DEDUCCIÓN NATURAL VERSUS COMPUTACIÓN

Faraón Llorens, Rosana Satorre, Francisco Escolano, Pilar Arques

I'GIA (Grupo de investigación Gráficos, Visión e Inteligencia Artificial)
Dpto. Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
e-mail: [faraon,rosana,sco,arques]@dccia.ua.es

Resumen: la lógica nos proporciona métodos de cálculo que nos permiten inferir nuevas fórmulas a partir de las conocidas, por simple manipulación sintáctica. Uno de estos métodos es la Deducción Natural, cuyo mecanismo está muy cercano al razonamiento intuitivo del ser humano. De forma sencilla, a partir de las fórmulas dadas, y mediante la aplicación de reglas, obtenemos nuevas fórmula. Podemos considerar la deducción como una forma de computación, ya que ¿un programa no es una deducción en la que a partir de unas entradas (premisas) debemos obtener unas salidas determinadas (conclusiones)?

1.- INTRODUCCIÓN.

Uno de los principales problemas al que nos enfrentamos los profesores de materias "no informáticas" (o al menos eso es lo que piensan los alumnos) en las titulaciones de informática es la actitud, en un principio, a la defensiva y/o de rechazo por parte de nuestros alumnos. La gran mayoría de los alumnos que eligen alguna de las ingenierías en informática (tanto las técnicas como la superior) solo piensan en "tocar" ordenador. Cualquier materia que se salga de esto no "va con ellos" y se limitan a intentar aprobarla a "trancas y barrancas". Este es el caso de la asignatura obligatoria de primer curso *Lógica de Primer Orden*. Aunque nadie pone en duda la gran vinculación de la informática a la lógica y pese al enfoque eminentemente computacional dado a la asignatura [Llorens96],

[Llorens98], el alumno de primer curso la ve como un escollo que debe salvar para poder dedicarse a lo que le gusta: programar.
En este trabajo se plantea un enfoque metodológico que permite acercar ambos campos: lógica y programación. Se trata de estudiar uno de los temas fundamentales de cualquier curso de lógica, el de la *inferencia* o *deducción*, comparándolo con la programación. De esta forma, por un lado, mantenemos el carácter formal de la lógica de manera que el alumno se acostumbra a trabajar con rigor y de forma abstracta. Por otro lado, su visión computacional motivará al alumno y, al bajar en nivel de abstracción, le ayudará a comprenderlo mejor.
Una deducción la podemos ver como un algoritmo que partiendo de unos valores de entrada (*premisas*) obtiene unas determinadas salidas (*conclusiones*) utilizando un conjunto dado de instrucciones (*reglas*).

2.- DEDUCCIÓN NATURAL

Uno de los aspectos fundamentales de la lógica es que, además de un lenguaje de representación de conocimiento, nos proporciona unas técnicas de razonamiento o inferencia, que nos permiten obtener nuevo conocimiento a partir del que ya poseemos. En particular vamos a trabajar con la *Deducción Natural* [Garrido95],[Reeves90], sistema formal que a partir de una premisas y con el único apoyo de unas reglas básicas, se puede llegar a determinadas conclusiones. Así, si asumimos las premisas y cada paso elemental que damos lo justificamos con una regla básica, iremos obteniendo nuevas fórmulas lógicas que podemos asumir como conclusiones derivadas de las premisas. Y ¿qué es un programa de ordenador sino una deducción?

- Un *programa* es un conjunto de instrucciones básicas que a partir de unos valores de entrada proporciona unos determinados valores de salida.
- Una *deducción* es un conjunto de pasos (cada uno consistente en la aplicando de una determinada regla) que a partir de unas premisas se obtiene una conclusión.

Reglas básicas

Para determinar las *reglas básicas* de la Deducción Natural nos basaremos en el cálculo de Gentzen [Gentzen34] que propone dos reglas (una de introducción y una de eliminación) para cada símbolo lógico (conectivas y cuantificadores). Si la regla básica introduce en su conclusión una conectiva o cuantificador que no aparece en sus premisas será una regla de

introducción; si elimina de su conclusión una conectiva o cuantificador que aparece en sus premisas será una regla de eliminación. Intuitivamente podemos ver que si disponemos de procedimientos para añadir o quitar los distintos símbolos lógicos, podemos por pura manipulación sintáctica transformar las premisas en la conclusión.

	regla introducción	regla eliminación
$\wedge$ (conjunción)	IC	EC
$\vee$ (disyunción)	ID	ED*
$\neg$ (negación)	IN*	EN
$\rightarrow$ (implicación)	II*	EI*
$\forall$ (cuantificador universal)	IU	EU
$\exists$ (cuantificador existencial)	IE	EE

* Estas reglas se conocen con nombres propios:

ED	Prueba por casos	IN	Reducción al absurdo
II	Teorema de deducción	EI	Modus Ponens

Cada línea de nuestra deducción (y por tanto la fórmula lógica escrita en ella) estará "justificada" por la aplicación de una regla básica a alguna o algunas líneas anteriores.

