

LoTREC

(Logical Tableaux Research Engineering Companion)



VNiVERSiDAD DE SALAMANCA

Álvaro Félix García Sánchez 5ºISI

1.~ INTRODUCCIÓN

Lotrec es un sistema creado con la finalidad de probar teoremas para lógicas modales y descriptivas. No se limita a una lógica concreta ya que el sistema está diseñado de tal manera que el usuario defina su propia lógica mediante una sintaxis creada al efecto.

Lotrec está implementado siguiendo el paradigma de la orientación a objetos, haciendo uso del lenguaje de programación Java.

2.~ DEFINIR UNA LÓGICA

2.1.~ DEFINICIÓN DE CONECTORES

Lo primero que hay que hacer es definir los conectores lógicos que se quieren usar en la pestaña “Connectors and Rules”. Se sigue el siguiente esquema para la definición de cada conector:

connector *name nb arguments associativity output priority*

connector – palabra reservada que indica que se va a definir un conector.

name – nombre que usa el programa para el conector.

nb – número de argumentos que utiliza el conector.

associativity – valor booleano (true o false) que indica si el conector es asociativo.

output – representación gráfica del conector, los argumentos se sustituyen por el símbolo “_”.

priority – Prioridad del conector respecto al resto de conectores.

Ejemplo:

connector and 2 true "_&" 3

Se define el operador lógico and, es asociativo y toma dos parámetros. La representación en pantalla se hará mediante el símbolo “&”. Se le asigna una prioridad 3 respecto al resto de conectores.

2.2.~ DEFINICIÓN DE LAS REGLAS SEMÁNTICAS DE LA LÓGICA

Las reglas consisten en definiciones de condiciones y acciones a llevar a cabo con el cumplimiento de las condiciones definidas. La sintaxis de definición es la siguiente:

```
rule nombre_de_la_regla
  if condición1
  ...
  if condiciónN
  do acción1
  ...
  do acciónk
end
```

2.2.1.~ CONDICIONES

Las condiciones se expresan siguiendo la siguiente sintaxis:

```
if nombre_de_la_condición argumentos (parametros)
```

Por ejemplo, si se quiere definir una regla para comprobar si un nodo contiene una expresión de la forma "p" escribimos la condición de la siguiente manera:

```
if hasElement node nec p
```

2.2.2.~ ACCIONES

Para definir las acciones hay que seguir la siguiente sintaxis:

```
do nombre_de_la_acción [argumentos]
```

Por ejemplo, si se desea añadir la expresión "expresión" al nodo "nodo0" habría que escribir lo siguiente:

```
do add nodo0 expresión
```

2.3.~ DEFINICIÓN DE LA ESTRATEGIA DE BÚSQUEDA

Una vez definida la semántica de la lógica hay que decirle a Lotrec como deseamos que se combinen y apliquen las reglas. La definición de las estrategias de búsqueda se realiza en la pestaña “Strategies” en la pantalla principal de Lotrec. Es posible que simplemente deseemos que las reglas definidas se apliquen secuencialmente, esta estrategia se expresaría de la siguiente forma:

repeat regla1; regla2; ... ; reglaN **end**

Con Lotrec se pueden definir estrategias mucho más sofisticadas, su definición debe adaptarse a la siguiente gramática:

```
GlobalStrategy ::= strategy NameOfStrategy Block end.
Block          ::= rule|
                  rule Block |
                  repeat Block end|
                  firstRule Block end|
                  allRules Block end.
```

Fig. 1.- Gramática normalizada para la definición de estrategias.

Se usa **firstRule** cuando se desea ejecutar únicamente la primera regla aplicable de un bloque y se usa **allRules** cuando queremos que se ejecuten todas las reglas aplicables del bloque según el orden en el que aparecen.

Ejemplo:

firstRule regla1; ...; reglaN

Si las reglas aplicables son la 1 y la 3, usando **firstRule** sólo se aplicaría **regla1**.

allRules regla1; ...; reglaN

Si las reglas aplicables son también la 1 y la 3, usando **allRules** se aplicaría primero la **regla1** y después la **regla3**.

3.~ TABLEAUX

Una vez definida la lógica sobre la que queremos trabajar podemos empezar a generar tableaux de las fórmulas que queramos. Para la introducción de las fórmulas hay que pasar a la pestaña “Formula” de la pantalla principal de Lotrec.

Hay que tener en cuenta que la fórmula debe ser introducida siguiendo la notación polaca, y precediendo hay que indicar la estrategia que se desea utilizar para generar el tableau de la fórmula.

