

Sistema Demostrador automático de teoremas basado en resolución.

Juan Carlos Fernández Durán, Victor Gualdrás de la Cruz y Omar Khanji.

Profesor y director del proyecto: Pascual Julián Iranzo.

Universidad Castilla la Mancha, Escuela Superior de Informática
Ciudad Real, España

Índice general

Sistema Demostrador automático de teoremas basado en resolución.	1
<i>Juan Carlos Fernández Durán, Victor Gualdrás de la Cruz y Omar Khanji. Profesor y director del proyecto: Pascual Julián Iranzo.</i>	
1. Introducción	3
2. Documentación del código	3
2.1. Resolutor	3
2.2. Traductor	4
2.3. Entrada y Salida	4

1. Introducción

La demostración automática de teoremas (de siglas ATP, por el término en inglés: Automated theorem proving) que también puede ser denominada Deducción automatizada, es actualmente el subcampo más desarrollado del razonamiento automático, y se encarga de la demostración de teoremas matemáticos mediante programas de ordenador.

La construcción de un programa de demostración automatizado significa proporcionar una descripción algorítmica a un cálculo formal para que pueda ser implementado en una computadora para demostrar teoremas de manera eficiente.

La idea de resolución se fundamenta en que si un conjunto de cláusulas es insatisfacible entonces a partir de él podrá deducirse la cláusula vacía. La estrategia de resolución lineal es una herramienta que nos permite localizar esa cláusula vacía. Su funcionamiento es el siguiente. Comenzando con una cláusula, se resuelve esta contra otra para obtener un resolvente que, a su vez, será resuelto con otra, repitiendo el proceso anterior tantas veces como sea necesario.

2. Documentación del código

A continuación se presentarán los diferentes módulos de los que está compuesto la aplicación

2.1. Resolutor

La regla principal del resolutor se llama, valga la redundancia, resolutor. Se invocaría primero resolutor/0 (donde 0 es la aridad) coge la primera cláusula perteneciente al conjunto de cláusulas que haya en la base de hechos, e invoca a resolutor/2 con la cláusula seleccionada. Esta regla va probando, cogiendo cláusulas de la base de hechos para comprobar si llega a la cláusula vacía, en caso contrario prueba con las cláusulas laterales, y si tampoco tiene éxito retrocede hasta resolutor/0 y prueba a elegir otra cláusula. Cada vez que se aplica el proceso de resolución, se toma la cláusula resultante que se denomina resolvente, y se prueba a aplicar de nuevo este proceso, probando primero con las cláusulas de la base de hechos o si no consigue nada, probando con las cláusulas añadidas previamente denominadas cláusulas laterales. Este proceso se repite hasta dar con la cláusula vacía.

El funcionamiento interno del resolutor es buscar dentro de dos cláusulas dos proposiciones iguales, una negada y la otra no, para poder aplicar el mecanismo de resolución. Además se asegura de que no se introducen proposiciones iguales (por ejemplo: $p \wedge \neg p$), y que no se introducen cláusulas que ya existan en el conjunto de las cláusulas laterales para no incurrir en bucles.

A continuación se muestra una breve API de las funciones que se encuentran en el módulo Resolutor:

- resolutor/0: Selecciona una cláusula de las centrales e invoca Resolutor/2.

-resolutor/2: Toma una cláusula perteneciente al conjunto de cláusulas laterales e invoca resolutor/3 con la cláusula central y lateral seleccionadas y la lista de cláusulas laterales.

-resolutor/3: Aplica el mecanismo de resolución lineal mediante las cláusulas seleccionadas por Resolutor/2.

-recorrer/3: Esta regla recorre la lista de cláusulas laterales para intentar buscar una con la que llegar a la cláusula vacía.

-repeticion/3: Se asegura de que no se introducen cláusulas con proposiciones repetidas.

-esta/2: Comprueba si ya existe una cláusula determinada en el conjunto de laterales para no incurrir en bucles.

2.2. Traductor

El traductor es el paso intermedio entre las fórmulas de usuario a cláusulas que podrán ser manejadas por el resolutor. Están programadas basadas en una recursión sobre patrones, `formula2clausula/1` traduce la fórmula y la inserta en el resolutor como una cláusula que puede aceptar.

Los predicados `f2c` adicionales transforman de forma recursiva la fórmula hasta que queda totalmente lista. También se define la precedencia de los operadores para que se tengan en cuenta en la transformación.

2.3. Entrada y Salida

Para leer el archivo de entrada al ejecutar `input('nombredelarchivo')`, el programa abre el fichero en modo lectura, lo almacena en Stream, y va separando por líneas, las cuales va almacenando en la lista L. Al encontrar con que ha leído todo el fichero mediante `at_end_of_stream/1`, por cada elemento de la lista, es decir, por cada línea, ejecuta `formula2clausula/1`. A continuación llama al resolutor y por último cierra el Stream.