

SELFA-Pro: Añadiendo nuevas entidades y funcionalidades para mejorar la enseñanza y el aprendizaje de TALF

Alfredo Rodríguez, José Jesús Castro-Schez, Ester del Castillo

Dpto. Tecnologías y Sistemas de Información

Escuela Superior de Informática

Universidad de Castilla-La Mancha

Pº Universidad, 4 – 13071 Ciudad Real

alfredo@alfredonline.es, josejesus.castro@uclm.es, ester.castillo@uclm.es

Resumen

En este trabajo se presenta una herramienta software desarrollada sobre una existente, SELFA, cuyo objetivo es facilitar la enseñanza y el aprendizaje de la asignatura Teoría de Autómatas y Lenguajes Formales. Esta herramienta mejora a la anterior tanto en los contenidos que abarca como en usabilidad, reducción del tiempo de aprendizaje de uso y corrección de algunas carencias. La nueva herramienta mantiene características positivas de su predecesora como el de ser desarrollada empleando tecnologías Web o el de disponer de mecanismos que permitan el seguimiento del trabajo realizado por los alumnos durante el periodo de estudio de la asignatura.

1. Motivación

La utilización de herramientas software como complemento en la enseñanza y el aprendizaje de una materia, puede ser muy útil para mejorar la calidad formativa.

En este trabajo se describe SELFA-Pro, la herramienta creada para mejorar la enseñanza y el aprendizaje de la asignatura de Teoría de Autómatas y Lenguajes Formales (TALF) impartida en la Escuela Superior de Informática (ESI) de la Universidad de Castilla-La Mancha (UCLM).

Existen actualmente herramientas que son de interés como soporte al aprendizaje de esta asignatura, como por ejemplo THOTH [5; 9;], JFLAP [8; 10;], JCT [6] o el Proyecto SEPa! [7]. No cabe duda que estas herramientas constituyen un buen apoyo, sin embargo presentan una serie de desventajas:

- Trabajan usualmente de manera gráfica para facilitar su uso por parte de los usuarios y, aunque esto pudiera parecer una ventaja a

priori, se convierte en una desventaja cuando queremos realizar una cantidad algo mayor de ejercicios e incluso enlazar resultados. Esto puede resultar más sencillo si se hace por medio de un lenguaje, utilizando variables para almacenar resultados intermedios.

- Requieren la instalación de manera local o bien de algún entorno de trabajo para poder utilizarlas. Esto hace que la herramienta no sea utilizable en numerosos lugares públicos donde la instalación de software es algo restringido solamente a los administradores.
- No dejan constancia de su utilización de manera alguna; se limitan a computar sin gestionar información de uso.
- No detallan, cómo se desearía, los pasos u operaciones seguidas para alcanzar un resultado.

Una herramienta que soluciona los inconvenientes anteriormente descritos es SELFA [4], con la que se puede trabajar desde cualquier ordenador conectado a internet por medio de un navegador web, sin necesidad de instalación. Además, permite registrar quién la usa y como lo hace.

SELFA ha sido de gran utilidad, en vista de los resultados obtenidos en los cursos 05/06 y 06/07 [12]. Por tanto, la motivación principal es la de, partiendo de la herramienta software SELFA, que sirve para el estudio de la asignatura de TALF, desarrollar una herramienta más completa en cuanto a los contenidos o conceptos que cubre de la materia, que a su vez sea potente y fácil de utilizar, potenciando las virtudes de SELFA y trabajando en la mejora de sus carencias.

En la siguiente sección se presenta brevemente el problema que se pretende solucionar. A continuación se detalla de manera resumida el punto de partida de este trabajo, los

objetivos marcados y cómo se han alcanzado estos. Para finalizar, se mostrarán las conclusiones.

2. El contexto de la herramienta

La asignatura Teoría de Autómatas y Lenguajes Formales es de gran importancia en la formación de buenos ingenieros informáticos ya que proporciona conocimientos teóricos muy importantes en la disciplina [1,2,3].

