trabajo con JFLAP			10%
T2.3-T3	13 dic — 16 ene	Teoría + ejercicios Lecturas	14 h	10 h		AE7 -1 h 12 de enero;		15%
Totales			60 h	65 h			100%	



Autómatas y Lenguajes Formales

Curso 2006-07 - Grupo 16

Evaluación continua modelo ECTS

AE1

Apellidos, Nombre

Horas presenciales y semipresenciales para la preparación de esta actividad:

Asistencia	Todas	He faltado a 1	He faltado a 2	He faltado a 3 ó más	
He asistido a	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	clases
Participación	No	Si			
He salido a la pizarra a	<input type="checkbox"/>	<input type="checkbox"/>			
Horas de estudio	5 ó más	más de 3 y menos de 5	más de 1 y menos de 3	1 o menos	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

PALABRAS

1. Definición inductiva de palabra sobre un alfabeto Σ

2. La propiedad asociativa de la concatenación dice

$\forall u, v, w \in \Sigma^* (u \bullet v) \bullet w = u \bullet (v \bullet w)$ la demostración de esta propiedad se hace por inducción sobre:

- a) u
- b) v
- c) w
- d) indistintamente u, v ó w

3. La siguiente propiedad se demuestra por inducción. Justifica cada uno de los pasos de la demostración. Propiedad: $\forall \mathbf{x} \in \Sigma^* \ n \in \mathbb{N} \ (\mathbf{x}^n \bullet \mathbf{x}^m = \mathbf{x}^{n+m})$

Demostración por inducción sobre m:

$$\text{P.B.: } \overset{1}{\mathbf{x}^n \bullet \mathbf{x}^0} = \overset{2}{\mathbf{x}^n \bullet \varepsilon} = \overset{3}{\mathbf{x}^n} = \mathbf{x}^{n+0}$$

$$\text{H.I.: } \mathbf{x}^n \bullet \mathbf{x}^k = \mathbf{x}^{n+k}$$

$$\text{P.I.: } \overset{4}{\mathbf{x}^n \bullet \mathbf{x}^{k+1}} = \overset{5}{\mathbf{x}^n \bullet (\mathbf{x}^k \bullet \mathbf{x})} = \overset{6}{(\mathbf{x}^n \bullet \mathbf{x}^k)} \bullet \mathbf{x} = \overset{7}{\mathbf{x}^{n+k}} \bullet \mathbf{x} = \overset{8}{\mathbf{x}^{(n+k)+1}} = \mathbf{x}^{n+(k+1)}$$

Paso 1:

Paso 2:

Paso 3:

Paso 3:

Paso 4:

Paso 5:

Paso 6:

Paso 7:

Paso 8:

4. Definir inductivamente la operación **separar**: $\Sigma^* \times \Sigma^* \rightarrow \Sigma^*$ cuya definición formal es

$$\text{separar}(\mathbf{x}, \mathbf{y}) = \begin{cases} \mathbf{z} & \text{sufijo}(\mathbf{x}, \mathbf{y}) \wedge \mathbf{y} = \mathbf{z} \bullet \mathbf{x} \\ \varepsilon & \text{c.c.} \end{cases}$$

Ejemplos: $\text{separar}(\mathbf{ab}, \mathbf{baab}) = \mathbf{ba}$, $\text{separar}(\varepsilon, \mathbf{ab}) = \mathbf{ab}$, $\text{separar}(\mathbf{abc}, \mathbf{aabba}) = \varepsilon$, $\text{separar}(\mathbf{ab}, \mathbf{ab}) = \varepsilon$

LENGUAJES

5. Considera el alfabeto $\{\mathbf{a}, \mathbf{b}\}$. Indica si las siguientes igualdades son ciertas o falsas:

a) $\{\mathbf{a}\}^* \cup \{\mathbf{b}\}^* = (\{\mathbf{a}\} \cup \{\mathbf{b}\})^*$

b) $\{\mathbf{a}\}^* \bullet \{\mathbf{b}\}^* = (\{\mathbf{a}\} \bullet \{\mathbf{b}\})^*$

c) $\{\mathbf{a}\}^* = \{\mathbf{a}\}^+$

d) $\{\mathbf{a}, \varepsilon\}^* = \{\mathbf{a}\}^+$

6. Considera el lenguaje $L = \{x \in \{0,1\}^* : |x|_{01} = 0\}$. Indica si las siguientes afirmaciones son ciertas o falsas:

- a) $01 \notin L \bullet L$
- b) $L^* = \{0,1\}$
- c) $L^* = L$
- d) $L^* = L - \{\epsilon\}$

7. Sea L cualquier lenguaje. Señala cuál de las siguientes afirmaciones es falsa:

- a) $\epsilon \in L^*$
- b) $\forall x(x \in L \Rightarrow x \in L^*)$
- c) $\exists x(x \notin L \wedge x \in L^*)$
- d) $L^* = \bigcup_{n \geq 0} L^n$

8. Queremos definir un lenguaje de códigos para los departamentos de la UPV-EHU con el siguiente formato: dos letras mayúsculas indicando uno de los tres campus BI, GI, VI, un número de tres dígitos, un guión y un código de tres letras mayúsculas con al menos una vocal. Define formalmente el lenguaje utilizando operaciones de lenguajes y los alfabetos $\Sigma_1 = \{A, B, C, \dots, Z\}$, $\Sigma_2 = \{A, E, I, O, U\}$ y $\Sigma_3 = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Ejemplo: BI127-CLA

9. Sea $L_1 = \{00, 11\}^*$ y $L_2 = \{x \in \{0, 1\}^* : |x|_{00} = 0 \wedge |x|_{11} = 0\}$. Escribe dos ejemplos de palabras pertenecientes a cada uno de estos lenguajes y otros dos de palabras no pertenecientes a cada uno de ellos:

- a) $L_2 - L_1$
- b) $L_1 \bullet L_2$
- c) L_1^*
- d) L_2^R

Competencia

Definir formalmente lenguajes asociados a datos, instrucciones,



Autómatas y Lenguajes Formales

Curso 2006-07 - Grupo 16

Evaluación continua modelo ECTS

AE2 - Actividad en grupo

Plazo máximo de entrega 13 de Noviembre

Apellidos, Nombre

Apellidos, Nombre

Horas totales del grupo para la realización de esta actividad:

Para la instalación _____

Para la comprensión a través de ejemplos del funcionamiento de JFLAP _____

Para el desarrollo del punto 5 de la actividad _____

Para escribir la documentación _____

Otros (especificar) _____

Valoración de JFLAP:

Respecto de su facilidad de uso:

Considero la herramienta

Sencilla

Difícil

Respecto de su utilidad para el aprendizaje en ALF:

Considero la herramienta

Muy útil

Poco útil

Problemas detectados en el uso **No** **Si** (Especificar) _____

VALORACIÓN DE LA ACTIVIDAD

La realización de la actividad me ha servido para

Utilizar diferentes modelos formales para describir lenguajes concretos **No** **Si**

Aprender los algoritmos de transformación entre modelos **No** **Si**

Entender mejor el funcionamiento de autómatas, gramáticas y expresiones regulares **No** **Si**

USO DE LA HERRAMIENTA JFLAP

El objetivo de esta actividad es utilizar una herramienta de visualización que permita asimilar mejor la definición y funcionamiento de algunos de los mecanismos de definición de lenguajes regulares, así como los algoritmos de transformación entre estos modelos formales. Entre las herramientas existentes para ello se ha seleccionado JFLAP (Java Formal Languages and Automata Package) por ser una de las más completas y en actualización continua

También podremos ver que ninguna de las herramientas existentes nos proporciona de manera automática un modelo concreto para definir el lenguaje.

Para la realización de esta actividad los pasos a seguir son:

- T1. Acceso a la página oficial de **JFLAP**: <http://www.jflap.org/>
- T2. Descarga de la herramienta
- T3. Comprensión del funcionamiento de JFLAP para los modelos asociados a lenguajes regulares: AFD, AFN, ϵ -AFN, GRD, GLD y ER a través de la ejecución de ejemplos sencillos.
- T4. Detectar y documentar las posibles diferencias entre las definiciones formales usadas por JFLAP y las definiciones formales vistas en clase.
- T5. Usar JFLAP para demostrar la equivalencia de dos expresiones regulares o de un autómata y una expresión regular. Para ello se ha de seleccionar un lenguaje (no trivial y no resuelto en clase), buscar dos expresiones regulares diferentes que lo generen o un autómata y una expresión regular y comprobar con JFLAP que los dos modelos construidos son equivalentes (viendo que los autómatas mínimos coinciden).