Subdeducciones

De manera más formal, diremos que una deducción $P_1, P_2, P_3 \Rightarrow Q$, se puede leer de forma declarativa como "si se cumplen las premisas P_1, P_2 y P_3 entonces también se cumplirá la conclusión Q" o lo que es lo mismo "se cumplirá Q si se dan P_1, P_2 y P_3". Si hacemos una lectura procedimental diremos que "para resolver el problema Q primero resolveremos los subproblemas P_1, P_2 y P_3".

El párrafo anterior nos introduce otro de los aspectos cruciales de la deducción natural: las *subpruebas* (subdeducciones o subderivaciones). En cualquier paso de nuestra deducción podemos introducir un *supuesto provisional*, que debe ser cancelado en alguna línea posterior. Desde el supuesto hasta la cancelación tendremos una subdeducción. Los supuestos provisionales son una herramienta muy potente ya que nos permiten suponer lo que nosotros queramos. Pero debemos pagar un alto precio por ello: para poder finalizar una demostración deberemos haber cancelado todos los supuestos que hayamos hecho. Por tanto, la cancelación de supuestos provisionales se convierte en una pieza clave de las deducciones naturales.

La utilización de subdeducciones nos permite "modularizar" nuestras deducciones, planteándonos subobjetivos más sencillos que el objetivo final, y que en su conjunto nos lleven a la conclusión que buscamos.

Reglas derivadas

Cuando vemos que algunas de estas subdeducciones se repiten con bastante asiduidad, podemos definir con carácter genérico dicho esquema de deducción, es decir, a partir de estas reglas básicas, podremos obtener otro tipo de reglas que se llaman *reglas derivadas*, que nos servirán como "atajos" para acortar nuestras deducciones. Así, aunque las reglas básicas son por sí solas suficientes para resolver cualquier problema de deducción formal que se presente dentro del cálculo de predicados, lo que hacemos es crear combinaciones de reglas básicas y así obtener lo que se llaman *reglas derivadas*. Una vez demostrada una regla derivada ya podemos utilizarla en nuestras deducciones. Con estas reglas vamos a conseguir algo muy importante que es el reducir el número de líneas y la complejidad de la derivación. Podemos ver a estas reglas derivadas como "atajos" que tomamos para llegar al destino, de forma que lo único que aportan son rapidez para alcanzar el destino. Serían similares a los procedimientos que nos podemos definir en cualquier lenguaje de programación. No hay nada que podamos demostrar con reglas derivadas que no podamos demostrar utilizando reglas básicas únicamente.

3.- EJEMPLO

Veamos un ejemplo de deducción natural. Supongamos que sabemos que "hay una mujer que gusta a todos los hombres". Esta sería nuestra premisa, que escrita en el lenguaje de la lógica de primer orden quedaría:

$$\exists y \{ mujer(y) \wedge \forall x [hombre(x) \rightarrow leGustaA(x,y)] \}$$

Queremos demostrar que de aquí se deduce que a "todos los hombres les gusta alguna mujer", que en forma de fórmula bien formada sería:

$$\forall x \{ hombre(x) \rightarrow \exists y [mujer(y) \wedge leGustaA(x,y)] \}$$

En las líneas siguientes aparece la deducción natural correspondiente:

1	$\exists y \{ \mathbf{mujer(y) \wedge \forall x [hombre(x) \rightarrow leGustaA(x,y)] \}}$	Premisa
2	$mujer(b) \wedge \forall x [hombre(x) \rightarrow leGustaA(x,b)]$	Supuesto
3	$mujer(b)$	EC 2
4	$\forall x [hombre(x) \rightarrow leGustaA(x,b)]$	EC 2
5	$hombre(a) \rightarrow leGustaA(a,b)$	EU 4
6	$hombre(a)$	Supuesto
7	$leGustaA(a,b)$	MP 5,6

8	mujer(b) $\wedge$ leGustaA(a,b)	IC 3,7
9	$\exists$y [mujer(y) $\wedge$ leGustaA(a,y)]	IE 8
10	hombre(a) $\rightarrow$ $\exists$y [mujer(y) $\wedge$ leGustaA(a,y)]	TD 6-9
11	hombre(a) $\rightarrow$ $\exists$y [mujer(y) $\wedge$ leGustaA(a,y)]	EE 1,2-10 *
12	$\forall$**x { hombre(x) $\rightarrow$ $\exists$y [mujer(y) $\wedge$ leGustaA(x,y)] }**	IU 11 **

* La Eliminación del Existencial, cumple las dos restricciones :

- el individuo genérico elegido en la línea 2, *b*, no aparece en ningún supuesto previo pendiente de cancelar ni premisa.
- en la fórmula donde aplicamos la cancelación (línea 10) no aparece el individuo genérico elegido, *b*.

