

Experiencias de uso y evaluación de una herramienta de apoyo a la enseñanza de Teoría de Autómatas y Lenguajes Formales

Resumen

Una materia habitual en los planes de estudios de Ingeniería Informática es la Teoría de Autómatas y Lenguajes Formales. Los alumnos suelen tener problemas a la hora de afrontar la preparación de esta materia, ya que la encuentran muy abstracta y apartada de otras materias que estudian en paralelo. Para facilitar la preparación de esta materia, se desarrolló la herramienta SELFA-Pro, que permite resolver ejercicios de la mayor parte del temario de la materia. En este artículo se presenta la experiencia reciente de uso de la herramienta en un entorno real, tanto en cuanto a estadísticas y su relación con los resultados académicos, como en cuanto a opinión subjetiva de los alumnos.

Abstract

A subject that is usually found in the studies of Computer Science is the Theory of Automata and Formal Languages. Students usually have some problems when facing this subject, as they find it very abstract and distant from other subjects they study at the same time. In order to make easier the preparation of the subject, the SELFA-Pro tool was developed. This tool allows students solving problems of most themes in the subject. In this paper we show some recent experiences of use of this tool in real studies. We include both statistics of use with their possible relationship with academic results and subjective opinion of the students that work with the tool.

Palabras clave

Teoría de Autómatas y Lenguajes Formales, Herramienta on-line, Experiencias de uso.

1. Introducción y motivación

La Teoría de Autómatas y Lenguajes Formales es un campo clásico y multidisciplinar dentro de los planes de estudios de Ingeniería Informática de la mayoría de universidades, tanto en el marco actual como en anteriores regulaciones de los planes de estudios universitarios. Se trata de una materia que provee los fundamentos teóricos de la disciplina de la Ingeniería Informática y ayuda a conocer su potencia y sus limitaciones. Los alumnos que cursan esta materia suelen encontrarse problemas a la hora de afrontarla, por tratarse de una materia que maneja numerosos conceptos que requieren de una gran capacidad de abstracción y que en ocasiones pueden resultarles alejados del resto de materias del plan de estudio, aunque realmente no sea así.

Con el fin de ayudar a los alumnos a encarar el estudio y la preparación de la materia, y en particular del manejo de los conceptos antes comentados y de sus operaciones y algoritmos asociados, se desarrolló la herramienta SELFA y posteriormente su extensión SELFA-Pro. Esta herramienta incluye la definición de un Lenguaje Específico del Dominio que permite la definición de gramáticas, expresiones regulares, autómatas finitos y autómatas con pila, así como la aplicación de los algoritmos y operaciones que los alumnos van a manejar durante su estudio de la materia. Con respecto a otras herramientas similares que afrontan estos objetivos, SELFA-Pro presenta como uno de sus puntos fuertes el uso de una notación basada en variables para encadenar la aplicación de algoritmos y operaciones, lo cual permite a los estudiantes utilizar una notación similar a la que utilizan en los lenguajes de programación para resolver problemas complejos y experimentar con los resultados obtenidos.

En este artículo se presentan las experiencias de uso de la herramienta SELFA-Pro en una asignatura que cubre la materia descrita durante cuatro cursos

académicos. Se analizan el uso real que se ha hecho de la herramienta por parte de los estudiantes, las operaciones que realmente estos emplean, la opinión subjetiva de los estudiantes y la relación de todo ello con los resultados académicos obtenidos en este tiempo. De esta forma, se ha emitido un primer balance de evaluación de la herramienta, obteniendo sus puntos fuertes y débiles y sus posibilidades de mejora de cara a posibles nuevas versiones de la misma.

El resto del artículo se estructura como sigue: en la Sección 2, se realiza una breve descripción de la materia de Teoría de Autómatas y Lenguajes Formales para enmarcar el trabajo. En la Sección 3, se comentan trabajos relacionados con el actual, tanto propuestas similares como otras que pueden resultar de interés. En la Sección 4, se describe la herramienta SELFA-Pro, tanto desde el punto de vista de su desarrollo como desde su uso para la preparación de la materia. En la Sección 5 se comentan los resultados obtenidos durante los cuatro cursos académicos en los cuales se viene utilizando la herramienta como ayuda en la asignatura de Teoría de la Computación del Grado en Ingeniería Informática de la Universidad de xxxxx¹. Finalmente, la Sección 6 incluye las conclusiones obtenidas durante el desarrollo de las experiencias comentadas y las líneas de trabajo futuro que se van a explorar próximamente en este sentido.

