SldDraw: Un trazador de árboles SLD

Francisco Gutiérrez
Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga
e-mail: pacog@lcc.uma.es

Mª del Carmen de Castro
Dpto. de Lenguajes y Sistemas Informáticos
Universidad de Cádiz
e-mail: maricarmen.decastro@uca.es

Resumen

Se presenta una aplicación software, en el ámbito de la Programación Lógica, que dibuja árboles de ejecución del método de resolución SLD. La aplicación está diseñada tanto como herramienta didáctica para el aprendizaje del método de resolución SLD con todas sus variantes como herramienta para el tutor que le permite encontrar árboles SLD con las características deseadas. Está destinada a alumnos y profesores que imparten docencia en niveles universitarios en asignaturas relacionadas con la Programación Lógica.

1. Introducción

En un principio, se pensó diseñar una herramienta de ayuda al profesor a la hora de proponer ejercicios que tracen árboles de ejecución por el método SLD [1] (árboles SLD). Antes de pasar a la realización de la aplicación se buscaron utilidades que ya resolvieran ese problema. La más parecida es la que se ofrece en un paquete junto el texto "Computational Intelligence. A Logical Approach" [2]. De todas formas, aunque esta herramienta genera árboles SLD, su propósito no es el mismo que el que guía a la aquí presentada.

SldDraw[3] pretendía cubrir una necesidad docente. Normalmente, a la hora de proponer ejercicios de traza de árboles SLD, el profesor se encuentra con la dificultad de encontrar árboles adecuados, en espacio y complejidad. El trazado a mano de los árboles es una tarea de cierta complejidad y encontrar esos árboles adecuados requiere mucha experiencia y sobre todo tiempo. La herramienta conseguida permite generar árboles SLD de forma automática a partir de un programa y un objetivo, e ir modificando tanto el programa como el objetivo hasta conseguir el árbol de tamaño adecuado.

Posteriormente se pensó que la aplicación también podía servir como herramienta didáctica en manos de los alumnos y profesores por lo que se le incorporaron más facilidades como cambios de reglas de búsqueda y selección, descripción detallada de las unificaciones que aparecen en cada rama del árbol, ejecuciones paso a paso, de éxito en éxito, puntos de ruptura, etc.

Con objeto de poder hacer trazas de ejecución de programas escritos en Prolog, se incorporan también características propias del lenguaje como aritmética extralógica, listas, predicados sobre metaprogramación entre los que cabe destacar el predicado =.., corte, etc.

Con SldDraw, el profesor puede seleccionar ejercicios en el que se signifiquen las diferencias que se producen al cambiar de regla de búsqueda, de regla de selección, al reordenar los objetivos, al realizar distintos usos de un predicado, etc. Por otro lado, al alumno le vale como test para comprobar que entiende los mecanismos que comporta la realización de árboles SLD y en particular, el modelo de ejecución de Prolog, así como el uso de ciertos predicados específicos de este lenguaje de programación o el efecto del corte.

SldDraw no incluye un intérprete de Prolog eficiente sino simplemente un simulador. Por tanto, no es posible realizar trazas de programas

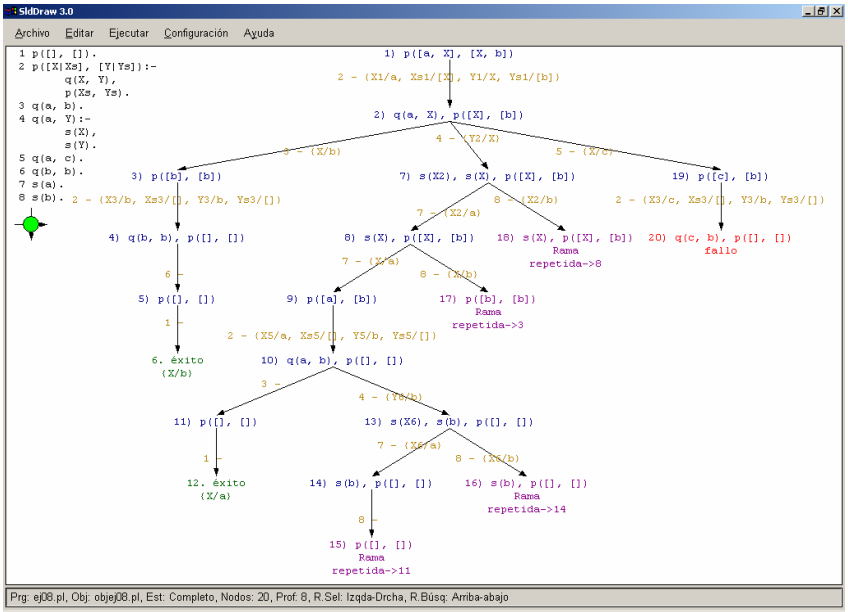

Figura 1. Ejemplo de ejecución de SldDraw

Prolog que generen una gran cantidad de nodos. Como ejemplo puede decirse que un árbol con 600 nodos y profundidad 40 se consigue visualizar en un tiempo razonable.