ENTREGA

La documentación a entregar debe contener el coste y valoración de la actividad de este documento y los siguientes apartados:

- 1) un breve apartado sobre JFLAP, sus características y posibilidades relacionado con el punto cuatro
- 2) un segundo apartado con alguno de los ejemplos simples de uso de la herramienta que se piden en el punto tres
- 3) un tercer apartado con todos los pasos seguidos en el desarrollo del punto cinco

Se enviará por correo electrónico a través del correo institucional. El mensaje llevará por título AE2 y el apellido de las dos personas del grupo y contendrá también los ficheros creados en JFLAP perfectamente identificables.

EVALUACIÓN

Se tendrán en cuenta tanto los ejemplos trabajados como la documentación presentada. La complejidad del lenguaje elegido para el paso 5 exigirá el uso de un AF mínimo con al menos cuatro estados.



Autómatas y Lenguajes Formales

Curso 2006-07 - Grupo 16

Evaluación continua modelo ECTS

AE3 - Actividad en grupo

Plazo máximo de entrega 13 de Noviembre

USO DE EXPRESIONES REGULARES EN JAVASCRIPT

Las **expresiones regulares** son descriptores de lenguajes regulares. Una de sus utilidades más conocidas es en la realización de búsquedas de las cadenas o palabras que describen o que se ajustan a dicho “patrón”, como lo hacen, en mayor o menor medida, los editores de textos. Otra utilidad es la comprobación de que una cadena de texto cumple ciertos requisitos, por ejemplo a la hora de rellenar formularios. Los patrones o descripciones que utilizan estas aplicaciones están relacionados con las expresiones regulares.

Ya hemos visto la definición y ejemplos de expresiones regulares. Ahora se trata de utilizarlas con una finalidad y en un contexto concreto.

El objetivo específico de esta actividad es conocer el manejo de expresiones regulares que permite un lenguaje concreto: **JavaScript**.

Para la realización de esta actividad los pasos a seguir son:

T6. Búsqueda de información sobre JavaScript y las expresiones regulares en Javascript.

Algunos ejemplos de direcciones útiles para esta tarea:

- Página web que describe el lenguaje JavaScript
<http://www.unav.es/cti/manuales/TutorialJavaScript/indices/>
- Página web que describe las expresiones regulares y sus aplicaciones
http://es.wikipedia.org/wiki/Expresi%C3%B3n_regular
- Página web que describe la sintaxis de las expresiones regulares y sus métodos
http://developer.mozilla.org/es/docs/Gu%C3%ADa_JavaScript_1.5:Expresiones_Regulares
- Capítulo 5 del libro de varios autores “Programación de Aplicaciones Web” que incluye creación de páginas Web interactivas: JavaScript

T7. Desarrollo de al menos un programa con JavaScript

T8. Desarrollo de al menos un programa con JavaScript que incluya el uso de expresiones regulares.

T9. Creación de un documento HTML directamente procesable en un navegador, que solicite un correo electrónico y detecte si responde al patrón de correo utilizado por la UPV-EHU para asignar direcciones de e-mail a sus estudiantes.

T10. Documentar el programa de la tarea T4 y las expresiones regulares utilizadas

T11. Reflexionar sobre las diferencias entre nuestra definición de expresión regular y las expresiones regulares utilizadas en JavaScript

T12. Estimar justificadamente el coste (en tiempo) que supondría realizar la tarea T4 con un lenguaje de programación que no disponga de expresiones regulares.

ENTREGA

Se entregará el documento HTML de la tarea T4 y otro documento que contendrá un apartado por cada una de las tareas especificadas y el coste y valoración de la actividad en el modo descrito en la siguiente página.

Se enviará por correo electrónico a través del correo institucional. El mensaje llevará por título AE3 y los apellidos de las dos personas del grupo.

EVALUACIÓN

Se tendrán en cuenta tanto las tareas realizadas como la documentación presentada.