2. La materia de Teoría de Autómatas y Lenguajes Formales

La materia de Teoría de Autómatas y Lenguajes Formales (TALF) está normalmente presente en los planes de estudios de Ingeniería Informática, ya que proporciona a los alumnos los fundamentos teóricos de la disciplina de la Informática y ayuda a conocer y a estudiar los fundamentos de la Computación, su potencia y sus limitaciones [5, 6, 7, 8]. Actualmente, esta materia se estudia en ocasiones en asignaturas con el nombre de Teoría de la Computación, para marcar más el carácter de fundamento de las Ciencias de la Computación que se le viene dando en los últimos tiempos, como se observa por ejemplo en el plan de estudios del Grado en Ingeniería Informática de la Universidad de xxxxxx. Aun así, la materia tiene también una gran utilidad a la hora de enseñar una serie de conceptos presentes en los temas que desarrollan dichos conceptos teóricos. Por ejemplo, la descripción de lenguajes mediante expresiones regulares, que se utiliza en numerosos ámbitos de la Informática como mecanismo para describir patrones, la definición de autómatas finitos, que permite ejemplificar el uso de máquinas de estados, o la descrip-

ción de lenguajes mediante gramáticas y el posterior desarrollo de procesadores de lenguajes para reconocer los lenguajes así descritos. Este último era, precisamente, el enfoque que más se le daba hasta hace unos años a este tipo de asignaturas, enfocadas entonces al desarrollo de compiladores.

Con el replanteamiento del mapa de titulaciones que se llevó a cabo en España a partir de la década anterior, la asignatura en la cual se tratan los temas de TALF se ha ubicado en momentos distintos dentro de los planes de estudios, y en ocasiones incluso con carácter distinto. Por poner dos ejemplos, mientras que en la Universidad de xxxxxx la asignatura Teoría de la Computación es considerada de Formación Básica y se ubica en el primer cuatrimestre del segundo curso del Grado, en la Universidad de Castilla-La Mancha la asignatura Teoría de Autómatas y Computación forma parte de la intensificación de Computación y se ubica en el segundo cuatrimestre del tercer curso del Grado. Esta disparidad está en parte provocada por el hecho de que el Libro Blanco sobre el título de Grado en Ingeniería Informática publicado por la ANECA² y que se tomó como referencia inicial para la elaboración de los planes de estudios no hacía referencia expresa a TALF.

Los planes de estudios de las asignaturas de Teoría de Autómatas y Lenguajes Formales o Teoría de Computación suelen estar organizados de la siguiente forma:

- Introducción a los lenguajes formales y las gramáticas. Inicialmente, se definen los conceptos de lenguaje formal, alfabeto y palabra, junto con las operaciones básicas sobre palabras y lenguajes. A continuación, se introduce el concepto de gramática y la clasificación de las mismas según los trabajos de Noam Chomsky.
- Lenguajes y gramáticas regulares. Este bloque trata sobre los lenguajes regulares, las gramáticas que los definen y los mecanismos de descripción y reconocimiento asociados. En concreto, se presentan las expresiones regulares y los autómatas finitos, en sus diversas variantes.
- Lenguajes y gramáticas libres de contextos. Al igual que en el bloque anterior, se trata sobre las distintas formas de trabajar con lenguajes libres de contexto, definiéndolos con gramáticas de dicho tipo y reconociéndolos con autómatas con pila.
- Máquinas de Turing. Se describe el modelo básico de Máquina de Turing, usándolo como puente entre los bloques anteriores, más centrados en el manejo de lenguajes formales, y el posterior, centrado ya en el concepto de computabilidad.

¹ Se omite el nombre de la Universidad para cumplir los requisitos marcados por la organización.

²http://www.aneca.es/var/media/150388/libroblanco_jun05_informatica.pdf

- Introducción a la computabilidad y la complejidad computacional. En este bloque, y utilizando como fundamento lo aprendido acerca de Máquinas de Turing en el bloque anterior, se discute sobre los límites de la computabilidad, se presenta la Tesis de Turing-Church y se tratan aspectos relacionados con la categorización de problemas según su decidibilidad y según el tiempo de resolución.