Presionando el botón “Run” se genera el tableau correspondiente en forma de nodos, muy útil para la representación de tableaux en lógica modal. Además indica que ramas se encuentran abiertas y cuales cerradas.

4.~ EJEMPLO DE FVNCIONAMIENTO

Vamos a configurar la herramienta para que se puedan realizar demostraciones de teoremas de lógica clásica de primer orden.

En primer lugar hay que definir los conectores, para ello nos dirigimos a la pestaña “Connectors and Rules” y definimos los conectores que se usarán tal y como se indica en el apartado 2.1.

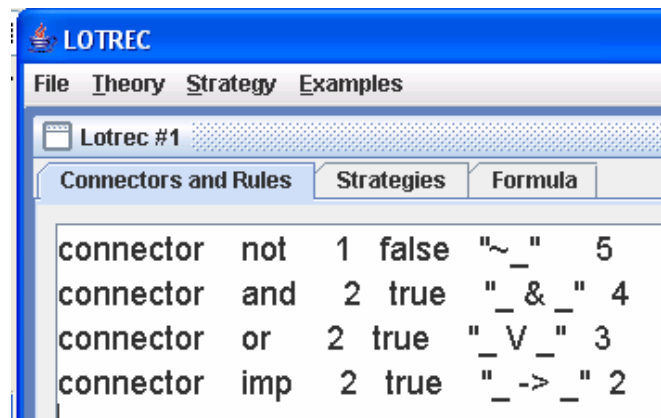


Fig 2.- Definición de conectores.

Después de los conectores, en la misma pestaña, hay que definir las reglas de resolución de tableaux. En la siguiente figura se muestran las reglas básicas en la lógica clásica de primer orden:.

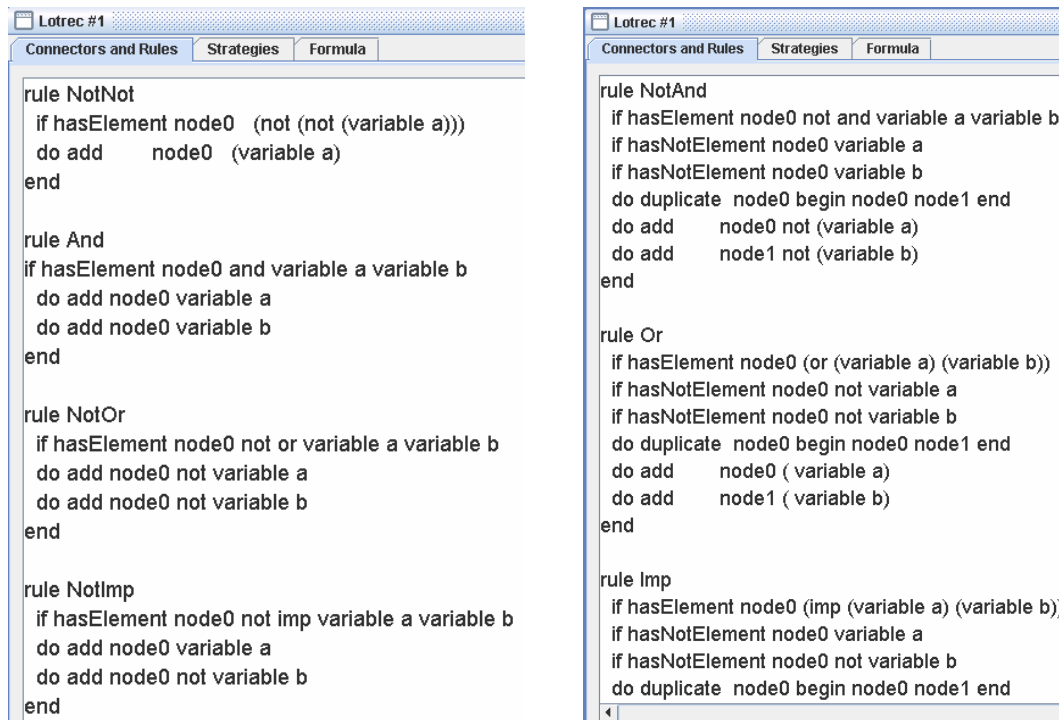


Fig 3.- Definición de reglas.

Por último, hay que definir el tipo de estrategia que se desea seguir. Para ello debemos dirigirnos a la pestaña “Strategies”, la definición se debe realizar de acuerdo a la gramática detallada en la figura 1.