P1: Apellidos, Nombre

P2: Apellidos, Nombre

COSTE DE LA ACTIVIDAD AE3

Horas totales para la realización de esta actividad:

	más de tres		entre dos y tres		entre una y dos		una o menos		Total
	P1	P2	P1	P2	P1	P2	P1	P2	
T1:	<input type="checkbox"/>	___							
T2:	<input type="checkbox"/>	___							
T3:	<input type="checkbox"/>	___							
T4:	<input type="checkbox"/>	___							
T5:	<input type="checkbox"/>	___							
T6:	<input type="checkbox"/>	___							
T7:	<input type="checkbox"/>	___							

	más de tres		entre dos y tres		entre una y dos		una o menos	
	P1	P2	P1	P2	P1	P2	P1	P2
Elaborar la documentación:	<input type="checkbox"/>							
Total	___							
Otros (especificar):	<input type="checkbox"/>							
Total	___							

Total P1: ___

Total P2: ___

Total Grupo: ___

VALORACIÓN DE LA ACTIVIDAD

La realización de la actividad me ha servido para	P1		P2	
Entender mejor la definición de expresiones regulares	No	Si	No	Si
Comprender la utilidad real de las expresiones regulares	No	Si	No	Si

AUTOEVALUACIÓN

Valora tu capacidad para el desarrollo de la competencia trabajada

Soy capaz de desarrollar fragmentos de código basados en una notación formal para resolver problemas de reconocimiento de patrones	P1		P2	
	Si	No	Si	No



Autómatas y Lenguajes Formales

Curso 2006-07 - Grupo 16

17 de Noviembre de 2006

AE4

Apellidos, Nombre

1. Definición inductiva de expresión regular sobre un alfabeto Σ

2. Asocia cada función de transición con el autómata correspondiente

- | | |
|-----------------------|--|
| e) AFD | 1) $\delta: Q \times \Sigma \cup \{\varepsilon\} \longrightarrow \wp(Q)$ |
| f) AFDt | 2) $\delta: Q \times \Sigma \longrightarrow \wp(Q)$ |
| g) AFN | 3) $\delta: Q \times \Sigma \longrightarrow Q$ |
| h) ε -AFN | 4) $\delta: Q \times \Sigma \rightarrow Q$ |

3. Di si las siguientes afirmaciones son ciertas o falsas según el tipo de autómata al que se refieran:

- | | ε -AFN | AFN |
|---|--------------------|-----|
| a) $\varepsilon \in L(M) \Leftrightarrow q_0 \in F$ | | |
| b) $\delta(q,s) = \delta^*(q,s)$ | | |
| c) $x \in L(M) \Leftrightarrow \delta^*(q,x) \cap F \neq \emptyset$ | | |
| d) $\delta^*(q,\varepsilon) = q$ | | |

4. Sea G una gramática cuyas reglas de producción son $S \rightarrow \mathbf{a}bS$ | **a**. Describe $L(G)$.

5. Escribe una expresión regular que genere el lenguaje $\{x \in \{a,b\}^* : |x|_a \leq 3\}$

6. Sea δ la función de transición de un AFD. A continuación tienes una demostración de la siguiente propiedad:

$$\forall x,y \in \Sigma^* \delta^*(q, x \bullet y) = \delta^*(\delta^*(q, x), y)$$

$$\text{Paso básico: } \delta^*(q, x \bullet \epsilon) \stackrel{1}{=} \delta^*(q, x) \stackrel{2}{=} \delta^*(\delta^*(q, x), \epsilon)$$

$$\text{Hipótesis de inducción: } \delta^*(q, x \bullet y) = \delta^*(\delta^*(q, x), y)$$

Paso de inducción:

$$\delta^*(q, x \bullet ys) \stackrel{3}{=} \delta(\delta^*(q, x \bullet y), s) \stackrel{4}{=} \delta(\delta^*(\delta^*(q, x), y), s) \stackrel{5}{=} \delta^*(\delta^*(q, x), ys)$$

Justifica cada uno de los pasos de la demostración.

Paso 1:

Paso 2:

Paso 3:

Paso 4:

Paso 5:

7. Demuestra que el lenguaje sobre el alfabeto $\{0,1\}$ formado por aquellas palabras que contienen la subpalabra **00** pero no contienen tres **0**'s seguidos es **regular**.