Como se observa, el temario abarca aspectos muy distintos a aquellos de las asignaturas que pueden compartir temporalidad dentro del plan de estudios. Por ejemplo, dentro del plan de estudios de la Universidad de xxxxxxx, esta asignatura se imparte a la vez que Estructuras de Datos y Algoritmos, Programación de Sistemas Concurrentes y Distribuidos, Redes de Computadores y Sistemas Operativos. Es decir, dos asignaturas englobadas dentro del área de la programación y las primeras asignaturas que introducen a los estudiantes en las materias de Sistemas Operativos y Redes. Además, del análisis de los resultados de las encuestas de evaluación de la enseñanza de la Escuela xxxxxxx³ se observa que a los alumnos les cuesta comprender la importancia de esta materia y contextualizarla en la titulación. Así pues, se hace necesario replantear la enseñanza de esta asignatura y utilizar mecanismos familiares para los alumnos para que así puedan asimilar mejor los conceptos de la asignatura.

3. Trabajos relacionados

Existen numerosas herramientas para el manejo de autómatas finitos que han sido utilizadas o lo están siendo en el ámbito de la docencia sobre Autómatas Finitos y Lenguajes Formales. Muchas de estas herramientas son gráficas, lo cual aunque en principio podría suponer una ventaja, también puede convertirse en una limitación, como comentaremos más adelante.

Una de las herramientas más populares para la docencia de Autómatas y Lenguajes Formales es JFLAP [11]. Es una herramienta gráfica bastante completa que permite trabajar con autómatas finitos, autómatas a pila, máquinas de Turing, gramáticas y expresiones regulares, implementando un buen número de algoritmos sobre esos componentes. Otra herramienta gráfica interesante es THOTH [3], que también permite trabajar con la mayoría de estos elementos de la materia de autómatas y lenguajes formales y está muy orientada al aprendizaje de los algoritmos que implementa. También merece la pena destacar la herramienta descrita en [2], que se desarrolló con un gran énfasis en una interfaz de usuario amigable, pero

³ Se omite el nombre del centro para cumplir con los requisitos de la organización.

que se ha especializado en autómatas finitos, y no cubre otros conceptos presentes en la materia. Finalmente, la herramienta PFM_UOC [4] trabaja también sobre autómatas finitos, y permite su representación gráfica a partir de la tabla de transiciones, aunque carece de otras funcionalidades como la simulación del funcionamiento del autómata.

Como antes comentábamos, el hecho de que estas herramientas sean fundamentalmente gráficas tiene sus ventajas de cara a su mayor usabilidad, pero eso puede redundar en problemas en ciertas situaciones en las cuales se pretenda enlazar la ejecución de diversos algoritmos sobre elementos diferentes, como por ejemplo la minimización de autómatas finitos obtenidos a partir de una expresión regular o gramática regular. Por otro lado, existen varios trabajos que han probado que el uso de notaciones formales y el hecho de asemejar ciertos elementos de tales notaciones a los lenguajes de programación, que es algo ya conocido y trabajado por parte de los alumnos, puede redundar en beneficio del proceso de aprendizaje [9, 10].

Considerando todo lo anteriormente, y otros aspectos que algunas de las herramientas ya mencionadas no cubren, como la recopilación de estadísticas para el seguimiento de los alumnos, se desarrolló la herramienta SELFA [1], y posteriormente su extensión SELFA-Pro [12]. En la siguiente sección se va a hablar más en detalle de la herramienta SELFA-Pro, para posteriormente mencionar algunas experiencias de uso y resultados obtenidos en la Sección 5.

4. La herramienta SELFA-Pro

Como se ha comentado ya, SELFA-Pro es la versión actual de la herramienta para el aprendizaje de la materia de TALF, y evolucionó de una versión anterior denominada SELFA. Desde el primer momento del desarrollo de esta herramienta, se buscó como objetivo el disponer de un sistema que aceptase ejercicios de dicha materia y los resolviese, mostrando las soluciones de la manera más clara y sencilla posible de cara al estudiante. Es una herramienta *on line*, de forma que no necesita instalación y es accesible a cualquiera a través de su navegador *web*. Además, SELFA-Pro incluye el registro e identificación de usuarios, de forma que posteriormente se registra la actividad de dichos usuarios en cuanto a número de ejercicios ejecutados de cada tipo, permitiendo así al profesor tener un control personalizado del uso real de la herramienta, así como estadísticas más globales del uso de la misma.