La teoría de autómatas y lenguajes formales es considerada habitualmente como una disciplina compleja por su gran base teórica y por la gran cantidad de conceptos que maneja. Para poder aprender y asimilar los contenidos de la asignatura (conceptos y algoritmos), los alumnos deben practicar intensamente, realizando el mayor número de ejercicios posible. Sin embargo, realizar tal cantidad de ejercicios durante las clases se antoja muy difícil ya que en las horas lectivas no se dispone del tiempo que esto requeriría. Es por tanto el alumno el responsable de entrenarse fuera de las aulas.

Sería deseable que durante este aprendizaje, mediante la realización de ejercicios, se contara con la supervisión y ayuda del profesor, ya que, para saber si un ejercicio se ha realizado correctamente, no basta con comparar el resultado sino que se necesita entender una metodología de resolución y aplicarla correctamente en cada caso determinado.

Dado el elevado número de alumnos que existen o pueden existir, es muy difícil para los docentes dedicar el tiempo que cada uno requiere en este proceso de aprendizaje. Para solucionar este problema surge la idea de diseñar y desarrollar una herramienta software con fines educativos que hiciera las veces de profesor, esto es, que sirva al alumno para ejercitarse y a la vez comprobar el grado de acierto en sus prácticas, comparando tanto los resultados finales como todo el procedimiento algorítmico que se ha seguido para llegar a los mismos. Además, esto haría que el número de ejercicios posibles quedase limitado únicamente a la capacidad inventiva del alumno ya que cualquier ejercicio que se le ocurriera podría ser comprobado.

Adicionalmente, esta herramienta podrá ser usada para explicar conceptos y algoritmos en clase. El profesor utiliza los resultados

proporcionados de forma automática en lugar de tener que desarrollarlos a mano, con la pérdida de tiempo que esta tarea conlleva.

En este trabajo se presenta SELFA-Pro [11], una herramienta que proporciona soluciones a los ejercicios introducidos por el alumno, de manera que este pueda comprobar el grado de acierto de sus soluciones y que puede ser usada por el profesor para explicar los conceptos y algoritmos de la materia

Para el desarrollo de SELFA-Pro se parte de SELFA, la cual solucionaba parte de la problemática anteriormente descrita pero presentaba algunas carencias que comentaremos en posteriores secciones.

3. Diseño de SELFA-Pro

Como se ha comentado anteriormente, SELFA-Pro se ha desarrollado partiendo de SELFA. La principal funcionalidad de la nueva herramienta sigue siendo la misma que en la anterior versión: *aceptar ejercicios y calcular las soluciones a los mismos, mostrándolas de una forma clara y sencilla de manera que se permita comprender cómo se han alcanzado*. Además, esta nueva herramienta mejora la usabilidad, el tiempo de aprendizaje de uso y lo que es más importante, amplía la cantidad de conceptos de la materia abarcados, de entidades, operaciones y transformaciones que se pueden realizar.

3.1. Nuevos aspectos tratados por SELFA-Pro

La asignatura de TALF se estructura según los siguientes cuatro bloques:

- Introducción a los lenguajes y gramáticas.
- Lenguajes y gramáticas regulares (expresiones regulares y autómatas finitos).
- Lenguajes y gramáticas libres de contexto (autómatas con pila).
- Máquinas de Turing.

Aunque la herramienta SELFA ofrecía soluciones para algunos aspectos de los bloques anteriores, no contempla otros, como son las expresiones regulares, los autómatas a pila y algunas operaciones entre los otros tipos de entidades ya existentes (autómatas finitos y gramáticas). Es por esto por lo que se han ampliado las funcionalidades de SELFA con otras nuevas con

las que cubrir aspectos del temario de la asignatura.

Esto condujo a la necesidad de permitir definir nuevas entidades con las que operar: expresiones regulares y autómatas a pila.