** La Introducción del Universal también cumple la restricción: el individuo a generalizar, *a*, no aparece en ningún supuesto previo pendiente de cancelar ni premisa.

En la deducción anterior podemos observar tres partes (en columnas):

1. Numeración de las líneas, para poder hacer referencia a ellas
2. Fórmulas lógicas que vamos obteniendo. Las sangrías indican que entramos en un nuevo supuesto (hacia la derecha) o que lo cancelamos (vuelta a la izquierda). Así, el supuesto de la línea 2 es cancelado en la 11, de forma que las líneas comprendidas entre la 2 y la 10 conforman una subprueba. De la misma forma, el supuesto de la línea 6 es cancelado en la 10, siendo de la línea 6 a la 9 otra subprueba.
3. Justificación de la fórmula obtenida mediante la aplicación de una regla básica a una o más fórmulas anteriores. Por ejemplo, *EE 1,2-10* significa que esa fórmula la hemos obtenido porque tenemos un fórmula cuantificada existencialmente en la línea 1, suponemos en la línea 2 que un individuo genérico del dominio (*b*) cumple dicha fórmula y llegamos en la línea 10 a una conclusión que no depende de la elección, por lo que podemos concluir la fórmula de la línea 11.

Siguiendo con nuestra comparación con la computación, podemos ver la columna de las justificaciones como las instrucciones de nuestro programa, de forma que siguiendo esos pasos podemos realizar cualquier deducción que tenga la misma estructura (por ejemplo, nos servirá para demostrar que "cualquier número natural tiene otro menor que él" a partir de la premisa "existe un número natural menor que todos los demás"). La columna de las

fórmulas lógicas conformaría la *traza* del algoritmo (deducción) para una entrada concreta (si cambiamos las premisas, cambiarían las fórmulas). La premisa sería la entrada al programa y la conclusión la salida obtenida.

4.- REFLEXIONES

En este trabajo se ha pretendido presentar un tema marcadamente "no informático" para los alumnos utilizando una metodología que "enganche" y motive a los alumnos de informática. La idea básica es "que la deducción es una forma de computación". Aquí tenemos una tabla comparativa:

Deducción	Computación
premisas	valores de entrada
conclusión	valores de salida
reglas básica	conjunto de instrucciones del lenguaje
fórmulas de la líneas de derivación	traza del programa
justificaciones de las líneas de derivación	líneas (instrucciones) del programa
reglas derivadas	procedimientos
subdeducciones	modularización

Pero no olvidemos que la realidad es la inversa, y hay que hacerla notar a los alumnos: "la programación es una forma de demostración matemática". Por acercar el tema de la deducción a la realidad que conocen no debemos perder de vista el carácter formal y fundamental de esta disciplina. Si los planes de estudio de informática tienen materias matemáticas en sus cursos iniciales es porque deben aprender a trabajar de forma rigurosa, y aunque intentemos hacerlas didácticamente más amenas no debemos desvirtuarlas y convertirlas en puramente prácticas.

Para cerrar estas cortas reflexiones, comentar que en ambos casos (tanto la deducción como la programación) se trata de ser capaces, con la única ayuda de las herramientas formales que se nos proporcionan, de transformar unas premisas dadas en unas conclusiones determinadas.

5.- BIBLIOGRAFÍA

[Garrido95] Manuel Garrido, *Lógica simbólica*. Ed. Tecnos, 3ªedición, 1995

[Gentzen34] Gerhard Gentzen, "Untersuchungen über das logische Schliessen" (Investigaciones sobre la deducción lógica), Mathematische Zeitschrift, vol. 39, 1934.

[Llorens96] F. Llorens y Mª J. Castel. *Lógica de Primer Orden en las Ingenierías Informáticas*. II Jornadas Nacionales de Innovación en las Enseñanzas de las Ingenierías, I.C.E. Universidad Politécnica de Madrid, 1996.

[Llorens98] F. Llorens, F. Escolano, M. Pujol y O. Colomina. *Formalización del Razonamiento*. JENUI'98. IV Jornades sobre l'ensenyament universitari de la Informàtica, Sant Julià de Lòria (Principat d'Andorra) 1998

[Reeves90] S. Reeves y M. Clarke, *Logic for Computer Science*. Ed. Addison-Wesley, 1990