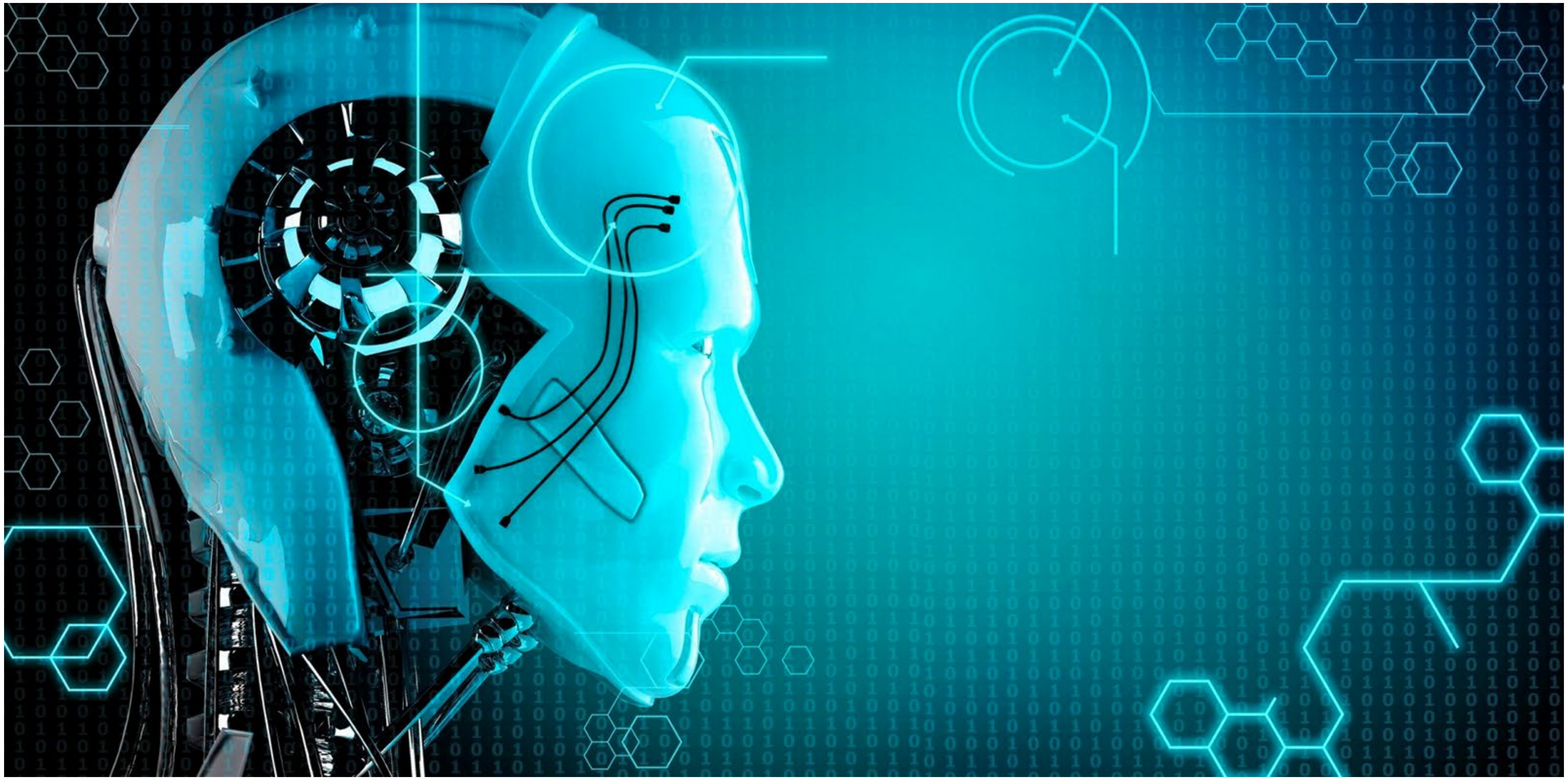
Inteligencia Artificial

Lógica de primer orden



[Transparencias adaptadas de Dan Klein and Pieter Abbeel: CS188 Intro to AI, UC Berkeley (ai.berkeley.edu)]

Universidad Rey Juan Carlos

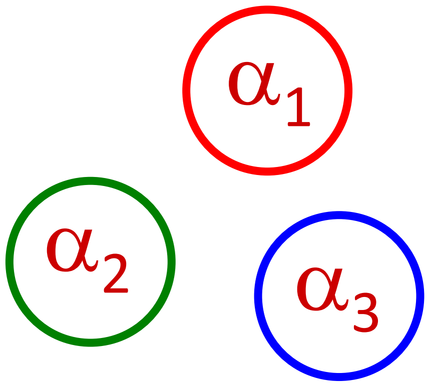
Repaso: Conocimiento

- **Base de conocimiento (KB)** = conjunto de *sentencias* (que representan afirmaciones sobre el mundo) en un lenguaje formal
- Enfoque declarativo para crear un agente (u otro sistema):
  - **Dile** lo que necesita saber (o pídele que **aprenda** el conocimiento)
  - A continuación, puede **preguntarse** qué hacer: las respuestas deben obtenerse de la base de conocimiento
- Los agentes pueden verse a **nivel de conocimiento**
  es decir, lo que **saben**, se implemente como se implemente
- Un único algoritmo de inferencia puede responder a cualquier pregunta que se pueda responder