En la Tabla 1 se pueden ver las entidades y las operaciones que soportaba SELFA.

Autómatas
Convertir λ -N DFA en N DFA
Convertir FA en DFA
Obtener el DFA de un FA dado
Realizar la unión de dos FA
Realizar la intersección de dos FA
Obtener el FA inverso de uno dado
Obtener el FA complemento de uno dado
Comprobar si dos FA son equivalentes
Ver si una cadena es aceptada por un FA
Gramáticas
Limpieza de una gramática
Pasar GLC a Forma Normal de Chomsky
Pasar GLC a Forma Normal de Greibach
Algoritmo de Cocke-Younger-Kasami

Tabla 1. Listado de entidades y operaciones soportadas por SELFA

En la Tabla 2 se muestra cuales son las nuevas operaciones que aporta la versión Pro de la herramienta, recordando que conservará también las existentes en SELFA. Con estas nuevas aportaciones ahora sí se cubre el temario de la

asignatura, a excepción de las Máquinas de Turing, mejorando así la herramienta tanto para alumnos como para profesores.

Autómatas
Reconocimiento visual de una cadena
Obtener la RG equivalente a un autómata
Obtener la RE equivalente a un autómata
Simulación visual del reconocimiento de una cadena
Gramáticas
Obtener el FA equivalente a una RG
Obtener el PDA equivalente a una CFG
Ver si dos RGs son equivalentes
Autómatas con pila
Obtener la CFG equivalente a un PDA
Convertir PDA criterio de los estados finales
Convertir PDA criterio de pila vacía
Reconocimiento visual de una cadena
Reconocimiento en forma de árbol
Simulación visual del reconocimientote una cadena
Expresiones Regulares
Obtener el N DFA equivalente a una RE
Comprobar si dos RE son equivalentes
Ver si una cadena es aceptada por una RE
Realizar la clausura de Kleene de una RE
Realizar la unión de dos RE
Realizar la concatenación de dos RE

Tabla 2. Listado de entidades y operaciones añadidas por SELFA-Pro

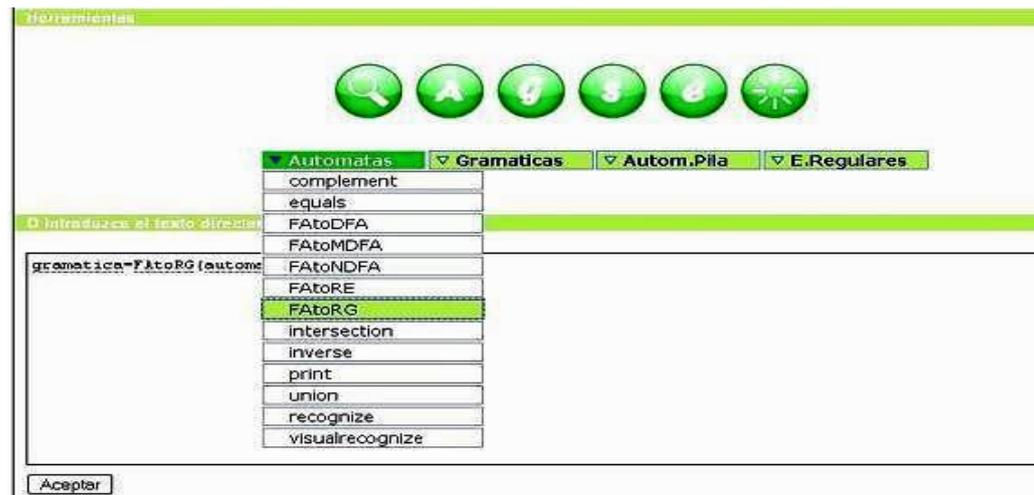


Figura 1. Botones de definición de entidades y especificación de operaciones

Estos nuevos cambios han sido realizados conservando las siguientes características ya existentes:

