

INTRODUCCIÓN A AGENTES LÓGICOS

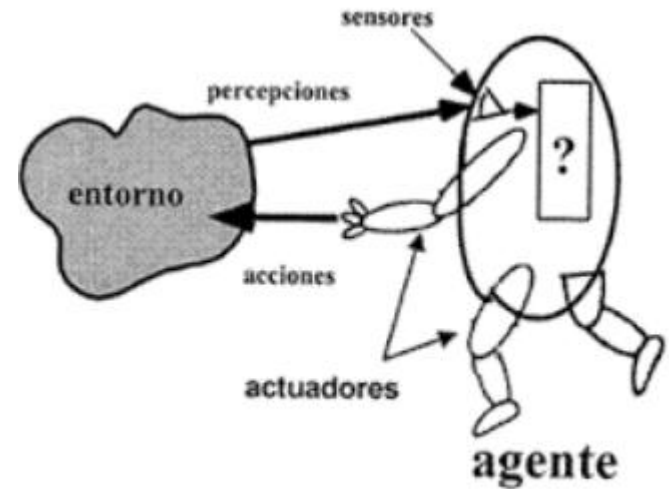
Inteligencia Artificial

!

Agentes Lógicos

Son agentes que pueden formar representaciones de un mundo complejo, utilizar un proceso de inferencia para derivar nuevas representaciones sobre el mundo y usar estas nuevas representaciones para deducir qué hacer.

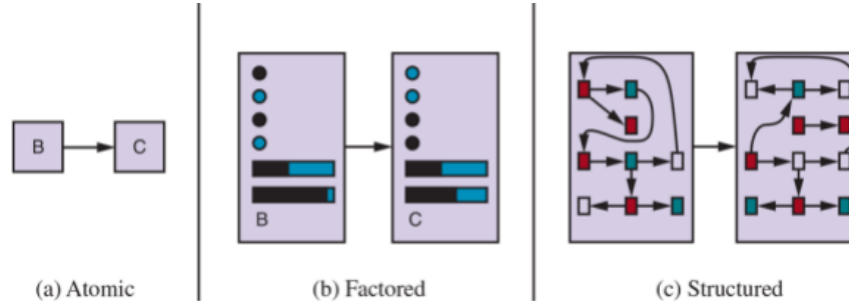
En inteligencia artificial, los agentes basados en conocimiento utilizan un proceso de razonamiento sobre una representación interna del conocimiento para decidir qué acciones tomar.



Agentes Lógicos

Los agentes de resolución de problemas son muy limitados al igual que las representaciones atómicas que utilizan.

En un entorno parcialmente observable, por ejemplo, la única opción que tiene un agente de resolución de problemas para representar lo que sabe sobre el estado actual es enumerar todos los estados concretos posibles.



Para los agentes lógicos desarrollamos la lógica sobre una clase general de representaciones para apoyar a los agentes basados en conocimiento.

Agentes Lógicos

Estos agentes pueden combinar y recombinar información para adaptarse a una multitud de propósitos. Esto puede estar muy alejado de las necesidades del momento, como cuando un matemático demuestra un teorema.

Los agentes basados en conocimiento pueden aceptar nuevas tareas en forma de objetivos explícitamente descritos; pueden alcanzar competencia rápidamente al ser informados o al aprender nuevos conocimientos sobre el entorno; y pueden adaptarse a cambios en el entorno actualizando el conocimiento relevante.



AGENTES LÓGICOS- Agentes basados en conocimiento

Agentes Lógicos-Agentes basados en conocimiento

El componente central de un agente basado en conocimiento es su base de conocimiento, o KB (por sus siglas en inglés, Knowledge Base).

Una base de conocimiento es un conjunto de oraciones (No definidas en el sentido idiomático).

Cada oración se expresa en un lenguaje llamado **lenguaje de representación del conocimiento** y representa alguna afirmación sobre el mundo. Cuando la oración se toma como dada sin derivarse de otras oraciones, la llamamos un **axioma**.

Agentes Lógicos-Agentes basados en conocimiento

Debe haber una manera de **agregar nuevas oraciones** a la base de conocimiento y una manera de **consultar lo que se sabe**. Los nombres estándar para estas operaciones son **TELL** y **ASK**, respectivamente.