Base de conocimiento	Hechos específicos del dominio
Motor de inferencia	Código genérico

Repaso: Lógica

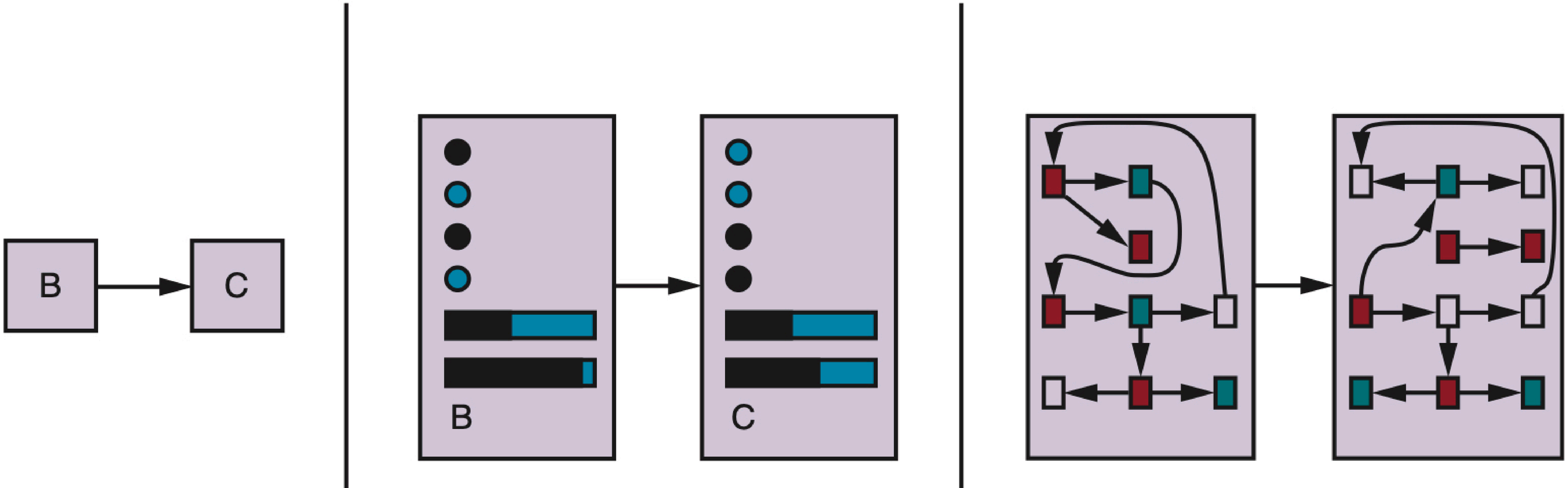
- **Sintaxis**: ¿Qué sentencias están permitidas?
- **Semántica:**
  - ¿Cuáles son los **mundos posibles**?
  - ¿Qué frases son **verdaderas** en qué mundos (modelos)? (es decir, **definición** de verdad)
    - Se dice que el modelo m "satisface α" o "es un modelo de α"
    - $M(\alpha)$ representa todos los modelos de α



Sintaxisland

Semanticaland

Espectro de representaciones



(a) Atómica | (b) Factorizada | (c) Estructurada

Problemas de búsqueda

Planificación, **lógica proposicional**, redes Bayesianas, redes neuronales

Lógica de primer orden, bases de datos, programas lógicos, programas probabilísticos

Poder expresivo

- Objetivo: adoptar las bases de la lógica proposicional (independiente del contexto, sin ambigüedades) y construir sobre ellas una lógica **más expresiva tomando ideas del lenguaje natural** y evitando sus defectos.
- Reglas del ajedrez
  - 100.000 páginas en lógica proposicional
  - 1 página en lógica de primer orden
- Reglas de Pacman:
  - $\forall t\ Alive(t) \Leftrightarrow$
    $[Alive(t-1) \wedge \neg\exists\ g,x,y\ [Ghost(g) \wedge At(Pacman,x,y,t-1) \wedge At(g,x,y,t-1)]]$

Elementos del lenguaje natural

- Sustantivos y grupos nominales: **Objetos**
- Verbos: **Relaciones** que pueden ser *unarias* (**propiedades**, adjetivos) o n-arias (binarias, ternarias, cuaternarias, etc.)
- Algunas de estas relaciones son **funciones**: solo hay un *valor* para una determinada *entrada* (mapeo entre objetos)