- Disponibilidad temporal.
- Disponibilidad geográfica.
- La usabilidad ha sido mejorada ya que se han añadido botones de herramientas que ayudan a la definición de entidades y a la inserción de ejercicios a modo de plantillas. Por ejemplo, si se elige la pestaña “autómatas”, se ofrece un menú en el que se puede escoger la operación a realizar con autómatas, generándose la estructura de la instrucción (ver Fig. 1). También se han renombrado las operaciones y se han reestructurado algunas secciones.
- Facilita la entrada de ejercicios, añadiendo la posibilidad de utilizar resultados intermedios y poder realizar operaciones sobre ellos. El uso de resultados intermedios para aplicarlos en otras operaciones, cobra una mayor importancia ya que en la anterior herramienta solamente se podía obtener entidades del mismo tipo a la que se aplicaban operaciones. Por ejemplo, si se hacía una operación sobre gramáticas, el resultado (si lo tuviera) sería una gramática, pero ahora puede que el resultado sea de una entidad distinta a la de entrada, como el caso de obtener el autómata finito que reconoce el lenguaje dado por medio de una expresión regular.

A continuación, se pueden ver los tipos de entidades con los que se podía trabajar usando SELFA (ver Fig. 3) y con la nueva versión SELFA-Pro (ver Fig. 2).

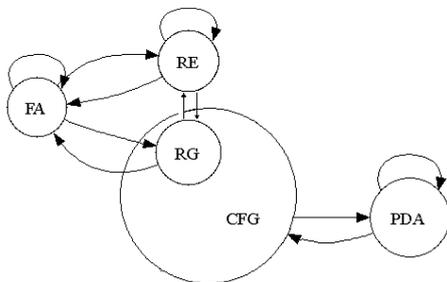


Figura 2. Entidades y relaciones de SELFA-Pro.

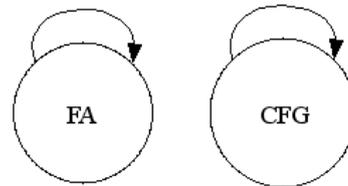


Figura 3. Entidades y relaciones de SELFA.

Como se puede comprobar, con SELFA-Pro se tiene una cantidad mayor de entidades y una interconexión total entre ellas.

Desde el punto de vista de los profesores, sigue cumpliendo que es ejemplar, es decir, que la propia herramienta sirve como ejemplo de desarrollo y reconocimiento de lenguajes; es accesible, funcional, transparente y ha sido desarrollada con software libre.

Además de los cambios descritos, tanto de contenidos como características, se añaden otras características nuevas, que son:

- La interfaz multilinguaje: Esto es, se ha dotado a la herramienta de una interfaz fácilmente modificable para hacer que la herramienta pueda trabajar en otros idiomas y se ha particularizado con el caso del inglés, estando la herramienta disponible tanto en castellano como en inglés, lo que hace que su nivel de posible utilización tenga un ámbito de carácter internacional en lugar de estar restringida a usuarios hispanoparlantes.
- Gráfica: Existen nuevas operaciones que permiten obtener resultados que reflejan el proceso del algoritmo paso a paso de forma visual. Por ejemplo, se podrá ver los estados en los que se encuentra un autómata a pila según va procesando la entrada, así como saber el contenido de la pila, todo de forma muy comprensible y atractiva (ver Fig. 4).

