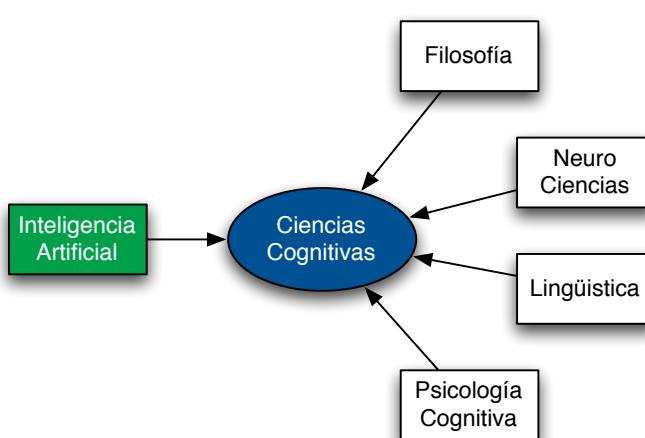


La Inteligencia Artificial (IA) tiene como objeto de estudio a las entidades inteligentes y su comportamiento; pero a diferencia de la filosofía, la psicología, las neurociencias, y demás disciplinas que comparten este objeto de estudio, su meta no tiene que ver únicamente con la comprensión de tales entidades, sino con su construcción. De hecho, como se muestra en la Figura 1.1, Varela [22] ubica a la IA como parte de las Ciencias Cognitivas resaltando esta asimetría: Su interés, único en el área, por la síntesis de entidades inteligentes. La **construcción de agentes racionales** constituye la idea central del enfoque adoptado en este curso, propuesto por Russell y Norvig [17], y curiosamente calificado de moderno<sup>1</sup>.

*Agencia e IA*



**Figura 1.1:** La IA como parte de las Ciencias Cognitivas, según Varela [22], resalta por su interés en la síntesis de entidades inteligentes y no solo en su característico análisis, presente en otras disciplinas del área.

Aunque muchos practicantes de la IA se consideran a si mismos científicos de la computación, como bien señala Kayser [10] parafraseando al famoso cuadro de Magritte (Fig. 1.2) –*Ceci n'est pas de l'informatique*. O más precisamente, la no es reducible a la computación. Sus intereses, representación de conocimientos incluida, son anteriores a ella y más amplios.

*IA y Computación*

A propósito del cuadro, Magritte nos ofrece un comentario sobre el mismo, ideal para iniciar este curso:

La famosa pipa. ¡Como la gente me reprochó por ello! Y sin embargo, ¿Podría usted llenar mi pipa? No, sólo es una representación ¿No es así? ¡Así que si hubiera escrito en mi cuadro “Esto es una pipa”, habría estado mintiendo! [21].

En esta introducción abordaremos intuitivamente, algunas de las cuestiones planteadas por el cuadro de Magritte y su relación con el tema central de este curso: ¿Qué queremos decir por conocimiento, representación y razonamiento?, ¿Qué es la verdad? y ¿Porqué creemos que estos conceptos son útiles para la construcción de sistemas inteligentes? Para ello, nos guiarímos con el capítulo uno del libro sobre representación del conocimiento de Brachman y Levesque [1] y otras lecturas complementarias [14, 6].

<sup>1</sup> AIMA (*Artificial Intelligence: A Modern Approach*) es el libro de texto más usado en el mundo sobre IA. Se tiene registro de su uso en más de 1500 universidades: <https://aima.cs.berkeley.edu/index.html>



**Figura 1.2:** De la serie la traición de las imágenes de René Magritte, el texto dice “Esto no es una pipa”.

Siendo así las cosas, la **Representación del Conocimiento** es el área de la IA que estudia cómo el conocimiento puede ser representado y procesado simbólicamente mediante programas que razonan automáticamente –El estudio del pensamiento como un proceso computacional.

*Representación de Conocimiento*

Si bien, tal estudio está ligado a la cuestión de cómo es que un agente utiliza su conocimiento para satisfacer sus metas en su medio ambiente, aquí focalizaremos en el **conocimiento**, no en el conocedor<sup>2</sup>: Nos preguntaremos qué necesita saber un agente –humano, animal, electrónico, ó mecánico, para comportarse inteligentemente; y qué clase de mecanismos computacionales le permiten acceder a ese conocimiento cuando así lo requiere.