- *"Los cuadrados colindantes al Wumpus son malolientes"*
- *"El malvado rey Juan gobernó Inglaterra en 1200"*
- *"Uno más dos es igual a tres"*

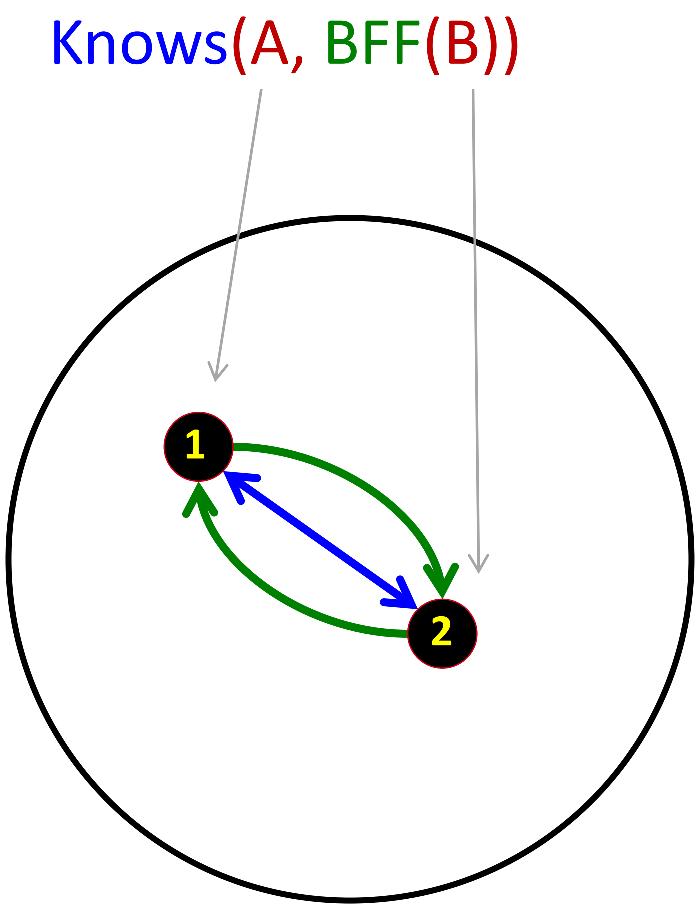
La lógica de primer orden se construye sobre objetos y relaciones. Además, permite expresar hechos sobre *algunos* o *todos* los objetos del universo.

Lógica de primer orden (FOL) vs. proposicional

- Distinto *compromiso ontológico*: se diferencian en lo que asumen sobre la naturaleza de la realidad.
- **Lógica proposicional**: supone que hay **hechos que se dan o no se dan** en el mundo. Cada hecho puede estar en uno de dos estados: verdadero o falso, y cada modelo asigna verdadero o falso a cada símbolo de proposición.
- **Lógica de primer orden**: supone algo más; en concreto, que el mundo está formado por objetos con ciertas **relaciones** entre ellos **que se mantienen o no**. Los modelos formales son más complicados que los de la lógica proposicional.

Mundos posibles

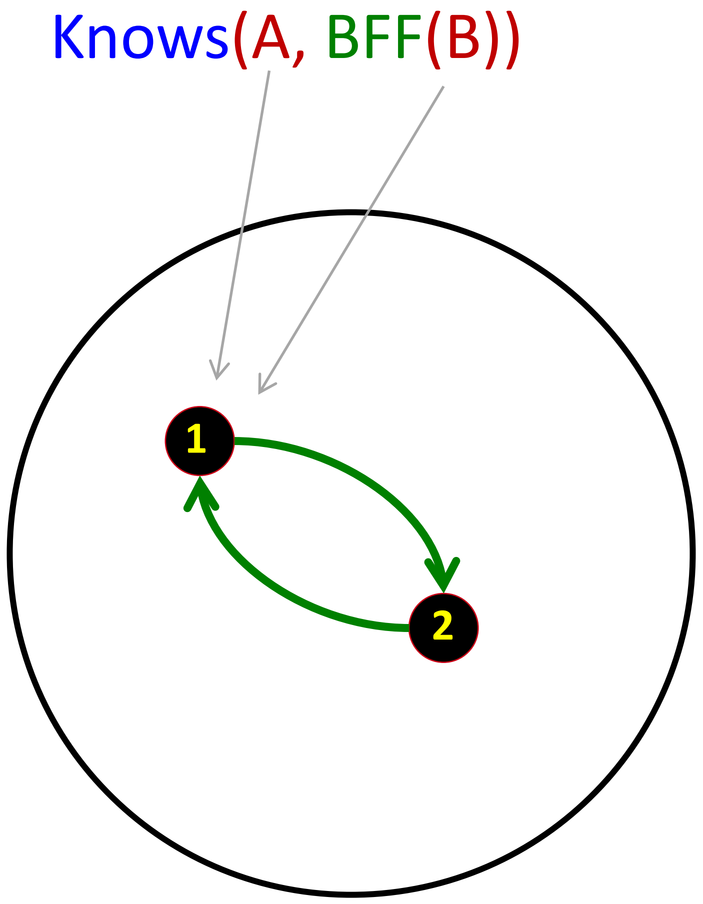
- Un mundo posible en lógica de primer orden (FOL) está formado por:
  - Un conjunto no vacío de **objetos**
  - Para cada **predicado (relación)** k-ario en el lenguaje, un conjunto de tuplas de k objetos (es decir, el conjunto de tuplas de objetos que satisfacen el predicado en este mundo)
  - Para cada **función** k-aria en el lenguaje, un mapeo de las tuplas de k objetos a objetos
  - Para cada **símbolo constante**, un objeto particular (se pueden considerar las constantes funciones 0-arias)