Ambas operaciones pueden involucrar **inferencia**, es decir, derivar nuevas oraciones a partir de las antiguas. La inferencia debe cumplir con el requisito de que cuando se realiza una consulta (**ASK**) a la base de conocimiento, la respuesta debe derivarse de lo que se ha dicho (**o TELLed**) a la base de conocimiento previamente.

Agentes Lógicos-Agentes basados en conocimiento

El esquema general de un programa de agente basado en conocimiento, como todos nuestros agentes, toma una percepción como entrada y devuelve una acción. El agente mantiene una base de conocimiento, que puede contener inicialmente algún conocimiento de fondo.

```
function KB-AGENT(percept) returns an action  
  persistent: KB, a knowledge base  
             t, a counter, initially 0, indicating time  
  
  TELL(KB, MAKE-PERCEPT-SENTENCE(percept, t))  
  action  $\leftarrow$  ASK(KB, MAKE-ACTION-QUERY(t))  
  TELL(KB, MAKE-ACTION-SENTENCE(action, t))  
  t  $\leftarrow$  t + 1  
  return action
```


Agentes Lógicos-Agentes basados en conocimiento

MAKE-PERCEPT-SENTENCE: construye una oración que afirma que el agente percibió el percepto dado en el momento dado.

MAKE-ACTION-QUERY: Se pregunta qué acción se debe realizar en el momento actual.

MAKE-ACTION-SENTENCE: construye una oración que afirma que la acción elegida fue ejecutada.

Agentes Lógicos-Agentes basados en conocimiento

Un agente basado en conocimiento se puede construir simplemente informándole (TELL) lo que necesita saber. Comenzando con una base de conocimiento vacía, el diseñador del agente puede agregar oraciones una por una hasta que el agente sepa cómo operar en su entorno.

Esto se llama el **enfoque declarativo** para la construcción de sistemas. En contraste, el **enfoque procedimental** codifica los comportamientos deseados directamente como código de programa.

Un agente exitoso a menudo combina elementos tanto declarativos como procedimentales en su diseño



AGENTES LÓGICOS- Especificación del entorno

Agentes Lógicos-Agentes basados en conocimiento

Para los agentes basados en conocimiento, normalmente se debe especificar la medida de desempeño, el entorno, los actuadores y sensores del agente. Normalmente, esto recibe el nombre de **PEAS** (Performance, Environment, Actuators, Sensors).

Al diseñar un agente, el primer paso siempre debe ser especificar el entorno de tarea tan completamente como sea posible.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

Agentes Lógicos-Agentes basados en conocimiento

Performance: ¿cuál es la medida de desempeño a la que nos gustaría que nuestro conductor automatizado aspirara? Ejm : llegar al destino correcto; minimizar el consumo de combustible y el desgaste del vehículo; minimizar el tiempo o costo del viaje.

Environment: ¿cuál es el entorno de conducción al que se enfrentará el agente? Ejm: variedad de carreteras, que van desde caminos rurales y callejones, peatones, semáforos, otros vehículos, obras en via, huecos, etc.

Actuators: Acciones disponibles para ser ejecutadas por el agente que impacten el entorno
Ejm: control sobre el motor a través del acelerador y control sobre la dirección y el frenado.

Sensors: ¿Cuáles son las capacidades sensoriales del agente? Ejm: cámaras de video para que pueda ver, así como sensores lidar y de ultrasonido para detectar distancias con otros vehículos y obstáculos

