

Laboratorio Programación Lógica

Paradigmas y Lenguajes de Programación - Trelew

Lic. Pablo Toledo Margalef

Lic. Lautaro Pecile

Año 2023

Blackjack

El blackjack, también llamado veintiuna o veintiuno, es un juego de cartas, propio de los casinos, que consiste en obtener 21 puntos mediante la suma de los valores de las cartas. Las cartas numéricas suman su valor, las figuras suman 10 y el As es un 11 o un 1, según como convenga al jugador. Si se consigue 21 con sólo dos cartas, se obtiene un blackjack y se gana automáticamente. En el juego participan un conjunto de jugadores que juegan en simultaneo pero individualmente contra el crupier, quien reparte las cartas. El crupier inicia el juego repartiendo dos cartas a cada jugador y luego dos para si mismo, luego los jugadores juegan por turnos hasta plantarse, finalmente juega el crupier. Terminada la ronda los jugadores que hayan superado al crupier ganan un monto igual al de su apuesta, quien haya obtenido un blackjack recibe un pago de 3 a 2.

Entorno

- Descargar e instalar el entorno SWI-Prolog.
- Probar desde la línea de comandos `swipl -s test.pl play`, debería responder Tu comando es "play".

Objetivos

Objetivos preliminares

Dada la base de datos `cards.pl` que contiene la representación de las cartas, y sin alterar la misma, se deben implementar el siguiente conjunto de reglas para asistir al razonamiento del juego:

- `value(card(Number, Suit), Value)`: Dada una carta, indica el valor de la misma. Para las numéricas su mismo valor, para la figuras (J, Q, K) siempre 10 y para el As 11 o 1.

Una mano es representada como una lista de cartas. A partir de aquí, definimos `Hand` para hacer referencia a una mano de la forma `[card(Number, Suit)]`, donde `Number` indica el número o figura de la carta y `Suit` el palo.

- `hand(Hand, Value)`: Dada una mano, indica el valor o valores de la mano.
- `twentyone(Hand)`: Indica si la mano suma exactamente 21.
- `over(Hand)`: Indica si la mano se pasó de los 21.
- `blackjack(Hand)`: Indica si la mano es un blackjack. O sea, suma 21 con sólo dos cartas.

Objetivos intermedios

En función de las reglas básicas del juego, es necesario darle entidad al crupier. El mismo tiene una política muy estricta de juego que respeta las siguientes reglas:

- Con una mano de 16 o menos, juega.
- Con una mano de 17 o más, se planta.

Existen dos posibles variantes a estas reglas cuando el crupier obtiene un 17 soft, esto es, su mano suma 17 con un As otorgándole el valor de 11. Se deben implementar entonces dos reglas:

- `soft_dealer(Hand)`: Se planta con una mano de 17 suave. Esto es, el As valiendo 11.
- `hard_dealer(Hand)`: Sigue jugando con una mano de 17 suave.

Donde `Hand` es la mano del crupier.

Objetivo principal

Implementar una regla `play` (`Hand`, `Crupier`, `Cards`) que reciba tres listas de cartas. `Hand` y `Crupier` representan las manos del jugador y crupier respectivamente, `Cards` representa la lista de cartas que ya fueron puestas en lo que va de la ronda, tanto para todos los jugadores como para el crupier. Por simplicidad, se establece de base que el juego se realizará con un único mazo de cartas a diferencia de lo que se acostumbra en los casinos al utilizar múltiples barajas.

Esta regla debe realizar una deducción lógica indicando verdadero en el caso de seguir jugando y pedir más cartas o falso si decide plantarse. Es requisito primordial que el razonamiento se realice utilizando el total de parámetros indicados (Mano del jugador, mano del crupier y cartas que salieron en todo el juego) y cada uno de ellos debe hacer un aporte significativo a la lógica de juego. Este razonamiento además puede considerar la lógica del crupier, pero no es necesario que se haga una diferencia entre una política `soft` o `hard`.

Finalmente es necesario realizar las siguientes llamadas al programa desde la línea de comandos:

1. `swipl -s black.prolog play <hand> <dealer> <cards>`: Ejecuta la lógica de un jugador e imprime `Hit` si el jugador decide jugar, o `Stand` si decide plantarse.
2. `swipl -s black.prolog dealer soft <hand>`.
3. `swipl -s black.prolog dealer hard <hand>`.

Condiciones de aprobación

Para la aprobación del trabajo el alumno debe entregar:

- Todos los códigos fuente de los programas.
- Un documento de texto donde explique en detalle y en lenguaje natural la estrategia de razonamiento utilizada y los pasos seguidos para su implementación de forma lógica.