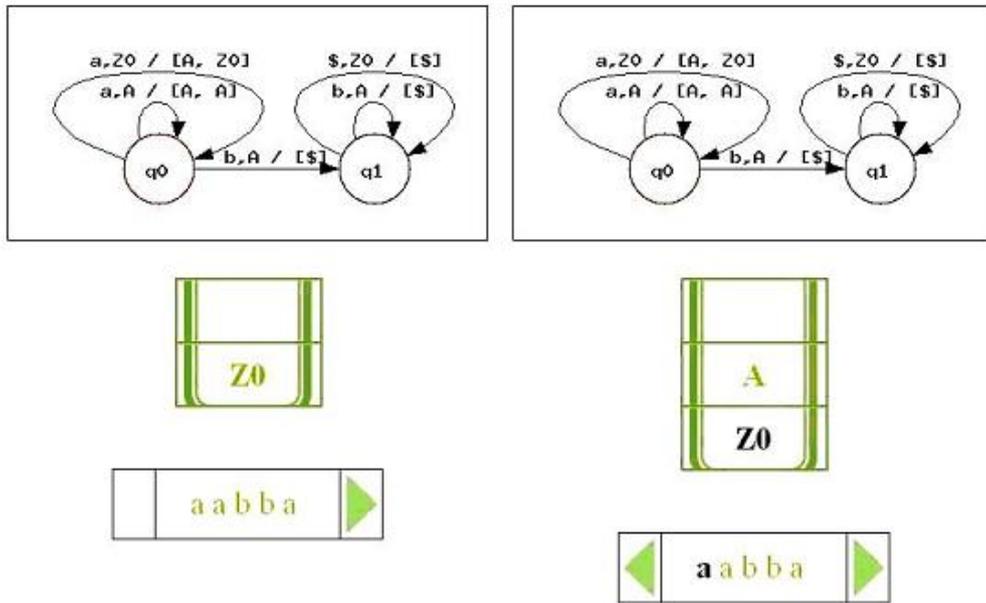


Figura 4. Simulación gráfica del reconocimiento de una cadena por medio de un PDA.

3.2. Interacción con el usuario

La entrada por parte del usuario queda definida por la especificación de una o varias entidades (autómatas, gramáticas, autómatas a pila o expresiones regulares) y la especificación de una serie de operaciones sobre dichos elementos, las cuales pueden utilizar o no los resultados intermedios generados.

La entrada a la herramienta será un texto escrito en un determinado lenguaje (una extensión del lenguaje de entrada de SELFA), en el que, por una parte se definen las entidades con las que se quiere trabajar, definiendo su nombre, y sus elementos por medio de una serie de palabras reservadas lo más intuitivas y memorizables que sea posible, y por otra parte se definen las operaciones que tienen una sintaxis y unas palabras reservadas para invocarlas.

Veamos cómo se especifican o definen las nuevas entidades:

La definición de un autómata a pila $P=(Q,\Sigma,\beta,\delta,q_0,Z_0,F)$, implica determinar el conjunto de estados (Q), el alfabeto de entrada (Σ), el alfabeto de la pila (β), la función de

transición (δ), el estado inicial (q_0), el símbolo inicial en la pila (Z_0) y el conjunto de los estados finales (F) si los tuviera. La existencia de estados finales determina el criterio por el que se aceptan palabras (por pila vacía o estados finales).

La definición de una expresión regular implica construir la misma por medio de operadores básicos, alternación o unión (\cup), concatenación (operador implícito) y clausura (*), o por medio de otros operadores como el de clase ([]), clausura positiva (+) u opcionalidad (?). Los posibles elementos de las expresiones son las letras del alfabeto americano, tanto mayúsculas o minúsculas, cifras del 0 al 9 o combinaciones de los anteriores entre comillas simples (''), a los que podemos aplicar los anteriores operadores.

Para facilitar el trabajo con la herramienta se ha definido un menú de ayuda dónde se indica cómo se ha de trabajar, existe también una zona de ejemplos dónde se pueden ver ejemplos de uso de cada una de las posibles operaciones. También se ha incorporado una serie de botones y menús en la zona de compilación que hacen que simplemente pulsando sobre ellos aparezca ya escrito un ejemplo de la entidad seleccionada que sirva como esqueleto para modificar adaptándolo

a las necesidades particulares (ver Fig. 1). Del mismo modo, se ha definido un menú que nos permite elegir la operación que se desea hacer entre todas las disponibles. Todo lo anterior, por tanto, permite trabajar sin ni siquiera tener que recordar una sola palabra reservada del lenguaje, lo cual hace que la facilidad de uso aumente considerablemente. De este modo se mejora SELFA, que tenía, como uno de sus problemas de uso, la necesidad de aprender un nuevo lenguaje, lo cual podría ser una dificultad para alumnos de primeros cursos.