Agentes Lógicos-Agentes basados en conocimiento

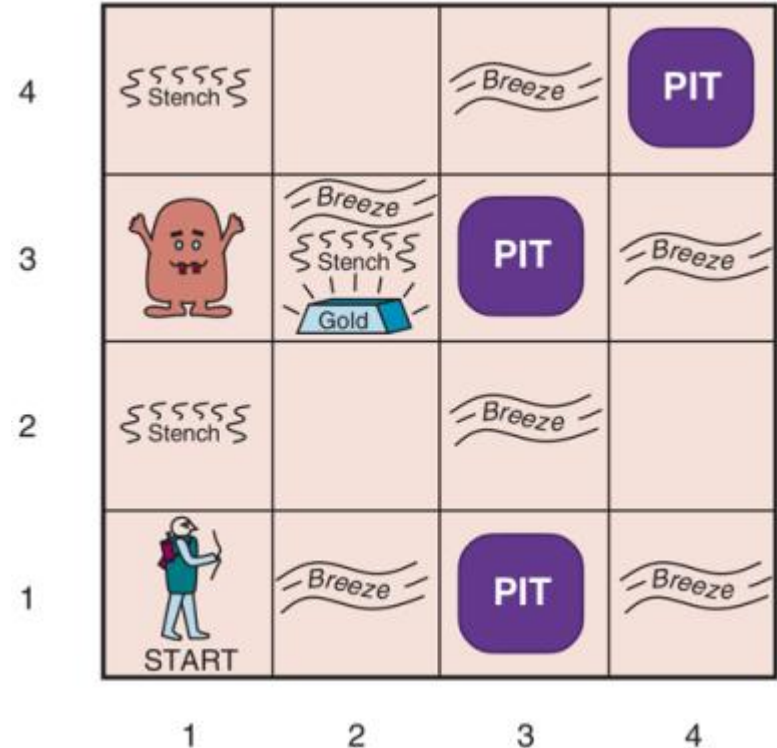
Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings
Satellite image analysis system	Correct categorization of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, tactile and joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, feedback, speech	Keyboard entry, voice



AGENTES LÓGICOS- El mundo de Wumpus (Introducción)

AGENTES LÓGICOS- El mundo de Wumpus (Introducción)

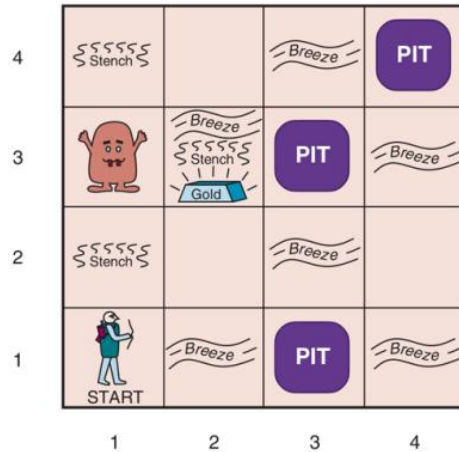
El mundo de Wumpus es uno de los ejemplos mas usados para el entendimiento de agentes lógicos. Se trata de una cueva compuesta por habitaciones conectadas por pasadizos. Acechando en algún lugar de la cueva se encuentra el terrible Wumpus, una bestia que devora a cualquiera que entre en su habitación. El Wumpus puede ser abatido por un agente, pero el agente solo tiene una flecha. Algunas habitaciones contienen pozos sin fondo que atraparán a cualquiera que entre en ellas (excepto al Wumpus, que es demasiado grande para caer en ellos). La única característica redentora de este sombrío entorno es la posibilidad de encontrar un montón de oro.



AGENTES LÓGICOS- El mundo de Wumpus (Introducción)

MEDIDA DE DESEMPEÑO: +1000 por salir de la cueva con el oro, -1000 por caer en un pozo o ser devorado por el Wumpus, -1 por cada acción tomada, y -10 por usar la flecha.

ENTORNO: Una cuadrícula de habitaciones, con paredes que rodean la cuadrícula. El agente siempre comienza en el cuadrado etiquetado como [1,1], mirando hacia el este. Las ubicaciones del oro y del Wumpus se eligen aleatoriamente, con una distribución uniforme, de entre los cuadrados distintos al cuadrado de inicio. Además, cada cuadrado, aparte del inicio, puede contener un pozo, con una probabilidad de 0.2.



El juego termina cuando el agente muere o cuando el agente sale de la cueva.

AGENTES LÓGICOS- El mundo de Wumpus (Introducción)

ACTUADORES: El agente puede avanzar (Forward), girar a la izquierda (TurnLeft) en 90 grados, o girar a la derecha (TurnRight) en 90 grados. El agente muere si entra en un cuadrado que contiene un pozo o un Wumpus vivo (es seguro entrar en un cuadrado con un Wumpus muerto).