*Agencia y Conocimiento*

En lo que sigue, la Sección 1.1 introduce una serie de conceptos básicos a los que haremos referencia continuamente: Conocimiento, proposición, valor de verdad, actitud proposicional, creencia, representación, símbolo, razonamiento, e inferencia entre otros. Posteriormente, la Sección 1.2 aborda la cuestión de la relevancia de la representación del conocimiento para la IA, discutiendo la postura intencional e introduciendo una hipótesis de trabajo. La Sección 1.3 introduce lo que será nuestro interés tecnológico principal: Los Sistemas Basados en el Conocimiento. La Sección 1.4 abunda sobre la relevancia de usar una base de conocimiento en estos sistemas y la Sección 1.5 hace lo propio con respecto al razonamiento y los mecanismos de inferencia. El capítulo no podría terminar sin una pequeña nota sobre el uso de la lógica en estas tareas, de lo cual da cuenta la Sección 1.6.

*Organización*

## 1.1 CONCEPTOS BÁSICOS

La cuestión sobre qué es el **conocimiento** es anterior a la IA y a la computación. Se ha abordado desde la antigua Grecia y sigue siendo una cuestión de interés. No entraremos aquí en detalles filosóficos, pero si intentaremos ofrecer una versión informal de lo que se supone es el conocimiento, útil para los propósitos del curso.

*Conocimiento*

Primero, entre otras cosas, el conocimiento es una relación entre un agente y una **proposición** –Una idea expresada como un enunciado declarativo. Observen que es común decir “Ana sabe que...” donde a los puntos suspensivos sigue una proposición. Por ejemplo, “Ana sabe que la clase de representación de conocimiento es martes y jueves”.

*Proposición*

<sup>2</sup> Un enfoque con énfasis en el conocedor, el agente, se ofrece en nuestro curso sobre Sistemas MultiAgentes: <https://www.uv.mx/personal/aguerra/sma/>

Una parte del misterio acerca del conocimiento se debe a la naturaleza de las proposiciones. Por ahora, lo que nos interesa de las proposiciones es que son entidades abstractas sujetas a **valores de verdad**: Pueden ser verdaderas o falsas, correctas o incorrectas. Decir que “Ana sabe  $P$ ” equivale a decir que “Ana sabe que  $P$  es verdad”. Observen que esto refleja un juicio donde el agente se da cuenta de que su medio ambiente es de cierta forma (Uno donde  $P$  es el caso), y no de otras (Donde  $P$  no es el caso).

*Valor de verdad*

Algo similar sucede con frases como “Ana desea que la clase de representación de conocimiento sea los martes y jueves”. Se trata de la misma proposición del ejemplo anterior, pero su relación con el agente es diferente. Los verbos como saber, creer, desear, intentar, etc., denotan lo que se conoce como **actitudes proposicionales**. Independientemente de la actitud, lo que importa con respecto a la proposición es su valor de verdad. Si “Ana desea que  $P$ ”, entonces Ana desea que el medio ambiente sea tal, que  $P$  es el caso.

*Actitudes proposicionales*

Las actitudes proposicionales **creer** y **saber** guardan cierta relación. Usamos la primera cuando deseamos expresar que el juicio del agente no es necesariamente preciso, o que no se sostiene por las razones adecuadas. A veces también se usa para expresar que el agente no está completamente convencido de que la proposición sea el caso. Lo importante aquí es que ambas actitudes comparten una idea básica sobre el conocimiento: Expresan que el agente asume que el mundo es de cierta forma y no de otra.

*Creencia*

El concepto de representación es tan escurridizo como el de conocimiento. En líneas generales, una **representación** es una relación entre dos dominios, donde se pretende que el primero denote o tome el lugar del segundo. Normalmente el primer dominio es más concreto, inmediato o accesible que el segundo. Por ejemplo, el diagrama de la bomba de gasolina en los avisos de la carretera es más concreto que la frase “Gasolinera”.

*Representación*

De especial interés para nosotros es la representación basada en **símbolos formales**, esto es, un carácter o secuencia de caracteres tomada de algún alfabeto. Por ejemplo, el dígito 6 representa al número seis, lo mismo que la secuencia de letras VI y, como no, seis y six. Como en toda representación, se asume que es más fácil contender con los símbolos, que con lo que representan. Observen que el caso de las proposiciones es una representación simbólica.