La herramienta ofrece información intermedia, mostrando las entidades que se van obteniendo como pasos intermedios, así como los resultados finales haciendo que el proceso se entienda de principio a fin. En el caso de que la entrada no sea correcta, se mostrarán los oportunos mensajes de error.

Además, las nuevas operaciones visuales añadidas (visualrecognize para autómatas y autómatas a pila) permiten al profesor enseñar como evoluciona el autómata mientras está reconociendo una frase. Esto se puede conseguir, por ejemplo, usando en el aula pizarras electrónicas.

3.3. Lenguaje de entrada a SELFA-Pro

El lenguaje de entrada a SELFA-Pro conservó la sintaxis de SELFA casi inalterada en las partes que se encontraban ya implementadas, a excepción de un cambio de nombres de las operaciones para facilitar su recuerdo.

El lenguaje diseñado tiene las siguientes características:

- Permite la definición en un solo archivo de todos los ejercicios que el usuario quiera realizar, pudiendo trabajar de manera off-line requiriendo solamente un acceso a Internet a la hora de compilar y obtener los resultados.
- Los resultados intermedios de las operaciones se pueden almacenar en variables con las que se puede volver a operar.
- Existe una operación de imprimir con la que podemos mostrar por pantalla todo tipo de entidades.
- La definición de las nuevas partes del lenguaje, ya sean autómatas a pila, expresiones regulares o nuevas operaciones conservan el estilo del lenguaje que ya estaba

definido para que la dificultad de recordar las nuevas incorporaciones sea mínima.

La sintaxis en notación EBNF del lenguaje propuesto se muestra en [11].

El lenguaje tiene nuevas restricciones que no pueden ser expresadas sintácticamente y que el procesador de lenguaje deberá detectar. Estas son:

- Toda entidad debe tener un identificador único.
- En la definición de un autómata finito o autómata a pila, los elementos que se definen como estados (states) no pueden definirse como elementos del alfabeto y viceversa.
- El estado declarado como inicial debe haber sido definido previamente como estado, lo mismo ocurre con los estados finales.
- El símbolo inicial de la pila (en los autómatas a pila) debe haber sido declarado como elemento del alfabeto de la pila.
- En las transiciones de los autómatas el primer identificador debe haber sido declarado como un estado, el segundo como un elemento del alfabeto y el resto de identificadores deben ser elementos del conjunto de estados.
- En el caso de las transiciones de los autómatas a pila, el primer elemento debe ser un estado, el segundo un elemento del alfabeto y el tercero un símbolo del alfabeto de pila. En la parte derecha, el primer símbolo ha de ser un estado y el resto deben ser también elementos del alfabeto de la pila.
- Las operaciones de un determinado tipo de entidad deben ser aplicados sobre entidades de dicho tipo salvo excepciones de operaciones sobrecargadas que aceptan varios tipos, como por ejemplo imprimir, reconocimiento de cadenas (incluida la forma visual) o la igualdad.