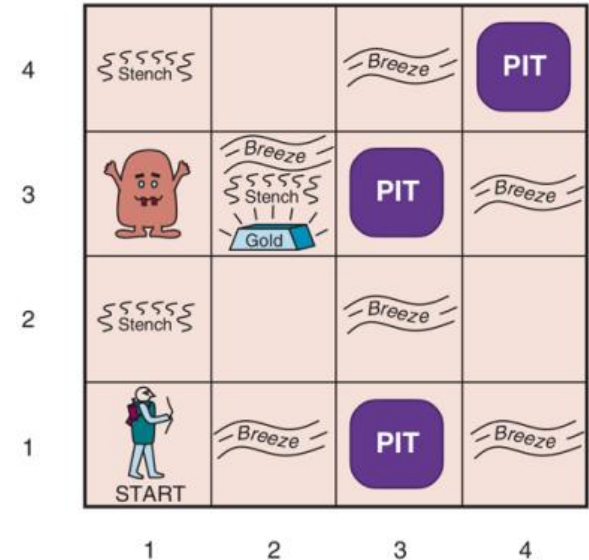
Si un agente intenta avanzar y choca con una pared, entonces el agente no se mueve.

La acción "Agarrar" (Grab) se puede usar para recoger el oro si está en el mismo cuadrado que el agente.

La acción "Disparar" (Shoot) se puede usar para disparar una flecha en línea recta en la dirección en la que mira el agente. La flecha continúa hasta que golpea (y por lo tanto mata) al Wumpus o golpea una pared.

El agente solo tiene una flecha, por lo que solo la primera acción de disparo tiene algún efecto.

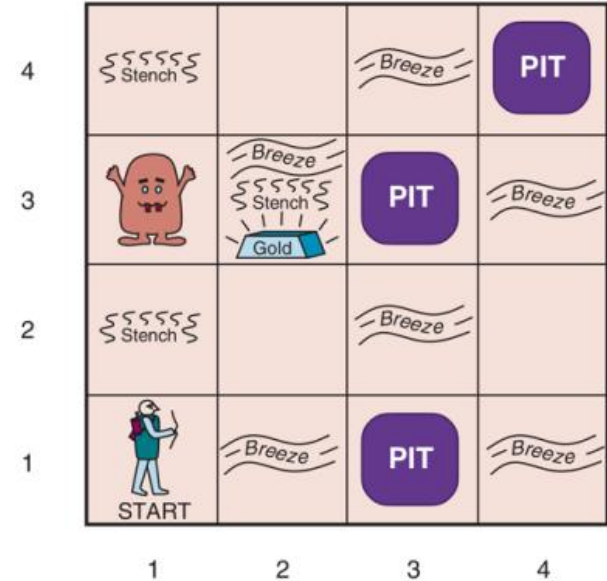
Finalmente, la acción "Escalar" (Climb) se puede usar para salir de la cueva, pero solo desde el cuadrado [1,1].



AGENTES LÓGICOS- El mundo de Wumpus (Introducción)

SENSORES: El agente tiene cinco sensores, cada uno de los cuales proporciona un solo bit de información:

- En los cuadrados directamente adyacentes (no en diagonal) al Wumpus, el agente percibirá un "Hedor" (Stench).
- En los cuadrados directamente adyacentes a un pozo, el agente percibirá una "Brisa" (Breeze).
- En el cuadrado donde está el oro, el agente percibirá un "Brillo" (Glitter).
- Cuando un agente choca con una pared, percibirá un "Golpe" (Bump).
- Cuando el Wumpus es abatido, emite un "Grito" (Scream) que se puede percibir en cualquier lugar de la cueva.



Las percepciones se entregarán al programa del agente en forma de una lista de **cinco símbolos**; por ejemplo, si hay un hedor y una brisa, pero no hay brillo, golpe ni grito, el programa del agente recibirá [**Stench, Breeze, None, None, None**].

AGENTES LÓGICOS- El mundo de Wumpus (Introducción)

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
OK			
1,1	2,1	3,1	4,1
A OK	OK		


A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2 P?	3,2	4,2
OK			
1,1	2,1 A B OK	3,1 P?	4,1
V OK	OK		

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2	3,2	4,2
OK	OK		
1,1	2,1 B V OK	3,1 P!	4,1
V OK	OK		

A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
OK	OK		
1,1	2,1 B V OK	3,1 P!	4,1
V OK	OK		



AGENTES LÓGICOS- Lógica y representación del conocimiento

Lógica-Oraciones (Syntaxis)

Las bases de conocimiento consisten en oraciones. Estas oraciones se expresan de acuerdo con la syntaxis del lenguaje de representación, que especifica todas las oraciones que están bien formadas. La noción de syntaxis es bastante clara en la aritmética ordinaria:

$x + y = 4$ es una oración bien formada,
mientras que:
 $x4+y=$ no lo es.

