



## Tarea 1

### Aspectos generales

#### Formato y plazo de entrega

El formato de entrega son archivos con extensión .lp con un PDF para las respuestas teóricas. El lugar de entrega es en el repositorio de la tarea, en la *branch* por defecto, hasta el **martes 18 de abril a las 23:59 hrs.** Para crear tu repositorio, debes entrar en el enlace del anuncio de la tarea en Canvas.

#### Integridad Académica

Este curso se adhiere al Código de Honor establecido por la universidad, el cual tienes el deber de conocer como estudiante. Todo el trabajo hecho en esta tarea debe ser **totalmente individual**. La idea es que te des el tiempo de aprender estos conceptos fundamentales, tanto para el curso, como para tu formación profesional. Las dudas se deben hacer exclusivamente al cuerpo docente a través de las *issues* en GitHub.

Por otra parte, sabemos que estás utilizando material hecho por otras personas, por lo que es importante reconocerlo de la forma apropiada. Todo lo que obtengas de internet debes citarlo de forma correcta (ya sea en APA, ICONTEC o IEEE). Cualquier falta a la ética y/o a la integridad académica será sancionada con la reprobación del curso y los antecedentes serán entregados a la Dirección de Pregrado.

#### Comentarios adicionales

El objetivo de esta tarea es que puedan desarrollar la capacidad de modelar problemas a partir del lenguaje natural y resolverlos utilizando ASP en Clingo. Es fundamental que pongan énfasis en las justificaciones de sus respuestas, cuidando la redacción, ortografía; manteniendo el código ordenado y comentado. Aquellas respuestas que solo presenten resultados o código (sin contexto ni comentarios) no serán consideradas, mientras que tareas desordenadas pueden ser objeto de descuentos.

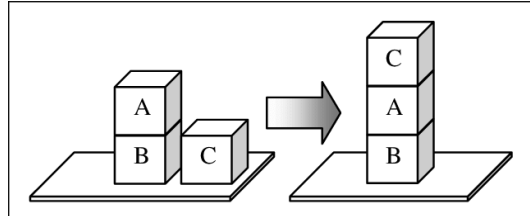
## 1. Reflexión y Teoría (2 pts.)

1. Existen formas de argumentar que el podcast de Lex Fridman en YouTube es imperdible para todo quien se dedique a la inteligencia artificial. El recién pasado sábado 25, Lex publicó una entrevista a Sam Altman, CEO de OpenAI, quien desarrolla chatGPT. Para contestar esta pregunta, primero ve la sección AGI del video [https://www.youtube.com/watch?v=L\\_Guz73e6fw](https://www.youtube.com/watch?v=L_Guz73e6fw) (47:33–1:09:05).
  - a) En la entrevista, Altman se refiere a que chatGPT, su forma actual, no será capaz de hacer descubrimientos científicos. Fridman está en desacuerdo. Argumenta en qué sentido ambos pueden tener razón. Da ejemplos y analiza cómo es que se podrían superar las actuales limitaciones para lograr la AGI, en tal como lo define Altman.
  - b) Describe el argumento de Sutskever en la versión de Altman y discútelo. En tu discusión entrega tu propia definición de conciencia e intenta argumentar por qué un modelo de lenguaje podría o no podría tener esa capacidad.

Si no has formado tu propia opinion sobre el concepto de conciencia, ve al menos los primeros 19 minutos de este video. También puedes consultar otras opiniones como ésta. Para explorar otras concepciones, puramente experienciales/místicas del concepto de conciencia, podrías también revisar este otro video (aunque recomiendo verlo completo, puedes saltarte el rango 1:06-7:24).
2. En nuestras clases de definiciones, definimos exactamente cómo calcular modelos (o *answer sets*) de programas disyuntivos con negación, pero no dimos una definición para las *restricciones de cardinalidad*. Propón una definición. Ilustra su funcionamiento usando al menos 3 programas distintos.

## 2. DCCubos (2 pts.)

El problema de *Mundo de Bloques* es un clásico dominio ilustrativo de la inteligencia artificial. Consiste en que hay bloques sobre una mesa, que pueden estar apilados sobre otros o directamente sobre la mesa. Existe un brazo robótico sobre la mesa que mueve un bloque a la vez. El robot puede mover un solo bloque a la vez y el bloque que mueva no debe tener ningún otro bloque sobre él. La misión es mover los bloques que se encuentran en una determinada configuración inicial hasta que terminan en una configuración objetivo. Un ejemplo de este problema se puede ver en la imagen de abajo.



En el archivo `DCCubos.lp` encontrarás la representación de un dominio simple de bloques usando *answer set programming*. En esta parte de la tarea, se pide que modifiques menormente esa implementación para que permita que los bloques puedan ser manipulados por múltiples manos robóticas al mismo tiempo y que evalúes experimentalmente el impacto que esto tiene en el tiempo de computación y en la calidad de solución.

Para usar el programa debes correr el comando `DCCBloques.lp -c bound=N` donde `N` representa a la cantidad de unidades de tiempo en los que se puede resolver el problema. Este valor es un número que se puede modificar para probar el programa.