Como ejemplo de uso de SELFA-Pro se muestra un ejercicio que ilustra el gran potencial de la herramienta, trabajando con operaciones sobre expresiones regulares y autómatas finitos. En este ejercicio se define una expresión regular (e1) que será mostrada por pantalla (print), se calculará el autómata finito equivalente a la expresión (REtoFA, empleando el algoritmo de Thompson) y posteriormente se obtendrá la expresión regular equivalente al autómata obtenido por la anterior operación (FAtoRE, empleando el algoritmo de Han y Word [13]). Seguidamente se comprobará

que se trata de la misma que se tenía inicialmente (equals).

```

regexp e1{ ([A-C]*) (b?) }
print (e1);
a1=REtoFA (e1);
ne1=FAtoRE (a1);
equals (e1, ne1);

```

Después de realizar las operaciones de conversión, se observa que la operación FAtoRE nos devuelve como resultado la expresión $((A|B|C)^*)(\$b)$, que es una expresión regular equivalente a la de entrada. No es literalmente la misma ya que este algoritmo trabaja con operadores básicos (unión, concatenación y clausura) pero es equivalente. De hecho, al realizar la operación equals se comprueba que son equivalentes.

Como se ha podido observar, estas operaciones son muy potentes y devuelven expresiones regulares muy cortas y legibles al contrario que hacen el resto de herramientas disponibles que trabajan en esta materia.

Más ejemplos de ejercicios escritos en el lenguaje de entrada de SELFA pueden ser consultados en [11].

3.4. Tratamiento de la entrada

Para comprobar si el texto de entrada tiene una sintaxis correcta y se cumplen las restricciones anteriormente comentadas, además de extraer la información relevante de los ejercicios para poder solucionarlos, es necesario que SELFA-Pro tenga un procesador de dicho lenguaje. Se partió del procesador de SELFA, y se fue modificando el analizador léxico, el sintáctico y el analizador semántico. La integración de los tres analizadores constituye el procesador de lenguaje que tiene las siguientes características:

- Portable: Está implementado en Java por lo que es multiplataforma.
- Integro: El procesador funciona obteniendo soluciones cuando la entrada es correcta e informando del mayor número de errores cuando la entrada no es correcta.
- Extensible: Se pueden añadir nuevas características, como nuevas operaciones.

Los algoritmos implementados en Java han sido integrados en el procesador de lenguaje por medio de la interfaz, que llamará a los métodos oportunos a través de la misma.

3.5. Salida de la herramienta

Una vez determinado como se introducen los ejercicios y como trabaja el procesador de lenguaje invocando a los algoritmos que resuelven las operaciones adecuadas, queda determinar como será la salida de SELFA-Pro.

Las soluciones se generan utilizando XML como formato interno, que es fácilmente utilizable por aplicaciones externas. Utilizando XML las soluciones pueden ser formateadas para obtener los resultados en HTML, lo que será de mucha utilidad para integrar esta parte con la interfaz de la herramienta.

3.6. Arquitectura de la herramienta

Uno de los requisitos que se querían conservar de SELFA era que su instalación y actualización debían ser lo más sencillas posibles, incluso, como es el caso, que no necesitaran instalación. Además se quería conservar, enriqueciéndola, la capacidad de registrar el uso que los usuarios hacían de la misma. Por lo tanto se siguió con la filosofía de que la aplicación fuese centralizada utilizando la arquitectura cliente/servidor Web.

Se conservó la arquitectura multicapa Web en la que cada capa ofrece sus servicios a la capa inmediatamente superior y recibe los servicios de la capa inferior. La arquitectura Web posee las siguientes ventajas:

- Toda la carga de trabajo recae sobre el servidor haciendo de SELFA-Pro una aplicación ligera al lado del cliente.
- No necesita instalación local, simplemente un navegador con acceso a Internet.
- Siendo una aplicación centralizada se permite registrar el uso de la herramienta mediante una base de datos en el servidor.

Por tanto, al estar centralizada, las actualizaciones son muy sencillas e inmediatas, y estos cambios repercuten directamente en todos los sitios en los que se utilice.

La parte Web ha sido desarrollada utilizando PHP y la base de datos MySQL, que es muy ligera y gratuita, con tiempos de respuesta lo suficientemente bajos como para soportar con creces los requisitos que la herramienta requiere. Las soluciones se devuelven en formato HTML.