Lógica-Oraciones (Semántica)

La lógica también debe definir la **semántica** o el significado de las oraciones. La semántica define la verdad de cada oración con respecto a cada posible mundo.

Por ejemplo, la semántica para la aritmética especifica que la oración " $x + y = 4$ " es verdadera en un mundo donde $x=2$ y $y=2$, pero falsa en un mundo donde $x=2$ y $y=1$. En los estándares de la lógica, cada oración debe ser verdadera o falsa en cada posible mundo: no hay términos intermedios.

Lógica-Oraciones (Semántica)

Los modelos son abstracciones matemáticas, cada una de las cuales tiene un valor de verdad fijo (verdadero o falso) para cada oración relevante.

Ejemplo:

Tenemos x hombres y y mujeres sentados en una mesa jugando al bridge,

la oración $x+y=4$

Cuando es verdadera?

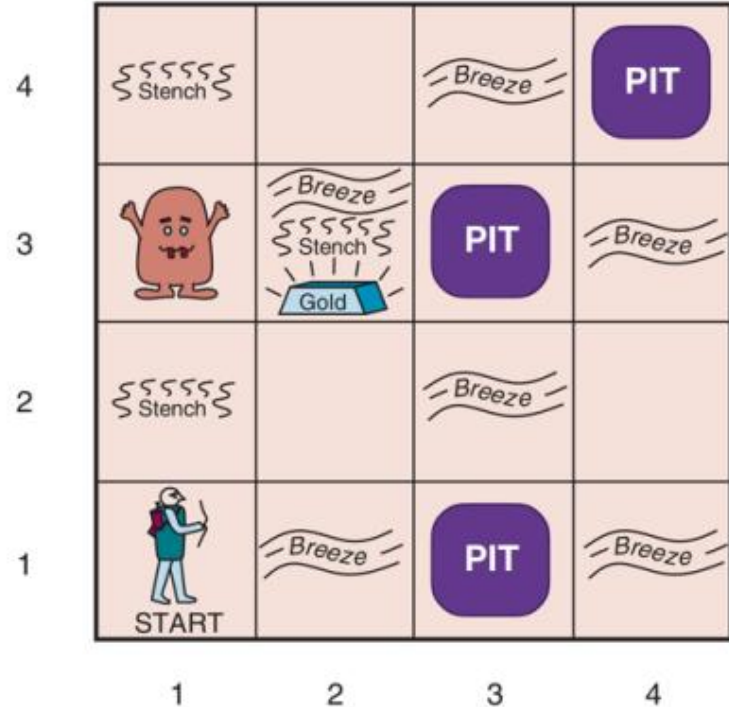
Lógica-Oraciones (Semántica)

COMO RESUMEN:

Un modelo es un conjunto de asignaciones que indican si una oración es verdadera o falsa en un "mundo posible". Por ejemplo, si tienes una oración que dice "Hay un pozo en la casilla [2,2]", el modelo sería la asignación que dice si esta oración es verdadera o falsa en ese mundo.

Lógica-Oraciones (Modelos)

- Si una oración es verdadera en un modelo dado, decimos que el modelo **satisface** esa oración. Por ejemplo, si el modelo en el mundo del Wumpus asigna verdadero a la oración "Hay un Wumpus en la casilla [1,3]", entonces ese modelo **satisface** esa oración.
- $M(\alpha)$ es la notación utilizada para referirse al conjunto de todos los modelos que satisfacen la oración α



Lógica-Oraciones (Modelos)

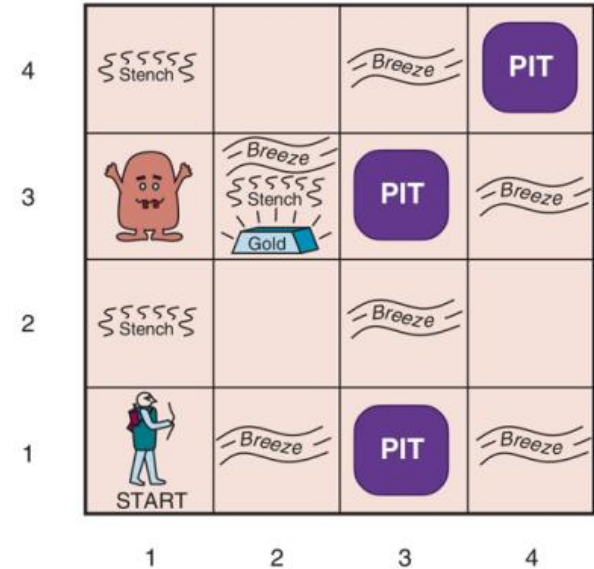
Supongamos que tenemos la siguiente información:

- Hay una brisa en la casilla [2,1].