Más precisamente, tu informe deberá contener:

1. Exactamente cómo debe ser modificado el archivo `DCCubos.lp` para que admita el uso de  $n$  brazos robóticos. Esto significa que en el mismo instante de tiempo, hasta  $n$  bloques podrían ser movidos a otro lugar. Tu implementación debe respetar aspectos de la física de este problema. Tampoco debes considerar posible que, al mismo tiempo, se ubique o se retire más de un bloque sobre una misma torre de bloques.
2. Una evaluación experimental en la cual evalúes la solución propuesta en al menos 20 problemas de diversa dificultad (con al menos 40 bloques), que usted debe diseñar<sup>1</sup>. Cada problema deberá ser ejecutado con 1, 3 y 5 brazos robóticos. Deberás reportar, en un gráfico, el tiempo de solución obtenido por cada variante en cada problema, y, en otro gráfico, el instante de tiempo en el que se logra el objetivo, para cada variante y en cada problema.
3. Un párrafo de conclusiones que analice los resultados experimentales. Para obtener buenas conclusiones, use problemas en los que el estado inicial tenga entre 1 y 5 torres y donde el estado final también tenga entre 1 y 5 torres.
4. En el libro ([click acá](#)) se explica el keyword `minimize` de Clingo. Leyendo el libro, asegúrate que entiendes la diferencia entre estas dos sentencias:

```
#minimize {X : p(X)}  
#minimize {1,X : p(X)}
```

---

<sup>1</sup>Se recomienda que usted intercambie problemas con otros alumnos.

Ahora muestra cómo generar una versión de tu programa que produzca un plan con mínimo número de acciones. Evalúa experimentalmente los tiempos de ejecución en un par de las instancias que generaste arriba.

### 3. DCCConnect de Pou (2pts.)

En esta parte deberás construir un programa en lógica para resolver el puzzle *Connect*, disponible en el juego Pou<sup>2</sup>. Pou es un juego orientado a niños, disponible para dispositivos móviles, y, por lo tanto, se debiera esperar que este puzzle sea “sencillo” para cualquier persona. Se recomienda verifiques esto personalmente.

El puzzle se juega sobre una grilla de  $m \times n$  en la cual inicialmente hay pares de celdas coloreados, cada par de un color distinto. El objetivo es encontrar un camino entre cada par de celdas del mismo color de forma tal que los caminos no se crucen (compartan celdas). Además, **se requiere que los caminos usen todas las celdas**.



(a) Grilla al comienzo del juego.



(b) Una (casi) solución.

#### 3.1. Representaciones

Esta parte de la tarea se corregirá automáticamente, por lo cual deberás suponer que el problema se define de la siguiente manera:

---

<sup>2</sup><https://www.pou.me/>

### 3.1.1. Instancia

Una instancia de  $Y_{max} \cdot X_{max}$  se describirá de la siguiente forma:

$$\begin{aligned} &fila(1..Y_{max}) \\ &columna(1..X_{max}) \\ \\ &color(c_1, y_{origen_1}, x_{origen_1}) \\ &color(c_1, y_{destino_1}, x_{destino_1}) \\ &\vdots \\ &color(c_k, y_{origen_k}, x_{origen_k}) \\ &color(c_k, y_{destino_k}, x_{destino_k}) \end{aligned}$$

Supón que habrá un par de celdas asignadas para cada color  $c_i \in \{1...k\}$  y que el *input* es válido.

### 3.1.2. Resultado

El predicado

$$camino(C, Y_{origen}, X_{origen}, Y_{destino}, X_{destino})$$

deberá ser parte de un *answer set* si y solo si en su solución hay que conectar  $(Y_{origen}, X_{origen})$  con su celda vecina  $(Y_{destino}, X_{destino})$  usando el color  $C$ .

El orden de las celdas será irrelevante, ya que no es un camino dirigido. Bastará representar “en un sólo sentido”, pero usar ambos no está mal.

**Puedes usar todos los predicados auxiliares que quieras.**

### 3.1.3. Visualizador

Para facilitar la visualización de sus resultados en esta pregunta entregaremos el código de un visualizador, capaz de mostrar las soluciones obtenidas por clingo. Para hacer uso del visualizador, haz lo siguiente:

- Una vez tengas un programa capaz de solucionar el problema, digamos `solucion.lp`, debes mover dicho programa dentro de la carpeta `Visualizador`.
- Una vez dentro de la carpeta `Visualizador`, ejecuta el comando `'clingo solucion.lp | python parser.py output.json'` (sin comillas). Esto generará un archivo llamado `output.json` dentro de la carpeta.
- Abrir el archivo `index.html` y seleccionar el archivo `output.json` dentro de la página que se muestra. Esto mostrará una visualización de la solución encontrada.

**IMPORTANTE:** Tu programa será corregido por otro programa. Por lo tanto, es importante que utilices los mismos predicados sugeridos en este enunciado. Tu programa **no deberá** incluir ninguna regla que defina un puzle específico. Es decir, no debe incluir hechos que describan fila, columna y color.