Mundo posible

Mundos posibles

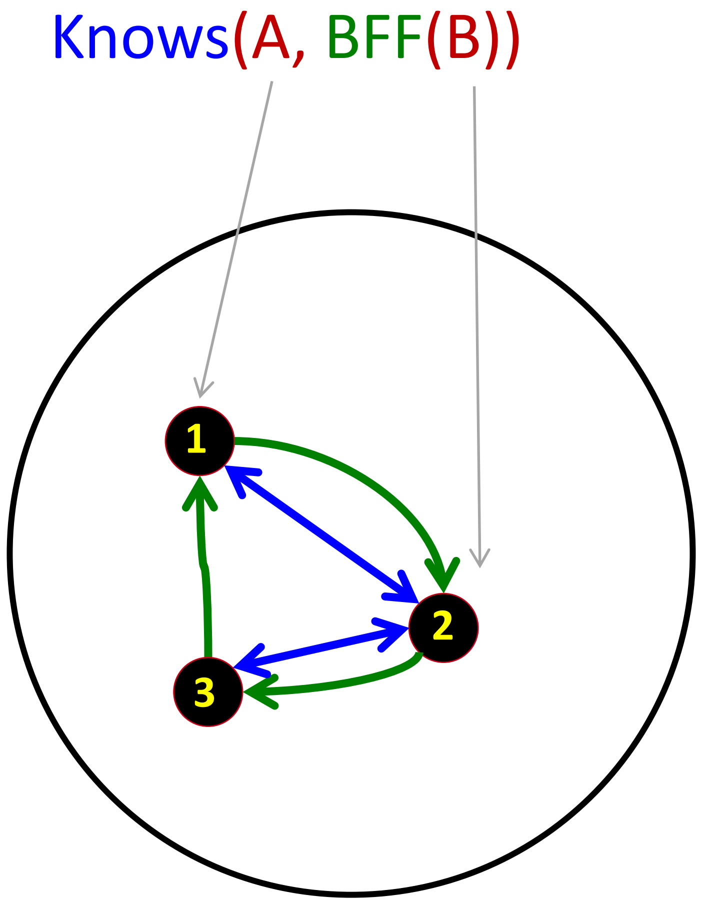
- Un mundo posible en lógica de primer orden (FOL) está formado por:
  - Un conjunto no vacío de **objetos**
  - Para cada **predicado (relación)** k-ario en el lenguaje, un conjunto de tuplas de k objetos (es decir, el conjunto de tuplas de objetos que satisfacen el predicado en este mundo)
  - Para cada **función** k-aria en el lenguaje, un mapeo de las tuplas de k objetos a objetos
  - Para cada **símbolo constante**, un objeto particular (se pueden considerar las constantes funciones 0-arias)



Mundo posible

Mundos posibles

- Un mundo posible en lógica de primer orden (FOL) está formado por:
  - Un conjunto no vacío de **objetos**
  - Para cada **predicado (relación)** k-ario en el lenguaje, un conjunto de tuplas de k objetos (es decir, el conjunto de tuplas de objetos que satisfacen el predicado en este mundo)
  - Para cada **función** k-aria en el lenguaje, un mapeo de las tuplas de k objetos a objetos
  - Para cada **símbolo constante**, un objeto particular (se pueden considerar las constantes funciones 0-arias)