La herramienta trabaja con las construcciones más habituales en TALF, como son las gramáticas, las expresiones regulares, los autómatas finitos y los autómatas con pila. Para cada construcción, se permite el planteamiento y resolución de un buen número

de ejercicios, que cubren prácticamente todos los algoritmos y ejercicios que se suelen tratar en las asignaturas que cubren esta materia. En el Cuadro 1 se detallan las operaciones que incluye SELFA-Pro, junto con el nombre que han recibido en la herramienta, que está en inglés. Se han utilizado las abreviaturas ER por expresión regular, GLC por gramáti-

ca libre de contexto, AFD por autómata finito determinista, AFND por autómata finito no determinista, AFND- ϵ por autómata finito no determinista con transiciones nulas y AF por autómata finito sin especificar el tipo.

| Nombre | Descripción |
|--|---|
| Operaciones sobre expresiones regulares | |
| alternation | Hacer la unión de dos ER |
| concatenation | Hacer la concatenación de dos ER |
| equals | Comprobar que dos ER son equivalentes. |
| print | Mostrar por pantalla una ER |
| recognize | Determinar si una cadena forma parte del lenguaje representado por la ER dada |
| REtoFA | Obtener el AFND- ϵ equivalente a una ER dada |
| star | Hacer la clausura de Kleene a una ER |
| Operaciones sobre gramáticas | |
| CFGtoPDA | Obtener un autómata con pila que reconozca el mismo lenguaje que una GLC dada |
| clean | Realizar la limpieza de una gramática, aplicando todos los algoritmos de limpieza |
| cyk | Determinar si una cadena es derivable a partir del símbolo inicial de la gramática |
| equals | Comprobar que dos gramáticas son equivalentes |
| fnc | Obtener una gramática equivalente a la dada en Forma Normal de Chomsky |
| fng | Obtener una gramática equivalente a la dada en Forma Normal de Greibach |
| nullable | Obtener los no terminales de la gramática que derivan en la cadena vacía |
| print | Mostrar por pantalla una gramática |
| reminnacc | Eliminar los símbolos de una gramática que sean inaccesibles desde el símbolo inicial |
| remnongen | Eliminar los símbolos de una gramática que no deriven en cadena de terminales |
| remnullable | Eliminar las producciones no generativas ($A \rightarrow$ cadena vacía) |
| remunit | Eliminar las producciones unitarias, es decir aquellas de la forma $A \rightarrow A$ |
| remuseless | Eliminar los símbolos inaccesibles y los que no generan terminales de la gramática |
| RGtoFA | Obtener el AF que reconoce el lenguaje expresado por medio de una gramática dada |
| Operaciones sobre autómatas finitos | |
| complement | Obtener el autómata que reconoce el lenguaje complemento al reconocido por el AF dado |
| equals | Comprobar que dos autómatas dados como argumentos reconocen el mismo lenguaje |
| FAtoFDA | Transformar un AFND- ϵ a AFD |
| FAtoMDFA | Transformar un AFND- ϵ a un AFD mínimo equivalente |
| FAtoNDFFA | Transformar un AFND a AFND |
| FAtoRE | Obtener la ER que represente el mismo lenguaje que el que reconoce el autómata dado |
| FAtoRG | Obtener la gramática que representa el mismo lenguaje que el que reconoce un AF dado |
| intersection | Obtener el AF que reconoce el lenguaje intersección de los reconocidos por los AF dados |
| inverse | Obtener el AF que reconoce el lenguaje inverso del reconocido por el AF dado |
| print | Mostrar por pantalla un autómata |
| union | Obtener el AF que reconoce el lenguaje unión de los reconocidos por los dos AF dados |
| recognize | Determinar si una cierta cadena es reconocida por un AF |
| visualrecognize | Determinar si una cierta cadena es reconocida por un AF de manera visual |
| Operaciones sobre autómatas con pila | |
| print | Mostrar por pantalla un autómata con pila |
| recognize | Determinar si una determinada cadena es reconocida por un autómata con pila |
| PDAtoCFG | Obtener la GLC que representa al lenguaje que reconoce un autómata con pila dado |
| PDAtoFPDA | Obtener un autómata con pila que trabaje bajo el criterio de estados finales |
| PDAtoEPDA | Obtener un autómata con pila que trabaje bajo el criterio de pila vacía |
| visualrecognize | Determinar si una cierta cadena es reconocida por un autómata con pila de manera visual |