Un modelo en este contexto podría ser la asignación de valores de verdad a las siguientes oraciones:

- "Hay un pozo en la casilla [2,2]": **Verdadero.**
- "Hay un pozo en la casilla [3,1]": **Falso.**
- "Hay un Wumpus en la casilla [1,3]": **Verdadero.**

Este modelo **satisface** la oración "Hay una brisa en la casilla [2,1]", ya que de acuerdo con las reglas del mundo del Wumpus, la brisa implica la presencia de un pozo en una de las casillas adyacentes.




Lógica-Oraciones (Modelos)

En el contexto de la lógica y la teoría de modelos, un modelo en sí no es "erróneo" o "correcto" en el sentido convencional, ya que simplemente es una asignación de valores de verdad a las oraciones dentro de un sistema formal. Sin embargo, un modelo puede no corresponder con la **realidad** del entorno que se está modelando, lo que podría llevar a conclusiones equivocadas si se usa como base para tomar decisiones.

Lógica-Oraciones (Modelos)

Un agente lógico intentará reducir el número de modelos posibles a medida que obtiene más información del entorno. Cuanta más información perciba, más restricciones podrá aplicar a los modelos, descartando aquellos que no se ajustan a las nuevas percepciones. Idealmente, el agente reduce los posibles modelos hasta que solo quede uno, que corresponde con la realidad del entorno.



LÓGICA- Razonamiento lógico

Lógica-Razonamiento Lógico

Implicación lógica entre oraciones: la idea de que una oración se sigue lógicamente de otra oración. En notación matemática, escribimos:

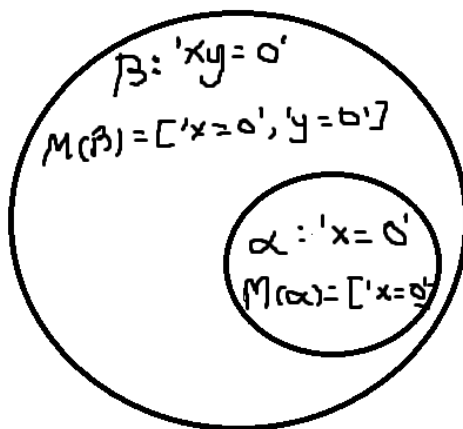
$$\alpha \models \beta$$

La definición formal de implicación es esta: $\alpha \models \beta$ si y solo si, en todo modelo en el cual α es verdadera, β también es verdadera. Usando la notación que acabamos de introducir, podemos escribir:

$$\alpha \models \beta \text{ si y solo si } M(\alpha) \subseteq M(\beta)$$

Lógica-Razonamiento Lógico

si $\alpha \models \beta$, entonces α es una afirmación más fuerte que β ; descarta más mundos posibles.) La relación de implicación es familiar en aritmética; estamos contentos con la idea de que la oración $x=0$ implica la oración $xy=0$. Obviamente, en cualquier modelo donde x es cero, es el caso que xy es cero (independientemente del valor de y).

