

# Kowalski, R. (2011) La lógica computacional y el pensamiento humano: cómo ser artificialmente inteligente



## Niveles de Conciencia

Hay un sentido en el que la conciencia puede ser entendida tanto en términos computacionales como en términos lógicos. Me refiero a cuando se dice que un agente está **conciente** cuando está **al tanto** de lo que está haciendo y del porqué lo está haciendo. Sea o no conciente un agente (en este sentido), la manifestación exterior de su conducta es la misma. Sin embargo, si el agente está conciente de lo que hace, entonces su conducta es “a propósito” y está bajo su control (NT: ¿Bajo “su reflexión”?). Si no está conciente, entonces su conducta es automática e instintiva.

La interpretación computacional de la conciencia como el “estar al tanto” dice que cuando un agente está conciente su conducta es generada por un programa de alto nivel, que manipula símbolos que tienen significado en el ambiente. Por otro lado, cuando un agente no es conciente, entonces su conducta es producida por un programa de bajo nivel o un dispositivo físico, cuya estructura está determinada, en todo caso, por las características del cuerpo del agente.

La interpretación lógica es que, cuando un agente es conciente, su conducta es producida por el razonar con metas y creencias. Cuando el agente no está conciente, entonces su conducta está determinada por asociaciones entrada-salida de bajo nivel. Estas asociaciones se puede representar, a su vez, a diferentes niveles, incluyendo un nivel lógico-simbólico y el más básico nivel físico del cuerpo

del agente.

Ambas interpretaciones se encuentran en la lógica computacional:

## **Conciencia en el metro**

Recuerde nuestra última versión del ejemplo del Aviso en el Metro:

*Metas:*

Si hay una emergencia entonces Yo obtengo ayuda.

*Creencias:*

Una persona obtiene ayuda si la persona alerta al conductor.

Una persona alerta al conductor si la persona presiona el botón de la señal de alarma.

Hay una emergencia si hay fuego.

Hay una emergencia si una persona ataca a otra.

Hay una emergencia si alguien se enferma gravemente.

Hay una emergencia si hay un accidente.

Hay fuego si hay llamas.

Hay fuego si hay humo.

Un pasajero puede usar la meta y las creencias

explícitamente, razonando hacia adelante a partir de las observaciones, para reconocer cuando hay una emergencia y derivar la meta de conseguir (ob-tener) ayuda, y luego, razonando hacia atrás, conseguir ayuda al presionar el botón de la señal de alarma.

Sin embargo, el pasajero podría llegar a la misma conducta usando asociaciones de entrada-salida de bajo nivel o reglas de condición-acción, que puede también ser representadas en forma lógica.

### **Metas:**

Si hay llamas entonces Yo presiono el botón de la señal de alarma.

Si hay humo entonces Yo presiono el botón de la señal de alarma.

Si una persona ataca a otra entonces Yo presiono el botón de la señal de alarma.

Si alguien se enferma gravemente entonces Yo presiono el botón de la señal de alarma.

Si hay un accidente entonces Yo presiono el botón de la señal de alarma.

La nueva meta es más eficiente que la meta y la creencia original. Sólo requieren un paso de razonamiento hacia adelante para asociar la acción apropiada de salida con la observación relevante en la entrada. Al respecto, se asemejan a un programa escrito en un lenguaje de bajo nivel, que es más eficiente que un programa con la misma funcionalidad en un lenguaje de nivel más alto.

De hecho, en este caso, ambos programas están escritos en

el mismo lenguaje lógico. El programa original está escrito a mayor nivel, lo cual requiere, no solamente más recursos computacionales, sino también mecanismos de razonamiento más sofisticados.

En Computación, los lenguajes de diferentes niveles y las representaciones de diferentes niveles en el mismo lenguaje tienen diferentes ventajas y son complementarios. Las representaciones de bajo nivel son más eficientes. Pero las de alto nivel son más flexibles, fáciles de desarrollar y fáciles de cambiar.