*Símbolo*

La **representación del conocimiento** es entonces el área de estudio concerniente al uso de los símbolos formales para representar una colección de proposiciones creídas por un agente putativo. No estamos afirmando que los símbolos representarán todas las proposiciones creídas por el agente, podría haber un número infinito de proposiciones creídas y solo un número finito de ellas representadas. Es el razonamiento el que tiende un puente entre lo representado y lo creído.

*Representación del conocimiento*

Por **razonamiento** entendemos la manipulación formal de los símbolos que representan colecciones de proposiciones creídas, para producir representaciones de nuevas proposiciones. Aquí explotamos el hecho de que el símbolo sea más simple que lo que representa: Las representaciones simbólicas deben ser lo suficientemente concretas para manipularlas (cambiarlas de sitio, eliminarlas, copiarlas, concatenarlas, etc.) de tal manera, que se posible construir nuevas representaciones.

*Razonamiento*

Brachman y Levesque [1] nos recuerdan una brillante analogía con la aritmética: Podemos pensar en la suma como una manipulación formal, de los símbolos 103 y 8, podemos construir 111 para denotar la suma de ellos. La manipulación (suma por columnas, acarreo, etc.) define el concepto suma. El razonamiento es similar: De proposiciones como “Todos los días martes hay clase de representación de conocimiento” y “El día de hoy es martes” podemos construir el enunciado “Hay clase de representación de conocimiento”. A esta forma de razonamiento le solemos llamar **inferencia lógica**, porque la proposición final es una conclusión lógica de las primeras. De forma que, como fue por primera vez expuesto por Leibniz, el razonamiento es una forma de **cálculo** que en lugar de operar sobre números, como la aritmética, lo hace sobre proposiciones.

*Inferencia lógica**Cálculo*

## 1.2 REPRESENTACIÓN DEL CONOCIMIENTO E IA

¿Porqué es relevante el conocimiento para la IA? Una primer respuesta sería que pareciera que los humanos somos capaces de exhibir un **comportamiento inteligente**, gracias a lo que sabemos.

*Comportamiento*

Otra respuesta es que a veces resulta útil describir el comportamiento de sistemas que son suficientemente complejos, usando un vocabulario que incluye términos como creencias, deseos, metas, intenciones, etc. Por ejemplo, al jugar ajedrez contra un programa podemos decir que nuestro contrincante movió su alfil porque creía que su reina era vulnerable, pero aún desea amenazar mi caballo. En términos de cómo el programa está construído, tal explicación se reduciría a que usando el procedimiento de evaluación  $P$  con la función de evaluación estática  $Q$ , se obtuvo un valor de +9 en una búsqueda alfa-beta minimax de profundidad 4. La segunda descripción, aunque precisa, está expresada en un nivel de detalle incorrecto y no nos ayuda a determinar que movimiento haré yo en consecuencia. La primera, que resulta más útil para este fin, descansa en lo que Dennett [7] denomina la **postura intencional**.

*Postura intencional*

Lo anterior no significa que la postura intencional sea siempre la adecuada y la clave está en la complejidad del sistema que se quiere representar: Asumir la postura intencional para describir el apagador de luz resulta innecesario, inadecuado y caricaturesco. McCarthy [12] estableció claramente que en computación la postura intencional es útil para contender con nuestra limitación para adquirir conocimiento, usarlo para predecir y establecer generalizaciones en términos de la estructura de un programa; siempre y cuando las actitudes proposicionales reflejen el uso que se les suele dar.

Ahora bien, ¿Es esto todo lo que entendemos por representación de conocimiento? Es solo hablar acerca del conocimiento. Observen que la postura intencional no dice nada acerca de qué está y qué no está representado simbólicamente. El programa de ajedrez representa simbólicamente la posición de las piezas en el tablero, pero, por ejemplo, su deseo de eliminar mi caballo lo más pronto posible puede no estarlo. Puede tratarse de una propiedad emergente de diferentes componentes del programa: sus funciones de evaluación, su librería de movimientos, etc. Y aún así, resulta útil para mi saber que el programa desea despachar a mi caballo. ¿Cuál es el rol entonces de la representación de conocimiento?

