

## Tarea 1

Fecha de entrega: Martes 18 de Abril 2023

ALUMNOS: BEATRIZ ERRÁZURIZ CAMUS NRO.19638906

### Parte 1

#### 1.

##### a)

Cuando Altman afirma que ChatGPT no es una AGI debido a que no es capaz de realizar descubrimientos científicos significativos, está en lo correcto. Las inteligencias artificiales actuales, incluyendo a ChatGPT, aún no pueden igualar el nivel de creatividad, intuición y juicio propio humano, por lo que los avances científicos más significativos seguirán siendo realizados por seres humanos.

Por otro lado, Friedman menciona que se pueden lograr grandes descubrimientos científicos con los datos en los que ha sido entrenado GPT. Si bien es cierto lo que mencionaba Altman, actualmente se están realizando grandes descubrimientos con la ayuda de inteligencias artificiales, como la identificación de nuevos fármacos o la modelización y predicción de comportamientos en sistemas complejos de la astrofísica. Se podría decir que ChatGPT podría aplicar el concepto de *zona de desarrollo próximo*, que se define como la distancia entre el nivel de desarrollo efectivo y el potencial, lo que sugiere que al trabajar junto a un ser humano, una máquina podría realizar nuevos descubrimientos importantes para el área científica.

Altman menciona que una de las limitaciones para la AGI es la necesidad de expandir los paradigmas de GPT, es decir, el enfoque de aprendizaje profundo utilizado en el modelo GPT. Él menciona que aún faltan muchas ideas para lograr esto de manera efectiva. Es difícil dar ejemplos concretos para superar las limitaciones mencionadas por el CEO de OpenAI porque, como él mismo afirma, las ideas aún son desconocidas. Sin embargo, una posible solución sería entrenar modelos generales de aprendizaje profundo pre-entrenados en lugar de modelos especializados para tareas específicas. De esta manera, se podría entrenar una red neuronal profunda que aprenda de cualquier tipo de datos y tenga un conocimiento mucho más amplio, lo que le permitiría realizar cualquier tarea intelectual humana, ya que conocerá tanto como nosotros.

##### b)

Para mí, la conciencia es la capacidad de tener conocimiento de uno mismo y de tener una percepción de las funciones y procesos propios. La conciencia se relaciona con una experiencia subjetiva, como las autopercepciones, las emociones, la comprensión del mundo y de la interacción con él, los pensamientos individuales y las experiencias personales.

Altman opina que GPT4 no es consciente, ya que según Sutskever (cofundador de OpenAI), si se entrena un modelo sin mencionar la conciencia ni nada relacionado con ella durante el proceso de entrenamiento, y se le pregunta acerca de cualquier cosa que no haya sido entrenada, responderá que no lo sabe. Sin embargo, si se le describe una experiencia subjetiva de lo que es ser consciente y él dice "sé exactamente de lo que me hablas", entonces podríamos considerar que es consciente, lo cual actualmente no sucede con GPT4.

Se entiende lo que dice Altman, pero por otro lado, Friedman también toca un punto importante. Se ha hablado mucho de la teoría de la simulación, según la cual los seres humanos no podríamos reconocer si somos parte de ella o no. Bajo esta teoría, ¿cómo podríamos asegurarnos de que una inteligencia artificial es realmente consciente y no está fingiendo?

Desde mi perspectiva, será casi imposible verificar si una inteligencia artificial es o no consciente, ya que esto está muy relacionado con lo propio de cada individuo, con opiniones subjetivas y con comportamientos y sensaciones internas. Por lo tanto, realmente solo uno mismo podría autoconvencerse de ser consciente, pero sería muy difícil convencer a alguien más. En conclusión, no estoy segura de si es posible generar conciencia o no, ya que su comprobación dificultaría la búsqueda real de alcanzarla y verificarla.

## 2.

Las restricciones de cardinalidad son restricciones lógicas que se utilizan para limitar el número de átomos que satisfacen cierta condición de un *answer set*. Para ilustrar esta definición, se mostrarán tres programas en modo de ejemplo:

### Programa 1:

Asignación de tareas entre un personal. En este programa se sabe quienes son parte del personal, que tareas existen y que personal puede realizar que tarea. Definiremos una regla a partir de restricciones de cardinalidad para asegurar que cada tarea deber ser realizada por una persona y que una persona puede realizar una tarea en un determinado tiempo (asumiendo que todas las tareas toman lo mismo)

```
% Definimos el personal
```

```
personal(Maria;Lupe;Miguel).
```

```
% Definimos una lista de tareas que deben realizarse
```

```
tareas(T1;T2;T3;T4;T5;T6).
```

```
% Definimos que persona puede realizar que tarea
```

```
puede_realizar(Maria, T1).
puede_realizar(Maria, T6).
puede_realizar(Lupe, T1).
puede_realizar(Lupe, T2).
puede_realizar(Lupe, T5).
puede_realizar(Miguel, T3).
puede_realizar(Miguel, T4).
puede_realizar(Miguel, T6).
```

```
% Definimos la regla que establece que cada persona solo puede realizar una tarea
% en un cierto tiempo
```

```
:- realiza(P, W1, T), realiza(P, W2, T), W1 != W2, personal(P).
```

```
% Definimos la regla que establece que cada tarea debe ser realizada por una
% persona
```

```
%% En este caso, la restriccion de cardinalidad se leeria de la siguiente forma
%% <<Cada tarea W puede realizarla solo una persona en un tiempo T
%% (si esta puede realizar W)>>
```

```
1 { realiza(P, W, T) : puede_realizar(P, W) } 1 :- tareas(W).
```

## Programa 2:

Elección de ramos. En este programa, se brinda al estudiante la opción de seleccionar los cursos que desea tomar durante el semestre. Es importante tener en cuenta que el número máximo de cursos que puede seleccionar es de 5. Además, es obligatorio seleccionar al menos 2 cursos para poder inscribirse en el semestre. No se permiten conflictos de horario entre los cursos seleccionados.

```
% Definimos la lista de ramos

ramo(iic1103; iic2343; iic2613; iic2154; iic1243).
ramo(iic3213; iic2437; iic1264; iic2432; iic1353).

% Definimos aquellos ramos con topes

topa(iic1103 , iic3213 ).
topa(iic3213 , iic1264 ).
topa(iic2154 , iic1353 ).
topa(iic1353 , iic3213 ).
topa(iic1353 , iic1264 ).

% Definimos la regla que establece que se deben elegir entre dos y cinco ramos
2 { elige(A) : ramo(A) } 5.

% Definimos la regla que establece que no se pueden elegir dos ramos con tope
:- elige(A), elige(B), topa(A, B).

% El programa muestra aquellos ramos elegidos
#show elige/1.
```

## Programa 3:

Ofrecer menús. En este programa, se brinda la opción de ofrecer menús a una lista de invitados específicos. Existen varias opciones de menú disponibles y cada una de ellas puede ser ofrecida hasta cuatro veces a los invitados.

```
% Definimos a los invitados

invitado(1..10).

% Definimos las opciones de menus

menu(carne; pollo; vegetariano).

% Se le debe ofrecer un menu a cada invitado
1{ofrecer(M,I): menu(M)}1 :- invitado(I).

% Cada menu puede ser ofrecido a maximo 4 invitados
0{ ofrecer(M,I) : invitado(I)}4 :- menu(M).

% Mostramos los menus ofrecidos
#show ofrecer/2.
```