En este ejemplo, la representación de bajo nivel “no está al tanto”, lo que si ocurre con la representación de mayor nivel, de la meta de conseguir ayuda que es el propósito de presionar el botón de la señal de alarma. Si algo sale mal con la representación de bajo nivel, por ejemplo si el botón no sirve o el conductor no consigue ayuda, entonces el pasajero podría no darse cuenta que hay un problema. Además, si el ambiente cambia, y se desarrollan métodos nuevos y mejores para resolver emergencias, será más difícil modificar la representación para adaptarla al cambio.

En Computación, ambos niveles de representación son útiles. Típicamente, la representación de nivel superior es la primera en ser desarrollada, algunas veces ni siquiera en la forma de un programa, sino como un análisis de los requerimientos del programa[1]. Esta representación de alto nivel es luego **transformada**, bien manualmente o por medio de programas que algunas veces se llaman **compiladores**, en una representación de más bajo nivel que se puede ejecutar más eficientemente.

El proceso inverso también es posible. Los programas de

bajo nivel pueden, en ciertas circunstancias, ser **decompilados** en programas equivalentes de mayor nivel. Esto es muy útil cuando el programa de más bajo nivel debe ser modificado, quizás porque debe operar en un ambiente diferente o porque se ha descubierto que tiene una falla de programación. La representación de alto nivel puede entonces ser modificada y **recompilada** en una nueva forma de bajo nivel mejorada.

Este proceso inverso, sin embargo, no siempre es posible. Los llamados sistemas legados, desarrollados directamente en lenguajes de bajo nivel y que han sido modificados durante años pueden carecer de la estructura necesaria para que se pueda identificar sus metas sin ambigüedades y se les pueda decompilar en la forma de nivel superior. Pero inclusive en estos casos, se les podría decompilar parcialmente y aproximar con programas de alto nivel. Este proceso de reconstrucción racional pueden servir para mejorar la mantenibilidad de los sistemas legados, aún cuando una re-implementación completa no sea posible.

## **Compilando por medio del razonamiento adelantado.**

Uno puede realizar parte del trabajo de compilar un programa de alto nivel en una forma de bajo nivel, realizando por adelantado algo de la computación que se realizaría al invocar el programa original.

En el ejemplo del metro, en lugar de aguardar a que ocurra la emergencia, un compilador puede anticipar la necesidad de reducir la conclusión de la meta de mantenimiento de nivel superior:

***Meta: Si hay una emergencia entonces consigue ayuda.***

Haciendo todo el razonamiento hacia atrás necesario, por adelantado. En dos de esos pasos de inferencia, la meta puede ser transformada en una versión declarativa de una regla de condición-acción:

***Meta: Si hay una emergencia entonces presiono el botón de la señal de alarma.***

De la misma manera, el compilador puede anticipar la necesidad de reconocer que hay una emergencia, haciendo todo el razonamiento hacia adelante por adelantado. En dos pasos de inferencia, que corresponden a las dos maneras de reconocer que hay fuego, el compilador puede derivar las nuevas creencias:

***Creencias:***

***Hay una emergencia si hay llamas.***

***Hay una emergencia si hay humo.***

En cinco pasos adicionales de inferencia, que corresponden a los cinco tipos de emergencia, el compilador obtiene las asociaciones simples de entrada-salida que vimos antes:

***Metas:***

***Si hay llamas entonces presiono el botón de la señal de alarma.***

***Si hay humo entonces presiono el botón de la señal de alarma.***

**Si una persona ataca a otra entonces presiono el botón de la señal de alarma.**

**Si alguien se enferma gravemente entonces, presiono el botón de la señal de alarma.**

**Si hay un accidente entonces presiono el botón de la señal de alarma.**

Esta representación es tan baja como puede serlo, pero preserva la forma lógica. Sin embargo, es posible bajar más aún, si esas asociaciones son implantadas con conexiones físicas directas entre las partes relevantes de los sistemas sensoriales y motores humanos. Esto último equivale a implantar el software en el hardware.