*Hipótesis*

La **hipótesis** subyacente en el área de representación de conocimiento es que podemos construir sistemas que contengan representaciones simbólicas con dos propiedades importantes: Que desde fuera, nosotros podamos entender qué representan las proposiciones del sistema; y que internamente, el sistema haya sido diseñado para comportarse como lo hace debido a estas representaciones.

## 1.3 SISTEMAS BASADOS EN EL CONOCIMIENTO

Para entender en qué consiste un sistema basado en el conocimiento usaremos dos programas muy simples en Prolog. Veamos el primero:

```

1 adivina_color(nieve) :- !, write("Es blanca.").
2 adivina_color(pasto) :- !, write("Es verde.").
3 adivina_color(cielo) :- !, write("Es amarillo.").
4 adivina_color(_) :- write("Me has ganado.").
```

Y he aquí el segundo:

```

1 adivina_color(X) :-
2   color(X,Y), !,
3   write("Es de color "), write(Y), write(".").
4 adivina_color(_) :- write("Me has ganado.").
5
6 color(nieve,blanca).
```

```

7 | color(cielo,amarillo).
8 | color(vegetacion,verde).
9 | color(X,Y) :-
10 |   hecho_de(X,Z),
11 |   color(Z,Y).
12 |
13 | hecho_de(pasto,vegetacion).
14 | hecho_de(selva, vegetacion).

```

Observen que ambos programas pueden adivinar el color de varios objetos, aunque se equivoque con respecto al cielo. Desde la postura intencional, podemos decir que ambos saben que el color de la nieve es blanco. Sin embargo, como veremos más adelante, solo el segundo programa está diseñado en conformidad con la hipótesis de la representación del conocimiento.

Consideremos la cláusula `color(nieve,blanca)`, por ejemplo. Se trata de una estructura simbólica que puede verse como una representación de la proposición “la nieve es blanca”; mejor aún, gracias a que sabemos como funciona Prolog, sabemos que el sistema imprimirá la frase adecuada debido a que el programa dará con la cláusula en el momento adecuado. Si eliminamos la cláusula el programa dejará de actuar de esa manera. Ahora bien, no existe una cláusula similar en el primer programa. Lo más cercano es la primer cláusula, pero no resulta natural interpretarla como una creencia.

De forma que lo que hace que un sistema esté basado en conocimiento, no es el uso de un formalismo lógico (Prolog en este caso); ni el hecho de que el sistema sea lo suficientemente complejo como para justificar el uso de la postura intencional y sus descripciones sobre el conocimiento; ni el hecho de lo que creemos es verdadero. Lo importante es la presencia de una **base de conocimientos**, una colección de estructuras simbólicas que representan lo que el agente cree y razona, durante su operación.

*Base de conocimiento*

Diversos sistemas propuestos por la IA funcionan de esta manera. Los **sistemas expertos** son un claro ejemplo de ello, pero se pueden encontrar bases de conocimiento en sistemas para el procesamiento de lenguaje natural, planeación, diagnóstico y aprendizaje. Otros sistemas incluyen en menor grado conocimiento, como algunos juegos y sistemas de visión de alto nivel (reconocimiento de objetos). Y por supuesto, hay una parte de la IA que no está basada en el conocimiento: Reconocimiento de habla de bajo nivel, visión, control motor, entre otros.

*Sistemas expertos*

## 1.4 ¿PORQUÉ UNA BASE DE CONOCIMIENTO?

Una pregunta obvia que surge al comparar los dos programas Prolog de la sección anterior es ¿Qué ventajas tiene, si acaso tiene alguna, usar una base de conocimiento? Los defensores del llamado **conocimiento procedimental** defenderían al primer programa: Una especie de compilación de la base de conocimientos del segundo programa que distribuye lo que debe saberse para actuar en cada módulo que así lo requiere. De hecho, el desempeño del sistema será mejor en este caso, ya que buscar los hechos en la base de conocimientos para posteriormente, decidir qué hacer, solo puede hacer que el sistema sea más lento.

*Conocimiento procedimental*

Sin embargo, desde el punto de vista del diseño de sistemas, la aproximación basada en el conocimiento del segundo programa tiene las siguientes **propiedades deseables**:

*Propiedades deseables*

- Podemos agregar nuevas tareas y fácilmente hacer que dependan del conocimiento previo. Por ejemplo, en nuestro programa podemos agregar la tarea de enumerar todos los objetos de un color dado e incluso pintar un dibujo, haciendo uso de la base de conocimientos previamente especificada para determinar los colores a usar.