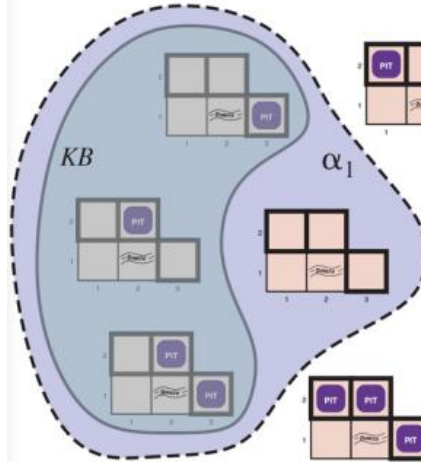


Lógica-Razonamiento Lógico

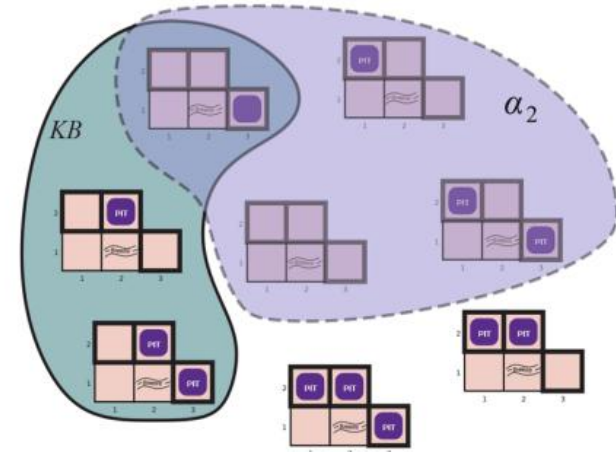
A = Agent
B = Breeze
G = Glitter, Gold
OK = Safe square
P = Pit
S = Stench
V = Visited
W = Wumpus

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2 P?	3,2	4,2
OK			
1,1	2,1 A B OK	3,1 P?	4,1

- α_1 = "No hay pozo en [1,2]."
- α_2 = "No hay pozo en [2,2]."



(a)



(b)



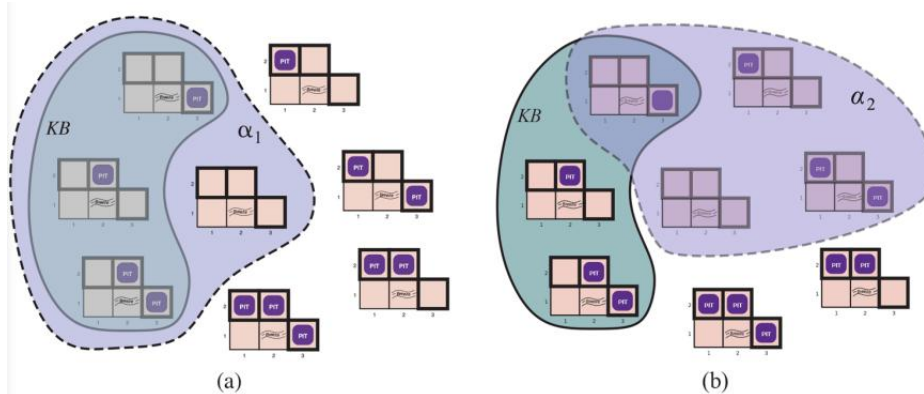
La base de conocimiento (KB) se puede pensar como un conjunto de oraciones o como una sola oración que afirma todas las oraciones individuales. La KB es falsa en los modelos que contradicen lo que el agente sabe, por ejemplo, la KB es falsa en cualquier modelo en el que $[1,2]$ contenga un pozo, porque no hay brisa en $[1,1]$. De hecho, solo hay tres modelos en los que la KB es verdadera.

Lógica-Oraciones (Modelos)

El KB es un **conjunto de oraciones** que representan lo que el agente sabe con certeza hasta el momento. Esas oraciones podrían ser algo como:

- "No hay pozo en [1,2]" es True.
- "No hay pozo en [1,1]" es True.
- "No hay pozo en [2,1]" es True.

Los **tres modelos posibles** a los que nos referimos son **diferentes "mundos posibles"** en los que el **KB es verdadero**. En otras palabras, los **modelos** son posibles configuraciones del entorno del agente que no contradicen el conocimiento que el agente tiene. Por lo tanto, a partir de KB podemos filtrar los mundos posibles



El algoritmo de inferencia ilustrado en la se llama verificación de modelos, porque enumera todos los modelos posibles para verificar que α es verdadero en todos los modelos en los que KB es verdadero, es decir, que $M(KB) \subseteq M(\alpha)$.

Lógica-Oraciones (Modelos)

Al entender la implicación y la inferencia, puede ser útil pensar en el conjunto de todas las consecuencias de **KB** como un pajar y en α como una aguja.

La implicación es como si la aguja estuviera en el pajar; la inferencia es como encontrarla.

Esta distinción se expresa en una notación formal: si un algoritmo de inferencia i puede derivar α de KB, escribimos:

$$KB \vdash_i \alpha ,$$

lo cual se pronuncia “ α se deriva de KB por i ” o “ i deriva α de KB”.

Las oraciones son configuraciones físicas del agente, y el razonamiento es un proceso de construir nuevas configuraciones físicas a partir de las anteriores. El razonamiento lógico debe asegurar que las nuevas configuraciones representen aspectos del mundo que realmente se derivan de los aspectos que representan las configuraciones anteriores.