## **Combinando el pensamiento racional con el pensamiento intuitivo.**

En la psicología cognitiva se ha hecho una distinción similar a la discutida en la sección anterior entre el **pensamiento racional** (*deliberative thinking*) y el **pensamiento intuitivo** (*intuitive thinking*). El pensamiento racional, conciente de sí mismo y controlado, incluye al pensar lógico. El pensamiento intuitivo, que es opaco y automático, lleva los procesos de percepción hasta los niveles subconcientes del pensar. Los psicólogos cognitivistas han propuesto varios modelos de procesamiento dual del pensamiento humano, en los cuáles interactúan esos dos tipos de pensamiento.

La interacción más simple ocurre el pensamiento racional se desplaza hacia el nivel intuitivo con el paso del tiempo, por ejemplo, cuando una persona aprende a usar el teclado,

tocar un instrumento musical o conducir un carro. Esta migración del pensamiento racional hacia el intuitivo es equivalente a compilar un programa de alto nivel en un programa de nivel inferior – pero no razonando por adelantado, sino razonando con los hechos. Luego de usar la misma combinación de procedimientos de propósito general de alto nivel durante mucho tiempo (o muchas repeticiones), la combinación colapsa en un “atajo computacional” de bajo nivel. Este atajo es un procedimiento especializado que alcanza el mismo resultado que el procedimiento general original, pero con mayor eficiencia.

Algunas veces podemos ir en la dirección opuesta, reflexionando sobre el conocimiento subconciente y representándolo concientemente en términos explícitos – Por ejemplo, cuando un lingüista trata de construir una gramática formal para un lenguaje. Mientras el hablante nativo de un lenguaje conoce la gramática tácita y subconcientemente, el lingüista intenta formular una gramática explícita. Un maestro podría usar la gramática resultante para enseñar a los estudiantes. Si estos practican lo suficiente, llegarán a compilar la gramática explícita en sus propios subconcientes y aprenderán a hablar el lenguaje con mayor correctitud y eficiencia.

De la misma manera que muchos programa de bajo nivel no pueden ser completamente decompilados, sino que sólo se les puede aproximar con programas de nivel superior, es probable que mucho de nuestro pensar subconciente sólo pueda ser aproximado hasta cierto punto por el pensar conciente. Las gramáticas formales de los lingüistas siguen siendo un buen ejemplo.

En el pensar humano, los dos niveles del pensar pueden operar en cascada. En el modelo dual de Kahneman-Tversky,



el nivel subconciente e intuitivo “rápidamente propone respuestas intuitivas a los problemas de elección en la medida en que surgen”, mientras que el nivel conciente, racional “supervisa la calidad de esas propuestas y puede apoyarlas, corregirlas o reemplazarlas”[\[2\]](#).

El uso de la gramática formal para supervisar y corregir el uso instintivo del lenguaje natural es un ejemplo familiar.

## **La lógica como el lenguaje de alto nivel del pensamiento.**

La imagen de la lógica que emerge de estas consideraciones es la de un lenguaje de alto nivel para el pensamiento racional, controlado y conciente. En ese alto nivel, el razonamiento lógico reduce metas a sub-metas, deriva las consecuencias de las observaciones e infiere las consecuencias de las candidatas a acciones. Este razonamiento se puede hacer cuando haga falta para actuar o se le puede hacer por adelantado. Cuando se hace por adelantado, sirve para transformar representaciones de alto nivel de metas y creencias en representaciones de bajo nivel más eficientes. En este nivel inferior de representación, la conducta es instintiva, automática y subconciente. Todas estas relaciones se pueden apreciar en esta figura:

[\[1\]](#) Como en el ejemplo de la diseñadora del piojo.

[\[2\]](#) “Representatividad revisada: Substitución de atributos en juicio intuitivo”- “Representativeness revisited: Attribute substitution in intuitive judgement”, Daniel Kahneman and