- Podemos extender el comportamiento del programa agregando nuevas creencias. Por ejemplo, al agregar una cláusula que especifique que el color de los canarios es amarillo, automáticamente propagamos esa información a cualquier rutina que la necesite.
- Podemos depurar el comportamiento erróneo del sistema localizando creencias erróneas en el mismo. Por ejemplo, corregir la cláusula que especifica que el color del cielo es amarillo.
- Podemos explicar y justificar de manera concisa el comportamiento del sistema. ¿Porqué dice mi programa que el pasto es verde? Porque cree que el pasto es una forma de vegetación y la vegetación es de color verde. Tal explicación se justifica plenamente, de hecho, si eliminamos alguna cláusula del programa, la explicación no se sostiene, pero el comportamiento del mismo será diferente.

Resumiendo, la ventaja de los sistemas basados en el conocimiento es que, por diseño, pueden ser informados de nuevos hechos sobre el mundo y ajustar su comportamiento en consecuencia. La propiedad de algunas de nuestras acciones de ser dependientes de lo que creemos, es lo que Zenon W. Pylyshyn llama **penetrabilidad cognitiva**. Por ejemplo, nuestra respuesta normal al oír una alarma contra incendios sería levantarnos y salir del lugar donde nos encontramos para ir a un área segura. Sin embargo, la respuesta es diferente si sabemos que la alarma está siendo probada. Nuestro reflejo de retirar la mano del calor, por el contrario, no parece ser cognitivamente penetrable.

*Penetrabilidad cognitiva*

## 1.5 ¿PORQUÉ EL RAZONAMIENTO?

La motivación detrás del razonamiento tiene que ver con el hecho de que nos gustaría que la actuación del sistema dependiera no únicamente de lo que está explícitamente representado, sino de lo que el sistema cree como consecuencia de ello. En el segundo programa Prolog, no hay una cláusula que diga que el pasto es verde, y aún así queremos que el sistema sepa esto. Mucho del conocimiento que guardamos en una base tiene que ver con hechos muy generales que necesitarán posteriormente aplicarse en situaciones particulares. Por ejemplo, podríamos querer representar las siguientes dos cláusulas de forma explícita:

1. Paciente X alérgico al medicamento  $M$ .
2. Cualquier alérgico al medicamento  $M$  también es alérgico al medicamento  $M'$ .

Cuando tratemos de decidir si es apropiado darle el medicamento  $M'$  al paciente X, ninguno de los hechos representados resuelven la cuestión individualmente. Sin embargo, juntos proveen la información de que X es alérgico a  $M'$ . Esto y otros hechos acerca de las alergias, son suficientes para desaconsejar tal medicación. No queremos condicionar el comportamiento solo a aquellos hechos que podamos colectar a manera de una **base de datos**, las creencias de un sistema basado en el conocimiento deben ir más allá.

*Base de datos*

¿Más allá a donde? Existe una respuesta simple a esta cuestión, aunque no siempre sea práctica: El sistema debe creer  $P$  si, conforme a las creencias que tiene representadas, el mundo que se imagina es uno donde  $P$  es verdadera. Si las cláusulas (1) y (2) están representadas en el sistema y el mundo es tal que son verdaderas, entonces en ese mundo

3. El paciente X es alérgico al medicamento  $M'$ .

también es una cláusula verdadera, aún cuando este hecho está representado solo implícitamente. Esto es ilustra el concepto de **consecuencia lógica**: Decimos que las proposiciones representadas por un conjunto de enunciados  $\Delta$  tienen como consecuencia la proposición representada por el enunciado  $P$  cuando la verdad de  $P$  está implícita en la verdad de los enunciados de  $\Delta$ . En otras palabras, si el mundo es tal que cada enunciado en  $\Delta$  es verdadero, entonces este es el caso también para  $P$ . Todo lo que necesitamos para obtener una noción de consecuencia, es un lenguaje donde esté definido que significa que un enunciado sea falso o verdadero, algo necesario en nuestro caso. De forma que cualquier lenguaje de representación de conocimiento, independientemente de otras características que pueda poseer y sean cuales sean su sintaxis y sus procedimientos de razonamiento, tendría que tener una noción de consecuencia bien definida.