Cuadro 1. Operaciones en SELFA-Pro

La manera que tiene el usuario de interactuar con la herramienta es mediante textos conformes a un lenguaje específico del dominio que se diseñó para permitir el manejo de las operaciones y construcciones del lenguaje de manera fácil. Esos textos incluirán por tanto la definición de construcciones de cualquiera de los cuatro tipos que maneja SELFA-Pro junto a la aplicación de operaciones sobre las operaciones definidas o sobre los resultados de otras operaciones. Los usuarios de la herramienta introducen los textos de entrada escribiéndolos en un área de texto en la pantalla *Compile* de la herramienta, o bien cargando un archivo de texto en el cual se incluya las construcciones y operaciones con las que se desea trabajar. Esto permite, por ejemplo, que el profesor distribuya ejemplos que pueden ser luego ejecutados por los alumnos en sus máquinas.

Uno de los puntos fuertes del lenguaje de entrada de SELFA-Pro es la posibilidad que tiene de encadenar operaciones, utilizando el resultado obtenido al aplicar una operación como entrada de otra operación diferente. Esto se consigue utilizando una sintaxis similar a la que poseen la mayoría de lenguajes de programación de alto nivel para almacenar resultados en variables. En la Figura 1 se puede observar un ejemplo de encadenamiento de operaciones en el propio entorno de la herramienta. En concreto, se definen una expresión regular y un autómata finito, posteriormente se obtiene el autómata equivalente a la expresión regular definida y posteriormente se comprueba la equivalencia entre ambos lenguajes. Se muestra así como con una sintaxis sencilla y similar a algo familiar a los alumnos, como son los lenguajes de programación de alto nivel, se pueden realizar cálculos y operaciones encadenados en una única entrada.

```

0 introduzca el texto directamente

regexp expresion{ (aa(b*)) }
automaton automata1{
    states q0,q1,q2;
    alphabet a,b;
    initial q0;
    final q2;
    transition{
        q0,a=q1;
        q1,a=q2;
        q2,b=q2;
    }
}
automata2=REtoFA(expresion);
equals(automata1,automata2);

Aceptar

```

Figura 1. Ejemplo de encadenamiento de operaciones en SELFA-Pro

El lenguaje integrado en la herramienta SELFA-Pro tiene un procesador de lenguaje asociado que forma parte de la herramienta y que se encarga de comprobar la corrección de los textos recibidos como entrada, así como de cargar las construcciones definidas y lanzar la ejecución de los algoritmos solicitados. El procesador de lenguajes está implementado en Java, y es fácilmente extensible para añadir nueva funcionalidad, como ocurrió al pasar de la versión inicial a la versión SELFA-Pro.

La herramienta está implementada como una herramienta *web* para, como se ha comentado, conseguir que cualquiera la pudiera utilizar al no necesitar instalación. La parte *web* de la herramienta está implementada con PHP y MySQL.

La herramienta, además de la funcionalidad básica de ejecutar operaciones, incluye una serie de pantallas adicionales para facilitar el trabajo a sus usuarios. En el menú de la herramienta se incluyen también una sección de ayuda, un *HowTo*, que es un resumen de la funcionalidad de la misma y una serie de ejemplos precargados para mostrar de una manera más clara cómo trabajan los distintos algoritmos implementados.

La propia pantalla de compilación posee también algunas ayudas para que sea más fácil trabajar con la herramienta y para no tener que partir de cero a la hora de escribir los textos conformes al lenguaje de SELFA-Pro. Así, se incluyen una serie de botones que introducen en el cuadro de texto un autómata finito, gramática, autómata con pila o expresión regular de ejemplo, y también unos desplegables que muestran las operaciones disponibles para cada construcción y al hacer clic en una, la escriben en el texto. Todo esto se observa en la Figura 2.



Figura 2. Elementos de ayuda en la pantalla de compilación de SELFA-Pro

Finalmente, cabe también mencionar, aunque ya se hizo con anterioridad, que SELFA-Pro recopila información de uso y posee la capacidad de extraer estadísticas acerca del trabajo de cada usuario, agrupándolos en grupos o en global. En las experiencias de uso que se comentan en la Sección 5 se mostrarán algunos ejemplos de las estadísticas obtenidas por la herramienta.