2. Descripción

Se trata de un trazador de árboles de ejecución SLD que muestra el árbol SLD y permite manipularlo hasta conseguir una representación adecuada. Al contrario del modelo de ejecución de Prolog, el árbol se genera por niveles y, dado que un árbol puede tener ramas infinitas, se fija inicialmente un nivel máximo de desarrollo pudiendo en cada momento modificarse. Un círculo verde o rojo nos indica claramente si el árbol está completo o no.

La aplicación admite como programas y objetivos a un subconjunto de los programas Prolog lo suficientemente amplio para que sea de utilidad. En particular, no incorpora características de entrada de datos y sólo incorpora los predicados básicos de salida. Está desarrollada en SWI-Prolog con XPCE [4] y este hecho resulta ser un aliciente para los alumnos que observan como el lenguaje que están aprendiendo es válido para la realización de aplicaciones de cierta complejidad. Existen versiones para Windows y para Linux: en realidad, esta es una facilidad de SWI-Prolog. Tiene un entorno similar al de cualquier aplicación actual basada en ventanas. Por otro lado, también se ha desarrollado una utilidad Java, llamada JsldDraw[5] bastante parecida a SldDraw que permite ser ejecutada como Applet y como aplicación.

Cada elemento del árbol es mostrado en un color diferente permitiendo distinguirlos y detectar fácilmente los distintos tipos de nodos; por ejemplo, un nodo fallo aparece en rojo y un nodo éxito en verde.

Se permite también modificar el programa o el objetivo mediante un editor que puede ser seleccionado por el usuario.

La aplicación dispone de un fichero de configuración en el que entre otras cosas, se permite modificar los colores de los nodos y unificaciones, el idioma de la aplicación, el editor utilizado, etc.

El algoritmo para la representación gráfica ha sido adaptado del propuesto para la representación de árboles en ML [6].

3. Ejecución

Para comenzar debemos disponer de un programa en Prolog que esté libre de errores y almacenado en un fichero de extensión *.pl*. Si el programa contiene errores, y se intenta abrir desde la aplicación se muestra una ventana de aviso indicando el error producido pero en la que la descripción del error no está suficientemente clara. A continuación se debe disponer de otro fichero con la misma extensión en el que estará escrita la consulta que queremos realizar.

Ya desde SLDdraw abrimos ambos ficheros, el que contiene el programa y el de la consulta y pedimos a la aplicación que nos represente el árbol. Otras opciones nos permiten iniciar el proceso de representación paso a paso.

Tanto el programa como el objetivo pueden modificarse desde la aplicación generando árboles para cada caso. Podemos imprimir el árbol resultante. En este caso podemos ajustar el número de páginas horizontales y verticales que ocupará la representación. Esto permite imprimir árboles cuya representación exceda del tamaño de una página. Además, podemos obtener un fichero encapsulado postscript con la representación.

Ya se ha mencionado que la herramienta no es un intérprete de Prolog. A diferencia de Prolog la aplicación trabaja con chequeo de ocurrencias (occur check) de manera que no produce unificación cuando esta situación se detecta. Por otro lado, detecta ramas infinitas y ramas cíclicas siendo la detección de estas últimas configurable.

4. Ámbito de utilización

Es una herramienta muy útil en las clases de prácticas de las asignaturas de programación lógica para el aprendizaje del lenguaje Prolog y en general de la resolución SLD. Desde el programa más simple se puede comprobar el árbol que genera mostrando unificaciones y respuestas computadas.

Dado que el aprendizaje de un lenguaje de programación declarativo como es Prolog suele resultar complicado, los alumnos agradecen disponer de una utilidad de estas características.

Es una aplicación relativamente pequeña y sencilla de aprender y utilizar. La distinción de los elementos que participan en la ejecución mediante colores permite desde el primer momento al alumno identificar cada parte y cada paso.