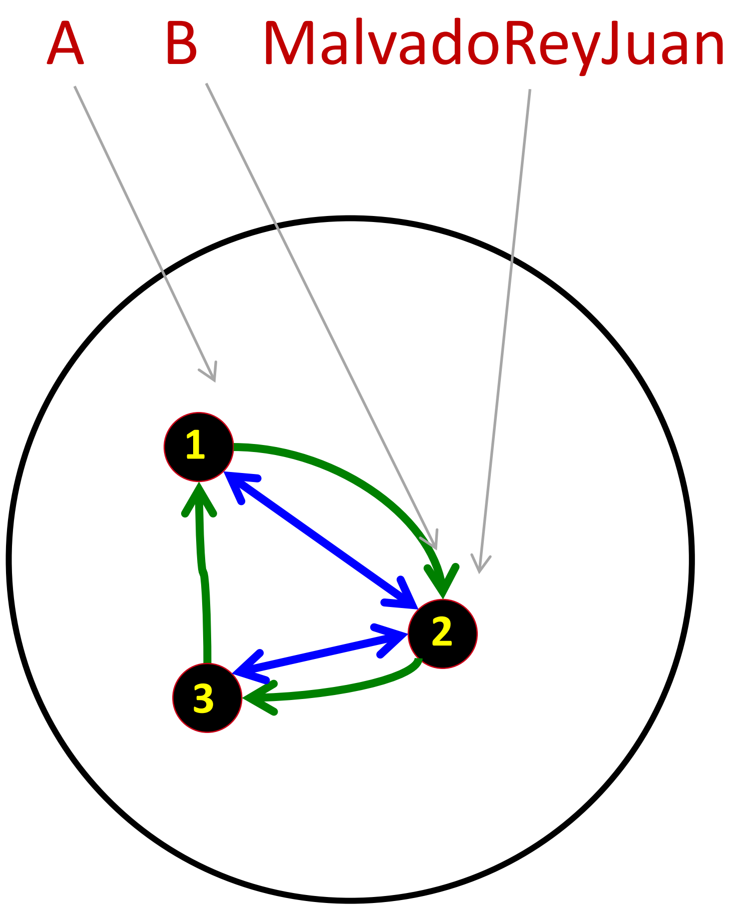


Mundo posible

¿Cuántos mundos posibles?

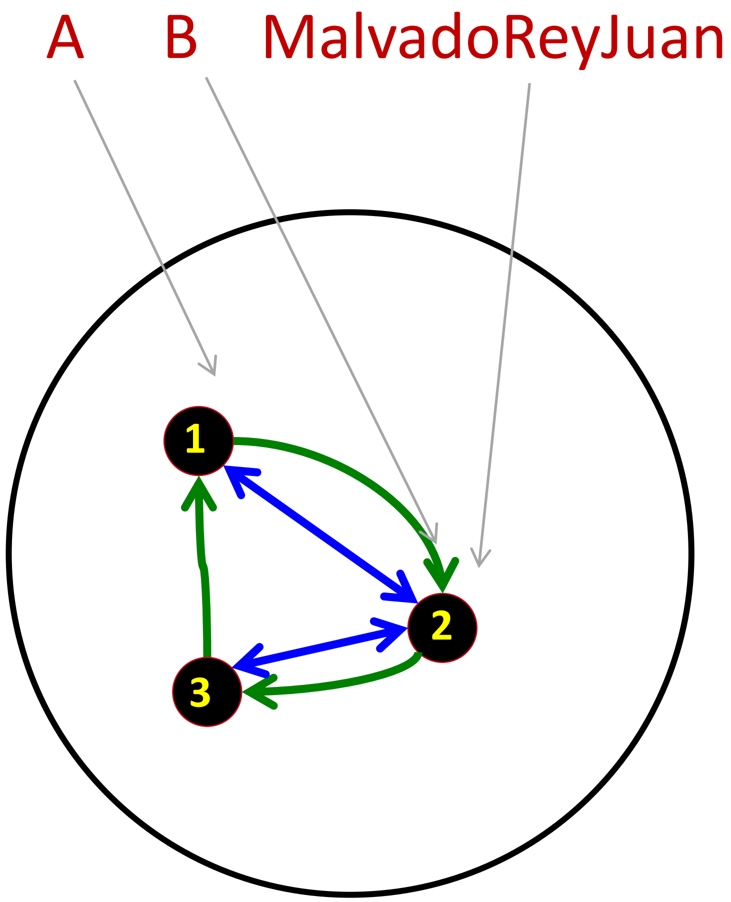
Sintaxis y semántica: términos

- Un término se refiere a un objeto; puede ser:
  - Un **símbolo constante**; por ejemplo: A, B, MalvadoReyJuan
    - El mundo posible fija estos referentes
  - Un **símbolo de función** con términos como argumentos; por ejemplo, BFF(MalvadoReyJuan)
    - El mundo posible especifica el valor de la función, dados los referentes de los términos
      - BFF(MalvadoReyJuan) -> BFF(**2**) -> **3**
  - Una variable lógica; por ejemplo, x
    - (lo vamos a ver justo después)



Sintaxis y semántica: sentencias atómicas

- Una sentencia atómica es una proposición elemental (vs. símbolos en log. proposicional)
  - Un símbolo de predicado con términos como argumentos; p. ej., Knows(A, BFF(B))
    - Verdadero si y solo si los objetos a los que se refieren los términos están en la relación a la que se refiere el predicado
    - Knows(A,BFF(B)) -> Knows(**1**,BFF(**2**)) -> Knows(**1**,**3**) -> **F**
  - Una igualdad entre términos; p. ej., BFF(BFF(BFF(B)))=B
    - Verdadero si y solo si los términos se refieren a los mismos objetos
    - BFF(BFF(BFF(B)))=B -> BFF(BFF(BFF(**2**)))=**2** -> BFF(BFF(**3**))=**2** -> BFF(**1**)=**2** -> **2**=**2** -> **V**