5. Experiencias y resultados de uso de Selfa-Pro

En esta sección vamos a detallar el uso que se le viene dando a la herramienta SELFA-Pro en el ámbito de la Escuela xxxxxxx, centro de la Universidad de xxxxxx. Como ya se comentó en la Sección 2, la materia de FLAT se cubre en la titulación de Graduado en Ingeniería Informática de dicha universidad en la asignatura Teoría de la Computación, asignatura considerada de Formación Básica y ubicada en el primer cuatrimestre del segundo curso de la titulación. En esta asignatura se viene utilizando SELFA-Pro desde el inicio de su impartición en el centro mencionado, en el curso 2011/12. En cada curso desde entonces se ha seguido más o menos el mismo procedimiento en cuanto al uso de la herramienta que es el siguiente:

- En una sesión de clase celebrada en una de las primeras semanas del cuatrimestre se ha presentado la herramienta y los alumnos han realizado unos primeros ejercicios guiados con ella.
- Posteriormente, durante el curso se ha ido explicando para cada concepto o algoritmo de la asignatura cómo se puede manejar con la herramienta, animando a los estudiantes a que utilicen la herramienta para practicar cada uno de los algoritmos.
- Finalmente, los alumnos terminan la asignatura con el desarrollo de un trabajo en grupo en el

cual deben proponer y resolver una serie de ejercicios de la asignatura, ayudándose de SELFA-Pro para ello.

En esta sección vamos a presentar, por un lado, algunas estadísticas de uso de la herramienta durante estos cursos y, por el otro lado, los resultados de un estudio en el cual recogimos la opinión de un grupo de alumnos acerca de la herramienta.

5.1. Estadísticas de uso de SELFA-Pro

Mediante la funcionalidad de recopilación de información y obtención de estadísticas de la herramienta, se ha recopilado dicha información para los cursos 2011/12 a 2014/15. De esta forma, se ha pretendido comprobar si los alumnos realmente están utilizando la herramienta y, dentro de ello, qué operaciones son las más empleadas por ellos. En el Cuadro 2 se muestra la siguiente información: el número total de ejercicios ejecutados por los alumnos, el número medio de ejercicios por alumno (para lo cual se ha tenido en cuenta el número de alumnos presentados a cualquiera de las dos convocatorias de examen) y los ejercicios más utilizados por los estudiantes, junto a dos datos propios de la asignatura: el número de alumnos que superaron la asignatura en cada curso (en cualquiera de las dos convocatorias) y la nota media de los alumnos de ese curso.

| Curso | Ejercicios | Ejercicios por alumno | Ejercicios más utilizados | % de alumnos aprobados | Nota media |
|---------|------------|-----------------------|--|------------------------|------------|
| 2011/12 | 1868 | 169,82 | remnongen remunit REtoFA | 72,73% | 6,44 |
| 2012/13 | 3357 | 176,68 | REtoFA remunit remnull | 73,68% | 5,77 |
| 2013/14 | 1765 | 126,07 | FAtoRE REtoFA visualrecognize (AF) | 84,62% | 6,02 |
| 2014/15 | 1635 | 56,38 | remunit FAtoDFA REtoFA | 55,17% | 4,43 |

Cuadro 2. Estadísticas de uso de SELFA-Pro y otros indicadores de la asignatura.

Los resultados mostrados en el Cuadro 2 no nos permiten todavía obtener unas conclusiones potentes acerca de cómo el uso de la herramienta repercute en los resultados académicos de la asignatura. Ahora bien, sí que se puede destacar que el descenso en el uso de la herramienta en el curso 2014/15 ha venido acompañado de un descenso en los resultados académicos de dicho curso. Este hecho debe considerarse

de cara a seguir recopilando esta información y poder extraer más adelante otras conclusiones más respaldadas por los datos.

Aparte, en lo que se refiere a las operaciones más utilizadas por los alumnos, se observan bastantes diferencias a lo largo de los cursos. Ahora bien, sí que se observan dos grupos de operaciones que suelen estar entre los más utilizados. Se trata de los algoritmos

mos de limpieza de gramáticas (*remunit*, *remnongen*, *remnull*) y de los algoritmos para alternar entre expresiones regulares y autómatas finitos (*FAtoRE*, *REtoFA*).

