

# Introducción al Razonamiento Deductivo y Clingo

Jorge Baier

Departamento de Ciencia de la Computación  
Pontificia Universidad Católica de Chile

Santiago, Chile



- Discutir sobre el sistema 1 y 2 del pensamiento
- Introducir el Razonamiento Deductivo Automático
- Introducir el uso de Clingo para resolver problemas de Answer Set Programming



# Psicología del pensamiento: Sistema 1 y Sistema 2

Desde el área de la psicología/economía podemos encontrar algo de inspiración

- **Sistema 1:** Aquellas conclusiones que nuestro cerebro parece hacer en forma automática (sin esfuerzo consciente).
- **Sistema 2:** Aquellas conclusiones que nos cuesta obtener (con esfuerzo consciente).

Vemos el video “Thinking Fast, Slow – Daniel Kahneman”



# Razonamiento Deductivo (Sistema 2)

- En la primera parte del curso nos concentramos solo en algunos problemas asociados al razonamiento.
- Los seres humanos somos capaces de resolver una gran variedad de *problemas de razonamiento* que van más allá del aprendizaje.



# Razonamiento Deductivo (Sistema 2)

- En la primera parte del curso nos concentramos solo en algunos problemas asociados al razonamiento.
- Los seres humanos somos capaces de resolver una gran variedad de *problemas de razonamiento* que van más allá del aprendizaje.
- ¿Pero qué es *razonar*?

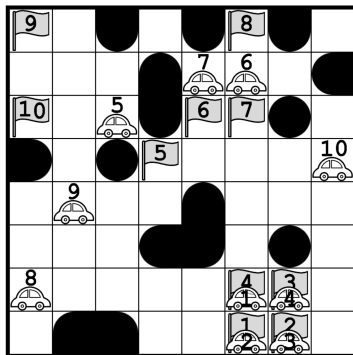


# Razonamiento Deductivo (Sistema 2)

- En la primera parte del curso nos concentramos solo en algunos problemas asociados al razonamiento.
- Los seres humanos somos capaces de resolver una gran variedad de *problemas de razonamiento* que van más allá del aprendizaje.
- ¿Pero qué es *razonar*?
- Definición RAE: *Ordenar y relacionar ideas para llegar a una conclusión*
- No daremos una definición ahora, en vez, veamos ejemplos



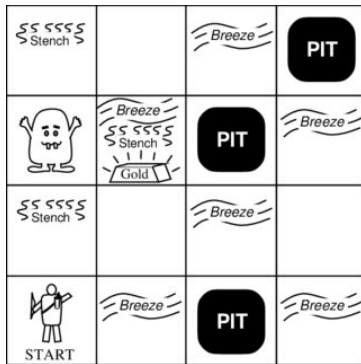
# Problemas de Razonamiento: Planificación de Rutas



*Encuentre una secuencia de movimientos que lleve desde la situación inicial a la final*



# Problemas de Razonamiento: Mundo Desconocido



*¿Cómo encontramos el oro?*





# Problemas de Razonamiento: Diagnóstico



*Juan enciende el interruptor y la linterna no produce luz. Sin cambiar la posición del interruptor, Juan cambia las baterías y ahora sí la linterna produce luz. ¿Qué puedo inferir sobre la linterna?*



- Tipo de programación lógica que usa un lenguaje declarativo para resolución de problemas de búsqueda difíciles
- Los programas de ASP se construyen a partir de reglas lógicas
- Las reglas expresan deducciones o restricciones que se deben cumplir sobre lo deducido por el programa
- Aquello que se deduce de un programa es un **modelo**
- A los modelos también se les llama *answer sets*.



## Definición

Una *regla* en programación en lógica es un objeto de la forma:

$$Head \leftarrow Body,$$

donde *Head* y *Body* son conjuntos de *átomos*. Esta regla indica que si *Body* se encuentra en el modelo, entonces *Head* también se encuentra en él.

Ejemplo:  $\{u\} \leftarrow \{t, r\}$

En el sistema *clingo* esta regla se anota así:

$u :- t, r.$

Y significa que si *t* y *r* se encuentran en el modelo, entonces *u* también



Sintácticamente un átomo es de la forma:

- $p$ , donde  $p$  es un string que comienza con minúscula.
- $p(c_1, c_2, \dots, c_n)$ , donde  $p$  es un string que comienza con minúscula y que se llama *predicado* y  $c_1, c_2, \dots, c_n$  son *constantes*. Cada constante es un string que comienza con minúscula.



## Definición

Un *hecho* en programación en lógica es una regla cuyo *Tail* es vacío.

En clingo se ven de la forma:

q.



# Qué es un programa básico

## Definición

Un *programa* es un conjunto de reglas básicas.



Informalmente, un *modelo* contiene lo que se “deduce” de un programa.

Para construir un modelo  $M$  de un programa básico  $\Pi$ :

- 1 Agregamos todos los hechos del programa  $\Pi$  a  $M$ .
- 2 Si hay una regla  $Head \leftarrow Tail$  en  $\Pi$  que cumple:
  - (a)  $Head$  no está contenido en  $M$
  - (b) Todos los elementos de  $Tail$  están en  $M$entonces agregamos  $Head$  a  $M$ .
- 3 Si el paso 2 agrega algo a  $M$  volvemos al paso 2. En caso contrario, terminar.



Un ejemplo de lo anterior es

```
gato(tom).  
animal(tom) :- gato(tom).
```

La segunda linea significa que si tom es gato, entonces tom es un animal. A partir de esto, el modelo resultante es  
{gato(tom), animal(tom)}





# Ejemplo

Tomando el ejemplo anterior, si consideramos que hay otro gato, garfield, y queremos decir que el hecho de que garfield sea gato, implica que garfield también es animal, escribiremos:

```
gato(tom).  
gato(garfield).  
animal(tom) :- gato(tom).  
animal(garfield) :- gato(garfield).
```

Para no escribir una regla para cada hecho podemos usar **variables**.



## Variables

Una *variable* es una forma de generar una familia de reglas.

En Clingo, para las variables comienzan con mayúscula.

De esta manera, podemos generar los mismos modelos de este ejemplo:

```
gato(tom).  
gato(garfield).  
animal(tom) :- gato(tom).  
animal(garfield) :- gato(garfield).
```

con el siguiente programa:

```
gato(tom).  
gato(garfield).  
animal(X) :- gato(X).
```

Donde la tercera línea indica que todo profesor forma parte del curso.



- Introducir el Razonamiento Deductivo Automático
- Introducir el uso de Clingo para resolver problemas de Answer Set Programming