Sintaxis y semántica: sentencias complejas

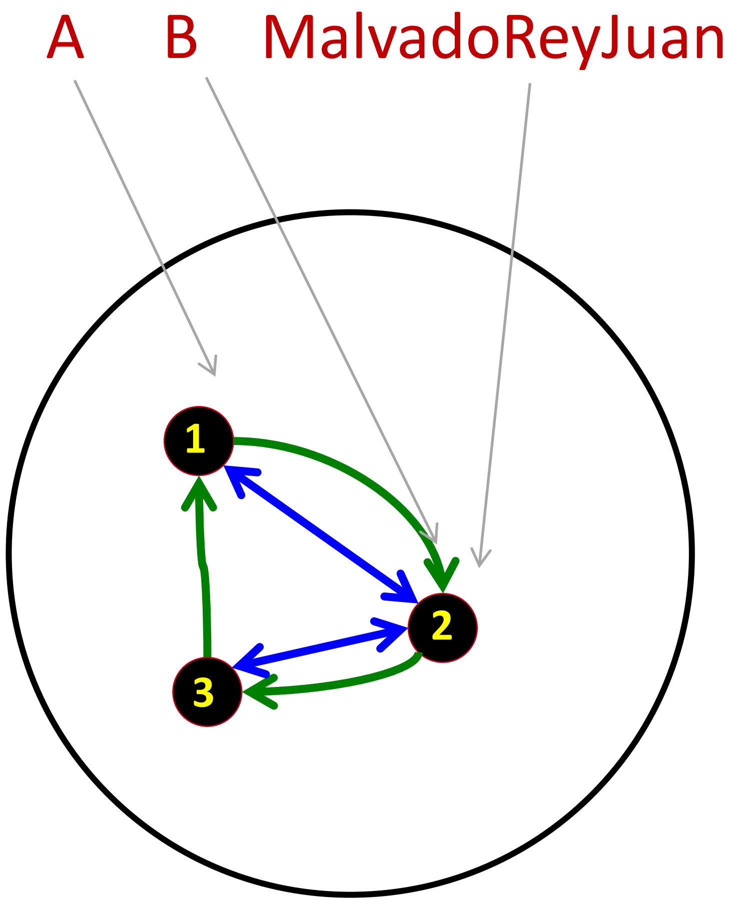
- Sentencias con conectores lógicos

  $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \Rightarrow \beta$, $\alpha \Leftrightarrow \beta$

- Sentencias con cuantificadores universales o existenciales; por ejemplo:
  - $\forall$x Knows(x, BFF(x)):

    Verdadero en el mundo w si y solo si es Verdadero **en todas las extensiones** de w donde x se refiere a un objeto en w

    - x -> **1**: Knows(**1**, BFF(**1**)) -> Knows(**1**,**2**) -> **V**
    - x -> **2**: Knows(**2**, BFF(**2**)) -> Knows(**2**,**3**) -> **V**
    - x -> **3**: Knows(**3**, BFF(**3**)) -> Knows(**3**,**1**) -> **F**



Sintaxis y semántica: sentencias complejas

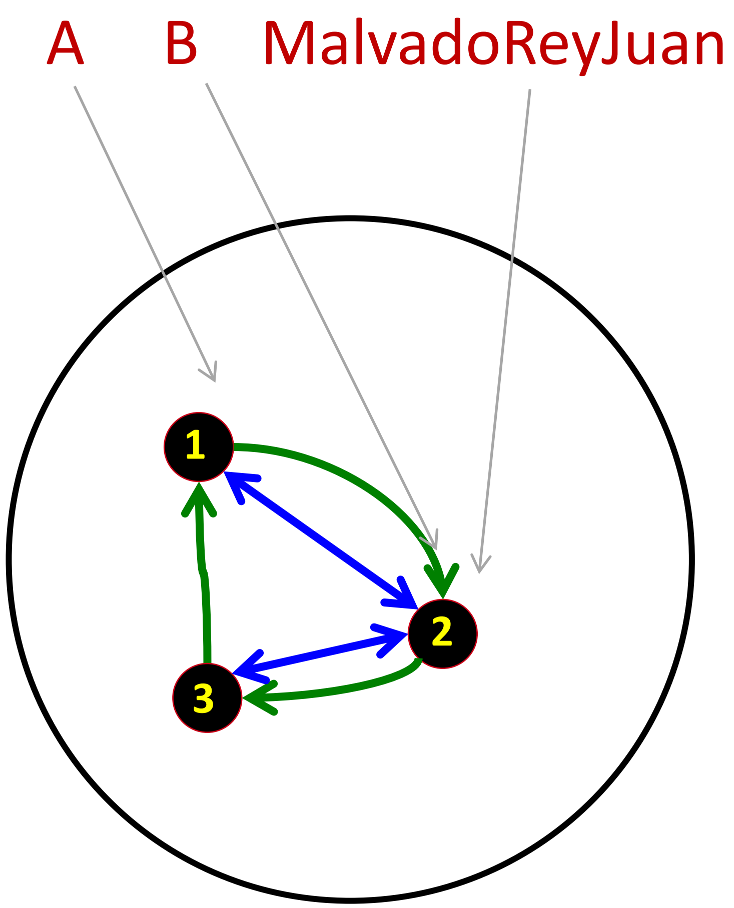
- Sentencias con conectores lógicos

  $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \Rightarrow \beta$, $\alpha \Leftrightarrow \beta$