5.2. Estudio práctico

Para recoger la opinión de los alumnos acerca de SELFA-Pro, llevamos a cabo un estudio en el cual varios estudiantes de la asignatura de Teoría de la Computación realizaron una serie de ejercicios tanto a mano como con la herramienta y posteriormente respondieron a una serie de preguntas en un cuestionario. En esta subsección vamos a describir ese estudio y a obtener algunas conclusiones de él. Así, las cuestiones que queríamos aclarar con este estudio eran:

- ¿Encuentran los alumnos interesante el utilizar una herramienta como SELFA-Pro?
- ¿Qué funcionalidad de SELFA-Pro encuentran más importante los alumnos?
- ¿Ayuda SELFA-Pro a prepararse y superar una asignatura de FLAT?

Un total de 13 alumnos de la Escuela xxxxxxxx, todos ellos alumnos de la asignatura de Teoría de la Computación, participaron voluntariamente en el estudio. La tarea que realizaron consistía en determinar si dos lenguajes, uno dado en la forma de un autómata finito no determinista y otro dado en la forma de una expresión regular, eran equivalentes. Los alumnos tenían que realizar esa tarea tanto a mano como con la herramienta SELFA-Pro. La resolución de ese problema implica la aplicación de diversos algoritmos: la obtención del autómata finito equivalente a una expresión regular, el paso de un autómata finito a autómata finito determinista, la comprobación de la equivalencia entre autómatas, etc.

Los alumnos llevaron a cabo esta tarea de manera individual en una sesión de 90 minutos de duración. Esta sesión se llevó a cabo al final del cuatrimestre, con lo cual los participantes ya habían trabajado con la herramienta y no necesitaron explicaciones adicionales sobre uso. Después de finalizar la actividad, los alumnos contestaron un cuestionario formado por ocho bloques de preguntas. En los primeros tres bloques, los alumnos tenían que valorar de 0 a 5 varias características de SELFA-Pro. En los otros bloques, se trataba de ordenar varias características de SELFA-Pro de la que se consideraba más importante a la que menos. En el Cuadro 3 se muestran los resultados de los primeros bloques del cuestionario.

| Característica | Media | Desviación típica |
|--|-------|-------------------|
| Presentación de las opciones; claridad de los menús | 4,08 | 0,49 |
| Información sobre ejecución de operaciones y algoritmos | 4,31 | 0,63 |
| Simplicidad al identificar y corregir errores | 3,69 | 0,75 |
| Ayuda disponible | 4,00 | 0,71 |
| Ejemplos prefijados | 4,31 | 0,75 |
| Uso de la herramienta para mostrar autómatas gráficamente | 4,77 | 0,44 |
| Uso de la herramienta para reconocer cadenas de manera visual | 4,46 | 0,52 |
| Uso de una notación basada en variables para encadenar operaciones | 3,85 | 0,69 |
| Facilidad de entendimiento y manejo del lenguaje | 3,92 | 0,86 |
| Utilidad para aprender los algoritmos de la materia | 4,62 | 0,51 |
| Utilidad para preparar la materia | 4,62 | 0,65 |

Cuadro 3. Resultados del primer bloque del cuestionario.

De los valores obtenidos en estas preguntas del cuestionario podemos obtener algunas conclusiones sobre cómo los alumnos utilizan la herramienta. Los alumnos otorgaron una buena puntuación a todas las preguntas, así que parece que realmente la consideran útil para los objetivos para los cuales fue diseñada. De hecho, los alumnos mostraron su certeza de que SELFA-Pro les estaba ayudando para aprender la materia de FLAT. Específicamente, las características visuales de SELFA-Pro recibieron una valoración bastante buena. El elemento menos valorado en este bloque fue el referido a la identificación y corrección de errores. Este resultado, junto con el análisis de la realimentación dada por los alumnos de la herramienta, nos lleva a pensar que habría que enriquecer la herramienta por ejemplo con un editor más rico que podría incluir aspectos tales como coloreado de sintaxis, números de línea u otros similares que permitiesen una mejor interacción con el alumno que introduce los textos para lanzar la ejecución de algoritmos y operaciones.

En cuanto al segundo bloque del cuestionario, primero se preguntó a los alumnos acerca de qué grupo

de operaciones de las incluidas en la herramienta era más interesante según su opinión. Las más valoradas resultaron ser las operaciones sobre autómatas finitos, seguidas de las operaciones sobre gramáticas. Las operaciones sobre autómatas con pila y sobre expresiones regulares recibieron una valoración algo peor. Estos resultados están en la línea de lo recogido en las estadísticas de uso de la herramienta, donde las operaciones que solían aparecer dentro de las más utilizadas pertenecían a los dos primeros grupos de operaciones.