*Consecuencia lógica*

De manera que una respuesta simple a la cuestión de que creencias debe poseer un sistema basado en el conocimiento, sería: Todas aquellas y solo aquellas que son consecuencia lógica de sus representaciones explícitas. La tarea de un mecanismo de razonamiento es entonces computar las consecuencias lógicas de una base de conocimiento. La razón por la cual tal respuesta no es siempre práctica tiene que ver con que comúnmente hay buenas razones para no computar tales consecuencias. Una razón es que puede ser muy difícil, costoso computacionalmente hablando, calcular que enunciados son consecuencia de la base de conocimiento que nos interesa usar. Todo procedimiento que siempre nos de una respuesta en un tiempo razonable, ocasionalmente perderá algunas consecuencias o nos dará una respuesta equivocada. En el primer caso, decimos que el proceso de razonamiento es lógicamente **incompleto**; en el segundo decimos que el procedimiento no es **sólido** (en inglés, *unsound*).

*Compleitud  
Solidez*

Tenemos razones para tomar en consideración el razonamiento no sólido e incompleto. Un ejemplo clásico: Supongamos que todo lo que sé acerca de Piolín es que es un ave. Podría tener un cierto número de hechos acerca de las aves en mi base de conocimiento, pero difícilmente estas tendrían como consecuencia que Piolín vuela (podría ser el caso que Piolín fuese una avestruz). Y sin embargo, es razonable creer que Piolín vuela. Estamos ante un razonamiento no sólido, donde las cláusulas de la base de datos pueden ser verdaderas en el mundo, mientras que Piolín no vuela.

*Consistencia*

Otro caso a considerar es aquel en donde el sistema integra en su base de conocimiento hechos provenientes de varias fuentes que resultan **inconsistentes**, es decir, no pueden ser verdaderos juntos. En tal caso, es inapropiado computar la consecuencia lógica completa del sistema, porque entonces todo enunciado sería creído: Puesto que no hay mundos donde la base de conocimientos sea verdadera, el consecuente se vuelve trivialmente verdadero. En este caso es evidente que una forma de razonamiento incompleto sería más conveniente, al menos hasta que se pueda resolver la contradicción.

Evidentemente, en algunos casos la respuesta simple es la más adecuada y aunque sea un error equiparar razonamiento en una base de conocimientos con el concepto de inferencia lógica completa y sólida, tal aproximación puede ser un buen comienzo.

## 1.6 EL PAPEL DE LA LÓGICA

La razón por la cual la **lógica** es relevante para la representación de conocimiento y el razonamiento es que, al menos conforme a un punto de vista, ésta es el estudio de las relaciones de consecuencia –lenguajes, condiciones de verdad y reglas de inferencia. Es por tanto normal, que usemos una gran cantidad de herramientas y técnicas provenientes de la lógica simbólica formal. Particularmente, haremos uso de la **lógica de primer orden** (FOL, por sus siglas en inglés), aunque claro, esto solo es un punto de partida y consideraremos también lenguajes bien distintos en forma y significado.

*Lógica  
FOL*

Donde la lógica reditúa realmente es en lo que Newell [14] llama *the knowledge level*. La idea es que todo sistema basado en el conocimiento puede entenderse en al menos dos niveles diferentes. En el nivel del conocimiento, hacemos preguntas concernientes a el lenguaje de representación y su semántica. Al nivel del símbolo, por otra parte, hacemos preguntas concernientes a aspectos computacionales. Claramente hay asuntos pertinentes a cada nivel: Al nivel de conocimiento contendemos con la expresividad del lenguaje de representación, las características de su relación de consecuencia, incluyendo su complejidad computacional intrínseca; a nivel del símbolo, contendemos con la arquitectura computacional usada, las propiedades de las estructuras de datos empleadas y los procedimientos de razonamiento, incluyendo su complejidad algorítmica. La Figura 1.3 ilustra las posibles descripciones de una computadora a diferentes niveles.

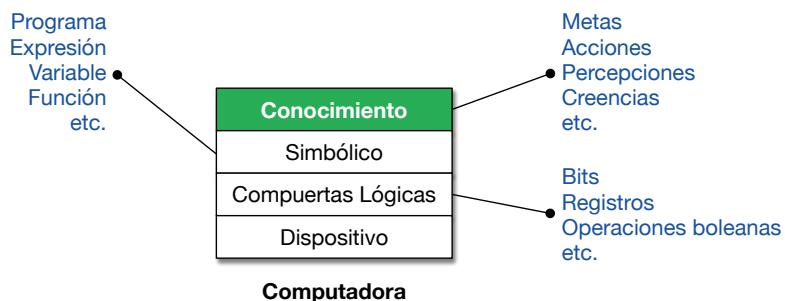


Figura 1.3: Niveles de descripción de una computadora según Newell [14].