- Sentencias con cuantificadores universales o existenciales; por ejemplo:
  - ∃x Knows(x, BFF(x)):

    Verdadero en el mundo w si y solo si es Verdadero **en alguna extensión** de w donde x se refiere a un objeto en w

    - x -> **1**: Knows(**1**, BFF(**1**)) -> Knows(**1**,**2**) -> **V**
    - x -> **2**: Knows(**2**, BFF(**2**)) -> Knows(**2**,**3**) -> **V**
    - x -> **3**: Knows(**3**, BFF(**3**)) -> Knows(**3**,**1**) -> **F**



Ejemplos de sentencias

- Todo el mundo conoce a Obama
  - $\forall n\ Person(n) \Rightarrow Knows(n,Obama)$

- Hay alguien al que todo el mundo conoce
  - $\exists s\ Person(s) \wedge \forall n\ Person(n) \Rightarrow Knows(n,s)$
- Todo el mundo conoce a alguien
  - $\forall x\ Person(x) \Rightarrow \exists y\ Person(y) \wedge Knows(x,y)$

Más ejemplos de sentencias

- Dos personas cualesquiera de la misma nacionalidad hablan una lengua común
  - Nationality(x,n) – x tiene nacionalidad n
  - Speaks(x,l) – x habla el idioma l
  - $\forall x,y\ (\exists\ n\ Nationality(x,n) \wedge Nationality(y,n)) \Rightarrow$

    $(\exists\ l\ Speaks(x,l) \wedge Speaks(y,l))$

Inferencia en lógica de primer orden (FOL)

- La implicación se define exactamente igual que en lógica proposicional:
  - $\alpha \models \beta$ ("α implica β") sii en cada mundo donde α es verdad, β también
  - P. ej., $\forall x\ Knows(x,Obama)$ implica $\exists y \forall x\ Knows(x,y)$
- En FOL, podemos ir más allá de responder "sí" o "no"; dada una consulta existencialmente cuantificada, se devuelve una sustitución (o ligadura) para la(s) variable(s) de tal manera que la frase resultante esté implicada: KB = $\forall x\ Knows(x,Obama)$
  - Pregunta = $\exists y \forall x\ Knows(x,y)$
  - Respuesta = Sí, $\sigma = \{y/Obama\}$
  - Notación: $\alpha\sigma$ significa aplicar la sustitución σ a la sentencia α
    - Si $\alpha = \forall x\ Knows(x,y)$ y $\sigma = \{y/Obama\}$, entonces $\alpha\sigma = \forall x\ Knows(x,Obama)$

Ejemplo: el mundo de Wumpus

Percepciones

- Vector de percepciones de cinco elementos.
- Sentencia en la KB de FOL: el vector y el **tiempo** en el que se percibió

$$Percept([Stench, Breeze, Glitter, None, None], 5) \leftarrow \text{para t = 5}$$

- La percepción implica hechos sobre el estado actual

$$\forall\, t, s, g, m, c \;\; Percept([s, Breeze, g, m, c], t) \;\Rightarrow\; Breeze(t)$$
$$\forall\, t, s, b, m, c \;\; Percept([s, b, Glitter, m, c], t) \;\Rightarrow\; Glitter(t)$$

Ejemplo: el mundo de Wumpus

Acciones

$$Turn(Right), \quad Turn(Left), \quad Forward, \quad Shoot, \quad Grab, \quad Climb$$

- ¿Cómo determinar cuál es la mejor acción?

$$\textsc{AskVars}(\exists\, a \;\; BestAction(a, 5))$$

- En FOL, esto devuelve algo como: $\{a/Grab\}$ (sustitución)
- Pueden implementarse comportamientos reflejos con sentencias condicionales

$$\forall\, t \;\; Glitter(t) \;\Rightarrow\; BestAction(Grab, t)$$

Ejemplo: el mundo de Wumpus

Términos

- Las casillas no serán $Casilla_{1,2}$ sino [1, 2]
  - Esto nos permite, por ejemplo, definir la adyacencia de forma concisa:

$$\forall x, y, a, b \;\; Adjacent([x, y], [a, b]) \;\Leftrightarrow \\ (x = a \land (y = b - 1 \lor y = b + 1)) \lor (y = b \land (x = a - 1 \lor x = a + 1))$$