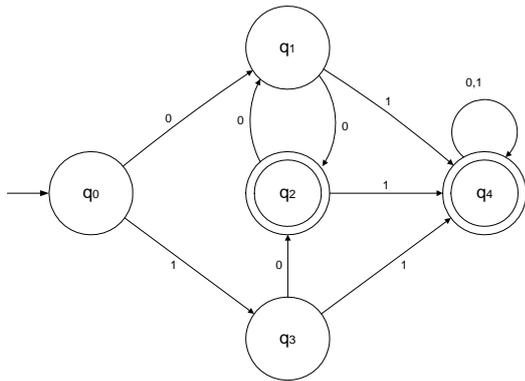
8. Resuelve las siguientes ecuaciones sobre expresiones regulares:

a) $X = aX \cup b^+aX \cup bX$

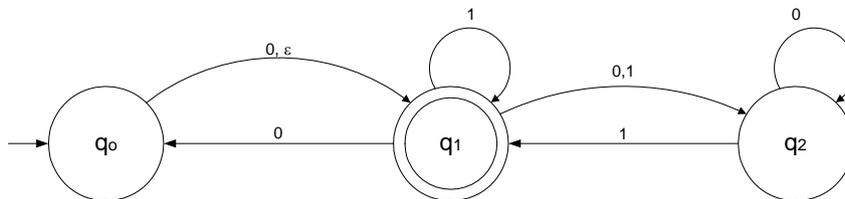
b) $X = abX \cup b \cup aa$

9. Dado el siguiente autómata decide y **justifica** si los siguientes pares de estados son equivalentes o no:

- a) q_1 y q_3
- b) q_2 y q_4



10. Aplica el **algoritmo** visto en clase para eliminar las transiciones vacías del siguiente ϵ -AFN:



11. Considera la siguiente expresión regular de JavaScript

$\text{^b(a|b)*aa?a?}$$

Decide si las siguientes palabras son generadas por dicha expresión:

- | | | | |
|---------------|-------|----------------|-------|
| a) aba | Si No | b) baaa | Si No |
| b) ba | Si No | c) aa | Si No |
| c) bb | Si No | d) ϵ | Si No |

12. Sobre el funcionamiento de **JFLAP** indica si las siguientes afirmaciones son ciertas:

- a) Para dibujar un autómata con JFLAP se debe seleccionar el tipo de autómata (AFD, AFN, ϵ -AFN) que se pretende dibujar.
- b) Las expresiones regulares utilizan el punto para la concatenación.

Horas presenciales y semipresenciales para la preparación de esta actividad:

Asistencia	Todas	He faltado entre una y cuatro	He faltado entre cinco y diez	He faltado a más de diez
He asistido a	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> clases
Participación	No	Entre una y tres veces	Cuatro o más veces	
He salido a la pizarra a	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Horas de estudio	Más de diecisiete	Entre trece y dieciséis	Entre seis y doce	Entre una y cinco
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

No contabilizar las de JFLAP(AE2) y JavaScript (AE3)

AUTOEVALUACIÓN

Valora de 0 a 3 tu capacidad para el desarrollo de las competencias trabajadas

Utilizar diferentes modelos formales (AF, AFN, ER, ...) para describir lenguajes regulares concretos	0 1 2 3
Describir formalmente dispositivos abstractos asociados a los lenguajes regulares (AF, AFN, ER, ...)	0 1 2 3
Aplicar algoritmos de transformación entre los modelos formales asociados a los lenguajes regulares	0 1 2 3



Autómatas y Lenguajes Formales

Curso 2006-07 - Grupo 16

Evaluación continua modelo ECTS

AE5 - Actividad en grupo

Plazo máximo de entrega 13 de Diciembre

USO DE LA HERRAMIENTA JFLAP PARA GIC

El objetivo de esta actividad es utilizar la herramienta JFLAP (Java Formal Languages and Automata Package) pero ahora para asimilar mejor la definición y funcionamiento de algunos de los conceptos relacionados con los **lenguajes independientes de contexto**.

