EJERCICIO CLASE: Caballeros y Bribones

Trasfondo

En 1978, el lógico Raymond Smullyan publicó «¿Cuál es el nombre de este libro?», un libro de acertijos lógicos. Entre los acertijos del libro había una clase de acertijos que Smullyan llamó acertijos de «Caballeros y Bribónes».

En un puzzle de Caballeros y Bribónes, se da la siguiente información: Cada personaje es un caballero o un Bribón. Un caballero siempre dirá la verdad: si el caballero dice una frase, esa frase es cierta. Por el contrario, un Bribón siempre mentirá: si un Bribón dice una frase, esa frase es falsa.

El objetivo del rompecabezas es, dado un conjunto de frases pronunciadas por cada uno de los personajes, determinar, para cada personaje, si es un caballero o un bribón.

Por ejemplo, consideremos un acertijo simple con un solo personaje llamado A. A dice «Soy tanto un caballero como un bribón».

Lógicamente, podríamos razonar que si A fuera un caballero, entonces esa frase tendría que ser cierta. Pero sabemos que la frase no puede ser cierta, porque A no puede ser a la vez caballero y bribón: sabemos que cada personaje es o caballero o bribón, pero no ambas cosas. Así que podríamos concluir que A debe ser un bribón.

Este enigma era el más sencillo. Con más personajes y más frases, los acertijos pueden volverse más complicados.

Tu tarea en este problema es determinar cómo representar estos rompecabezas utilizando la lógica proposicional, de modo que una IA que ejecute un algoritmo de comprobación de modelos pueda resolver estos rompecabezas por nosotros.

Entendiendo

Echa un vistazo a **puzzle.py**. En la parte superior, hemos definido seis símbolos proposicionales. CaballeroA, por ejemplo, representa la sentencia que «A es un caballero», mientras que BribonA representa la sentencia que «A es un bribón». También hemos definido símbolos proposicionales similares para los caracteres B y C.

Lo que sigue son cuatro bases de conocimiento diferentes, knowledge0, knowledge1, knowledge2 y knowledge3, que contendrán el conocimiento necesario para deducir las soluciones a los próximos acertijos 0, 1, 2 y 3, respectivamente. Observa que, por ahora, cada una de estas bases de conocimiento está vacía. ¡Ahí es donde entras tú!

La función principal de este **puzzle.py** recorre todos los puzzles, y utiliza la comprobación de modelos para calcular, dado el conocimiento para ese puzzle, si cada personaje es un caballero o un bribón, imprimiendo cualquier conclusión que el algoritmo de comprobación de modelos sea capaz de hacer.

Especificaciones

Añade conocimientos a las bases de conocimiento knowledge0, knowledge1, knowledge2, knowledge3 y knowledge4 para resolver los siguientes puzzles.

El acertijo 0 es el acertijo del Fondo. Contiene un único personaje, A.

A dice «Soy a la vez un caballero y un bribón».

El acertijo 1 tiene dos personajes: A y B.

A dice «Los dos somos bribones».

B no dice nada.

El acertijo 2 tiene dos personajes: A y B.

A dice «Somos de la misma clase».

B dice «Somos de distinta clase».

El acertijo 3 tiene tres personajes: A, B y C.

A dice «Soy un caballero» o «Soy un bribón», pero no sabes cuál.

B dice «A dijo “Soy un bribón”».

B dice entonces «C es un bribón».

C dice «A es un caballero».

El acertijo 4 tiene tres personajes: A, B y C

A dice: «O yo soy un bribón, o tanto B como C son caballeros.»

B dice: «Exactamente uno de nosotros es un bribón».

C no dice nada.

El acaertijo 5 tiene tres personajes A, B C

A dice: «Si C es un caballero, entonces B es un bribón.»

B dice: «A miente o yo soy un caballero».

C dice: «O todos nosotros somos b, o exactamente dos de nosotros somos caballeros».

En cada uno de los acertijos anteriores, cada personaje es un caballero o un bribón. Cada frase pronunciada por un caballero es verdadera, y cada frase pronunciada por un bribón es falsa.

Una vez que haya completado la base de conocimientos para un problema, debes ser capaz de ejecutar python puzzle.py para ver la solución al rompecabezas.

Sugerencias

Para cada base de conocimiento, es probable que desees codificar dos tipos diferentes de información: (1) información sobre la estructura del problema en sí (es decir, la información dada en la definición de un acertijo de Caballero y bribon), y (2) información sobre lo que los personajes realmente dijeron.

Considere lo que significa que un personaje diga una frase. ¿En qué condiciones es verdadera? ¿En qué condiciones es falsa? ¿Cómo se puede expresar como una frase lógica?

Hay varias bases de conocimiento posibles para cada acertijo que calcularán el resultado correcto. Debes intentar elegir una base de conocimiento que ofrezca la traducción más directa de la información del acertijo, en lugar de realizar un razonamiento lógico por tu cuenta. También debes considerar cuál sería la representación más concisa de la información del acertijo.

Por ejemplo, para el acertijo 0, establecer knowledge0 = CaballeroA daría como resultado una respuesta correcta, ya que a través de nuestro propio razonamiento sabemos que A debe ser un bribón. **Pero hacerlo iría en contra del espíritu de este problema: el objetivo es que tu IA razone por ti.**

No deberías necesitar (ni deberías) modificar **logic.py** en absoluto para completar este problema.