Se han añadido también pequeñas utilidades que facilitan el uso de la herramienta escritas en

javascript, de forma que puedan seguir siendo utilizadas con cualquier navegador.

3.7. Control y registro de datos

Se ha conservado la posibilidad de gestionar los usuarios que utilizan la aplicación, así como gestionar noticias o el repositorio de ejemplos prácticos como entrada a SELFA-Pro.

Todo esto está adaptado a las nuevas operaciones, añadiendo nuevos ejemplos al repositorio con las nuevas operaciones y haciendo que se puedan recoger estadísticas, tanto de forma tabular como gráfica, de todas las operaciones.

4. Conclusión

En este documento se ha mostrado el trabajo realizado en la elaboración de la herramienta SELFA-Pro, la cual mejora la herramienta SELFA y busca la mejora de la enseñanza de la asignatura de TALF, impartida en la ESI (UCLM).

Esta herramienta permite que se cubran las necesidades formativas de la asignatura, haciendo posible que los alumnos se entrenen en todos los contenidos de la materia, a excepción de las máquinas de Turing, mejorando la calidad tanto en el aprendizaje como en la enseñanza, ya que el profesor puede emplearla para explicar la mayoría de conceptos y algoritmos de la asignatura. SELFA-Pro se convierte en un instrumento muy útil para resolver cualquier ejercicio propuesto y poder observar las soluciones y mecanismos seguidos para obtenerlas.

El diseño cliente/servidor, que permite su uso a través de Internet, es una ventaja muy importante que diferencia a esta herramienta del resto de aplicaciones existentes.

Así mismo tiene una amplia gama de operaciones sobre varios tipos de entidades mucho mayor que en la mayoría de aplicaciones existentes para estos fines. Algunas operaciones como aquella que obtiene la expresión regular equivalente a un autómata, devuelve expresiones más cortas y legibles que en otras herramientas disponibles en la actualidad.

Agradecimientos

A la ESI, a la UCLM y a los profesores y alumnos del curso 2006/07 y 2007/08 que han participado activamente en la utilización de la herramienta

permitiendo probar y mejorar la misma a lo largo de su desarrollo.

Referencias

- [1] D. Kelley, *Automata and Formal Languages: An Introduction*. Prentice Hall, 1998.
- [2] J.G. Brookshear, *Theory of Computation: Formal Languages, Automata, and Complexity*. Benjamin-Cummings Publishing Company, 1989.
- [3] J.E. Hopcroft, R. Motwani, J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2001.
- [4] J.J. Castro-Schez, E. del Castillo, J. Hortolano, *Una herramienta para la enseñanza y aprendizaje de lenguajes formales y teoría de autómatas*. XIII Jornadas de la Enseñanza Universitaria de la Informática (JENUI 2007), 379-386, 2007.
- [5] C. García Osorio, A. Arnáiz Hosena, 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.
- [6] The Java Computability Toolkit, <http://humboldt.sunyit.edu/jct/>
- [7] Proyecto SEPa!, <http://www.ucse.edu.ar/fma/sepa/>
- [8] Proyecto JFLAP, <http://www.jflap.org>
- [9] Proyecto THOTH, <http://pisuerga.inf.ubu.es/cgosorio/THOTH/>
- [10] S.H. Rodger and Tomas W. Finley, *JFLAP: An Interactive Formal Languages and Automata Package*. Jones & Bartlett Publishers, 2006.
- [11] A. Rodríguez, J.J. Castro-Schez, SELFA-Pro, <http://apps.oreto.inf-cr.uclm.es/selfa/>
- [12] J.J. Castro-Schez, E. Castillo, J. Hortolano, A. Rodríguez. *Designing and Using Software Tools with Educational Purposes: FLAT a case of study*. Aceptada en IEEE Transactions on education, para su publicación en 2008.
- [13] Y-Han, D. Wood. Obtaining shorter regular expressions from finite-state automata. *Theoretical Computer Science*, 370:110-120, 2006