La realización de esta actividad incluye dos tareas:

- T1. Comprensión del funcionamiento de JFLAP para gramáticas independientes de contexto (GIC) a través de ejemplos. Elige algún lenguaje independiente de contexto no resuelto en clase y construye una gramática que lo genere explicando el razonamiento seguido para su construcción. Comprueba con varios ejemplos qué palabras genera y cuáles no. Redactarlo como **apartado 1** del documento a entregar.
- T2. Usar JFLAP para repasar cada uno de los pasos necesarios para simplificar una gramática independiente de contexto.

T2.1. Busca una gramática con producciones nulas, unitarias y símbolos inútiles y simplifícala aplicando y explicando todos los pasos necesarios.

T2.2. Haz lo mismo utilizando JFLAP siguiendo las transformaciones que realiza paso a paso y comprueba si el método y la gramática simplificada coinciden con lo realizado en el punto T2.1 o hay diferencias.

Redactar lo realizado en T2 como **apartado 2** del documento a entregar.

ENTREGA

La documentación a entregar debe contener los apartados especificados en T1 y T2 y el coste y valoración de la actividad de este documento. Identifica el documento con el nombre de los autores.

Se enviará por correo electrónico a través del correo institucional. El mensaje llevará por título AE5 y el apellido de las dos personas del grupo y contendrá también los ficheros creados en JFLAP perfectamente identificables.

EVALUACIÓN

Se tendrán en cuenta tanto los ejemplos trabajados como la documentación presentada.

Apellidos, Nombre

Apellidos, Nombre

Horas totales del grupo para la realización de esta actividad:

Para el desarrollo del punto T1 _____

Para el desarrollo del punto T2 _____

Otros (especificar) _____

Valoración de JFLAP para lenguajes independientes de contexto:

Respecto de su facilidad de uso:

Considero la herramienta

Sencilla

Difícil

Respecto de su utilidad para el aprendizaje en ALF:

Considero la herramienta

Muy útil

Poco útil

Problemas detectados en el uso **No** **Si** (Especificar) _____

VALORACIÓN DE LA ACTIVIDAD

La realización de la actividad me ha servido para

Comprender mejor el funcionamiento de las GIC

No Si

Aprender el algoritmo de simplificación de GIC

No Si



Autómatas y Lenguajes Formales

Curso 2006-07 - Grupo 16

15 de Diciembre de 2006

AE6

Apellidos, Nombre

1. Define cuando un símbolo X es un símbolo útil en una gramática G

2. Busca un ejemplo de palabra perteneciente a $L(G)$, y un ejemplo de palabra no perteneciente a $L(G)$ y describe $L(G)$.

$$G = (\{S, A, B\}, \{a, b\}, \{S \rightarrow aSb \mid bBA \mid BAa; A \rightarrow Aa \mid \varepsilon; B \rightarrow bB \mid \varepsilon\})$$

3. Los dos lenguajes definidos a continuación son independientes de contexto. Elige uno de ellos y demuéstralo.

e) $\{xcy: x, y \in \{a, b\}^* \wedge x^R \text{ es subpalabra de } y\}$

f) $\{a^n b^m: n=m \vee n=3*m\}$

4. Cuál de las siguientes definiciones se corresponde con la función de transición de los autómatas con pila definidos en clase

- c) $\delta: Q \times \Sigma \times \Gamma \cup \{\perp\} \rightarrow \wp(Q \times \Gamma^*)$
- d) $\delta: Q \times \Sigma \times \Gamma \cup \{\perp\} \cup \{\varepsilon\} \rightarrow Q \times \Gamma^*$
- e) $\delta: Q \times \Sigma \times \Gamma \cup \{\perp\} \rightarrow Q \times \Gamma^*$
- f) $\delta: Q \times \Sigma \times \Gamma \times \{\perp\} \rightarrow \wp(Q \times \Gamma^*)$

5. Di si las siguientes afirmaciones son ciertas o falsas.

- g) Ningún lenguaje independiente de contexto es regular
- h) La derivación de una palabra por una gramática en FNG tiene tantos pasos como su aceptación por un AP.
- i) Si G es una gramática tal que $\varepsilon \notin L(G)$, entonces G no puede tener producciones nulas.