Dentro de las operaciones sobre autómatas finitos, las más valoradas por los alumnos fueron las que llevan a cabo transformaciones entre los distintos tipos de autómatas, como la que obtiene un autómata determinista a partir de otro no determinista y otras similares. En cuanto al grupo de operaciones sobre gramáticas, las que obtienen autómatas a partir de gramáticas regulares fueron las que resultaron ser las más interesantes para los alumnos, aunque en este caso por un estrecho margen con respecto a otras operaciones de este bloque.

En lo que se refiere a las operaciones sobre autómatas con pila, aquellas operaciones que transformaban un autómata para que reconociese cadenas por un cierto criterio (estados finales o pila vacía) fueron las más valoradas. Y finalmente, en el grupo de operaciones sobre expresiones regulares, la obtención de autómatas finitos partiendo de expresiones regulares fue muy bien valorada. Esto no puede ser una sorpresa, ya que dicha operación se encuentra habitualmente entre las más utilizadas por los alumnos, como se vio con anterioridad. En general, parece haber una correlación entre las operaciones más utilizadas por los alumnos durante el curso y las más valoradas por ellos al recoger su opinión subjetiva. Esto debe llevarnos a pensar que los alumnos están satisfechos con la manera en la cual la herramienta maneja las operaciones que ellos van a manejar en la práctica.

6. Conclusiones y trabajos futuros

En este artículo se han presentado los resultados de varias experiencias de uso con SELFA-Pro, una herramienta para la enseñanza de la materia de Teoría de Autómatas y Lenguajes Formales. Se ha descrito la herramienta y se han presentado estadísticas de uso y los resultados de un estudio realizado con dicha herramienta. De la observación de toda esa forma hemos concluido que la herramienta parece estar cumpliendo satisfactoriamente los objetivos para los cuales fue desarrollada, aunque obviamente hay algunos puntos que pueden ser mejorados.

Como trabajo futuro, se plantea mejorar la herramienta introduciendo un editor de textos más rico. También se está estudiando la posibilidad de desarrollar un editor gráfico de autómatas que pueda generar el texto que luego procese la herramienta, como ayuda a la hora de definir las construcciones con las cuales se quiere operar. En ese sentido, se está considerando la posibilidad de desarrollar ese editor como un *plugin* del entorno de programación Eclipse, conocido por los alumnos y que no debería suponer un obstáculo el utilizar como contenedor para este tipo de editores.

Referencias

- [1] xxxxxxxxx
- [2] A. Dzib-Tun, C. González Segura, M. García García. Una herramienta didáctica interactiva para la enseñanza-aprendizaje de los autómatas finitos deterministas. V Congreso Internacional de Computación y Telecomunicaciones (COMTEL 2013), 2013.
- [3] C. García Osorio, A. Arnáiz Moreno, A. Arnáiz González. Enseñanza asistida de teoría de autómatas y lenguajes formales mediante el uso de THOTH. XIII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2007), 425-432, 2007.
- [4] A. Gutiérrez. Estudio y evaluación de una herramienta para representar máquinas de estado finitas. Trabajo Fin de Master, Universitat Oberta de Catalunya (UOC), 2013.
- [5] J.E. Hopcroft, R. Motwani, J.D. Ullman. Introduction to automata theory, languages and computation. Addison Wesley, 2001.
- [6] D. Kelley. Automata and Formal Languages: An Introduction. Prentice-Hall, 1998.
- [7] H.R. Lewis, C.H. Papadimitriou. Elements of theory of computation. Prentice-Hall, 1997.
- [8] J.C. Martin. Introduction to languages and the theory of computation. McGraw-Hill, 2003.
- [9] J. Rocha. Verificación de autómatas y gramáticas. IX Jornadas de la Enseñanza Universitaria de la Informática (JENUI 2003), 2003.
- [10] J. Rocha. Autómatas de pila y máquinas de Turing estructurados. XI Jornadas de la Enseñanza Universitaria de la Informática (JENUI 2005), 2005.
- [11] S.H. Rodger, T.W. Finley. JFLAP: An Interactive Formal Languages and Automata Package. Jones & Bartlett Publishers, 2006.
- [12] xxxxxxxxx