Las herramientas de la lógica simbólica parecen adecuadas para el análisis a nivel del conocimiento. En los siguientes capítulos abordaremos la lógica proposicional y de primer orden, dejando de lado por ahora los aspectos computacionales.

## 1.7 LECTURAS Y EJERCICIOS SUGERIDOS

Esta introducción esta basada en las ideas expresadas por Brachman y Levesque [1], cuyo libro servirá de base a este curso. Una vista rápida de esta visión de la IA puede obtenerse en un artículo previo de Brachman [2]. Levesque [11] ofrece una introducción a nivel de inicio de licenciatura en su libro sobre el pensamiento computacional. Una reseña de mi autoría<sup>3</sup> sobre este texto, puede consultarse en la revista Komputer Sapiens (8):3, de la Sociedad Mexicana de Inteligencia Artificial. Otra perspectiva interesante para introducirse en el área de la representación del conocimiento es la ofrecida por Davis, Shrobe y Szolovits [6]. Harmelen, Lifschitz y Porter [9] proveen una colección de artículos a la manera de un *handbook* muy completo sobre el área. También he tomado en cuenta el libro de Kayser [10] y las conversaciones de pasillo y metro con él, durante mi estancia doctoral en Paris 13.

Casi todos los libros sobre IA incluyen una sección sobre representación del conocimiento. En nuestro caso, el contenido del curso está influenciado por la parte III y parcialmente la IV del libro de Russell y Norvig [18]. El concepto de agente aquí discutido también está influenciado por el libro de Wooldridge [23]. Una excelente introducción al uso de los formalismos lógicos en la IA se encuentra en el libro de Genesereth y Nilsson [8].

La referencia obligada a Prolog es el libro de Bratko [3], que puede complementarse con el de Clocksin y Melish [5]. Otra alternativa es el excelente libro de Sterling y Shapiro [20]. Para revisar las técnicas avanzadas de este lenguaje se recomiendan los libros de Clocksin [4] y de Shoham [19]. Las notas de nuestro curso de Programación para la IA<sup>4</sup> también pueden ser de ayuda en este aspecto. Negrete-Martínez,

*El nivel del conocimiento*

<sup>3</sup> [http://smia.mx/komputersapiens/download.php?file=ks83\\_14MB\\_extensa.pdf](http://smia.mx/komputersapiens/download.php?file=ks83_14MB_extensa.pdf)

<sup>4</sup> <https://www.uv.mx/personal/aguerra/pia/>

González-Pérez y Guerra-Hernández [13] ofrecen una aproximación a los sistemas basados en el conocimiento con implementaciones en Lisp. Una gran parte del libro de Norvig [16] también aborda el uso de Lisp en estas tareas.

## Ejercicios

**Ejercicio 1.1.** Explique la diferencia entre creencia y conocimiento. Nuestro primer ejemplo de sistema basado en el conocimiento implementado en Prolog ¿Sabe o cree que ciertos objetos son de cierto color?

**Ejercicio 1.2.** ¿Porqué razones la cláusula `bel(color(cielo,amarillo))` es incorrecta para expresar en Prolog que creemos que el color del cielo es amarillo?

**Ejercicio 1.3.** ¿Qué diferencia hay entre la manipulación simbólica que define el razonamiento en este capítulo y la expresada en la hipótesis del sistema simbólico físico propuesta por Newell y Simon [15]?

**Ejercicio 1.4.** Describa un caso en el que la postura intencional nos podría resultar de utilidad de computación.

**Ejercicio 1.5.** Si vemos a Prolog como un sistema basado en el conocimiento, ¿Dónde reside su base de conocimientos? ¿Cuál es su método de inferencia? ¿Se trata de un método completo, sólido y consistente?

**Ejercicio 1.6.** Describa con detalle las ideas detrás del nivel de conocimiento propuesto por Newell [14] y su relevancia para la IA en general, y este curso en particular.