6. Sea $M = (\{q_0\}, \{a, b\}, \{A\}, \delta, \{q_0\})$ un autómata con pila. Di si las siguientes afirmaciones son ciertas o falsas.

- g) Como su único estado es final el autómata acepta cualquier palabra
- h) $\varepsilon \in L(M)$
- i) Si $(q_0, A) \in \delta(q_0, a, \perp)$ entonces $L(M) \neq \{\varepsilon\}$.

7. Clasifica los siguientes lenguajes indicando si son regulares (R) o no (NR) y si son independientes de contexto (IC) o no (NIC):

- | | R | NR | IC | NIC |
|---------------------------------|---|----|----|-----|
| j) $\{a^i b^i : i > 0\}$ | | | | |
| k) $\{a^i b^j a^i : i, j > 0\}$ | | | | |
| l) $\{a^i b^i a^i : i > 0\}$ | | | | |
| m) $\{a^i b^j a^k : i, j > 0\}$ | | | | |

8. Demuestra que la siguiente gramática es ambigua y busca otra equivalente no ambigua:

$$G = (\{S\}, \{a, b, +, *\}, \{S \rightarrow a \mid b \mid S+S \mid S^*S\})$$

9. Sabiendo como funciona el programa JFLAP para simplificar una gramática, indica si las siguientes afirmaciones son ciertas o falsas:

- n) Permite elegir el orden de aplicación de los algoritmos
- o) Permite recursión del símbolo inicial
- p) Permite eliminar la recursión a la izquierda

10. Considera el autómata con pila $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ con $Q = \{q_0, q_f\}$; $F = \{q_f\}$; $\Gamma = \{A\}$; $\Sigma = \{a, b\}$ y δ como sigue:

$$\begin{array}{ll} \delta(q_0, a, \perp) = \{(q_0, A), (q_f, \epsilon)\} & \delta(q_f, a, A) = \{(q_f, \epsilon)\} \\ \delta(q_0, a, A) = \{(q_0, AA), (q_f, A)\} & \delta(q_f, b, A) = \{(q_f, \epsilon)\} \\ \delta(q_0, b, \perp) = \{(q_0, A)\} & \delta(q_0, b, A) = \{(q_0, AA)\} \end{array}$$

- a) Da todos los cálculos posibles de M para la palabra **aba**
- b) Demuestra que **abb** no pertenece a $L(M)$ y **baa** pertenece
- c) Describe $L(M)$

11. Transforma la gramática G cuyo conjunto de reglas viene dado a continuación para obtener una gramática equivalente sin recursión a la izquierda:

$$\begin{array}{ll} S \rightarrow AC & B \rightarrow Bb \mid Cb \\ A \rightarrow AaB \mid AaC \mid B \mid a & C \rightarrow cC \mid c \end{array}$$

Horas presenciales y semipresenciales para la preparación de esta actividad:

Asistencia He asistido a	Todas <input type="checkbox"/>	He faltado a una, dos o tres <input type="checkbox"/>	He faltado a cuatro, cinco o seis <input type="checkbox"/>	He faltado a más de seis <input type="checkbox"/>	clases
Participación He salido a la pizarra a	No <input type="checkbox"/>	Entre una y tres veces <input type="checkbox"/>	Cuatro o más veces <input type="checkbox"/>		
Horas de estudio No contabilizar las de JFLAP(AE5)	Más de catorce <input type="checkbox"/>	Entre once y catorce <input type="checkbox"/>	Entre seis y diez <input type="checkbox"/>	Entre una y cinco <input type="checkbox"/>	

AUTOEVALUACIÓN

Valora de 0 a 3 tu capacidad para el desarrollo de las competencias trabajadas

Utilizar gramáticas para describir lenguajes independientes de contexto	0 1 2 3
Utilizar autómatas con pila para describir lenguajes independientes de contexto	0 1 2 3
Aplicar algoritmos de transformación a las gramáticas	0 1 2 3
Clasificar un lenguaje dentro de la jerarquía de Chomsky	0 1 2 3
Describir formalmente dispositivos abstractos asociados a los lenguajes independientes de contexto	0 1 2 3



Autómatas y Lenguajes Formales

Curso 2006-07 - Grupo 16

12 de Enero de 2007

AE7

Apellidos, Nombre

10. Define la forma de las reglas en una gramática G de tipo 0 ó recursivamente enumerable

11. Enuncia la Tesis de Church y sus implicaciones.

12. Enuncia un ejemplo de problema no computable.

13. Representa gráficamente la Jerarquía de Lenguajes de Chomsky indicando los tipos de lenguaje y su relación de contenido.