En la figura se muestra la definición de la estrategia común de resolución de tableaux para la lógica clásica de primer orden, en la que primero se aplican las α -reglas y después las β -reglas hasta que no queden ramas abiertas o no haya fórmulas sin aplicar en la rama.

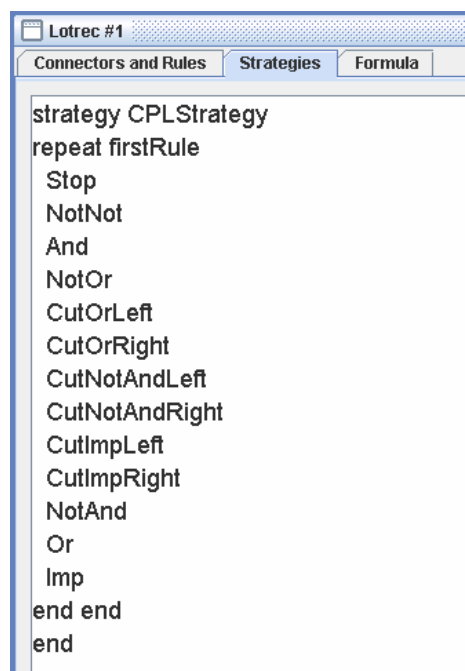


Fig 4.- Definición de la estrategia.

Finalmente llegamos a la pestaña “Formula”, dónde podremos introducir la fórmula que queremos demostrar. Debe tenerse en cuenta que la fórmula debe ser introducida siguiendo la notación polaca y precediéndola el nombre de la estrategia a usar.

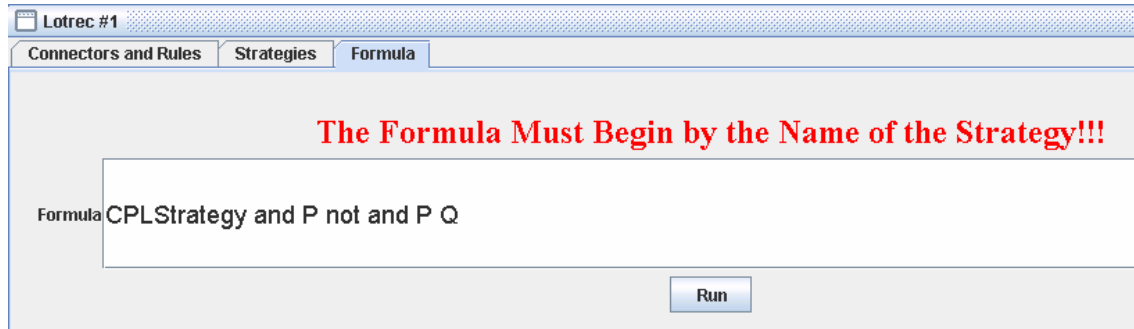


Fig 5.- Introducción de la fórmula.

Presionando el botón “Run” obtenemos el tableaux equivalente de la fórmula introducida indicándose si el tableaux está abierto o cerrado.

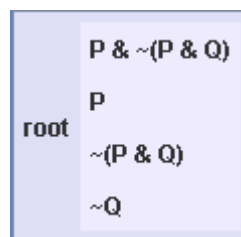


Fig 6.- Tableaux de la fórmula.

5.~ CONCLVSIONES

Lotrec es un sistema muy flexible para trabajar con varias lógicas, pero también cuenta con algunas deficiencias.

En mi opinión el uso de un lenguaje de definición de la lógica propio lo hace algo engorroso de utilizar, sobretodo al principio cuando no se está familiarizado con la herramienta. Además, podría haberse elegido un lenguaje standard de marcas haciendo así que el código generado sea más flexible.

Por otra parte, no dispone de ninguna función para el almacenamiento de las definiciones generadas, lo que obliga a usar un editor externo si no se desea perder el trabajo realizado.

Para terminar, la presentación de los tableaux resultantes de la ejecución es poco atractiva y nada intuitiva aunque es la manera más flexible y permite una visualización aceptable de tableaux para un gran número de lógicas.

Una gran ayuda en el uso de la herramienta es el repositorio de descripciones de lógicas que puede encontrarse en la página web de Lotrec (<http://www.irit.fr/Lotrec/>), ayudan a comprender el funcionamiento del mecanismo de descripción además de ser de gran utilidad para el trabajo con las lógicas que recoge.