El hecho de que las unificaciones que se realizan aparezcan escritas en las ramas también hace más rápida su comprensión y aprendizaje. Así mismo, al desplegarse, aparece una ventana con la descripción paso a paso del proceso de obtención del unificador. Esto también ayuda a la comprensión del algoritmo de unificación de Robinson.

En la actualidad se está utilizando con éxito en dos universidades, la de Málaga y la de Cádiz. Así mismo, sabemos que se está utilizando también en algunas universidades de Centroamérica.

En particular, en Málaga se utiliza desde hace dos años, tanto en la Ingeniería Informática como en las Ingenierías Técnicas y aunque no se ha realizado ningún test para comprobar la ventaja que ha supuesto el uso de la herramienta, los alumnos, y sobre todo los repetidores que no la habían usado en años anteriores, han manifestado su reconocimiento por poder contar con una herramienta de estas características. Según sus comentarios, y en los laboratorios se pone de manifiesto, usan la aplicación a menudo para aclarar significados, variaciones según las formas de uso, etc. Con respecto al profesorado, ha aumentado la variedad y diversidad de ejercicios relativos a la

creación de árboles SLD tanto en las prácticas como en los exámenes.

En Cádiz, es el primer año que los alumnos de Ingeniería Técnica en Informática de Gestión están utilizando la herramienta. Se les ha pasado una encuesta al final del cuatrimestre y, en su mayoría, les parece muy positivo contar con una aplicación complementaria al intérprete que les facilite el aprendizaje. Les resulta fácil de manejar, y la consideran práctica para detectar errores y depurar los programas. También están de acuerdo, en su mayoría, en que comprenden mejor el método de Resolución SLD y la forma en que se ejecutan los programas en Prolog gracias a la aplicación.

En realidad el ámbito de utilización se podría extender hacia cualquier persona que necesite programar en Prolog .

5. Problemas o dificultades que resuelve

A continuación enumeramos las dificultades que a nuestro entender resuelve la herramienta:

- Dificultad inicial para comprender este tipo de programación por parte de personas que siempre (o en la mayoría de los casos) han programado y aprendido a programar en lenguajes basados en otro paradigma (imperativo, etc.).
- Comprensión de que la ejecución de programas en Prolog se realiza mediante el método de resolución SLD. Es decir, la búsqueda de soluciones se realiza en *búsqueda primero en profundidad con retroceso* sobre el árbol, una vez fijadas las reglas de selección y de búsqueda. Esto aclara de inmediato el porqué de la no completitud de Prolog.
- La depuración de programas mediante esta traza gráfica es mucho más clara que la depuración tradicional sobre programas lógicos.
- Aprendizaje del efecto de los predicados de control, en especial del corte.

6. Ventajas

Comentamos a continuación, algunos elementos de la aplicación que hacen atractivo su uso y extensible a otros ámbitos:

- Facilidad de instalación y uso, así como su reducido espacio.
- Posibilidad de configurar el idioma.
- Existencia de versiones para varios sistemas operativos.
- Inclusión de numerosos ejemplos.
- Posibilidad de imprimir o guardar el árbol.
- La modificación de la visualización permite obtener el mismo árbol de diferentes formas.

7. Conclusiones y posibles mejoras

En definitiva es una herramienta útil y didáctica dentro del ámbito de la Programación Lógica, que sirve tanto de apoyo al docente en su trabajo como al alumno en la tarea de aprender y asimilar.

Para el futuro se pretende dotar a la aplicación de un analizador sintáctico que resuelva errores en la construcción de programas y objetivos, la posibilidad de no desarrollar trozos del árbol que no sean significativos y la posibilidad de abrir ventanas con subramas (contemplada en JsldDraw).

8. Referencias

[1]J.W. Lloyd. Foundations of Logic Programming. Second, Extended Edition. Springer-Verlag, 1987.

[2] D. Poole, A. Mackworth y R. Goebel. Computational Intelligence. A Logical Approach. Oxford University Press, 1998.

[3]Antonio Barranquero Campos. Visualización de árboles SLD. Proyecto Fin de Carrera. Dpto. de Lenguajes y Ciencias de la Computación. Universidad de Málaga. 2002.

[4] Jan Wielemaker. SWI-Prolog/XPCE. University of Amsterdam.

[5]Antonio Jesús Paredes García. JApplet para visualizar árboles de refutación SLD. Proyecto Fin de Carrera. Dpto. de Lenguajes y Ciencias de la Computación. Universidad de Málaga. 2002.

[6] Andrew J. Kennedy. Drawing Trees. Journal of Functional Programming, Cambridge University Press, Mayo 1996.