14.Cuál de las siguientes definiciones se corresponde con la función de transición de los autómatas de Turing definidos en clase

q) $\delta: Q \times \Sigma \cup \Gamma \longrightarrow \wp(Q \times \Sigma \cup \Gamma) \times \{R, L, S\}$

r) $\delta: Q \times \Sigma \cup \Gamma \longrightarrow \wp(Q \times \Sigma \cup \Gamma \times \{R, L, S\})$

s) $\delta: Q \times \Sigma \cup \Gamma \longrightarrow Q \times \Sigma \cup \Gamma \times \{R, L, S\}$

t) $\delta: Q \times \Sigma \cup \Gamma \longrightarrow Q \times \Sigma \cup \Gamma \cup \{R, L, S\}$

15. Sea $M = (\{q_0\}, \{a, b\}, \{\square\}, \delta, \{q_0\})$ un autómata de Turing. Di si las siguientes afirmaciones son ciertas o falsas.

- a) Como su único estado es final el autómata acepta cualquier palabra
- b) Como q_0 es final $\varepsilon \in L(M)$
- c) Si $\delta(q_0, \square) = (q_0, \square, S)$ entonces $L(M) \neq \emptyset$.
- d) M puede tener cómputos infinitos

16. Clasifica los siguientes lenguajes indicando si son regulares (R) o no (NR) y si son independientes de contexto (IC) o no (NIC) o recursivamente enumerable (RE). Rellena cada casilla con SI ó NO:

	R	NR	IC	NIC	RE
u) $\{a^i b^j : i > 0\}$					
v) $\{a^i b^j a^i : i, j > 0\}$					
w) $\{a^i b^j a^i : i > 0\}$					
x) $\{a^i b^j a^k : i, j > 0\}$					

17. Busca una gramática tal que $L(G) = \{a^n b^n c^n a^n : n \geq 0\}$, y da una derivación para la palabra $a^4 b^4 c^4 a^4$

18. Sea $M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{a, b, c\}, \{\square, \#\}, \delta, \{q_6\})$ un autómata de Turing cuya función de transición δ se encuentra a continuación. Comprueba (con los cómputos correspondientes) que la palabra $abacaba \in L(M)$ y $abcabc \notin L(M)$. Describe $L(M)$ y explica brevemente cuál es el cometido de los estados q_2 y q_6 .

	a	b	c	#	\square
q_0	(q_1, \square, R)	(q_2, \square, R)	(q_6, \square, R)	-	(q_0, \square, S)
q_1	(q_1, a, R)	(q_1, b, R)	(q_3, c, R)	-	(q_0, \square, S)
q_2	(q_2, a, R)	(q_2, b, R)	(q_4, c, R)	-	(q_0, \square, S)
q_3	$(q_5, \#, L)$	(q_0, \square, S)	(q_0, \square, S)	$(q_3, \#, R)$	(q_0, \square, S)
q_4	(q_0, \square, S)	$(q_5, \#, L)$	(q_0, \square, S)	$(q_4, \#, R)$	(q_0, \square, S)
q_5	(q_5, a, L)	(q_5, b, L)	(q_5, c, L)	$(q_5, \#, L)$	(q_0, \square, R)
q_6	(q_0, \square, S)	(q_0, \square, S)	(q_0, \square, S)	$(q_6, \#, R)$	(q_6, \square, S)

19. Demuestra que la siguiente función es Turing-computable mediante un diagrama de máquinas de Turing. Explica en un párrafo el algoritmo elegido para construir la máquina.

$$f: \{a, b, c\}^* \rightarrow \{a, b, c\}^*$$

$$f(x, y) = a^{|x|} a^{|y|} b$$

Por ejemplo $f(abac, abacab) = aabb$, $f(\varepsilon, aaca) = \varepsilon$, $f(cac, ccc) = a$, $f(cccb, cbc) = bb$