- Predicado unario *Pit* que se evalúa a verdadero en las casillas con hoyos
- Constante *Wumpus*

Ejemplo: el mundo de Wumpus

Físicas (ejemplos)

- Los objetos solo pueden estar en una posición en un instante

$$\forall x, s_1, s_2, t \;\; At(x, s_1, t) \land At(x, s_2, t) \;\Rightarrow\; s_1 = s_2$$

- Si el agente está en una casilla y percibe una brisa (*Breeze*), entonces esa casilla es ventosa (*Breezy*)

$$\forall s, t \;\; At(Agent, s, t) \land Breeze(t) \;\Rightarrow\; Breezy(s)$$

- Una vez descubierto qué lugares son ventosos (o malolientes) y, muy importante, no ventosos (o no malolientes), el agente puede deducir dónde están los pozos (y dónde el Wumpus)

$$\forall s \;\; Breezy(s) \;\Leftrightarrow\; \exists r \;\; Adjacent(r, s) \land Pit(r)$$

- También podemos cuantificar sobre el tiempo:

$$\forall t \;\; HaveArrow(t+1) \;\Leftrightarrow\; (HaveArrow(t) \land \neg Action(Shoot, t))$$

Inferencia en FOL: Proposicionalización

- Convertir (KB $\wedge \neg\alpha$) a lógica proposicional (PL), y utilizar un solucionador SAT de PL para comprobar la (in)satisfacibilidad
  - Idea: sustituir variables por términos básicos, convertir sentencias atómicas en símbolos (definiciones formales en siguiente diapositiva)
    - $\forall$x Knows(x,Obama) y Democrat(Biden)
      - Knows(Obama,Obama) y Knows(Biden,Obama) y Democrat(Obama)
      - Knows_Obama_Obama $\wedge$ Knows_Biden_Obama $\wedge$ Democrat_Biden
    - pero $\forall$x Knows(Mother(x),x)
      - Knows(Mother(Obama),Obama) y Knows(Mother(Mother(Obama)),Mother(Obama))...

Anidamiento infinito...

  - Truco: para *k* = 1 a infinito, utilizar todos los posibles términos con un anidamiento de llamadas a la función hasta *k*
    - Si hay implicación, encontraremos una contradicción para una k finita (Herbrand); si no, puede continuar para siempre: ***semidecidible***

Inferencia en FOL: Proposicionalización

- Para convertir a PL cuando hay variables:
- Instanciación universal (UI):
  - Podemos inferir cualquier sentencia obtenida sustituyendo un término base (un término sin variables) por la variable

$$\frac{\forall x\ King(x) \land Greedy(x) \Rightarrow Evil(x)}{\begin{array}{c} King(John) \land Greedy(John) \Rightarrow Evil(John) \\ King(Richard) \land Greedy(Richard) \Rightarrow Evil(Richard) \\ King(Father\ (John)) \land Greedy(Father\ (John)) \Rightarrow Evil(Father\ (John)) \\ \dots \end{array}}$$

- Instanciación existencial:
  - La variable se sustituye por un **nuevo** símbolo constante

$$\exists x\ Crown(x) \land OnHead(x, John)$$

$$\Downarrow$$

$$Crown(C_1) \land OnHead(C_1, John)$$

Inferencia en FOL: Inferencia por elevación

- Aplicar reglas de inferencia directamente a sentencias de primer orden; por ejemplo:
  - KB = Person(Socrates), $\forall$x Person(x) $\Rightarrow$ Mortal(x)
  - concluir Mortal(Socrates)
  - La regla general es una versión "elevada" de Modus Ponens:
    - Dados $\alpha \Rightarrow \beta$ y α', donde $\alpha'\sigma = \alpha\sigma$ para alguna sustitución σ, concluir $\beta\sigma$
      - En el ejemplo, σ es {x/Socrates}
    - Dado Knows(x,Obama) y Knows(y,z) $\Rightarrow$ Likes(y,z)
      - σ es {y/x, z/Obama}, concluir Likes(x,Obama)
- **Unificación**: encontrar sustituciones que hacen que expresiones lógicas diferentes parezcan idénticas

  UNIFY(Knows(John, x), Knows(y, Mother(y))) = {y/John, x/Mother(John)}

  - Cuando se compara una variable con un término complejo, se debe comprobar si la propia variable se encuentra dentro del término; si es así, la comparación falla porque no se puede construir un unificador coherente.

Resumen

- La lógica de primer orden (FOL) es un lenguaje formal muy expresivo
- Muchos ámbitos del sentido común y del conocimiento técnico pueden escribirse en FOL:
  - circuitos, software, planificación, legislación, impuestos, protocolos de red y seguridad, descripciones de productos, transacciones de comercio electrónico, sistemas de información geográfica, Google Knowledge Graph, web semántica, etc.
- En general, la inferencia es semidecidible; en la práctica, muchos problemas pueden resolverse eficientemente.